Discovering Social Media Experts by Integrating Social Networks and Contents

Zhao Zhang  Bin Zhao  Weining Qian  Aoying Zhou

Institute of Massive Computing, Software Engineering Institute, Shanghai Key Laboratory of Trustworthy Computing, East China Normal University
Shanghai 200062, P.R. China
Emails: {zhzhzhang,wnqian,ayzhou}@sei.ecnu.edu.cn zhaobin@nju.edu.cn

Abstract

Social media are media contributed by common users and distributed in social networks. There may exist thousands of answers to a single question provided by different users. However, it is difficult to evaluate the authority of a user to a specific question. We introduce a new method for identifying experts in social media. Both the structure of the social network and content of the media are used in a unified graph model for evaluation of users. Extensive experiments show that our approach can determine authority experts on specific domains.

1 Introduction

Social media contains huge volume of information. However, it is difficult to filter noise and low quality data out from social media. To find experts to a specific topic and then collect content contributed by experts to the topic is a natural way to acquisition of high-quality knowledge. It has proved to be an effective, and attracts much attention in social media mining research [10, 6]. However, existing methods are usually designed for a specific type of social media, such as blogs [8], online forums [19], and microblogs [9]. The structure of social networks and contents of the media are considered separately. We take another approach, which aims at the general problem of expert finding in social media. It relies on a unified graph model. The expert finding problem is then solved via a mutual reinforcement process in the network.

In this paper, our work provides comprehensive expertise analysis in general social media. The following two questions should be answered to solve the problem.

Who are experts to a specific topic? We call this task as expert finding. That is to say, given a topic query (describing the area in which expertise is being sought), a ranked list of user names is returned.

Which topics is she an expert in? We call this task as expert profile finding. In other words, given a user query (describing the user in which expertise is being sought), a ranked list of topics is returned.

There are several challenges to expertise analysis. First, social media is the mixture of social network and content. A typical social media site is shown in Figure 1. It contains several posts and part of their comments. A set of users build relationships by commenting the post. In other words, social networks and contents are mutually reinforcing. That is to say, if a user is an expert of a specific area, the users who has strong social relationship have high probability to be experts on the same area. So we need to find a model to represent this indirect relationship. In this paper, we propose a novel graph model to represent contents and social networks simultaneously.

Second, the lists of friendship and followship embed much noise. Since there are a large number of inactive users in social media sites, we cannot rely on friendship list to build social networks for social media sites. We have to extract active social relationships among users. In this paper, we only consider active social networks built by posting and commenting among users.

1.1 Our contributions

We made the following contributions to attack the problem of expertise search on social media in this paper.

• A tripartite graph model is introduced, which simultaneously represents features of social networks and contents in social media. This graph model makes our analysis simple and convenient.

• Active social relationships are used for expertise analysis. Since lists of friendship and followship embed much noise, only active social networks built by posting and commenting among users are used.

• Expert and expert profile finding are formally defined. Social media has been saturated with a large number of human generated contents. There exist many folk experts in social media. In this paper, we present formal definitions about expert and expert profile in social media sites.

• A random walk with restart (RWR) algorithm for tripartite graph is presented. Many researches showed RWR is a good correlation measurement method between nodes in graph. However, the main challenge of RWR algorithm is its efficiency. In this paper, we present an improved RWR algorithm for large tripartite graph based on star schema.

• Extensive experiments over real life data sets are conducted. We compare our algorithm with the
initial RWR on two real data sets. Our experimental results (see section 6) show significant benefits in time consumption.

1.2 Paper organization

The rest of this paper is organized as follows. The problem of expertise analysis is formally defined in Section 2. Section 3 introduces the random-walk-with-restart (RWR) algorithm. In Section 4, a star-schema-based optimization technique for RWR in tripartite graph is presented. The procedures for expert and expert profile finding are introduced in Section 5. Experimental results are shown and analyzed in Section 6. The related work are introduced in Section 7, followed which Section 8 is for concluding remarks.

2 Problem statement

A unified tripartite graph model that represents both content and structure of social networks is introduced in this section. It is the basis of expert and expert profile finding, which is introduced in detail in Section 5. The symbols and notations that used are listed in Table 1.

2.1 Social media preliminaries

There are two types of entities in social media, i.e. users and pieces of information. Pieces of information may contain multimedia content. In this paper, only content of text is considered. Both text content and other types of multimedia content can be handled via semantic annotation.

Table 1: Notations used in this paper

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Definition and description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G$</td>
<td>tripartite graph</td>
</tr>
<tr>
<td>$S_i$</td>
<td>star schema</td>
</tr>
<tr>
<td>$G_s$</td>
<td>star graph is composed by star schema</td>
</tr>
<tr>
<td>$G_q$</td>
<td>query graph appended query node on star graph $G_s$</td>
</tr>
<tr>
<td>$V_i$</td>
<td>the ith component of vertices of the tripartite graph. $i = 1, 2, 3$</td>
</tr>
<tr>
<td>$E_i$</td>
<td>the ith component of edges of the tripartite graph. $i = 1, 2, 3$</td>
</tr>
<tr>
<td>$</td>
<td>V_i</td>
</tr>
<tr>
<td>$Q_e$</td>
<td>expert query is composed by the set of terms</td>
</tr>
<tr>
<td>$Q_p$</td>
<td>expert profile query is composed by the set of users</td>
</tr>
<tr>
<td>$E_t$</td>
<td>the set of users which is the query result of $Q_e$</td>
</tr>
<tr>
<td>$P_e$</td>
<td>the set of terms which is the query result of $Q_e$</td>
</tr>
<tr>
<td>$r(v_i, v_j)$</td>
<td>relevance score based on RWR between $v_i$ and $v_j$</td>
</tr>
<tr>
<td>$W$</td>
<td>a transition matrix which is column normalized</td>
</tr>
<tr>
<td>$(1-c)$</td>
<td>random particle that starts from node $1$</td>
</tr>
<tr>
<td>$\vec{v}_i$</td>
<td>a vector that the i-th element is 1 and other elements all are 0</td>
</tr>
<tr>
<td>$\vec{r}_i$</td>
<td>an vector which has $</td>
</tr>
</tbody>
</table>
Usually, users are connected via social networks. Relationships between users include, for example, followships in Twitter, or friendship in Facebook. However, these types of relationships are relatively static. We argue that active social networks are more important than static relationships. Here, active social networks are social networks in which relationships capture the interactions between users. Such kind of dynamic relationships include retweeting in Twitter, like in Facebook, and commenting in online forums.

Thus, there are four types of information that should be included in the unified model:

**Users** A user is essentially an identifier identified as the author of any pieces of information or entities involved in a social network.

**Texts** Text is a piece of information in text form. It is used to represent the original form of content contributed by users.

**Active social networks** An active social network is the social network that captures dynamic relationships implying interactions between users.

**Terms** A term is a semantically meaningful word or phrase that represent the semantics of texts. Note that a text may be annotated by several terms, while a term may be used to annotate multiple texts.

An expert query is a set of terms, while the result should be a ranked list of experts who are good at topics defined by those terms. The list is ranked in descendant order based on the goodness of experts. An expert is also a user. It is formally defined in Definition 1.

**Definition 1** An expert query $Q_e$ is a set of terms: \( \{t_1, t_2, \ldots, t_i\} \), in which each $t_i$ is a term. The result of $Q_e$, denoted as $R_{Q_e}$, is \(u_1, u_2, \ldots, u_k >\) satisfying that $r^e(u_i, Q_e) \geq r^e(u_{i+1}, Q_e)$. Here, $r^e(u_i, Q_e)$ is a score function that denotes the possibility of user $u_i$ being experts on the domain defined by $Q_e$.

Similarly, an expert profile query is a set of users (experts). The result should be a ranked list of terms which denotes the domain(s) those experts are good at. It is formally defined in Definition 2.

**Definition 2** An expert profile query $Q_p$ is a set of users \( \{u_1, u_2, \ldots, u_n\} \), in which each $u_i$ is a user. The results of $Q_p$, denoted as $R_{Q_p}$, is \(t_1, t_2, \ldots, t_i >\) satisfying that $r^p(t_i, Q_p) \geq r^p(t_{i+1}, Q_p)$, in which $r^p(u_i, Q_p)$ is a score function that denotes the authority degree of user group $Q_p$ on domain denoted by term $t_i$.

In real-life applications, the expert and expert profile queries are top-$k$ queries. Thus, only top-$k$ users and terms with highest score function values are to be returned.

Thus, the essence of the problem is a reasonable definition of score functions $r^e()$ and $r^p()$, and efficient search of $u_i$ and $t_j$ with top-$k$ values given queries and score functions.

**2.2 Conventional model for modeling contents**

A simple yet natural way for expert and expert profile finding is to directly analyze social media content, e.g. texts and terms. Bipartite graphs, as it is shown in Figure 2, are often used to model relationships between users and terms. The vertices on the left are users, while those on the right are terms. The edges are weighted, in which weights are term frequency of a user mentions a specific term. Bipartite graphs are often used in text mining. Since the structure of social networks, whether static ones or dynamic ones, are not used, we argue that this model may not capture the important features of users’ expertise.

**2.3 Tripartite graph model**

Intuitively, users and terms are not directly connected. Terms are actually associated with users’ actions, such as posting, commenting, or retweeting. Thus, we extend the conventional bipartite graph model to a tripartite graph model. A tripartite graph have three types of vertices and two types of edges. The first type are used to represent users, while the second type is for their actions, and the third one is for terms. It is formally defined in Definition 3.

**Definition 3** A tripartite graph $G$ is defined as a quadruple $\{V_1, V_2, V_3, E_1, E_2\}$, in which $V_1$ is the set of users \( \{u_1\} \), $V_2$ is the set of texts \( \{p_1\} \), and $V_3$ is the set of terms \( \{t_1\} \). $E_1$ is the set of edges \( \{(u_1, p_1)\} \subseteq V_1 \times V_2 \) while $E_2$ is the set of edges \( \{(p_1, t_1)\} \subseteq V_2 \times V_3 \).

There is an edge \((u_1, p_1)\) in $E_1$ if that user $u_1$ contributes the piece of information $p_1$, while the semantics of $p_1$ is represented by terms that are connected to $p_1$ by edges in $E_2$. Thus, actions of users can be represented by this tripartite graph. Furthermore, for dynamic relationships between users, such as commenting, retweeting, and etc., users interact with each other are both connected to the same $p_1$, and thus establish an indirect relationship in the tripartite graph. Thus, the active social network is successfully embedded into our tripartite graph.

A tripartite graph corresponding to the posts in Figure 1 is shown in Figure 3.
and expert profile queries are not implied intuitively. The problem of expert and expert profile finding are essentially ranking correlation score between users and terms.

Several link-based relevance functions have been proposed in graph, including simrank [5] and random walk with restart (RWR) [4]. SimRank can compute relevance of a node-pair \((a, b)\) based on similarity of multi-step neighborhoods. RWR can simultaneously obtain relevance scores between given node \(a\) and other nodes except for node \(a\) in a graph. Considering efficiency and effectiveness of the algorithm in large graphs, we adopt RWR approach in this paper.

We define \(r^e(u_i, Q_e)\) as the sum of \(r^e(u_i, t_j)\) where \(t_j \in Q_e\) are terms in the query, i.e.

\[
r^e(u_i, Q_e) = \sum_{t_j \in Q_e} r^e(u_i, t_j).
\]

Similarly, \(r^p(t_i, Q_p)\) is the sum of \(r^p(t_i, u_j)\) where \(u_j \in Q_p\) are users in the query, i.e.

\[
r^p(t_i, Q_p) = \sum_{u_j \in Q_p} r^p(t_i, u_j).
\]

Given the tripartite graph \(G\), RWR is a natural way for definition of \(r^e(u_i, t_j)\) and \(r^p(t_i, u_j)\) [15], which can be defined by Equation 1, in which \((1 - c)\) is a random particle that starts from vertex \(i\). Matrix \(w\) is a transition matrix for graph \(G'=(V_1 \cup V_2 \cup V_3, E_1 \cup E_2)\) transformed from tripartite graph \(G(V_1, V_2, V_3, E_1, E_2)\), with column normalized. Elements in each column sum up to 1. \(r^e_i\) is a vector that the \((i\text{-th})\)element is 1 and other elements all are 0. Equation (1) is convergence which has been proved in reference [13].

\[
\overrightarrow{r}_{i+1} = (1-c)w \overrightarrow{r}_i + (c) \overrightarrow{e}_i
\]

Figure 4: Transition Matrix of Figure 3

Actually, as it is stated in reference [11], gave a node \(u_i\), to compute the relevance score of \(t_j\), it can be obtained via several random walks starting from \(u_i\), and count the number of times that we visit \(t_j\). This count reflects the relevance of between \(u_i\) and \(t_j\). The probability of visiting \(t_j\) from \(u_i\) is the relevance score we need.

We use a \(||V_1|| + ||V_2|| + ||V_3||\) \times \(||V_1|| + ||V_2|| + ||V_3||\) matrix \(w\) to represent a tripartite graph. If there is an edge from node \(i\) to node \(j\), then \(w_{ij} = 1\), otherwise \(w_{ij} = 0\). Figure 4 shows an example 11 \times 11 transition matrix \(w\) of above tripartite graph (Figure 3).

Figure 6: Transformed graph with stars given the query \(a_3\).

4 Two-stage RWR in tripartite graph based on star schema

The efficiency of RWR is a big challenge when processing large scale graphs [13]. To optimize the performance of RWR over tripartite graphs, we present a two-stage method in this section.

A tripartite graph can always be decomposed into a series of stars each of which centered at a vertex in \(V_2\). For example, the tripartite graph in Figure 3 can be decomposed into three stars in Figure 5. Based on this observation, we propose a two-stage RWR algorithm for tripartite graphs. The two stages are:

1. Decomposition of the tripartite graph to stars.
2. Random walk with restart over stars.

Definition 4 A star \(S\) is a tripartite graph \((V_1, V_2, V_3, E_1, E_2)\) satisfying that \(V_2\) has only one element \(v_0\), and \(v_0 \in V_1\) and \(v' \in V_3\), \((v_0, v') \in E_1\) and \((v_0, v'_0) \in E_2\).

Since decomposition of a tripartite graph is straightforward, we omit the details here. In the second stage, we first transform the original tripartite graph to a new weighted graph as follows. Firstly, each star is treated as a new vertex. Two new vertices are connected by an edge if and only if their stars share at least one common vertex in the old tripartite graph. The weight on edge is the number of vertices their stars share.

Given a user or a term, to evaluate the score function \(r^e()\) or \(r^p()\) over other vertices, we only need to set up a new vertex in the transformed graph denotes the query, i.e. the user vertex or the term vertex. This new one is connected to those vertices whose original star contains the query vertex. Figure 6, for example, is the transformed graph of the original tripartite graph in Figure 3, given a query \(a_3\).

The relevance score \(r(v_i, v_j)\) of two vertices in a graph can be also represented by Equation 2, in which \(\pi\) is a path from \(v_i\) to \(v_j\), while its length is \(\text{length}(\pi)\), and transition probability is \(p(\pi)\).

\[
r(v_i, v_j) = \sum_{v_{i\rightarrow v_j}} p(\pi)(1-c)^{\text{length}(\pi)}
\]
To evaluate the score function, it can be observed that:

\[ r(u_i, v_j) = r(u_i, p_k) + r(S_k, S_l) + r(p_l, t_j), \]

in which \( r(u_i, p_k) \) and \( r(p_l, t_j) \) can be directly obtained given the stars, and \( r(S_k, S_l) \) denotes the relevance score in the new transformed graph where \( S_k \) and \( S_l \) are vertices in the new graph that correspond to stars centered at \( p_k \) and \( p_l \). Thus, the problem of evaluation of \( r(u_i, v_j) \) is transformed to the problem of evaluating \( r(S_k, S_l) \) over the new graph. Thus, the size of transition matrix is reduced from \( (|V_1| + |V_2| + |V_3|) \times (|V_1| + |V_2| + |V_3|) \) to \( |V_2| \times |V_2| \). This, the process of RWR can be much more efficient in the decomposed graph compared with that in the tripartite graph. Experimental results reported in Section 6 verify our approach’s efficiency and effective.

5 Expert and expert profile query processing

Given the relevance score evaluation method introduced in Section 4, in this section, we introduce the whole process for expert and expert profile query processing, which is made up of four steps, as it is illustrated in Figure 7. The details on those four steps are introduced as follows.

5.1 Step 1: Construction of the tripartite graph \( G \)

The social media content are parsed. The bipartite graph of users and texts are constructed. Then, after the semantic annotation of the texts, the whole tripartite graph is constructed.

5.2 Step 2: Construction of the transformed graph \( G_s \) given the tripartite graph \( G \)

Intuitively, a star is a summary of a portion of the original tripartite graph. In this step, firstly, we find all stars \( S_i \) in the tripartite graph. Then, the star graph \( G_s \) is constructed, where stars \( S_i \)’s are treated as vertices, while the relationships between vertices, i.e., edges, are established and weighted.

Algorithm 1 shows more details about the procedures of constructing star graph \( G_s \) given the tripartite graph. This is a costly procedure. However, the computation can be offline and incremental. Thus, it will not affect the query processing performance.

5.3 Step 3: Constructing query graph \( G_q \) on the basis of star graph \( G_s \)

When a query \( Q_e \) (or \( Q_p \)) is submitted, we only need to add a query node and corresponding edges to star graph \( G_s \). If \( Q_e \) (or \( Q_p \)) and a star \( S_i \) have common vertices, an edge that connecting query node \( Q_e \) (or \( Q_p \)) and star component \( S_i \) is added. The weight of this edge is the number of common vertices of query \( Q_e \) (or \( Q_p \)) and star \( S_i \).

Algorithm 2 shows more details about constructing query graph \( G_q \) based on star graph \( G_s \).

5.4 Step 4: Finding experts \( E_t \) or expert profile \( E_p \) in graph \( G_q \)

5.4.1 Finding expert \( E_t \)

After the star graph is constructed, an inverted list for search of stars \( S_i \) given a vertex \( u \in V_1 \) is constructed. When a query is posed, after conducting RWR on query graph \( G_q \), we can get relevance scores of query node and each star \( S_i \). Then, for all \( u \in V_1 \), we only need to accumulate the relevance score of query node to each star \( S_i \) in \( S_u \) contained vertex \( v \). The accumulation value is the relevance score \( r(u, Q_e) \). Afterwards, we rank all \( u \in V_1 \) based on the relevance scores, and get top \( k \) u’s. They are experts \( E_t \) to query \( Q_e \).

Algorithm 3 and 4 show more details about finding experts \( E_t \).

5.4.2 Finding expert profile \( E_p \)

The problem of expert profile query processing is symmetric to the expert query processing. The process of finding experts can be easily adapted for finding expert profiles. Therefore, we omit the details here.

6 Empirical study

In this section, we perform extensive experiments to evaluate the performance of our algorithm on two real-life datasets.

6.1 Datasets

Two real-life datasets are used. They are introduced as follows:

- **Chinese online forum dataset** The first dataset contains all posts (and replies and comments) from a Chinese online forum, namely the Liba BBS\(^1\) from June 25 to July 25 2011. There are 1025 topics, each of which have a post and a series of comments and replies. 3528 users are involved. Conventional natural language processing methods are used for Chinese word segmentation. 6897 terms are extracted. Thus, in the tripartite graph, \( V_t \) contains authors of posts or comments, vertices in \( V_2 \) are topics, while \( V_3 \) is for terms extracted.

- **DBLP dataset** The DBLP Bibliography dataset\(^2\) is used in experiments. 5534 papers with 8136 authors from three research areas, including database, data mining, and information retrieval, are used. 2018 terms from paper titles are used as terms. Similarly, in the tripartite graph, \( V_t \) contains authors of papers, \( V_2 \) is the set of papers, while vertices in \( V_3 \) are terms extracted.

---

\(^1\)http://bbs.liba.com/.

\(^2\)http://www.informatik.uni-trier.de/~ley/db/.
Algorithm 1: Star graph construction

**Input:** Tripartite graph $G = (V_1, V_2, V_3, E_1, E_2)$

**Output:** Star Graph $G_s(V_1, V_2, V_3, E_1, E_2)$

1. Initialize star graph $G_s$ as empty;
2. For $v_i$ in $V_2$ do
   3. $\text{star}[i] = \text{all paths of length 1 starting from } v_i \text{ to vertices in } V_1 \text{ and } V_3$
   4. End
5. Each star component $S_i$ in $\text{star}[i]$ is used as a vertex in $G_s$;
6. For $S_i$ in $\text{star}[i]$ do
   7. For $S_j$ in $\text{star}[i]$ do
   8. If $\text{V}(S_i) \cap \text{V}(S_j) <> \emptyset$ then
   9. Add an edge to star graph $G_s$ from star $S_i$ to star $S_j$;
   10. $w(S_i \rightarrow S_j) = |V(S_i) \cap V(S_j)|$
   11. End
   12. End
13. End
14. Return star graph $G_s$;

Algorithm 2: Constructing query graph given a star graph

**Input:** Star graph $G_s(V_1, V_2, V_3, E_1, E_2)$, the query $Q$

**Output:** Query graph $G_q(V_1, V_2, V_3, E_1, E_2)$

1. Initialize star graph $G_q$ as empty;
2. $G_q = G_s$;
3. Add a new node $Q$ to $G_q$;
4. For $S_i$ in $\text{star}[i]$ do
   5. If $\text{V}(S_i) \cap \text{V}(Q) <> \emptyset$ then
   6. Add an edge to $G_q$ from $S_i$ to $Q$;
   7. $w(S_i \rightarrow Q) = |V(S_i) \cap V(Q)|$
   8. End
9. End
10. Return query graph $G_q$;

Algorithm 3: Capturing experts $E_t$ by RWR in graph $G_q(V_1, V_2, V_3, E_1, E_2)$

**Input:** matrix $W$ of $G_q$, all star components $\text{star}[i]$, query $Q_e$, the number of experts $k$, restarting probability $c$

**Output:** the experts $E_t$

1. Initialize $\vec{r}_i = 0$ except that the $i$-th element is 1;
2. Initialize $\vec{r}_i = 0$ except that the $i$-th element is 1;
3. Construct adjacent and transition matrix $w = colnorm(M)$ of graph $G_q$;
4. Repeat
   5. $\vec{r}_{i+1} = cw \vec{r}_i + (1-c) \vec{r}_i$;
   6. Passing 4 parameters $r_i, Q_e, \text{star}[i]$ to Algorithm 4;
   7. $E_t = \text{the output of Algorithm 4}$;
5. Until not changes to $E_t$;
6. Return $E_t$;
6.2 Methods to be compared

We carry out 50 expert queries and 50 expert profile queries randomly. They are conducted over two real datasets. We compare three methods based on different graph models.

- **Bipartite graph model (BG)** We construct a bipartite graph based on authors and terms, and implement the RWR algorithm on this bipartite graph to find experts and expert profiles. We call this approach as bipartite graph model, denoted as BG.

- **Tripartite graph model (TG)** We construct a tripartite graph based on users, topics (papers), and terms, and implement the RWR algorithm on this tripartite graph to find experts and expert profiles. We call this approach as tripartite graph model, denoted as TG.

- **Star graph model (SG)** We construct a star graph based on the tripartite graph, and implement the RWR algorithm on this star graph to find experts and expert profiles. We call this approach as star graph model, denoted as SG.

6.3 Measurements

We evaluate above three approaches on two datasets. Several measurements are used to evaluate those three methods.

- **Efficiency** Both time consumption and iteration times are used to measure the efficiency of three approaches.

- **Effectiveness** The effectiveness is only evaluated over the DBLP dataset. We did not evaluate it over the online forum dataset since there is no ground truth, and the evaluation is subjective. For the DBLP dataset, we use search results of ArnetMiner\(^3\), a service for academic data search, mining and visualization, as ground truth to evaluate the effectiveness of three approaches.

  - **Expert finding** We use expert query results of BG, TG and SG, and respectively compute edit distance and overlap rate between the query results of ArnetMiner and query results of three approaches.

6.4 Experimental results

6.4.1 Efficiency

We respectively use three graph models to find expert and expert profile. Figure 8 and Figure 9 respectively show time consumption and iteration times for three graph models over the DBLP dataset, where time consumption and iteration times are averages over 50 randomly selected persons and 50 randomly selected terms. Figure 10 and Figure 11 respectively show time consumption and iteration times for three graph models over the online forum dataset.

From the four figures, we know our star graph model is more efficient than other two graph models. It is because our graph model reduces the size of the transition matrix. As well, the bipartite graph model is more efficient than tripartite graph model due to the size of the matrix. It shows that the matrix size dominates the efficiency of RWR.

6.4.2 Effectiveness on expert finding

Using ArnetMiner as ground truth, we evaluate the effectiveness based on edit distance and overlap rate between the query result of three approaches and the query result of ArnetMiner. It is noted that we

\(^3\)http://www.arnetminer.org

---

**Algorithm 4:** Accumulating relevance score on results of RWR on star Graph \(G_s\)

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
<th>Experts (E_t)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(r_i, \text{star}[i], Q_e)</td>
<td>Experts (E_t)</td>
<td></td>
</tr>
<tr>
<td>Initialize (P_v = \emptyset)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Build a vertex level inverted index list (L(V, \text{Star})), containing two columns, in which one column is vertex (v_3 \in V_3), and another one is stars (S_v[i]) containing vertex (v)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>for (v \in V_3) do</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(m_v = 0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>for (S_v[i]) do</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Find the ((i+1))th element (e_{i+1}) of vector (r_i)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(m_v = m_v + e_{i+1})</td>
<td></td>
<td></td>
</tr>
<tr>
<td>end</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(E_t = { \text{top } k \ v \in V_3 \text{ according to its correlation score } m_v } }. \text{Return } E_t;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
6.4.3 Effectiveness on expert profile finding

For the DBLP dataset, expert profiles of a person are his research areas. However, the expert profile finding result of ArnetMiner are not ranked, and only contains 3-4 research area. We list the results of three researchers returned by three graph models, as they are shown in Table 2. From this table, we know the result is similar of three graph models, and the result of ArnetMiner cover the results of three graph models.

7 Related work

Our work in this paper is broadly related to several areas. We review some in this section.

7.1 Expertise mining in social media.

In the past few years, experts and expert profile finding is a hot topic. Krisztian Balog [2, 1] discusses people search in the enterprise by a generative probabilistic modeling framework for capturing the expert finding and profiling tasks in a uniform way. Small-Blue [7, 3] mainly depends on social network among
company. It focuses on “who knows what?”, “who knows whom?” and “who knows what about whom?”

Recently, along with the growth of web 2.0 applications, more and more researchers are devoted to expertise finding problem in social media. Jun Zhang et al. [18] and Zhao Zhang et al. [19] studies the problem on online forum. The former work only considers reply networks in online forums, while the latter one only considers contents. Junjie Yao et al. [16] model users’ expertise in folksonomies of tagging systems. Xiaoling Liu et al. [8] studied the problem of identifying topic experts in the Blogspace. In this paper, our approach can handle all kinds of social media, and perfectly combine social networks with contents.

7.2 Random walk with restart and its improvement.

Faloutsos et al. treats RWR as a good means to score relevance between nodes in a graph [4]. Hanghang Tong and others present several good applications using RWR [11, 12]. The issue of efficiency is great challenge of RWR [13]. Reference [14] proposed fast solutions to this problem. It uses low-rank matrix approximation and the community structure in graph to increase the query response of RWR.

8 Conclusions and future work

In this paper, we have addressed the problem of finding expert and expert profile in social media. Our work distinguishes with others in three aspects. First, a unified tripartite graph model is used to capture both content and structure information in social media. We show that a single random walk with restart procedure can be used to evaluate the relevance of a user and a term based on this graph model.

Second, a star-based optimization method is proposed to accelerate the RWR computation over tripartite graphs. Analysis show that this method can greatly reduce the online computation cost since it reduces the size of transition matrix.

Last but not the least, extensive experimental results over two real-life datasets show that our method outperforms previous bipartite graph model based method and the native tripartite graph model approach in terms of both effectiveness and efficiency.

Our future work include the exploration of data management techniques for star-based tripartite graph indexing that support RWR computation, and applications of expert and expert profile query in recommendation systems and online advertisement.

Acknowledgement

This work is partially supported by National Science Foundation of China under grant numbers 60833003, 61070051 and 61170086, National Basic Research (973 program) under grant number 2010CB731402, and National Major Projects on Science and Technology under grant number 2010ZX01042-002-001-01.

References


