

Pushing Scientific Documents by Discovering Interest in Information Flow Within E-Science Knowledge Grid*

Lianhong Ding^{1,2}, Xiang Li^{1,2}, and Yunpeng Xing^{1,2,3}

¹ Key Lab of Intelligent Information Processing, Institute of Computing Technology,
Chinese Academy of Sciences, Beijing, China

² Graduate School of Chinese Academy of Sciences,
100080 Beijing, China

{lhding, xiangli, Ypxing}@kg.ict.ac.cn

³ Hunan Knowledge Grid Lab, Hunan University of Science and Technology,
Hunan, China

Abstract. The Knowledge Grid is an intelligent and sustainable Internet application environment that enables people and roles to effectively capture, publish, share and manage explicit knowledge resources. As an important function of the e-Science Knowledge Grid, this paper proposes an approach to effectively push scientific documents within research teams by detecting the social characteristics in self-organized network, discovering interest in information flow, and capturing dynamic changes of interests over time. The proposed approach can be used in any cooperative organizations.

1 Introduction

Keeping with up-to-date documents becomes increasingly important in scientific research [17, 20, 24]. Team members often search for the same documents. This leads to low efficiency of teamwork. An efficient and effective document sharing method can improve the efficiency and competitiveness of organizations.

The first step is to discover the common interest communities in large social network by graph analysis [11]. Corresponding to vertex “betweenness” proposed by Freeman [7], Girvan and Newman put forward the conception of edge betweenness and partition a graph into discrete communities of nodes based on the idea of edge betweenness centrality. The betweenness of an edge is defined as the number of shortest paths that run along it. Communities are discovered by repeatedly identifying and removing the edges of highest betweenness because the edges that connect highly clustered communities have higher edge betweenness [8].

Two rules that direct the partition process to stop or go on are proposed and applied to find communities of related genes and communities within an organization automatically [11, 13]. The first is: the component that is composed of no more than 5 vertices should not be partitioned. The second is: the partition process should stop when the highest betweenness is $N-1$, where N is the number of vertices. The second

* This work was supported by the National Basic Research Program of China (973 project No.2003CB317000) and the National Science Foundation of China (Grants 60273020 and 70271007).

rule ends the algorithm before the isolated vertex appears. They will be called minimum component rule and $N-1$ betweenness rule respectively in the following.

Zhuge's Knowledge Grid is an intelligent and sustainable Internet application environment that enables people and roles to effectively capture, publish, share and manage explicit knowledge resources [18, 23, 25]. A scientific document sharing approach supported by the Knowledge Grid is introduced in this paper. It can actively push scientific documents to members by discovering their interests from information flow within e-Science Knowledge Grid [17, 19, 20, 24, 26, 28].

2 General Architecture

The general architecture of the proposed method consists of the following core modules as shown in Fig. 1.

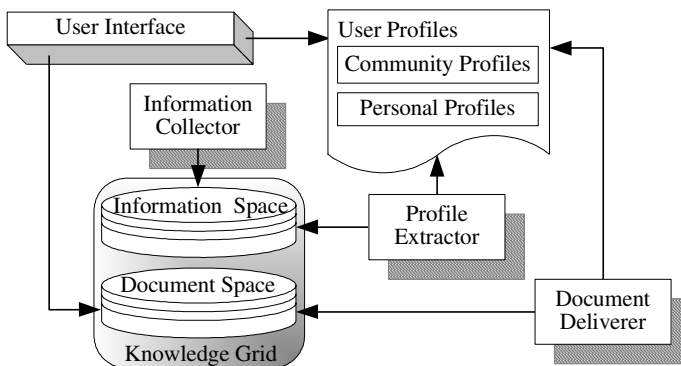


Fig. 1. General architecture of the proposed approach

Knowledge Grid manages distributed heterogeneous resources in a uniform way [23, 25]. It comprises information space and document space storing relevant contents of information flow and documents respectively [21, 22]. Document space has an *area-to-topic* structure, where each area is divided into many smaller and disjoint topics. An area review is stored in the area involved; a document emphasizing particularly on concrete issues is stored at the corresponding topic within certain area.

User profiles give *general-to-specific* description of member interests and provide supports for document deliverer. Community profile is an area covering the whole community interests. Personal profile is one or several concrete topics in the area, which exhibits difference between community members. Community profiles correspond to the area level of document space, while personal profiles correspond to the topic level.

Information collector gathers information flow in the organization and stores the useful contents into information space.

Profile extractor extracts user interests from information flow: discover community profiles by community detecting and learn personal profiles by mining.

User interface, by which members can tune their own profiles or upload scientific documents to the document space at anytime.

Document deliverer is responsible for pushing relevant documents to organization members actively by referring to user profiles [24].

3 Information Collection

There are many kinds of information flows in organizations, including email flow, flow of short message, flow in message board and flow in blog [19]. They are uniformly called messages if no special explanation.

Message like email is more than free-form text. It has additional features in the structured header in addition to the unstructured body. At regular intervals, information collector first parses each message into following six parts: *from*, *to*, *date*, *subject*, *body* and *attachment* (if any), uniforms the name of the senders and recipients and deletes quotations of other messages and signatures from message body, then stores them to information space, with each message as a record. Here, external messages and messages sent to a list of more than 10 recipients are neglected. Only the first attachment that the file type is doc, pdf, ps, html or plain text is reserved.

Junk messages mainly come from unknown people and only messages within an organization are collected, so they can be filtered out in this process.

4 Community Profiles and Community Structures

In practice, a community is formed by a group of members owning common interests. Community profile is an area that represents the community's common preference. It can be got from the understanding of anyone in the community. Here, it is approximately specified as the area that covers most of the member's interest points.

4.1 Social Network Construction

A social network is a map of the relationships between individuals where we can observe their social activities. Traditional generation of social network is time consuming and requires a large degree of cooperation from the subjects being studied.

Information flow in e-Science Knowledge Grid provides a cheaper, easier, and quicker way for social network data collection. Here, the social network for an organization is automatically constructed from information flow: vertices represent people; edges are added between pairs of correspondences that appear in the same message header. Different values can be assigned to the threshold that specifies the minimum number of messages passed between any two vertices.

4.2 Community Detecting

Our community detecting algorithm extends the idea of edge betweenness centrality to find social networks with different threshold values.

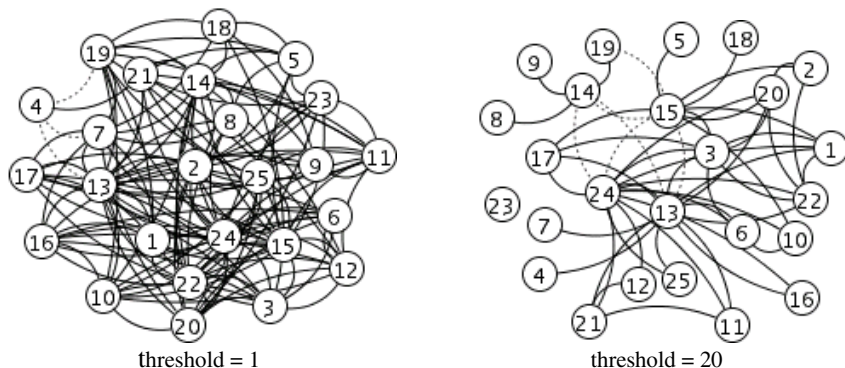


Fig. 2. Partition results following the $N-1$ betweenness rule

First, the partition algorithm follows the minimum component rule and $N-1$ betweenness rule. Fig. 2 illustrates the partition results, where dotted lines are the edges that have been removed by the algorithm. When the threshold=1, the algorithm stops with three edges removed for the $N-1$ betweenness rule. When threshold=20, it ends with six edges removed for the same reason, where node 23 is an isolated vertex in the social network itself. From above examples, we can see $N-1$ betweenness rule makes the partition process stop too early.

Then, the partition results of the algorithm discarding the $N-1$ betweenness rule are illustrated in Fig. 3. We can see the discarding of the $N-1$ betweenness rule leads to too many isolated vertices. So the rules proposed previously need modification, at least for the small-scale networks like ours.

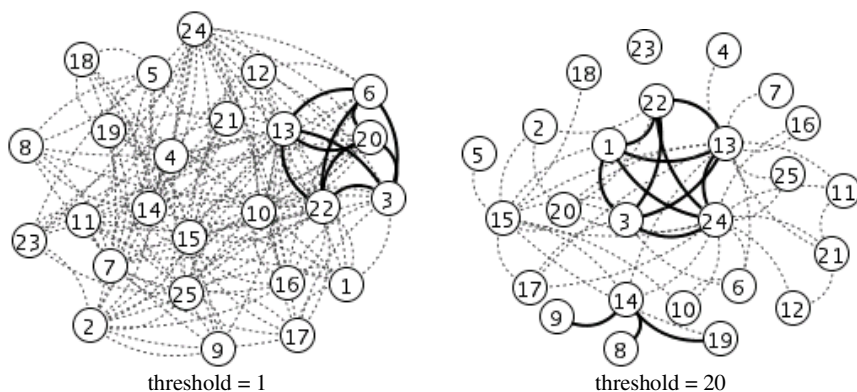


Fig. 3. Partition results discarding the $N-1$ betweenness rule

First, we replace the $N-1$ betweenness rule by the following rule: remove the edge with second highest betweenness, if the highest betweenness is $N-1$ and the component is still big enough. The number of the vertices in a component can be used to judge if the component is still big enough. It may be different for different scales and different aims.

Then, another rule that a complete graph should not be partitioned further is proposed. It can be judged if $E=N(N-1)/2$ holds, where E is the number of edges in the component and N is the number of vertices.

These two rules and the minimum community rule compose the new rules together.

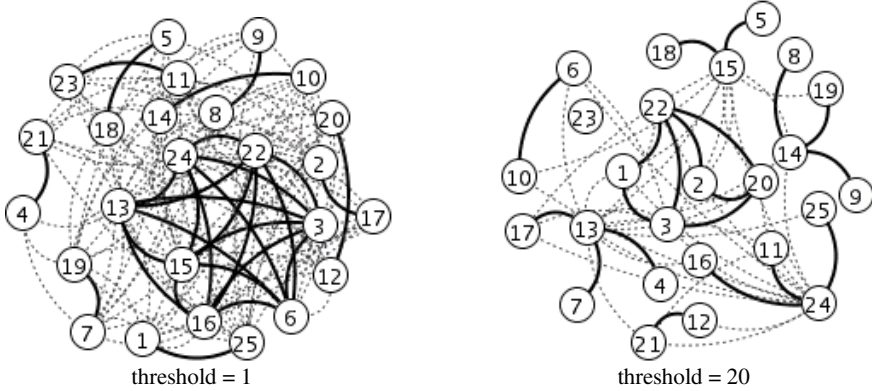


Fig. 4. Partition results following new rules

The partition results following the new rules are given in Fig. 4. Using threshold 1 as an example: after the three edges linking 4 and 13, 4 and 7, 4 and 19 are removed in order, the edge linking 4 and 21 becomes the one with the highest betweenness. Because the deletion of it will make 4 isolated, according to the new rules, the edge linking 13 and 21 is removed instead, which makes the algorithm continue. The complete graph rule newly introduced makes the community consisting of 3, 6, 13, 15, 16, 22 and 24 is reserved although there are more than five vertices.

4.3 Community Detecting of Weighted Social Network

The identification of shortest path is the key for the community discovering. The shortest path between two vertices is the fastest way from one to the other. Previous works find the shortest path on the assumption that each edge is equally long. In fact, a social network is a relational network, where edge length can illustrate the tie strength between two nodes: the shorter the edge is, the closer the relationship is or the more important the one is to the other [16]. Replacing equal edge length by different edge length will bring the shortest path more exact. Thereby, more accurate community structures are obtained.

One way is to define the edge length according to absolute importance of the edge, namely, how important the edge is for the whole network. Higher importance means shorter length. The length $Length_{ij}$ of the edge between vertex i and j is calculated as following.

$$Length_{ij} = \frac{t}{Num_{ij}} \tag{1}$$

where t is the threshold used to construct the social network from information flow and Num_{ij} is the number of messages that have been passed between i and j .

Another way is based on relative importance of the edge, that is, how important the edge is for the two vertices linked up by it.

$$Length_{ij} = \frac{Num_{i-all} \times Num_{j-all}}{Num_{i-j} \times Num_{j-i}} \tag{2}$$

where Num_{i-all} is the number of messages that i has sent to others, Num_{j-all} is the number of messages that j has sent to others; $Num_{i,j}$ is the number of messages that i has sent to j , and $Num_{j,i}$ is the number of messages that j has sent to i .

Table 1. Comparison between three methods. 1 means equal edge length, 2 means absolute importance-based edge length and 3 means relative importance-based edge length.

Method	t = 1	t = 10	t = 20	t = 30	t = 40
1	24%	72%	60%	72%	56%
2	44%	88%	92%	88%	80%
3	60%	72%	92%	84%	76%

Table 1 lists the partition precision when the three methods are adopted to calculate the edge length respectively. For each method medium thresholds bring higher precision. When the threshold is equal to 10, 20, 30 and 40, the absolute importance-based edge length method brings best partition results, then the relative importance-based edge length method, and then the equal edge length method. When the threshold is equal to 1, the relative importance-based method brings highest precious. In all, different edge length methods produce better results than the equal edge length method.

4.4 Network Generator and Community Detector

To automatically get community structures in an organization, we have developed network generator and community detector. The former takes messages collected as input, and related social network data stored in a Pajek .net file as output [1]. The threshold and specific method to calculate the length of edge can be chosen. Community detector inputs the Pajek .net file and displays the social network of the organization as an undirected graph. Community structures are given in the following form: the vertices of a community are linked up by solid edges and different communities in the organization are linked up by dotted edges. Location of each vertex can be changed to view the network and communities clearly.

5 Personal Profiles and Message Mining

Like user navigational data, messages that a member has read or written also implicate his or her preference. So members’ personal profiles can be got by tracing their daily using of all kinds of messages. As the Fig. 5 illustrates, it consists of unusable-message filtering, usable-message classification and personal profile computing.

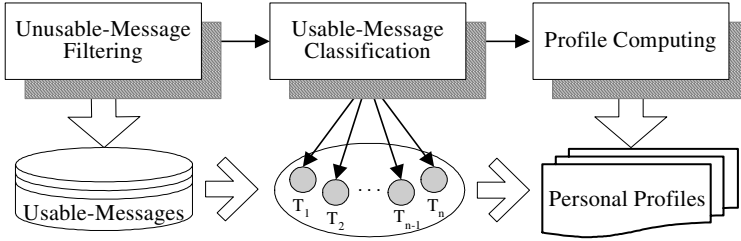


Fig. 5. Discovery of personal profiles

First, all messages stored are divided into usable-messages and unusable-messages. The former are the messages containing research information. Second, each usable-message is assigned to a specific topic by usable-message classification. The topic of a usable-message represents its main meaning. And at last, profile computing component determines personal profiles by a statistical evaluation of the results of usable-message classification. The principle directing this statistical algorithm is that the more usable-messages a member writes or reads about a topic, the more attention he or she pays to the topic.

Besides providing support to document deliverer, user profile can also benefit a research team by the following way: keeping track of what everyone is doing and has done by browsing current user profiles and history user profiles; and tuning one’s own profile to make interests-describing and document-delivering more accurately.

5.1 Message Representation

Each message m is represented as a weighed term vector $\vec{m} = (m^{(1)}, m^{(2)}, \dots)$ by the standard TFIDF function.

$$m^{(i)} = TF(w_i, m) \cdot \log\left(\frac{|D|}{DF(w_i)}\right) \tag{3}$$

where the term frequency $TF(w_i, m)$ is the number of times word w_i occurs in message m , $|D|$ denotes the total number of messages in the training set and the $DF(w_i)$ is the number of messages containing the word w_i at least one time.

Message is more than free-form text. The text in *subject* filed and *body* field of email is treated separately and identically in [14]: a word’s one time appearance in the *subject* and in the *body* is equal. Here, we also consider the text in *subject* field and *body* field of message separately but discriminatively.

Usually, *subject* is the outline of *body* contents, so words in *subject* filed are more descriptive and discriminative in contrast to the words in *body* field. That is, they are more important for the classification. Here, words in *subject* filed are assigned larger weights. When calculating the weight of word w_i , there is no change for $DF(w_i)$. For $TF(w_i, m)$, one time appearance in *subject* field equals to t times appearances in *body* field. The enhancement of $TF(w_i, m)$ strengthens the importance of w_i , while no change for $DF(w_i)$ ensures the tuning up of w_i will not be weakened. The Increase-ment of words’ weight reinforces their discriminative ability in turn [15].

Because message such as email is typically short and message *body*, *subject* and *attachment* normally express a common theme the text attachment reserved is treated as a part of message *body*.

5.2 Unusable-Message Filtering

All messages stored are divided into usable-messages and unusable-messages in this step. Usable-messages are messages whose contents are related to research information. Jokes or weekly meeting notices are typically unusable-messages. It can be regarded as a message classification where only usable-messages and unusable-messages exist.

Models of usable-messages and unusable-messages are represented as prototype vector \bar{m}_1 and prototype vector \bar{m}_2 respectively. For each model not only the positive examples but also the negative ones are taken into account.

$$\bar{m}_1 = \alpha \frac{1}{|M_1|} \sum_{\bar{m} \in M_1} \frac{\bar{m}}{|\bar{m}|} - \beta \frac{1}{|M_2|} \sum_{\bar{m} \in M_2} \frac{\bar{m}}{|\bar{m}|} \quad (4)$$

$$\bar{m}_2 = \alpha \frac{1}{|M_2|} \sum_{\bar{m} \in M_2} \frac{\bar{m}}{|\bar{m}|} - \beta \frac{1}{|M_1|} \sum_{\bar{m} \in M_1} \frac{\bar{m}}{|\bar{m}|} \quad (5)$$

α and β are parameters that adjust the relative impact of positive and negative examples. As recommended in [4], α and β are set to 16 and 4. M_1 and M_2 are the training messages of usable-messages and unusable-messages respectively; $|M_1|$ and $|M_2|$ are the number of messages in M_1 and M_2 . \bar{m} is the vector representation of message m and $|\bar{m}|$ denotes the Euclidean length of \bar{m} . Here, M_1 is positive examples for usable-messages and negative examples for unusable-messages. M_2 is positive examples for unusable-messages and negative examples for usable-messages. When transforming a training message into a feature vector, the words that occur with similar proportion of times in both M_1 and M_2 are deleted. Messages m is usable if cosine similarity between \bar{m} and \bar{m}_1 is higher than that between \bar{m} and \bar{m}_2 , otherwise, it is unusable.

5.3 Usable-Message Classification and Profile Computing

The usable-message classification specifies a topic for each usable-message by general document classification. First, a prototype vector for each topic in the document space is built. Documents stored in the topic level are utilized as training documents for corresponding topics. Then, for each usable-message within certain community, a prototype vector that gives the largest cosine of the message vector and the prototype vector itself is found. This topic is a good representation for the usable-message. Here, only the prototype vectors for the topics that belong to the area corresponding to the community profile are considered.

The extent that a member has paid attention to a topic can be reflected by how many usable-messages he or she has read or written about that topic. The more usable-messages a member reads or writes about a topic, the more the member is interested in the topic. Personal profile of user U_j takes the form as a finite set of $\langle topic_i,$

$energy_{ij} \gg$, where $energy_{ij}$ denotes the importance degree of $topic_i$ in the personal profile of U_j which can be obtained in the following way:

$$energy_{ij} = \frac{\alpha \sum_{(m \in from_j) \cap (m \in T_i)} 2^{-\frac{age(m)}{hl}} Sim(m, t) + \beta \sum_{(m \in to_j) \cap (m \in T_i)} 2^{-\frac{age(m)}{hl}} Sim(m, t)}{\alpha \sum_{(m \in from_j)} 2^{-\frac{age(m)}{hl}} Sim(m, t) + \beta \sum_{(m \in to_j)} 2^{-\frac{age(m)}{hl}} Sim(m, t)} \tag{6}$$

where $from_j$ is the usable-messages U_j has sent to others in the community and to_j is the usable-messages U_j has received from others in the community. T_i denotes the set of usable-messages associated with $topic_i$ and $Sim(m, t)$ is the cosine similarity between m and the topic to which it belongs. α and β are parameters that adjust the relative impact of usable-messages flowing from and to U_j separately. Stronger impact is specified to the usable-messages flowing from a member by a bigger α and a smaller β .

Since user interests often change, it is important to adjust the user profile incrementally [5]. A time factor $2^{-\frac{age(m)}{hl}}$ is introduced to adjust the contribution of usable-message for personal profile according to its age $age(m)$, which makes the descriptive ability of usable-message decay with time. $age(m)$ is the algebraic difference between the current date and the date when m was sent. The half-life span hl is set to 30 on the assumption that the effect of usable-messages on a topic reduces by 1/2 in one month [10]. Personal profile adapts to changes in member’s interests with the accumulation of messages and the decay of time. No techniques such as relevance feedback, user’s register and user’s ratings are employed.

6 Push Documents According to User Profiles

For each document, an owner list is maintained, which records all persons having owned the document. Each member can upload documents to the document space by user interface. One can only upload the document he or she owns, so the upload behavior itself shows that this document needn’t be pushed to him or her. Persons who try to upload a document are added to corresponding owner list, so are the members who have received that document. No record of a person in the owner list of a document is a necessary precondition for pushing a document for that person. A review in the area level of the document space is pushed to all members in the community whose community profile is in accord with its area. A document in certain topic is pushed to the persons whose personal profile includes this topic and corresponding energy value exceeds some threshold: first we find the community whose community profile is the area to which the topic belongs, and then choose right members in this community.

Each document in the document space is sent to members as email attachment by document deliverer regularly. Email is a “push” delivery mechanism in contrast to a “pull” mechanism, such as a web page searching. So group members do see the documents without any separate action, which makes our approach very low cost for organizations to adopt. If a member wants to share a document with others the only thing he or she needs to do is to upload that document into the document space.

7 Experimental Study and Findings

To verify the design concept and evaluate the performance of our approach, we carry out a study of our method and its effect. We present the survey results of the use of ‘profile extractor’ over a year and ‘document deliverer’ over six months working in our laboratory. The evaluation focuses on both the benefits and the costs.

Table 2. Median responses to selected questions from the survey. 1 means strongly disagree, 2 means disagree, 3 means neutral, 4 means agree and 5 means strongly agree.

	Questions	Response
Benefits	1. User profile helps me stay aware of what others are doing.	4.1
	2. User profile is useful when I want to find a person to discuss with.	4.3
	3. It is a good way to get useful documents.	4.2
	4. It's worth receiving documents to get valuable information.	4.2
	5. I would like to receive documents continuously.	3.9
	6. It is a good way to share documents with others.	4.4
	7. It is worth uploading documents to share them with others.	4.1
	8. I will upload documents continuously.	3.9
Concerns	9. Before user profiles, it was easy for me to keep track of what everyone else was doing.	2.1
	10. Receiving documents disturbs my daily routine.	2.1
	11. User profile extractor threatens my privacy.	1.3
	12. Uploading documents annoys me.	2.3
	13. Before that I can share documents with others easily.	1.1

Overall, we received a positive response. All members in our lab felt that we should continue to share documents by this way. Table 2 shows that using ‘profile extractor’ and ‘document deliverer’ results in group awareness and documents sharing at low cost.

Group awareness: Survey responses suggest that user profiles make them stay aware of what others are doing and it is useful when they want to find an appropriate person to discuss with (Question 1 and 2). Before that people did not find it was easy to know what others were doing (Question 9). Organization members did not have significant privacy concerns with user profile extractor (Question 11), because most of them didn't mind opening their research interests to others.

Documents sharing: Survey responses demonstrate that it is an easy way to get valuable documents and an effective method to share documents with others (Question 3, 6 and 7). Before that it was not easy to share documents with others (Question 13). The benefits of (selectively) reading documents received outweigh the cost of receiving and uploading documents (Question 4, 5 and 8). Members are also willing to upload more documents to receive documents (Question 8) and it doesn't disturb their routine work extensively (Question 10 and 12).

8 Comparison and Discussion

Learning and constructing user profiles without any intervention of human is an important property of our method. No techniques, such as relevance feedback, user's register and user's ratings are employed. A recommending system proposed in [12] asks users to evaluate a set of documents. Then a vector of keywords is extracted from these documents according to the evaluation results. The keywords vector is used to identify users. A study about document logistics in [24] has the descriptions supplied by users about their own interests as the core of user profiles initially. Then, a learning process is conducted to complement and tune user profiles until a steady state is reached.

Most recommending systems learn interests of users via relevance feedback and keep track of them daily using a search engine. It may satisfy a user's immediate information interests but usually is not sufficient for persistent interests because users are relatively poor at using them [9]. Today email has become one of the most popular communication fashions. Information flows such as email flow, short message flow and flow in message board supply a rich and persistent data resource for the learning of user profiles.

Usually, recommendations are given in the form of listing out the pages or documents when a user begins a new session of using for a system [2, 5, 9]. These occur only when a user chooses to use the system. It is an occasional and customizing behavior, not a persistent behavior. Our method pushes relevant documents by email for users at regular intervals, which ensures that users can get relevant documents in time and persistently. Users do see the documents without any separate action because email is a "push" delivery mechanism in contrast to the "pull" mechanism such as a web searching. It also makes our method very low cost for organizations to adopt.

Awareness means "understanding of the activities of others, which provides a context for your own activity" [6]. In [3] each group member has to write a 'today' message at the end of the day explaining what he or she did that day for the awareness of group. This type of awareness is immediate and short term (one day). The awareness in our work is long term. Through user profiles users can know what others have done, are doing and will do in a longer period. Members can quickly and accurately locate themselves in the whole organization by browsing community structures or user profiles. User profiles also tell us what we can learn from whom in which community.

9 Conclusion

The proposed approach has the following characteristics: (1) Introduce the community detection of social network into the interest discovering process. (2) Information flow is a rich and persistent data resource. Learning interest from information flow makes the description of user interests more accurate. (3) A time factor that weakens the impact of information flow on the user profiles with time elapsing is introduced, which makes user profiles adapt to changes in interests incrementally. (4) Actively push documents for members by email at regular intervals, which ensures that users can get valuable documents in time and persistently without any separate action. (5) Team members can know each others by observing community structures and user profiles. The approach will play a role in Zhuge's ideal of the future interconnection environment [27].

References

1. Batagelj, V., Mrvar, A.: Pajek - Analysis and Visualization of Large Networks. In: Jünger, M., Mutzel, P., (eds.): Graph Drawing Software, Springer-Verlag, Berlin (2003) 77-103
2. Bollacker, K.D., Lawrence, S., Giles, C.L.: A system for automatic personalized tracking of scientific literature on the Web. Proceedings of the Fourth ACM Conference on Digital Libraries. Berkeley, CA, USA (1999) 105-113
3. Bernheim, A.J.B., Borning, A.: 'Today' Messages: Lightweight Support for Small Group Awareness via Email. Proceedings of the 38th Annual Hawaii International Conference on System Sciences, Vol. 1. Waikoloa, Hawaii (2005)
4. Buckley, C., Salton, G., Allan, J.: The Effect of Adding Relevance Information in a Relevance Feedback Environment. Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Springer-Verlag, Dublin, Ireland (1994) 292-300
5. Chen, C.C., Chen, M.C., Sun, Y.: A Web Document Personalization User Model and System. User Modeling, Information Retrieval and User Modeling, Sonthofen, German (2001)
6. Dourish, P., Bellotti, V.: Awareness and Coordination in Shared Workspaces. Proceedings of the ACM Conference on Computer Supported Cooperative Work, Toronto, Canada (1992) 107-114
7. Freeman, L.: A set of measures of centrality based upon betweenness. *Sociometry*, 40(1) (1977) 35-41
8. Girvan, M., Newman, M.: Community structure in social and biological networks. Proceedings of the National Academy of Sciences of the United States of America, Vol. 99. USA (2002) 7821-7826
9. Somlo, G., Howe, A.: QueryTracker: An Agent for Tracking Persistent Information Needs. The Third International Joint Conference on Autonomous Agents and Multi-Agent Systems, New York, USA (2004) 488-495
10. Sugiyama, K., Hatano, K., Yoshikawa, M.: Adaptive web search based on user profile constructed without any effort from users. Proceedings of 13th conference on World Wide Web, New York, USA (2004) 675-684
11. Tyler, J.R., Wilkinson, D.M., Huberman, B.A.: Email as Spectroscopy: Automated Discovery of Community Structure within Organizations. Proceedings of the First International Conference on Communities and Technologies, Amsterdam, Netherlands (2003) 81-96
12. Vel, O., Nesbitt, S.: A Collaborative Filtering Agent System for Dynamic Virtual Communities on the Web. Proceedings of Conference on Automated Learning and Discovery, Pittsburgh, PA (1998)
13. Wilkinson, D.M., Huberman, B.A.: A method for finding communities of related genes. Proceedings of the National Academy of Sciences of the United States of America, Vol. 101. USA (2004) 5241-5248
14. Yang, J., Park, S.: Email Categorization Using Fast Machine Learning Algorithms. Proceedings of the 5th International Conference on Discovery Science, Lecture Notes in Computer Science, Vol. 2534. Springer-Verlag (2002) 316-323
15. Ye, Y., Ma, F., Rong, H., Huang, J.Z.: Enhanced Email Classification Based on Feature Space Enriching. 9th International Conference on Applications of Natural Languages to Information Systems, Lecture Notes in Computer Science, Vol. 3136. Salford, UK (2004)
16. Yee, J., Mills, R.F., Peterson, G.L., Bartczak, S.E.: Automatic Generation of Social Network Data from Electronic-Mail Communications. 10th ICCRTS, Track 1. Virginia (2005)

17. Zhuge, H.: Clustering Soft-Devices in Semantic Grid. *IEEE Computing in Science and Engineering*, 4 (6) (2002) 60-62
18. Zhuge, H.: A Knowledge Grid Model and Platform for Global Knowledge Sharing. *Expert Systems with Applications*, 22 (4) (2002) 313-320
19. Zhuge, H.: A Knowledge Flow Model for Peer-to-Peer Team Knowledge sharing and Management. *Expert Systems with Applications*, 23 (1) (2002) 23-30
20. Zhuge, H.: Active e-Document Framework ADF: Model and Platform. *Information and Management*, 41(1) (2003) 87-97
21. Zhuge, H.: Resource Space Grid: Model, Method and Platform. *Concurrency and Computation: Practice and Experience*, 16 (14) (2004) 1385-1413
22. Zhuge, H.: Fuzzy Resource Management in Semantic Grid: Model and Platform. *Journal of Systems and Software*, 73 (3) (2004) 389-396
23. Zhuge, H.: China's E-Science Knowledge Grid Environment. *IEEE Intelligent Systems*, 19(1) (2004) 13-17
24. Zhuge, H., .Li, Y.: Semantic Profile-based Document Logistics for Cooperative Research. *Future Generation Computer Systems*, 20(1) (2004) 47-60
25. Zhuge, H.: The Knowledge Grid. World Scientific, Singapore (2004)
26. Zhuge, H.: Exploring Epidemic and e-Epidemic with e-Science Environment. *Communications of the ACM*, 48(9) (2005)109-114
27. Zhuge, H.: The Future Interconnection Environment. *IEEE Computer*, 38 (4) (2005) 27-33
28. Zhuge, H.: Semantic Grid: Scientific Issues, Infrastructure, and Methodology. *Communications of the ACM*, 48 (4) (2005)117-119