IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 19, 2019, accepted July 25, 2019, date of publication August 19, 2019, date of current version September 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2935425

Search Personalization Based on Social-Network-Based Interestedness Measures

SHAYAN A. TABRIZI[1], AZADEH SHAKERY[1,2], MOHAMMAD ALI TAVALLAEI[3], AND MASOUD ASADPOUR[1], (Member, IEEE)

[1]School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran 14399-57131, Iran
[2]School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran 19538-33511, Iran
[3]Institute of Applied Intelligent Systems, University of Tehran, Tehran 14188-14751, Iran

Corresponding author: Azadeh Shakery (shakery@ut.ac.ir)

This work was supported in part by the Grant from the School of Computer Science, Institute for Research in Fundamental Sciences, under Grant CS1398-4-223.

ABSTRACT Personalized search methods try to tailor the search results to the user's needs and preferences. A popular source of information for personalization is social networks of users. In this paper, we aim at proposing a social-network-based personalized information retrieval (PIR) method that: *i)* is holistic (not just considering some local neighborhood of the searcher); *ii)* is efficient in term of computational/storage cost; *iii)* its personalization component has a probabilistic basis. To the best of our knowledge, no such method exists. We propose an interestedness measure and several variations of it for quantifying the interestedness of each user in another user based on a social network of users. The general idea is to try to mimic a searcher's behavior in the real world to estimate the interestedness. The measures are then exploited to personalize the retrieval results. We evaluate the resulting PIR methods and compare them in terms of retrieval effectiveness and computational/storage cost. We also compare them with some baseline methods. In summary, our analyses suggest that retrieval based on at least one of the measures performs well in terms of both retrieval effectiveness and computational/storage cost.

INDEX TERMS Information retrieval, personalization, personalized pagerank, social networks.

I. INTRODUCTION

Significant growth of Web 2.0 has led to the generation of huge volumes of data. In addition, because of the increasing role of the Internet in people's lives, almost any important event or accident has its own effect on the Web. There are also many new activities that are based on the Internet, which did not exist before. Furthermore, organizations publish huge volumes of data that did not previously exist digitally. Therefore, a significant amount of all the activities in the society are recorded on the Web.

The massive amount of data on the Web as well as the large number of users and their diversity bring up new challenges. From an information retrieval (IR) perspective, traditional one-size-fits-all solutions are not sufficient anymore. Different users with different information needs and preferences use the same queries in their searches. Thus, personalization methods, that try to tailor the search results to the user's needs and preferences, are proposed.

The associate editor coordinating the review of this article and approving it for publication was Chao Shen.

Different categories of personalization methods have been proposed and each category uses one or more information sources about users to perform the personalization (e.g., profiles, search histories, and social networks). Among different information sources, social networks are a popular type of information source and many methods are proposed based on them. One important benefit of using a social network as an information source is that it gives us much more information about the users than the relations of the individual users, and by analyzing the network, the information of other users can be used to enrich our information about a user. In this work, we focus on social networks as an information source for personalization.

A. OUR GOAL

In this paper, we address the following problem:

> In a collection of documents $\mathcal{D}$, in which each document d has a non-empty set of publishers P_d, we aim at personalizing search results for a query q according to the preferences of a searching user u_s, based on a social network of users (including publishers).

Our main goal is to propose a social-network-based personalized information retrieval (PIR) method that:

- is holistic (not just considering some local neighborhood of the searcher);
- is efficient in term of computational/storage cost;
- its personalization component has a probabilistic basis.

It is noteworthy that although in this paper we focus on publishing relationships between users and documents, our proposed method is not limited to such a setting. The method can be easily used for tasks with heterogeneous relationships between users and documents–like publishing, liking, sharing, and commenting–by defining proper aggregation functions, as will be discussed in Section IV-B.

B. OUR WORK

In this paper, we introduce an interestedness measure and several variations of it which quantify the interestedness of a user in another user in a social network. The general idea is to try to mimic a searcher's behavior in the real world to estimate the interestedness. The measures are based on two simplified explanations of the dynamics of information seeking/searching behavior in a society. In brief, according to the first explanation, the closer an individual is to the searcher or the more prominent (central) he is, the more likely is that he is the one who fulfills the information need of the searcher. Inspired by the explanation, an interestedness measure can weight users according to their centrality and closeness to the searcher.

According to the second explanation, within a society, a person is very likely to find his information needs in the most fine-grained community he belongs to, e.g., his family. If not found, it is then more likely that he can find his information needs in the community at the next level, e.g., his neighborhood, and this continues until he can find what he needs from a much further person in the hierarchy of the society. Inspired by the behavior, an interestedness measure can quantify the interestedness of users by taking the hierarchical structure of users into account.

We also propose a PIR method that exploits the introduced measures to personalize IR. We evaluate the proposed method by the PERSON evaluation framework [1], which makes repeatable and low-cost evaluation of PIR methods possible. We study the performances of the introduced interestedness measures in the PIR method and compare them in terms of retrieval effectiveness and computational/storage cost.

C. CONTRIBUTIONS

The contributions of this paper can be summarized as follows:

1) Proposing an interestedness measure and several variations of it that quantify the interestedness of a user in another user in a social network. The measures have three important features: *i)* They are holistic; *ii)* They are probability functions; *iii)* Some of them are efficiently computable;
2) Proposing a PIR method, based on the introduced interestedness measures, and showing that it obtains substantial results, while having a modest computational cost.

D. OUTLINE

The rest of this paper is organized as follows: Section II presents the related work. In Section III, we provide the intuition behind the proposed interestedness measures. Then, in Section IV, we introduce the base interestedness measure. We also describe the proposed PIR method in Section IV-B. Afterwards, in Section V, we introduce several variations of the base interestedness measure. Section VI presents the evaluation results. Finally, conclusions and areas of future work are discussed in Section VII.

II. RELATED WORK

Much research has been conducted on personalization in IR. Different personalization approaches have been proposed based on different information sources about users such as user-generated content (e.g., See [2], [3]), social annotations (e.g., See [4]–[7]), social networks (e.g., See [8]–[10]), and history of users' interactions (e.g., See [11]). Since the focus of this paper is on using social-networks as information sources, we focus on social-network-based PIR methods in the following and mention a number of them. It is noteworthy that these methods may use social network information in combination with other information sources. Readers are referred to surveys [12]–[16] for more information.

As mentioned in the introduction, one important benefit of using a social network as an information source is that by analyzing a social network, we can gain much more information about users than their direct relationships. This per se can help us mitigate the sparseness and cold-start problems since, e.g., only one relation between a new user and an old user can give us much information about the new user when our information about the old user and the rest of the network is considered. Also, social network information may be used in combination with other information sources to help mitigating the cold-start and sparseness problems in them.

One approach to using social networks information is to use the information of the neighbors of a user to enrich the model of the user. For example, Vosecky *et al.* [17] use the *individual user models* of the neighbors of a user to enrich his model. They compute a *collaborative user model* of a user by taking a weighted average of the individual user models of his friends, and combine his individual user model with the resulting collaborative user model. Although this approach may help in enriching the model of a user, it only considers the neighbors of the user and does not make use of the information about other users in the social network.

Another approach is to consider the whole network for the purpose of personalization, not only direct neighbors. For example, Yin *et al.* [18] personalize search results based on a *social influence* measure, which is defined based on the shortest path between the searcher and the publisher of a document. More precisely, they define social influence according to Eq. (1), in which $\mathcal{P}(u, v)$ denotes all paths from u to v in the network and $\delta(u, v)$ is the weight of the edge between u and v. They combine the normalized textual relevance score with the normalized social influence to calculate the score of a document with respect to a query and a searcher. As opposed to this method that considers shortest paths, our method employs personalized PageRank [19] (PPR) in calculating the personalization term, which is expected to yield better performance because PPR takes different paths between nodes into account, not only shortest paths.

$$SI_v(u) = \left(\min_{P(u,v)\in\mathcal{P}(u,v)} \left\{ \sum_{(n,n')\in P(u,v)} \delta(n, n') \right\} \right)^{-1} \quad (1)$$

A challenge in the whole-network approach is that the computational cost may be intractable. One approach to overcome this challenge is to consider the hierarchy of users in order to avoid, as much as possible, performing costly computations in the whole network, while still making use of information in the whole network. One method in this category is SNDocRank [8], which may be the most similar work to ours. This work defines a measure called Multi-Level Actor Similarity (MAS) and exploits it to personalize search results. MAS is based on a weighted version of LHN vertex similarity [20]. It applies weighted LHN vertex similarity at different levels of the clusters hierarchy and calculates the final similarities by combining the similarity values.

Our method is superior to SNDocRank in at least two aspects. The measures we define are probability functions over subsets of users, which makes them well-defined and convenient measures to be used in different applications. MAS values, on the other hand, are not necessarily in between [0, 1] and MAS may not be efficiently convertible into a probability function. Moreover, under some assumptions, the time complexity of our method (based on the proposed PHI measure) is $\mathcal{O}(n \log^2 n)$ while that of SNDocRank is estimated to be $\mathcal{O}(n^{1.688} \log n)$ [8], which is a major benefit of our method for large amounts of data.

Another approach is to use the information of interactions between users and documents for personalizing IR. For example, Khodaei and Shahabi [9] use the social network of users as well as the information of interactions (e.g., publishing, liking, sharing, and commenting on) between users and objects (documents) to personalize IR. Indeed, the information they use can be considered a heterogeneous network consisting of users, objects, and relationships between them. They define a social score according to Eq. (2), in which o is the object (document) to be scored; u_q is the searching user; U_o is the set of users with some action on o; and urf, uaf, and uwf are user relatedness, user action, and user weight functions, respectively. The functions may be defined in different ways in different applications. We mention the definitions we use in our experiments in Section VI-B. The social score is then combined with a textual score by a convex combination with an α coefficient. In comparison, the measures we define are probability functions over subsets of users, as opposed to the social score in this method, and thus have a stronger mathematical foundation. Also, our experimental results show that our method (based on the PHI measure) outperforms this method (with the definitions of urf, uaf, and uwf mentioned in Section VI-B) in term of retrieval effectiveness. Reference [21] can also be mentioned as a similar method with this approach.

$$socRel(o, q) = \sum_{v_i\in U_o} urf(u_q, v_i) \times uaf(v_i, o) \times uwf(v_i) \quad (2)$$

Khodaei et al. extend the above method in [10] to consider a network of documents (formed, e.g., by hyper-links between documents) in scoring the documents. They combine the social score of each document with a weighted sum of the social scores of the documents connected to it to obtain the new social score of the document. They show that such combination improves the retrieval results considerably.

A. PERSONALIZED IR EVALUATION

Evaluation of PIR methods is much more challenging than evaluation of traditional non-personalized methods. In a PIR problem, since the relevancy and irrelevancy of each document could be different for each individual, it is much harder to judge the results and demands much more manual work. The problem is worsened if we consider that the real searching users may be different from the users for whom we have relevance judgments and have different preferences. Such challenges have led to the conduction of dedicated research meetings for tackling the problem of PIR evaluation (e.g., [22], [23]).

Because of such challenges, there are not many evaluation datasets to be employed for evaluating PIR methods. Moreover, not all datasets could be employed for evaluating all PIR methods. For example, PIR methods that are based on social network relationships need users' relationships to be known, while many of evaluation datasets do not offer such information. Lack of widely-used standard datasets is also another concern in evaluating PIR. In addition, employing different sources of information in different methods (e.g., using or not using social network information) could make their comparisons unfair.

As a solution some indirect evaluation frameworks are proposed to address this problem. These frameworks "use surrogates of the PIR evaluation problem, instead of addressing it directly, to make PIR evaluation more feasible" [1]. Readers are referred to [1] for a survey of indirect evaluation frameworks. In this paper, we make use of PERSON [1] for the purpose of evaluation. The idea behind this framework is that in a dataset of scientific publications, in each paper (document) d, the cited papers are generally related to d from the perspective of d's author(s). Loosely speaking, in PERSON, a paper d is considered as a *query paper* and a query is extracted from it and the first author of it is considered as the searcher. The cited papers in d are then considered as the relevant documents from the perspective of the searcher. Although this assumption of relevancy is not accurate, [1] shows the consistency of the results of this framework with the results of human-judgment-based evaluation. We choose PERSON since it is studied and validated from different perspectives in [1] and also is capable of evaluating social-network-based PIR methods (as opposed to some other evaluation frameworks like ASPIRE [24]).

III. INTUITIONS

When an individual seeks some information or tries to find the answer of a question in a society (By "in a society", we mean by asking from others not, e.g., on the internet), he does not do that in a totally random fashion (i.e., he does not randomly draw an individual from all human beings and ask her the question). Instead, there are some phenomena/processes/etc. that govern his information seeking behavior. The behaviors also differ between individuals and depend on their characteristics/preferences/beliefs/etc., which makes their information seeking behaviors personalized. In this section, we introduce two explanations for information seeking/searching behaviors of individuals, which form the intuitions of the interestedness measures proposed in the next sections. Although the explanations are simplified and do not take many complexities of humans and their information seeking behaviors into account, they can be used to propose an efficient PIR method. To quote George Box, "All models are wrong, but some are useful" [25].

The first explanation assumes that the searching behavior of an individual is only governed by the social network of individuals. In this explanation, the searching individual is modeled as a random surfer who randomly traverses the social network of the society such that the closer an individual is to the searcher, the more probable is that he is the one who fulfills his information need.[1] This explanation for searching also has the implication that the more central a person is, the more likely is that he is the one who fulfills the need. This explanation is similar to the intuitive justification that Brin and Page [26] mention for the PageRank algorithm. We call this explanation of searching behavior the *random surfer explanation*.

The second explanation considers the communities of individuals. This explanation assumes a hierarchical structure in the society. In such a society the individuals form small groups, then these small groups form larger groups, and so on. In this explanation, the searching behavior of an individual looking for some information is assumed to be biased toward the communities he belongs to in the hierarchy of the communities.[2] The lower the community is in the hierarchy, the higher bias is assumed. For example, in a society with the network structure of Fig. 1, assume that the individual marked in red starts a search in the society. He starts searching

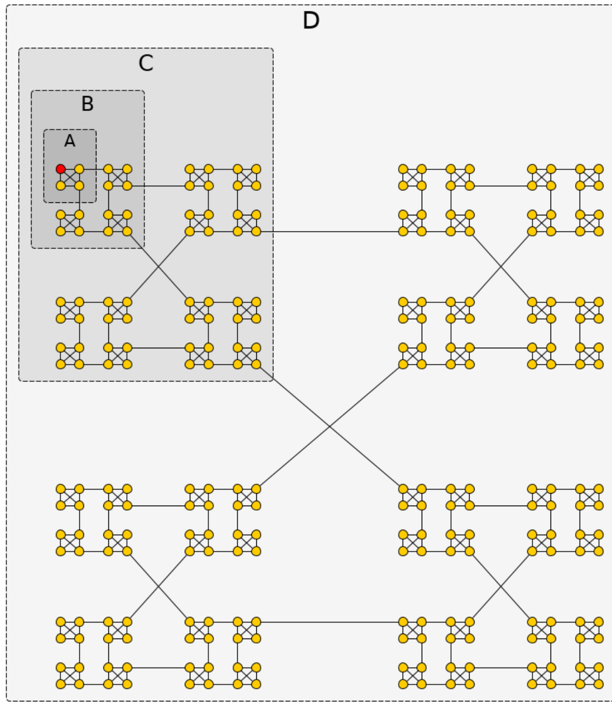

FIGURE 1. A sample social network, in which the individual marked in red is a member of four (hierarchical) communities A, B, C, and D.

in community A. If he could not satisfy his information need, then he searches in community B, and continues in this manner in C and D (In general, there may be biases and not necessarily strict orders). An observation from previous work that supports this explanation is what Wolek [27] states: "Interviewees preferred to contact some "non-expert" but close associates first even when the "expert" was a former acquaintance." We call this explanation for the searching behaviors the *hierarchical explanation*.

It is noteworthy that although assuming a hierarchical structure is a simplistic assumption, it is a prevalent simplifying assumption in different areas of research. There is also a theoretical foundation for this assumption in the previous work. Ravasz and Barabasi [28] state that "hierarchy is a fundamental characteristic of many complex systems".

IV. METHODOLOGY

In this section, we propose an interestedness measure based on social network information. The measure is grounded on the hierarchical explanation of searching behavior. Roughly speaking, the general idea is to imitate the behavior of a searcher who at the first step tries to satisfy his information need by exploring within the smallest community he belongs to, and continues the exploration up to the largest community he belongs to, i.e., the whole society. In practice, the documents are weighted so that the more common clusters a document's publisher(s) and the searcher belong to, the more weight the document is expected to gain. Although the measure is based on the hierarchical explanation of searching

[1] This explanation and the next one may explain the information seeking/searching behaviors in the pre-Internet world better than in the post-Internet world. However, we are using these explanations as intuitions and do not necessarily rely on their complete correctness.

[2] In some domains, this explanation may explain the reality better than in some others.

behavior, it is very flexible and can be easily used to imitate the random-surfer-explanation-based searching. In the next section, we show that it can be used to compute a variety of different interestedness measures.

In the following subsections, we first introduce Cluster-Sensitive Rank as a measure of the interestedness of a user in another user with respect to the social network structure. Then, we introduce a framework for using an interestedness measure to personalize search results in regard to the searcher attitudes.

A. CLUSTER-SENSITIVE RANK

In this subsection, we introduce Cluster-Sensitive Rank (CSR) that measures the interestedness of a user, say u_s, in a target user, say u_t, according to the social network hierarchy, $CSR(u_t|u_s)$. In the next section, we show that the measure can be easily used to compute a variety of different interestedness measures by just changing the input hierarchy, without any changes to the implementation. It is important to note that by interestedness we mean being interested and not having an agreement. In other words, u_s may disagree with the target user, but is still interested in what he writes or says. We can also interpret the interestedness as the importance of the target user to u_s.

For simplicity of presentation, we assume that the hierarchy of users has four properties: *i)* The children of each cluster (We use the terms cluster and community interchangeably hereafter) are either clusters or users, but not both; *ii)* Each user is a member of a cluster; *iii)* The hierarchy has at least two clusters; *iv)* The hierarchy is a rooted tree (not a forest). We call a hierarchy that meets these properties a proper hierarchy. It's important to note that any arbitrary hierarchy can be easily modified to meet these properties. If the first property does not hold, we put each user that its parent cluster has also cluster children into a separate cluster and make the new cluster a child of the parent cluster. For the second property, we put each user outside any cluster in a separate cluster. Then, if any of the last two properties does not hold, we add a dummy cluster to the hierarchy and consider it as the parent of all the root clusters of the forest. It is noteworthy that we assume the first property for simplicity of presentation and it is not essential in practice (We do not enforce it in our experiments).

In an interestedness measure $I(*|u_s)$, we intend to weight different users according to the interests of user u_s. Inspired by the hierarchical explanation of searching behavior, we define the interestedness measure by biasing the interestedness toward the communities of u_s.

To bias the interestedness, we use personalized PageRank [19] (PPR) scores that are biased toward the communities of u_s. PPR measures the stationary distribution of a random surfer in a network which starts at a random vertex with a probability distribution $\vec{s}$ and at each step follows an outgoing edge of the current node with probability d (damping factor) or restarts with probability $1 - d$ (It also restarts if the current node does not have any outgoing edges). We use $\vec{s}$

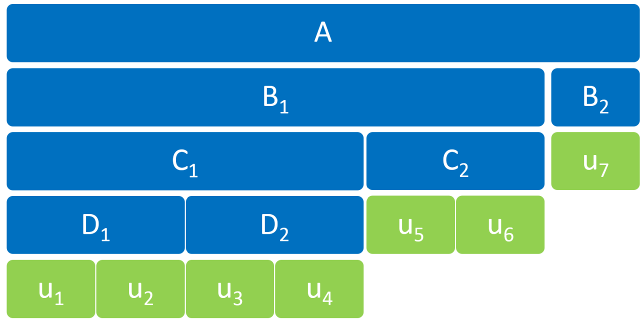

FIGURE 2. A sample hierarchy: A, B_*, C_*, and D_* are clusters and u_* are users.

for biasing the scores. Using PPR gives an inherent random-surfing nature to our measure, which will be discussed more in Section V. Hereafter, by saying PPR *from* a set of vertices S, we mean PPR by considering the vertices in S as the starting points of the random surfer with a uniform probability distribution (i.e., $\vec{s}_i = 1/|S|$ if vertex i is in S and $\vec{s}_i = 0$ otherwise). We denote the PPR score of a node u in an induced subgraph $G[C]$, in which G is the social network graph, from a child cluster $C' \subseteq C$ by $PPR(u, C'; C)$. We also define $PPR(S, C'; C)$ for a set S of vertices as $\sum_{u \in S} PPR(u, C'; C)$.

In a cluster C, by computing PPR scores from a child cluster C', the scores are biased toward C'. Thus, terms $PPR(C_i, C_i; parent(C_i))$, in which C_is are communities that u_s belongs to (in the hierarchy), can be used to bias the interestedness measure of u_s toward the communities of u_s. In this regard, a term $PPR(C_i, C_i; parent(C_i))$ can be viewed as an estimate of the likelihood that a user in community C_i is interested in a user in the same community rather than a user in the rest of $parent(C_i)$. Obviously, this estimate tends to give higher values to the users in C_i, which is exactly what is intended, i.e., giving higher scores to users in the same communities as the searcher.

Thus, CSR can be defined as a combination of the PPR scores from common cluster ancestors of u_s and the target user and the first non-common one. Strictly speaking, CSR for a searching user u_s, with its cluster ancestors denoted by $X_1, X_2, \ldots, X_i, Y_{i+1}, \ldots, Y_j$, and target user u_t with its cluster ancestors denoted by $X_1, X_2, \ldots, X_i, Z_{i+1}, \ldots, Z_k$, is defined as Eq. (3). We assume that there is at least one Y_*. Thus, if all the cluster ancestors of u_s are common with those of u_t, we consider X_i as non-common (See example $CSR(u_4|u_3)$ below).

$$\begin{aligned} CSR(u_t|u_s) = PPR(&u_t, Y_{i+1}; X_i) \\ &\times PPR(X_i, X_i; X_{i-1}) \\ &\times PPR(X_{i-1}, X_{i-1}; X_{i-2}) \\ &\times \ldots \\ &\times PPR(X_2, X_2; X_1) \end{aligned} \tag{3}$$

For example, in the hierarchy depicted in Fig. 2, if u_3 starts searching, the CSR scores of users u_1, u_4, u_5, and u_7

are calculated as shown in Eqs. (4) to (7).

$$CSR(u_1|u_3) = PPR(u_1, D_2; C_1) \times PPR(C_1, C_1; B_1) \times PPR(B_1, B_1; A) \tag{4}$$

$$CSR(u_4|u_3) = PPR(u_4, D_2; C_1) \times PPR(C_1, C_1; B_1) \times PPR(B_1, B_1; A) \tag{5}$$

$$CSR(u_5|u_3) = PPR(u_5, C_1; B_1) \times PPR(B_1, B_1; A) \tag{6}$$

$$CSR(u_7|u_3) = PPR(u_7, B_1; A) \tag{7}$$

We also define $CSR(S|u_s)$ for a set of vertices S as $\sum_{u \in S} CSR(u|u_s)$. From a probabilistic viewpoint, $CSR(S|u_s)$ can be interpreted as the probability that if u_s selects a user according to his interests, the user is in S. In Section A, we prove that $CSR(S|u_s)$ is a probability function. This property makes CSR a well-defined and convenient measure to be used in different applications.

An important design decision made in CSR is assuming that in computing the interestedness score of a user u_t with respect to a user u_s, only their common cluster ancestors and the first non-common one are important and the others are not that determinant. The intuition behind this decision is that if two individuals have more fundamental differences in their viewpoints/behaviors (belong to two different clusters in a higher level), the details of their viewpoints/behaviors may not be that relevant. For example, in the dimension of beliefs (which is one dimension that may bias the searching behavior), if a searcher, say s, is a theist and the target user, say t, is an atheist (See Fig. 3), whether s is, e.g., a Christian or a Jew may not change his views about t much. Similarly, if the searcher, s, is an atheist and the target user, t, is a theist, the detailed beliefs of t may not be that determinant for s. For instance, whether he is a Christian or Jew may not be that relevant to him, let alone more fine-grained categorizations like being a Catholic or a Protestant. Although, this assumption seams reasonable in many circumstances, it may not be valid in all cases. However, it allows CSR to be computationally efficient.

Another design decision made in CSR is using PPR locally. That is to say, for each cluster C, we compute PPR scores just for the nodes of its direct parent cluster with considering C as the starting points of the random surfer. For example, in the hierarchy illustrated in Fig. 2, using PPR locally means that when computing PPR from cluster D_2 we have to compute PPR only for the nodes of C_1, while not using PPR locally means that we have to compute PPR from D_2 for all of the nodes of the network. This decision allows CSR to have a small computational and storage footprint, as will be discussed in the next section.

It is also worth mentioning that PageRank can be considered as a non-personalized approach for defining an interestedness measure ($I(u_t|u_s) = PR(u_t)$), in which the amount of interestedness in a user is independent of u_s. In this sense, CSR can be considered as PageRank's personalized counterpart.

B. PERSONALIZED RETRIEVAL WITH CSR

In order to exploit an interestedness measure (like CSR) or other similar measures (e.g., MAS [8]) in retrieval, one needs a conversion from the users space to the documents space. In other words, a document interestedness measure $I(d_t|u_s)$ should be defined based on the user interestedness measure, $I(u_t|u_s)$. A simple definition for $I(d_t|u_s)$ can then be to consider the publisher of d_t, say u_{d_t} and define $I(d_t|u_s) = I(u_{d_t}|u_s)$. However, in a more general setting, a document may be published by more than one publisher. For example, about a scientific paper, each of the authors can be considered a publisher of the paper. Thus, in general, $I(d_t|u_s)$ can be defined in different forms that aggregate the interestedness scores of the individual publishers to a document-specific score. Denoting the set of publishers of a document d by P_d, and the most influential publisher of it (e.g., the first author) by $m(d)$, some possible definitions are:

- $I_{sum}(d|u_s) = \sum_{u_t \in P_d} I(u_t|u_s)$
- $I_{max}(d|u_s) = \max_{u_t \in P_d} I(u_t|u_s)$
- $I_{avg}(d|u_s) = \sum_{u_t \in P_d} I(u_t|u_s)/|P_d|$
- $I_{most}(d|u_s) = I(m(d)|u_s)$

The choice of the appropriate aggregation function depends on the specifics of the retrieval domain. It is noteworthy that the aggregation function may take heterogeneous relationships into account. For example, similar to Khodaei and Shahabi [9] (See Section II), it may consider relationship types other than publishing such as liking, sharing, and commenting in the conversion from the users space to the documents space. So, although in this paper our focus is on publishing relationships, our method can be used in other applications with other relationship types by defining proper aggregation functions.

The pseudocode of the algorithm for simultaneously computing $CSR(u_t|u_s)$ for all $u_t \in Publishers$ and $u_s = s$ is demonstrated in Algorithm 1. The values of the returned associative array are then given to the aggregation function to compute $I(d_t|u_s)$. In the algorithm, the CSR function initializes the auxiliary associative array P and then calls procedure ComputeCSR, which computes the CSR scores and stores them in the associative array S. Procedure ComputeCSR recursively computes CSR scores for the nodes in the set of keys of P (KeySet(P)). The SplitNodes(c, *nodes*, s) splits the nodes in set *nodes* into two groups: nodes that are in the same sub-cluster of c that the searcher node s is a member of (group C); other nodes (group N). The GetChild(c, n) function returns the sub-cluster of c that node n is a member of or NIL if no such sub-cluster exists. For each node in group N, the remaining term of the CSR formula (e.g., $PPR(u_4, D_2; C_1)$ in example $CSR(u_4|u_3)$ of Section IV-A) is multiplied in line 18, if c is the lowest level cluster that s is a member of; otherwise, the two remaining terms (e.g., $PPR(u_1, D_2; C_1) \times PPR(C_1, C_1; B_1)$ in example

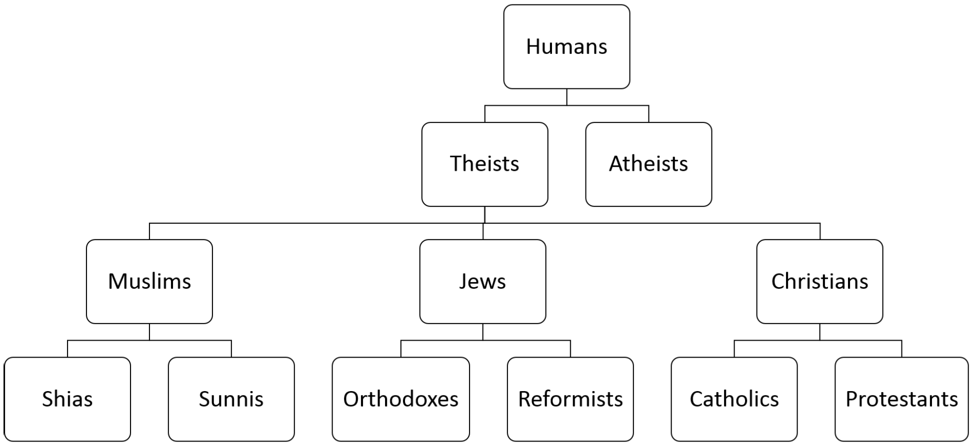

FIGURE 3. A sample hierarchy of humans according to their religions. The hierarchy is not exhaustive.

$CSR(u_1|u_3)$) are multiplied in line 16. ClusterPPR(c) computes $PPR(c, c; parent(c))$ and NodePPR(c, n) computes $PPR(n, c; parent(c))$. The final CSR scores for the nodes in group N are stored in S in line 20. For the nodes in group C, a term $PPR(c, c; parent(c))$ (e.g., $PPR(B_1, B_1; A)$ in example $CSR(u_1|u_3)$) is multiplied in line 25 and ComputeCSR is called in line 27 for performing the rest of the computations.

To perform PIR, one approach is to estimate $P(d_t|u_s, q)$ for each d_t, in which u_s is the searching user and q is the query. By assuming conditional independence, according to Eq. (8), we can decompose $P(d_t|u_s, q)$ into two terms: a personalization term $P(d_t|u_s)$ and a query likelihood term $P(q|d_t)$. The query likelihood term can be estimated as in a typical IR method (See [29] for a survey).

$$P(d_t|u_s, q) = \frac{P(u_s|d_t) \times P(q|d_t) \times P(d_t)}{P(u_s, q)} \propto P(d_t|u_s) \times P(q|d_t) \quad (8)$$

By using CSR as the personalization term, the final ranking score for a document d_t is then calculated as shown in Eq. (9).

$$Score(d_t|u_s, q) = \rho \log(CSR(d_t|u_s)) + \log(P(q|d_t)) \quad (9)$$

In practice, however, if a document does not match the query (when searching with an inverted index), it is not retrieved by the system even if it has a high value of the personalization term.

It must be noted that although CSR is a probability function over the subsets of users ($\sum_{u \in Users} CSR(u|u_s) = 1$), it is not necessarily a probability function over the subsets of documents, since documents do not necessarily have a one-to-one relation with users. Thus, we put a ρ coefficient in Eq. (9) for normalizing the personalization term, which can also be used to adjust the degree of personalization. However, assuming that each document has only a small number of publishers (e.g., authors in a typical paper), the range of the values of $CSR(d_t|u_s)$ is probably (depending on how $CSR(d_t|u_s)$ is defined) similar to that of $CSR(u_t|u_s)$. Thus, under such assumption, $CSR(d_t|u_s)$ can be combined with the query likelihood term without much tuning of ρ. Therefore, we use $\rho = 1$ in our experiments. In Section VI-F, we also experimentally show that $\rho = 1$ is an appropriate choice. This characteristic makes CSR a convenient measure to be used in personalization.

V. INTERESTEDNESS MEASURES

In the previous section, we proposed CSR based on a hierarchical explanation of searching behavior. However, CSR has also an inherent random-surfing nature because of its use of PPR, which per se has a random-surfing nature [19]. Because of this dual nature and also its flexibility to be employed on different hierarchical structures, it can be used to compute a variety of different interestedness measures. In the rest of this section, we explain some possible measures that can be computed with CSR, just by changing the input hierarchy.

In the following, we consider several assumptions in the computation of complexities. We assume that all PPR scores are computed based on an exact iterative computation (See [30] for a recent work on the approximate computation of PPR scores) with cost $\mathcal{O}(E \log V)$ (See Section B for more details). We also assume that the hierarchy of the network is roughly balanced and has a small maximum branching factor.

A. PPR INTERESTEDNESS (PI)

The first measure is fully based on the random-surfer explanation. It estimates the interestedness of a user u_s in a user u_t by the PPR score of u_t from u_s, i.e., $I(u_t|u_s) = PPR(u_t, \{u_s\}; S)$, in which S is the whole society. By the random-surfer interpretation of PPR, this is identical to the proportion of visits of a random surfer to u_t, when at each visit it may return to its starting point u_s with a probability α. Thus, it is totally consistent with the random surfer explanation of searching behavior. We call this measure PPR interestedness (PI). CSR can be used to compute PI simply by giving it an input

Algorithm 1 Pseudocode for Simultaneously Computing $CSR(u_t|u_s)$ for All $u_t \in Publishers$ and $u_s = s$

```
function CSR(r, s, Publishers)
INPUT: r: root cluster of the hierarchy
       s: the searcher node
       Publishers: set of publishers
    P ← new AssociativeArray
    for each p in Publishers do
        P{p} ← 0
    end for
    S ← new AssociativeArray
    ComputeCSR(r, s, P, S)
    return S
end function

procedure ComputeCSR(c, s, P, S)
INPUT: c: a cluster in the hierarchy
       s: the searcher node
       P: an auxiliary associative array
       S: associative array in which the final score of each node is stored
    key_set ← KeySet(P)
    <C, N> ← SplitNodes(c, key_set, s)
    next_cluster ← GetChild(c, s)
    for each node n in N do
        if next_cluster ≠ NIL then
            P{n} ← P{n} × ClusterPPR(c) × NodePPR(next_cluster, n)
        else
            P{n} ← P{n} × NodePPR(c, n)
        end if
        S{n} ← P{n}
    end for
    if C is not empty then
        A ← new AssociativeArray
        for each node n in C do
            A{n} ← P{n} × ClusterPPR(c)
        end for
        ComputeCSR(next_cluster, s, A, S)
    end if
end procedure

function SplitNodes(c, nodes, s)
    searcher_cluster ← NodeCluster(s)
    C ← ∅
    N ← ∅
    for each node n in nodes do
        child ← GetChild(c, n)
        if child ≠ NIL and child is equal to or an ancestor of searcher_cluster then
            C ← C ∪ {n}
        else
            N ← N ∪ {n}
        end if
    end for
    return <C, N>
end function
```

hierarchy in which each individual is put into a separate cluster that has a root cluster as its parent.

The computation of this measure can be costly, requiring a time complexity of $\mathcal{O}(EV \log V)$ for computing $I(u_t|u_s)$ for all values of u_t and u_s, and even storing the exact pre-computed scores requires $\mathcal{O}(V^2)$ memory.

B. HIERARCHICAL INTERESTEDNESS (HI)

The second measure is fully based on the hierarchical explanation. It estimates the interestedness by considering the hierarchical structure of the network and uses CSR on a hierarchy of clusters obtained according to the network structure, i.e., $I(u_t|u_s) = CSR(u_t|u_s)$. The clusters may be extracted by a clustering method or by considering the inherent properties of the society.

Computing the required PPR scores demands computing $\mathcal{O}(V \log V)$ scores, which can be pre-computed and stored for reasonably large networks. In addition, in Section VII, under some assumptions, we prove that the time complexity of computing the required PPR scores is $\mathcal{O}(E \log^2 V)$. Having the pre-computed scores, HI can be computed for each u_t in $\mathcal{O}(\log V)$ time at the query time.

C. CLUSTERED INTERESTEDNESS (CI)

An important drawback of the PPR measure is its high cost of computation and storage. One approach to overcome this drawback is to consider the flat (non-hierarchical) clusters of the society and employ a random-surfing approach with considering the cluster of the searcher as the starting points (i.e., $I(u_t|u_s) = PPR(u_t, C_{u_s}; S)$, in which C_{u_s} is the cluster of u_s). This measure can be computed with CSR by considering a hierarchy that consists of the flat clusters with a root cluster as their parent.

The cost of CI is highly dependent on the number of clusters (which may vary depending on the approach used for extracting clusters). If the number of clusters is very high, the cost can be similar to PI and a low number of clusters reduces the effect of personalization. If we assume the number of clusters is of $\mathcal{O}(\sqrt{V})$, the cost of computing all CI scores is of $\mathcal{O}(E\sqrt{V} \log V)$ and storing them requires $\mathcal{O}(V\sqrt{V})$ memory.

D. PERSONALIZED HIERARCHICAL INTERESTEDNESS (PHI)

Looking at Eq. (3), we observe that HI is indeed just considering the clusters of the searcher in measuring the interestedness, and not the individual preferences of him. Thus, this measure may be better called a *groupized* interestedness measure rather than a personalized measure. One extension to HI is to consider the individual preferences of the searcher when they are important. Similar to the design decision made in CSR, we assume that the particular preferences of the searcher are decisive only when his more fundamental viewpoints that are shared by his group are satisfied. By this assumption, we can use CSR to compute a personalized version of HI in a straightforward manner. If we put each individual in a separate cluster at the lowest level cluster he belongs to, the CSR also considers the particular preferences of the searcher when he shares the same cluster ancestors with the target user. In other words, if all the cluster ancestors of a searcher u_s are shared by a target user u_t, putting each individual into a separate cluster, causes B_{i+1} in term $PPR(u_t, B_{i+1}; A_i)$ of Eq. (3) to include only one node u_s, and thus causes the measure to take the individual preferences of u_s into account. We call this interestedness measure personalized hierarchical interestedness (PHI) hereafter.

TABLE 1. The required memory for storing and the time complexity of computing the required values for different interestedness measures under the mentioned assumptions.

Measure	Time Complexity	Memory
PI	$\mathcal{O}(EV \log V)$	$\mathcal{O}(V^2)$
HI	$\mathcal{O}(E \log^2 V)$	$\mathcal{O}(V \log V)$
CI	$\mathcal{O}(E\sqrt{V} \log V)$	$\mathcal{O}(V\sqrt{V})$
PHI	$\mathcal{O}(E \log^2 V)$	$\mathcal{O}(V \log V)$
PCI	$\mathcal{O}(E\sqrt{V} \log V)$	$\mathcal{O}(V\sqrt{V})$

Under the mentioned assumptions, the cost of computing and storing the required PPR scores for PHI is similar to HI. It is noteworthy that if the modified hierarchy has large clusters with many single-node clusters as their direct children, the efficiency of PHI is compromised (e.g., when the root cluster has many single-node children, which causes a PPR to be computed on the whole graph for each child). This can be easily solved at implementation by putting each node in a separate cluster only if its parent cluster contains not so many nodes.

E. PERSONALIZED CLUSTERED INTERESTEDNESS (PCI)

Similar to HI, CI can also be modified to be a personalized measure. Personalized clustered interestedness (PCI) can be computed with CSR simply by modifying the hierarchy used in CI such that each node in a cluster C is put into a separate cluster with C as its parent.

Again, the cost of PCI is highly related to the number of clusters and at its worst case can be similar to that of PI. If we assume that the number of clusters and the number of nodes in each cluster is of $\mathcal{O}(\sqrt{V})$ and the number of edges in each cluster is of $\mathcal{O}(\frac{E}{\sqrt{V}})$, the cost of computing the required PPR scores is of $\mathcal{O}(E\sqrt{V} \log V + V \frac{E}{\sqrt{V}} \log \frac{V}{\sqrt{V}}) = \mathcal{O}(E\sqrt{V} \log V)$. Also, the memory required for storing the PPR scores is of $\mathcal{O}(V\sqrt{V})$. It can be seen that if we have access to a roughly balanced hierarchy with a small maximum branching factor c (meaning that even the lowest level clusters have $\mathcal{O}(c)$ nodes as their children), PHI can be more efficient compared to PCI. The required memory for storing and the time complexity of computing the required values for different interestedness measures are summarized in Table 1.

VI. EXPERIMENTAL EVALUATION

In this section, we present our experimental results. In Section VI-A, we describe the experimental setup. We introduce the baseline methods in Section VI-B. Then, in Section VI-C, we evaluate the proposed PIR methods and compare them with the baselines. In Section VI-D, we try to crosscheck the results of Section VI-C by using another evaluation approach. We discuss the effect of using different aggregation functions on the results in Section VI-E. In Section VI-F, we investigate the effect of parameter ρ on the results. Finally, we analyze the time cost of using CSR in Section VI-G.

A. EXPERIMENTAL SETUP

We employ the PERSON evaluation framework for evaluating PIR methods. This framework is not only shown to be consistent with the traditional Cranfield-based evaluation, but also makes low-cost evaluation of social-network-based PIR methods possible [1].

We make use of a cleaned subset of AMiner's citation network V2 dataset.[3] [31], containing the information of about 100,000 papers. We performed some data cleaning/pruning steps on the original dataset to obtain the dataset we used in our experiments. One set of steps includes stripping diacritics from titles and abstracts, removing papers with no abstracts or no authors, etc. Readers are referred to [1] for more details on these steps. In addition, we performed some more steps to reduce the size of the dataset. One important reason for reducing the size of the dataset was that computing the PI measure was time-consuming, and thus by reducing the size of the dataset we could make the experimentation faster. To reduce the dataset size, we first filtered out the papers dated before 2003. Then, we iteratively removed authors and papers from the dataset so that at the end the following conditions hold in the dataset:

- The authors graph is a connected graph;
- Each paper is either cited by another paper or cites more than five papers. The reason for this condition is that PERSON does not consider papers with less than or equal to five references as query papers. Thus, not being cited by another paper, nor citing more than five papers means that the paper is almost useless in the evaluation procedure (although it may be useful in making the authors network richer as we used in our experiments);
- Each author (co-)authors at least one paper.

The statistics of the resulting dataset are depicted in Table 2. The dataset and an updated version of the source codes of our experiments are accessible at https://figshare.com/articles/PERSON_Dataset_V2/6958514 and https://github.com/shayantabrizi/CSR (The codes of the experiments of Section VI-G are accessible at branch "performance-test").

[3] https://aminer.org/citation (Accessed on June 28, 2019)

TABLE 2. Cleaned dataset statistics.

Total number of papers (documents)	102,923
Average title length	9.3 ± 3.2
Average abstract length	154.3 ± 60.4
Total number of citations	337,978
Total number of authors	122,063
Average number of authors	3.2 ± 1.7
Total number of co-authorships	494,436
Average number of references*	5.7 ± 4.8

* Only references to papers within the dataset; averaged over papers with at least one such reference.

We use the co-authorship network as the social network used for personalization. We employ PPC [32] v1.3.[4] for hierarchical clustering of the network. This method produces acceptable results at a low computational cost and also tends to produce a significantly lower number of clusters (2,051 clusters) compared to some other methods (Infomap [33] v0.19.20.[5] and Louvain [34] v0.3.[6]). In the proposed methods, we use f_{sum} as the aggregation function, unless otherwise stated.

We conduct our experiments using Apache Lucene 7.3.0,[7] unless otherwise stated. In PERSON, we use a similar set of settings to that of [1]: We use the title-based scheme for query extraction (using title of the paper as the query) and the modified abstract-based representation for extracting the textual representation of papers (a document consists of the combination of the title and the abstract of a paper); We also exploit the *inappropriate relevants*, *inappropriate searches*, *inappropriate queries*, and *publication-date-based filtering* heuristics (See [1] for details); The results reported are based on a total of 2,000 queries. In addition, similar to [1], we use normalized discounted cumulative gain (NDCG@k) [35], mean average precision (MAP), and precision at k (P@k) as evaluation measures. We consider $k = 100$ for NDCG@k and $k = 10$ for P@k throughout the experiments. Also, we use one-tailed paired Student's t-test with 99% confidence for statistical testing.

In the following, for simplicity, we refer to a retrieval method that is based on an interestedness measure just by the name of the measure.

B. BASELINES

In the experiments, we consider three baselines. The first baseline (LM), which is a non-personalized method, is based on the language modeling framework [36] with the Dirichlet prior smoothing [37] ($\mu = 400$). We use this method also for estimating the query likelihood term ($P(q|d_t)$) in the proposed methods.

The second baseline (Social-Textual) is an approximate implementation of the method of [9], mentioned in Section II.

[4] https://github.com/shayantabrizi/PPC (Accessed on June 28, 2019)
[5] http://www.mapequation.org/code.html (Accessed on June 28, 2019)
[6] https://louvain.sourceforge.io (Accessed on June 28, 2019)
[7] https://lucene.apache.org (Accessed on June 28, 2019).

The implementation is based on the implementation that the authors kindly provided to us. It considers only users with a limited distance from the searcher. We use two as the limit (i.e., only the searcher's friends and friends of friends are taken into account). We use Lucene's TF-IDF-based *ClassicSimilarity* to compute the textual scores (The original paper also uses a TF-IDF-based method). We define *urf* according to Eq. (10). In this definition, the last three cases are based on the provided implementation. The first case is added since we want other documents of the searcher to be considered in search, and thus we consider a β relatedness for the searcher himself, which we tune in our experiments. It is noteworthy that our proposed measures do not need such an extra parameter and they automatically give score to the searcher as well. Moreover, similar to [1], we define *uaf* and *uwf* according to Eqs. (11) and (12).

$$urf(u_q, u_i) = \begin{cases} \beta, & u_q = u_i \\ 1, & u_i \text{ is a friend of } u_q \\ \frac{1}{2}, & u_i \text{ is a friend of friend of } u_q \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

$$uaf(u_i, o_k) = \begin{cases} \left(\frac{1}{\#\text{ of authors of } o_k}\right)^{1/2}, & \\ \qquad \text{if } u_i \text{ is an author of } o_k & \\ 0, \quad \text{otherwise} & \end{cases} \tag{11}$$

$$uwf(u_i) = \log\left(1 + \min\left(\frac{\#\text{ of co-authors of } u_i}{100}, 1\right)\right) \tag{12}$$

The third baseline (SI) is based on [18], mentioned in Section II ([18] focuses on efficiency rather than retrieval effectiveness; However, its scoring function can be used to perform PIR). We use Lucene's TF-IDF-based *ClassicSimilarity* to compute the textual scores in this method (The original paper also uses a TF-IDF-based method) and consider the first authors in computing the shortest paths. We also make some modifications to the original method. First, in our experiments, we observed that considering edge weights in SI resulted in lower evaluation measures (We considered two ways of converting original edge weights to dissimilarities: $dis(w) = \frac{1}{w}$ and $\frac{1}{w+1}$). The reason may be that in this method when the neighbor of the searcher with the highest SI has a high value of SI, say 50, SIs of other nodes are divided by the value (for the purpose of normalization), which may highly reduce the effect of personalization (By contrast, our measures, which are probability functions overs subsets of users, do not need such heuristic normalizations, which can avoid such unwanted consequences). Thus, we ignore edge weights in this method. Second, we consider a β value for the SI of each node on itself, which we tune in our experiments (This value is not considered in the normalization of SI values).

It is noteworthy that although many PIR methods are proposed in previous work, not all of them can be used as baselines for our comparisons since they are diverse in the resources they use for personalization. For example, some methods use social annotations (e.g., See [38]) and some use user-generated content (e.g., See [2]) as sources of information about users. Comparing methods relying on different information sources may not be fair.

TABLE 3. Comparison of different IR methods. All the differences are significant except those specified.

#	Method	NDCG@100	MAP	P@10
1	LM	.320	.174	.117
2	Social-Textual	.408	.240	.153
3	SI	$.440^{6,8}$	.267	.163
4	PI	**.451**	**.279**	$\mathbf{.169}^{8}$
5	CI	.364	.210	$.134^{7}$
6	PCI	$.438^{8}$	$.271^{3,8}$	$.165^{3}$
7	HI	.359	.206	.134
8	PHI	.436	$.270^{3}$	$.166^{3,6}$

To emphasize the importance of the possibility of using a default ρ value in our method, it is worth mentioning that in both of the personalized baselines–Social-Textual and SI– there are two parameters to be tuned. According to our test results, we choose 0.8 and 0.2 for the convex combination coefficients in Social-Textual and SI, respectively. We also choose $\beta = \frac{1}{.09}$ in Social-Textual and $\beta = 2.5$ in SI. It is obvious that reaching these values requires a considerable amount of time and work. Also, our results showed that the choice of these parameters could have considerable effects on the retrieval performances. This can be contrasted with our method, which, under the mentioned assumption (Section IV-B), can probably be used without much tuning of ρ. In the experiments, we use a default ρ value of 1.0 (except in Section VI-F, in which we investigate the effect of different ρ values) and a default damping factor of 0.85 (a common damping factor for PageRank) in PPR computations.

C. COMPARISON OF PERSONALIZATION METHODS

In the first experiment, we compare the performances of our proposed methods and the baselines according to different evaluation measures. Table 3 depicts the results. All the differences in the table are significant except the ones specified. It can be seen that all the personalization methods outperform LM significantly and PI obtains the best results. In addition, CI and HI perform similarly, with CI performing slightly better. Both of the personalized measures (PCI and PHI) perform considerably better than their groupized counterparts (CI and HI), which shows the importance of considering the individual preferences of users rather than just considering their clusters. Moreover, the results of the clustered measures (CI and PCI) are similar to those of the hierarchical measures (HI and PHI), although mostly the clustered measures obtain subtly better results. The similar performances together with the better asymptotic cost of the hierarchical measures (in term of both the computational complexity and the storage; See Table 1) suggest that they may be more preferable in many applications.

Also, the Social-Textual baseline is outperformed by PI, PCI, and PHI and the SI baseline performs similar to PCI and PHI. It is important to note that the parameters of both Social-Textual and SI have been tuned on the dataset, while for our methods we used default parameter values. Moreover, it is noteworthy that the complexity of computing all pairs of shortest paths in SI is probably between $\mathcal{O}(V^2)$ and $\mathcal{O}(V^3)$ (depending on the algorithm used) and storing the exact values of shortest path lengths requires $\mathcal{O}(V^2)$ memory, which can be very costly for large networks.

In addition, although PI performs somewhat better than PCI and PHI in term of retrieval effectiveness, computing and storing the PI scores can be costly for large networks, at least by the basic exact computation (See [30] for a recent work on the approximate computation of PPR scores; It is noteworthy that methods for approximate computation of PPR scores may also be used in computing the PPR scores required for the other measures).

In summary, the results suggest that four methods SI, PI, PHI, and PCI are effective in term of retrieval performance. Moreover, the high retrieval performance of PHI together with the fact that its scores can be computed more efficiently (almost all of the time-consuming computations can be performed offline and the required values can be stored in a reasonable amount of memory), suggest that PHI is a preferable candidate for many PIR applications.

D. CROSSCHECKING THE RESULTS

Although the PERSON framework for evaluating PIR methods is studied and shown to be consistent with human-judgment-based evaluation, to be more confident about the results, we try to crosscheck them in this subsection. To this aim, we conduct an experiment and check if it confirms the results obtained in the previous subsection or not.

The intuition of this experiment is that the goal of any personalization method is tailoring the search results to the user's needs and preferences; Thus, roughly speaking, even if two personalization methods use different data sources to perform the personalization, they still have the same goal and if they are successful in achieving their goal (providing exactly what the user needs/likes) they must provide similar results. Therefore, if we have the results of a perfect personalization method (which ranks the result exactly according to the needs/preferences of the user) and our personalization method is also perfect, we expect our method to provide results that are similar to those of the perfect method. Thus, by measuring the similarity of the results of different personalization methods to those of a perfect method, we can obtain an estimate of the performance of each of them.

In practice, however, we do not have access to a perfect method to compare our results with. Thus, we use a presumably good method instead. But, we should note that when using a presumably good method we can not rely on the details of the results it retrieves (such as the exact rank of each result). Therefore, in this experiment, we extract some supposedly relevant results from the results of the presumably good method and evaluate other methods based on them. It is obvious that using a presumably good method instead of a perfect method may cause some errors in the evaluations. However, in this experiment, we do not aim at providing an independent evaluation method and just want to corroborate our previous results in general.

In this experiment, we consider the profile-based IRR method [39] (with $k = 20, p_0 = 0.66$) as the good method. This method reranks the original search results according to the results of a search with a query expanded based on the profile of the searching user. In this experiment, the profiles of users are constructed based on their papers (The query paper is not considered in the profile of the searcher; Thus, we only consider papers whose first authors have more than one paper as query papers). The details of our implementation of IRR are based on the implementation made in [1]. IRR is shown to outperform the baseline significantly in [39] (evaluated according to human judgments) and [1] (based on the PERSON framework). It is also the best performing profile-based method in [1] according to MAP. Thus, we assume that the method is indeed a good method and evaluate our methods based on it. It is important to note that although this method is not necessarily perfect, being good is enough to expect a higher similarity between its results and the results of better social-network-based personalization methods. Also, using a profile-based method as our basis of evaluation allows us to study if the improvements of our proposed methods reported in Table 3 are misevaluations caused by specific characteristics of the used personalization data source (co-authorship network as the social network of users) or they are supported by other data sources (profiles, here).

To extract the supposedly relevant results, we use the same settings of PERSON (e.g., the same queries and the same documents), and retrieve results based on four IR methods: *i)* IRR; *ii)* LM; *iii)* Lucene's implementation of the language modeling framework with Dirichlet prior smoothing with $\mu = 100$ (LM2); *iv)* Lucene's TF-IDF-based *ClassicSimilarity* (TF-IDF). The LM2 method is considered because the IRR method uses it for retrieval and TF-IDF is considered because the SI and Social-Textual methods use it. After retrieving top 100 results by each of these methods, for each result in IRR we consider it as relevant iff either its rank in IRR results is at least 10 places higher than its best rank in the other methods or it's among the top 10 results in IRR but is not among the top 10 in the results of the other methods (We consider the rank of a document not retrieved among top 100 as 101). In this way, we try to select relevant documents that IRR believes they deserve higher ranks than their non-personalized ranks. Indeed, we try to somewhat eliminate the biases of any of the three non-personalized methods and just consider the personalization effects in the relevant results selection, and thus be able to compare PIR methods using different non-personalized bases.[8] To evaluate PIR methods, we use

[8]Although this approach for extracting the supposedly relevant results is only one of possible approaches (e.g., we can also consider the results retrieved by all the methods in the top 10 results as relevant), finding the best approach is not our focus in this paper and can be a direction for future work.

TABLE 4. Comparison of different PIR methods according to the judgments extracted based on IRR. All the differences are significant except those specified.

#	Method	NDCG@100	MAP	P@10
1	Social-Textual	.322	$.119^{2}$	$.196^{5}$
2	SI	$.326^{1}$	.116	.181
3	PI	$\mathbf{.366}^{5,7}$	**.144**	**.206**
4	CI	.291	.094	.106
5	PCI	.364	.139	.196
6	HI	.287	.090	$.106^{4}$
7	PHI	$.365^{5}$	.137	$.197^{1,5}$

the same settings of PERSON and just replace the original PERSON judgments with the extracted relevant results.

Table 4 depicts the results. All the differences in the table are significant except the ones specified. It can be seen that the results are highly similar to those of Table 3 in term of rankings. For example, the ranking of the methods according to NDCG100 is exactly the same as the corresponding ranking in Table 3 except for a swap in the order of PCI and PHI, whose differences are very small and insignificant in both the results. Also, in term of MAP and P10, the rankings are again identical to the corresponding rankings of Table 3 except for a swap between the two baselines and also Social-Textual performing the same as PCI in term of P10 (We have considered ties in the tables to be exactly equal and ignored the subtle differences eliminated by rounding). Indeed, among 63 possible comparisons in the tables (for each pair of methods and for each measure), only four of them are different between the two tables. This supports the validity of the results of Table 3. It is worth mentioning that: *i)* The evaluation methods used in these two experiments have fundamentally different judgments. PERSON uses references for relevance judgment and the evaluation method of this experiment uses the results of a presumably good method for that. This supports that the results obtained in the previous subsection have meanings intrinsic to the performances of the PIR methods being compared and are not just consequences of the way PERSON extracts relevance judgments; *ii)* To obtain relevance judgments in the evaluation method of this experiment, we do not consider the social network of users. Thus, obtaining similar results in the evaluations conducted by it supports that the results obtained in the previous subsection are not just consequences of the characteristics of the used social network and are meaningful from the perspective of user profiles.

E. THE EFFECT OF AGGREGATION FUNCTION

To investigate the effect of the choice of aggregation function on the performances of PHI and PCI, we test them with different aggregation functions introduced in Section IV-B. The methods are chosen because they strike a good balance between efficiency and retrieval effectiveness. Table 5 shows the results. All the differences are significant except the ones specified. Moreover, the results corresponding to PCI and PHI are obtained by two separate PERSON runs. It can be seen that f_{sum} obtains the best results. This can be explained by considering that if there are multiple publishers (e.g., authors) for a document, each one can separately contribute to increasing the likelihood of the searcher being interested in that document through different social phenomena. For example, although having a close friend with some belief (or document) may highly expose the person to that belief (as considered in f_{max}), but still having an acquaintance with that belief can make the exposure higher (as considered in f_{sum}). Therefore, f_{sum} seems to be a reasonable choice in many scenarios.

F. THE EFFECT OF PARAMETER ρ

In this experiment, we study the effect of parameter $\rho \in (0, \infty)$ on the performances of PHI and PCI. Fig. 4 illustrates the results according to different measures. It can be seen that the best performances are obtained approximately when $\rho \in [.5, 1]$. This confirms our claim that under certain conditions (See Section IV-B) extensive parameter tuning is not required for the proposed PIR method and some default value, e.g., 0.5 or 1.0 (depending on the amount of personalization desired), can be used for ρ.

G. RUNNING TIME ANALYSIS

Although we provided time complexities for the proposed measures in the previous section, in calculating the complexities, we made some simplifying assumptions. Thus, it is worthwhile to study their performances in a real dataset. In this experiment, we focus on HI and PHI, which consider the full hierarchical structure of the network in computing interestedness values and compare them with PI. It is noteworthy that although PI can be computed via CSR, it indeed does not consider the hierarchical structure of the network and to compute $I(u_t|u_s)$ for all u_s and u_t, it requires computing $PPR(u_t, \{u_s\}; S)$ for all u_s and u_t, in which S is the whole graph. Thus, by comparing HI and PHI with it, we can obtain an understanding of how considering the hierarchical structure of network can affect the performance.

The experiments of this section are performed on a Dell Latitude laptop with a 64-bit Intel(R) Core(TM) i7-3540M CPU, 16GB of RAM and an SSD hard drive as the main storage device, running Ubuntu 18.04.2. As the database, we use MySQL 5.7.26. In the experiments, we use version 7.5.0 of Lucene.

For HI and PHI, we assume that computing PPR values is performed offline and the pre-computed values are then used at the query-time to compute the interestedness measures. Columns 3 and 4 in Table 6 show the number of PPR values computed and the time it took to compute them in our dataset, respectively. Also, if we want to similarly pre-compute exact PPR values in PI, we need to obtain $PPR(u_t, \{u_s\}; S)$ for all pairs of vertices. This requires computing V^2 (V is the number of vertices) values, which is shown in the table. Since pre-computing the PPR values for PI is costly, for PI, we report an estimate of the time required to compute the PPR values in column 4 instead of the actual time. To estimate

TABLE 5. Comparison of different aggregation functions. All the differences are significant except those specified.

		PHI			PCI		
#	Method	NDCG@100	MAP	P@10	NDCG@100	MAP	P@10
1	f_{max}	.505	.326	.170	.507	.328	.171
2	f_{sum}	**.510**^{3}	**.332**^{3}	**.172**^{3}	**.513**^{3}	**.333**^{3}	.171^1
3	f_{avg}	**.510**	.331	**.172**	.510	.330	**.172**^{1,2}
4	f_{most}	.490	.316	.164	.493	.319	.165

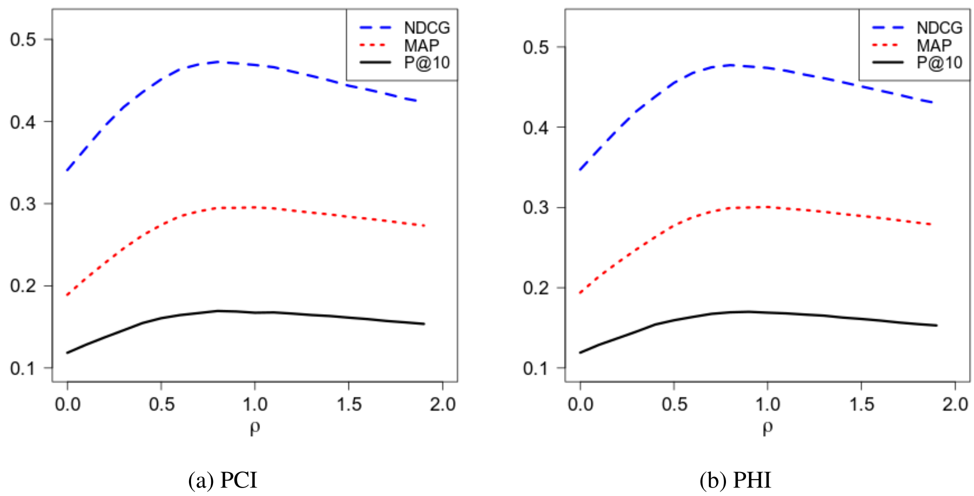

FIGURE 4. The performances of PCI and PHI w.r.t. ρ.

TABLE 6. Comparison of the costs of using different PIR methods. Columns 3 and 4 show the number of PPR values required to be computed and the time it took to compute them, respectively. The value marked with * is an estimation. Column 5 shows an estimate of the time spent for personalizing the retrieval for 500 queries.

#	Method	# of PPR values	Pre-computing time of PPRs (s)	Elapsed time (s)
1	HI	89 900 080	75	64.37
2	PHI	493 414 784	25 156	357.53
3	PI	14 899 375 969	911 945*	4 126.50

the time required, we calculated the time it took to compute PPR values for 100 randomly-chosen u_s and multiplied the time by $\frac{V}{100}$. It can be seen that the PPR values for HI and PHI can be computed in an affordable time and stored in an affordable amount of memory. But, computing and storing all the required PPR values for PI takes a lot of time and memory and may be infeasible for many practical purposes. Thus, for PI, we do not pre-compute the PPR values and compute the required values at the query-time.

For performing retrieval by HI and PHI, we load the pre-computed PPR values into the main memory at the start-up of the retrieval system for each cluster S satisfying $|parent(S)| < 5000 \times |S|$. For clusters not satisfying the condition, the values are stored in the database and are queried when needed.[9] We also load $PPR(S, S; parent(S)) = \sum_{u \in S} PPR(u, S; parent(S))$ for all clusters S into the main memory.

We then perform a retrieval for 500 queries with each of the methods separately and measure the elapsed times. Since our code performs tasks other than computing the required interestedness measures like textual retrieval and calculating retrieval measures, we also measure the elapsed time for the LM method and subtract it from the elapsed times of the methods being compared. This way we obtain an estimate of the time spent for personalizing the retrieval by each of the methods. The results are depicted in column 5 of Table 6. The results show that, on average, for each query almost $0.13s$, $0.72s$, and $8.25s$ are spent for personalizing the search by HI, PHI, and PI, respectively. It can be seen that PI, which does not consider the clusters' hierarchy of the network, is considerably slower than HI and PHI. The amount of time required for personalization by PI may not be acceptable in many of applications. According to the time complexities mentioned in Table 1, it is expected that the difference between the required time of PI and that of HI/PHI is even higher in larger datasets.

Moreover, there are some unnaturalities in the hierarchy of clusters extracted by the clustering algorithm we employed that have considerably increased the costs of using HI and PHI. For example, at the second level of the extracted hierarchy, the root cluster with 122 063 vertices is partitioned into a 60 412-vertex cluster, a 59 447-vertex cluster and 333 clusters

[9] For brevity, we do not mention some details and/or exceptions here.

TABLE 7. Comparison of the costs of using different PIR methods. Columns 3 and 4 show the average number of PPR values required to be computed and the average time it took to compute them per vertex (u_s), respectively. The value marked with * is an estimation. Column 5 shows an estimate of the time spent for personalizing the retrieval for 500 natural queries.

#	Method	# of PPR values	Pre-computing time of PPRs (s)	Elapsed time (s)
1	HI	16.1	0.0006	58.60
2	PHI	441.9	0.02	70.52
3	PI	122 063.0	7.47*	4 339.98

with an average number of vertices of 6.62, with a minimum of 1 (a total of 2 204 vertices); Or, it produces a cluster with 8 989 vertices that is not further divided into smaller clusters. These unnaturalities can increase the number of PPR scores to be calculated in HI and PHI considerably and reduce their performances. We claim that if a clustering algorithm is employed that extracts a natural hierarchy, we can obtain a much higher performance with PHI. Here, by natural we mean that the size of each cluster in the hierarchy is at least $1/k$ of the size of its parent.[10] We call k the unnaturality parameter and consider $k = 100$ here. We also call a vertex a *natural vertex* iff all of the cluster ancestors of it have at least $1/k$ of the number of vertices of their parents.

We perform another experiment to check the validity of the claim. We perform a retrieval for 500 queries with each of the methods again but we only consider queries whose corresponding searchers are natural. In other words, we consider a paper as a query paper only if its first author (searcher) is a natural vertex. We call the queries extracted from such query papers *natural queries*. Since in PHI we add an extra layer to the hierarchy, natural queries according to the hierarchy used in PHI are a subset of natural queries according to the hierarchy used in HI. Thus, in this experiment, we use queries that are natural according to the hierarchy used in PHI.

In this experiment, we only pre-compute the PPR values required for computing $CSR(*|u_s)$ when u_s is a natural vertex (Note that natural vertices for PHI are a subset of natural vertices for HI). Because in this experiment we have not computed PPR values for all u_s, to make the comparison between the results of the different methods possible, we divide the number of PPR values computed and the time it took to compute them by the number of vertices for which we computed PPR values. The average number of PPR values and the average time it took to compute them per vertex (u_s) for the methods are reported in columns 3 and 4 of Table 7, respectively.[11][12] If we calculate the same measures for HI and PHI in the previous experiment, they become 736.5 and 0.05 for HI and 4 042.3 and 0.2 for PHI. This confirms that the unnaturalities in the output of the clustering algorithm had imposed much overhead to HI and PHI (For example, about PHI, only inside the undivided cluster with 8 989 vertices, we had to compute $8\,989^2 = 80\,802\,121$ PPR values).

After pre-computing the PPRs, we perform a retrieval for 500 natural queries with each of the methods. It is noteworthy that in this experiment since all the pre-computed PPR values for PHI can be loaded into the main memory, we do not use a database.[13] The elapsed times are reported in column 5 of Table 7. The results show that, on average, for each query almost $0.12s$, $0.14s$, and $8.68s$ are spent for personalizing the search by HI, PHI, and PI, respectively. It can be seen that in this experiment the spent time for PHI is almost $1/5$ of the spent time in the previous experiment. This supports our claim that if the employed clustering algorithm extracts a natural hierarchy, we can obtain a much higher performance with PHI compared to the performance obtained in the previous experiment. Moreover, if we had considered a lower unnaturality parameter, we might have obtained an even better performance.

Overall, the results of this subsection suggest that in practice HI/PHI are efficient (compared to, e.g., PI) in term of computational/storage cost, especially when the hierarchical structure of the network does not have much unnaturalities.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, an interestedness measure and several variations of it were proposed based on two explanations of the dynamics of information seeking/searching behavior in a society. The intuition behind the measures is to try to mimic a searcher's behavior in the real world to estimate the interestedness of a user in another user. We proved that the interestedness measures are probability functions. We also proposed a PIR method based on the measures, in which we aggregate interestedness measures and combine them with textual scores. We evaluated the method based on different interestedness measures and compared them in term of retrieval performance and asymptotic cost of computation and storage. In summary, our analyses suggest that the PHI method is a good candidate for social-network-based personalization

[10]For PHI (not HI), since we add an extra layer to the hierarchy (single-node clusters), being natural also means that the maximum possible size for an undivided cluster is k.

[11]The results mentioned for PI are based on the estimations made in the previous experiment.

[12]It is noteworthy that for HI and PHI the average number and average time might have been even lower if all of the hierarchy was natural and we computed PPR values for all u_s because in that case the costs of PPR computations in the clusters higher in the hierarchy would have been divided by a more number of vertices.

[13]For PHI, since we have not pre-computed all of the PPR values required for computing $CSR(*|u_s)$ for all u_s and only pre-computed PPR values for natural vertices (u_s), we also load some dummy PPR values into the main memory so that looking up PPR values at the query-time is not much faster than in a real situation. For each cluster with k natural sub-clusters and containing n vertices, we added $\min(100, n) - k$ dummy values into the mapping of cluster to PPR value (See NodePPR(c, n) in Algorithm 1) for each of the n vertices. We do not add dummy values for HI since most of the vertices are natural in HI, and thus the required PPR values for them are already computed.

because of its retrieval effectiveness and computational efficiency. We also investigated the effect of the weight parameter for combining social and textual scores. The results show that due to the characteristics of the proposed interestedness measures, under a small number of publishers assumption, extensive parameter tuning can probably be avoided and some default value (e.g., 0.5 or 1.0 depending on the amount of personalization desired) be used instead.

In the following, we mention some directions for future work:

- In the hierarchical explanation of searching behavior, we assumed that individuals in a society form a hierarchy and based on the assumption we proposed CSR. In reality, however, human societies are much more complex than a single hierarchical structure. Specially, humans are not one-dimensional beings and in each dimension different social structures may form. For example, the structure of a society from a political perspective may vary greatly from its structure from a geographical perspective. Thus, extending the hierarchical explanation to take different dimensions into account is a direction for future work, which may be then used to extend CSR. Considering other structures such as directed acyclic graphs is also another direction for future work;
- In this paper, we proposed CSR based on PPR. Using other centrality measures, or more broadly, other measure in CSR can be considered in future;
- The focus in this paper was on social networks with a single relationship type between users. Extending our method to be able to use heterogeneous networks with multiple types of relationships is a direction for future work. A possible solution is to use multilayer-network-based versions of PageRank (e.g., see [40], [41]) in CSR;
- The applications of the introduced interestedness measures are not limited to PIR. Specially, PHI may be employed in other applications such as recommender systems and social networks analysis as a low-cost alternative to PPR. Furthermore, in PIR, it may be used in other personalization approaches as an efficient estimation of the similarity of users;
- Evaluating the proposed PIR method with user-based studies and in real-world applications is also another direction for future work;
- The method used for crosschecking our results in Section VI-D may be usable as an independent indirect evaluation method. Generalizing the method and studying its validity as an independent evaluation framework is also a direction for future work.

APPENDIX A
PROOF OF CSR BEING A PROBABILITY FUNCTION

In order for CSR to be a probability function over the subsets of users (We denote the set of all users by Ω), it must satisfy the following axioms:

- Non-negativity: $CSR(S|u_s) \geq 0$ for every $S \subseteq \Omega$;
- Additivity: For every disjoint subsets of users S and T, $CSR(S|u_s) + CSR(T|u_s) = CSR(S \cup T|u_s)$;
- Unit measure: $CSR(\Omega|u_s) = 1$.

It is obvious that the non-negativity axiom holds. Also, according to our definition of $CSR(S|u_s)$ based on $CSR(u|u_s)$, it is trivial that the additivity axiom holds. Thus, to prove CSR is a probability function, we only need to show that the unit measure axiom holds. In the rest of this proof, we assume $C_1, C_2 \ldots, C_r$ are the cluster ancestors of u_s.

We use strong induction on the height of the hierarchy, h, to prove $CSR(\Omega|u_s) = \sum_{u\in\Omega} CSR(u|u_s) = 1$. The axiom obviously holds for the base case of $h = 2$ (i.e., a set of clusters each containing some users and a root cluster as the parent of the clusters) since $\sum_{u\in\Omega} CSR(u|u_s) = \sum_{u\in\Omega} PPR(u, C_2; C_1) = 1$.

For the inductive step, we show that if the axiom holds for all hierarchies with height $\leq h$, it also holds for hierarchies with height $h+1$. For an arbitrary hierarchy with height $h+1$, illustrated in Fig. 5, we have

$$\begin{aligned} CSR(\Omega|u_s) &= \sum_{u\in\Omega} CSR(u|u_s) = \sum_{u\in C_2} CSR(u|u_s) \\ &\quad + \sum_{u\notin C_2} CSR(u|u_s) \\ &= PPR(C_2, C_2; C_1) \times \underbrace{\sum_{u\in C_2} \frac{CSR(u|u_s)}{PPR(C_2, C_2; C_1)}}_{X} \\ &\quad + \sum_{u\notin C_2} PPR(u, C_2; C_1) \\ &\stackrel{1}{=} PPR(C_2, C_2; C_1) \times 1 + \sum_{u\notin C_2} PPR(u, C_2; C_1) \\ &= \sum_{u\in\Omega} PPR(u, C_2; C_1) = 1, \end{aligned} \tag{13}$$

in which $\stackrel{1}{=}$ holds based on the this argument: If we consider H_1 in Fig. 5 as a new hierarchy, based on the definition of CSR (Eq. (3)), X in the above equation is equivalent to $\sum_u CSR(u|u_s)$ in the new hierarchy; According to the induction hypothesis, this equals 1 since the height of the new hierarchy is less than or equal to h.

Therefore, the axiom holds for all hierarchies with height $h + 1$, and thus by induction, $CSR(\Omega|u_s) = 1$ holds for all proper hierarchies. $\square$

APPENDIX B
COST OF COMPUTING CSR SCORES

To calculate the cost of computing $CSR(u_t|u_s)$ for all values of u_t and u_s, we assume that the hierarchy is roughly balanced and with a small branching factor c. Also, considering that graphs with more vertices tend to have lower PageRank scores, we assume a lower tolerance level τ for larger networks ($\tau = \frac{1}{1000\ V}$ in our experiments). Therefore, assuming that computing PPR scores requires $\frac{\log \tau}{\log \alpha}$ iterations (See [42]) and also assuming $V = \mathcal{O}(E)$, we consider the

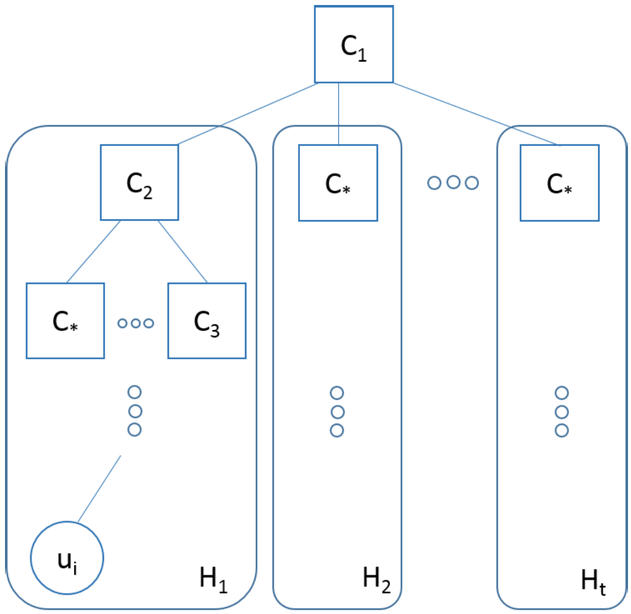

FIGURE 5. Hierarchy of users and clusters.

cost of computing PPR scores in a graph with V vertices and E edges as $\mathcal{O}(E \log V)$. Moreover, we assume that when a cluster with E edges is partitioned into c sub-clusters, each of the sub-clusters has $\mathcal{O}(E/c)$ edges.

Considering the above assumptions, for computing PPR scores in a cluster with depth i (i.e., at level $i+1$ of the hierarchy), a cost of $\mathcal{O}(c \times \frac{E}{c^i} \log \frac{V}{c^i})$ is required (one PPR computation for each of its sub-clusters). Thus, considering that there are $\mathcal{O}(c^i)$ clusters with depth i and the height of the hierarchy is of $\mathcal{O}(\log_c V)$, the cost of computing the CSR scores is

$$\begin{aligned}
&\mathcal{O}\Big(\sum_{i=0}^{\log_c V} c^i \times c \times \frac{E}{c^i} \log \frac{V}{c^i}\Big) \\
&= \mathcal{O}\Big(cE \times \sum_{i=0}^{\log_c V} \log \frac{V}{c^i}\Big) \\
&= \mathcal{O}\Big(cE \log \frac{V^{\log_c V+1}}{c^{\Sigma_{i=0}^{\log_c V} i}}\Big) \\
&= \mathcal{O}\Big(cE \log V^{\log_c V+1} - cE \log c^{\frac{\log_c V \times (\log_c V+1)}{2}}\Big) \\
&= \mathcal{O}\Big(cE \log V \times (\log_c V + 1) \\
&\quad - c\,E \log c \times \frac{\log_c V \times (\log_c V+1)}{2}\Big) \\
&= \mathcal{O}\Big(cE \log V \times (\log_c V + 1) - cE \frac{\log V \times (\log_c V + 1)}{2}\Big) \\
&= \mathcal{O}\Big(\frac{cE}{2} \log V \times (\log_c V + 1)\Big), \qquad (14)
\end{aligned}$$

which is effectively $\mathcal{O}\big(E \log^2 V\big)$.

ACKNOWLEDGMENT

Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

REFERENCES

[1] S. A. Tabrizi, A. Shakery, H. Zamani, and M. A. Tavallaei, "PERSON: Personalized information retrieval evaluation based on citation networks," *Inf. Process. Manage.*, vol. 54, no. 4, pp. 630–656, 2018.

[2] A. Younus, C. O'Riordan, and G. Pasi, "A language modeling approach to personalized search based on users' microblog behavior," in *Advances in Information Retrieval*. Cham, Switzerland: Springer, 2014, pp. 727–732.

[3] S. Samarawickrama, S. Karunasekera, A. Harwood, and R. Kotagiri, "Search result personalization in Twitter using neural word embeddings," in *Big Data Analytics and Knowledge Discovery*. Cham, Switzerland: Springer, 2017, pp. 244–258.

[4] M. R. Bouadjenek, H. Hacid, M. Bouzeghoub, and A. Vakali, "PerSaDoR: Personalized social document representation for improving Web search," *Inf. Sci.*, vol. 369, pp. 614–633, Nov. 2016.

[5] D. Zhou, X. Wu, W. Zhao, S. Lawless, and J. Liu, "Query expansion with enriched user profiles for personalized search utilizing folksonomy data," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 7, pp. 1536–1548, Jul. 2017.

[6] M. R. Bouadjenek, H. Hacid, and M. Bouzeghoub, "Personalized social query expansion using social annotations," in *Transactions on Large-Scale Data- and Knowledge-Centered Systems XL*. Berlin, Germany: Springer, 2019, pp. 1–25.

[7] B. Xu, H. Lin, Y. Lin, and Y. Guan, "Integrating social annotations into topic models for personalized document retrieval," *Soft Comput.*, pp. 1–10, Apr. 2019.

[8] L. Gou, H. Chen, J. Kim, L. Zhang, and C. L. Giles, "SNDocRank: A social network-based video search ranking framework," in *Proc. 19th Int. Conf. World Wide Web (WWW)*, 2010, pp. 367–376.

[9] A. Khodaei and C. Shahabi, "Social-textual search and ranking," in *Proc. 1st Int. Workshop Crowdsourcing Web Search*, 2012, pp. 3–8.

[10] A. Khodaei, S. Sohangir, and C. Shahabi, "Personalization of Web search using social signals," in *Recommendation and Search in Social Networks*. Cham, Switzerland: Springer, 2015, pp. 139–163.

[11] J. Teevan, S. T. Dumais, and E. Horvitz, "Personalizing search via automated analysis of interests and activities," in *Proc. 28th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, New York, NY, USA, 2005, pp. 449–456.

[12] M. R. Ghorab, D. Zhou, A. O'connor, and V. Wade, "Personalised information retrieval: Survey and classification," *User Model. User-Adapted Interact.*, vol. 23, no. 4, pp. 381–443, 2013.

[13] M. R. Bouadjenek, H. Hacid, and M. Bouzeghoub, "Social networks and information retrieval, How are they converging? A survey, a taxonomy and an analysis of social information retrieval approaches and platforms," *Inf. Syst.*, vol. 56, pp. 1–18, Mar. 2016.

[14] B. Steichen, H. Ashman, and V. Wade, "A comparative survey of personalised information retrieval and adaptive hypermedia techniques," *Inf. Process. Manage.*, vol. 48, no. 4, pp. 698–724, 2012.

[15] P. Brusilovsky, B. Smyth, and B. Shapira, "Social search," in *Social Information Access*. Cham, Switzerland: Springer, 2018, ch. 7, pp. 213–276.

[16] S. Han and D. He, "Network-based social search," in *Social Information Access*. Springer, 2018, ch. 8, pp. 277–309.

[17] J. Vosecky, K. W.-T. Leung, and W. Ng, "Collaborative personalized Twitter search with topic-language models," in *Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, New York, NY, USA, 2014, pp. 53–62.

[18] P. Yin, W.-C. Lee, and K. C. Lee, "On top-k social Web search," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, New York, NY, USA, 2010, pp. 1313–1316.

[19] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the Web," Stanford Digit. Library Technol. Project, Stanford, CA, USA, Tech. Rep. 1999-66, 1998. [Online]. Available: http://ilpubs.stanford.edu:8090/422/

[20] E. A. Leicht, P. Holme, and M. E. J. Newman, "Vertex similarity in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 73, no. 2, p. 026120, 2006.

[21] A. Dridi and Y. Slimani, "SonetRank: Leveraging social information for personalized search," *Social Netw. Anal. Mining*, vol. 7, no. 1, p. 16, 2017.

[22] L. Kelly, J. Kim, and D. Elsweiler, "Workshop on evaluating personal search," *ACM SIGIR Forum*, vol. 45, no. 2, pp. 81–86, Dec. 2011.

[23] G. Pasi, G. J. F. Jones, S. Marrara, C. Sanvitto, D. Ganguly, and P. Sen, "Overview of the CLEF 2017 personalised information retrieval pilot lab (PIR-CLEF 2017)," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction*. Cham, Switzerland: Springer, 2017, pp. 338–345.
[24] E. Vicente-López, L. M. de Campos, J. M. Fernández-Luna, J. F. Huete, A. Tagua-Jiménez, and C. Tur-Vigil, "An automatic methodology to evaluate personalized information retrieval systems," *User Model. User-Adapted Interact.*, vol. 25, no. 1, pp. 1–37, 2015.
[25] G. E. P. Box, "Robustness in the strategy of scientific model building," in *Robustness in Statistics*. New York, NY, USA: Academic, 1979, pp. 201–236.
[26] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," in *Proc. 7th Int. Conf. World Wide Web*, 1998, pp. 107–117.
[27] F. W. Wolek, "Preparation for interpersonal communication," *J. Amer. Soc. Inf. Sci.*, vol. 23, no. 1, pp. 3–10, 1972.
[28] E. Ravasz and A.-L. Barabasi, "Hierarchical organization in complex networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 67, p. 026112, Mar. 2003.
[29] C. Zhai, "Statistical language models for information retrieval a critical review," *Found. Trends Inf. Retr.*, vol. 2, no. 3, pp. 137–213, 2008.
[30] Q. Liu, Z. Li, J. C. Lui, and J. Cheng, "PowerWalk: Scalable personalized PageRank via random walks with vertex-centric decomposition," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, New York, NY, USA, 2016, pp. 195–204.
[31] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "ArnetMiner: Extraction and mining of academic social networks," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2008, pp. 990–998.
[32] S. A. Tabrizi, A. Shakery, M. Asadpour, M. Abbasi, and M. A. Tavallaie, "Personalized pagerank clustering: A graph clustering algorithm based on random walks," *Phys. A, Statist. Mech. Appl.*, vol. 392, no. 22, pp. 5772–5785, 2013.
[33] D. Edler and M. Rosvall. (2018). *The MapEquation Software Package*. [Online]. Available: http://www.mapequation.org
[34] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech., Theory Exp.*, vol. 2008, no. 10, p. P10008, 2008.
[35] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of IR techniques," *ACM Trans. Inf. Syst.*, vol. 20, no. 4, pp. 422–446, 2002.
[36] J. M. Ponte and W. B. Croft, "A language modeling approach to information retrieval," in *Proc. 21st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 1998, pp. 275–281.
[37] C. Zhai and J. Lafferty, "A study of smoothing methods for language models applied to information retrieval," *ACM Trans. Inf. Syst.*, vol. 22, no. 2, pp. 179–214, Apr. 2004.
[38] M. R. Bouadjenek, H. Hacid, M. Bouzeghoub, and A. Vakali, "Using social annotations to enhance document representation for personalized search," in *Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, New York, NY, USA, 2013, pp. 1049–1052.
[39] L. M. de Campos, J. M. Fernández-Luna, J. F. Huete, and E. Vicente-Lopez, "Using personalization to improve XML retrieval," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1280–1292, May 2014.
[40] S. Lee, S. Park, M. Kahng, and S.-G. Lee, "PathRank: A novel node ranking measure on a heterogeneous graph for recommender systems," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, New York, NY, USA, 2012, pp. 1637–1641.
[41] X. Tu, G. Jiang, Y. Song, and X. Zhang, "Novel multiplex PageRank in multilayer networks," *IEEE Access*, vol. 6, pp. 12530–12538, 2018.
[42] A. N. Langville and C. D. Meyer, "Deeper inside PageRank," *Internet Math.*, vol. 1, no. 3, pp. 335–380, 2004.

SHAYAN A. TABRIZI received the B.Sc. and M.Sc. degrees in computer engineering from the University of Tehran, where he is currently pursuing the Ph.D. degree in computer engineering. His research interests include information retrieval, text mining, social networks, and machine learning.

AZADEH SHAKERY received the Ph.D. degree in computer science from the University of Illinois at Urbana–Champaign, in 2008. She is currently an Associate Professor with the School of Electrical and Computer Engineering, College of Engineering, University of Tehran. Her research interests include text information management, information retrieval, and text mining.

MOHAMMAD ALI TAVALLAEI received the B.Sc. and M.Sc. degrees in software engineering from the University of Tehran, Tehran, Iran, in 2010 and 2013, respectively, where he is currently a Software Architect and the Manager with the Institute of Applied Intelligent Systems (IAIS). His research interests include social networks, information retrieval, text mining, and machine learning.

MASOUD ASADPOUR (M'05) was born in Lamerd, Iran, in 1975. He received the B.Sc. degree in computer software engineering from the Sharif University of Technology, Tehran, in 1997, the M.Sc. degree in AI and robotics from the University of Tehran, in 1999, and the Ph.D. degree in machine learning and collective robotics from EPFL, Switzerland, in 2006. He was a Researcher with the Institute for Research in Fundamental Sciences (IPM), Iran, from 1998 to 2001 and from 2010 to 2012. He was a Postdoctoral Researcher with the Biologically Inspired Robotics Group (BIRG), EPFL, in 2007. In 2008, he joined the Robotics and AI Group, Faculty of Electrical and Computer Engineering, University of Tehran, as a Faculty Member and a found Social Networks Laboratory. He is also an Adjunct Member of Software and IT groups. His research interests include social networks, collective robotics, machine learning, and bio-inspired robotics.