ISBN: 978-972-8939-25-0 © 2010 IADIS

TOWARDS A MEANINGFUL EXPLOITATION OF IMPORTANT RELATIONSHIPS AMONG COMMUNITY MEMBERS AND COLLABORATION ASSETS

George Gkotsis*, Nikos Karacapilidis*, Costas Pappis** and Nikos Tsirakis***
*Industrial Management and Information Systems Lab, MEAD, University of Patras, 26500 Rio Patras, Greece
**Department of Industrial Management, University of Piraeus, Piraeus 18534, Greece
***Computer Engineering and Informatics Department, University of Patras, 26500 Rio Patras, Greece

ABSTRACT

Numerous tools aiming at facilitating or enhancing collaboration among members of diverse communities have been already deployed and tested over the Web. Taking into account the particularities of online Communities of Practice, this paper introduces a framework for mining knowledge that is hidden in such settings. Our motivation stems from the criticism that contemporary tools receive regarding lack of active participation and limited engagement in their use, partially due to their inability of identifying and meaningfully exploiting important relationships among community members and collaboration-related assets. Particular attention is given to the identification of requirements imposed by contemporary communities and learning contexts.

KEYWORDS

Social Networking, Web 2.0, Collaborative Systems, Data Mining.

1. INTRODUCTION

As information diffusion is becoming enormous, contemporary knowledge workers are facing a series of problems. People are straggling when trying to filter relevant information, extract knowledge out of it, and apply specific practices on a problem under consideration. This phenomenon, broadly known as information overload, has currently raised new, challenging, but not fully addressed issues. At the same time, it is widely admitted that one of the best means to keep a knowledge worker's competence high is through continuous learning (Rosenberg, 2000). In fact, most organizations already support learning activities through seminars and other traditional learning activities. Nevertheless, these activities do not comply with every learning need. Collaborative environments aiming at supporting collaboration among groups of people forming Communities of Practice (CoPs) are believed to be one of the most promising solutions to promote what is commonly known as "collective intelligence" or "organizational memory" (Ackerman, 1998). The term CoP is used to define a group of people with "common disciplinary background, similar work activities, tools and shared stories, contexts and values" (Millen *et al.*, 2002).

Related to the above remarks, contemporary tools receive criticism as far as active participation and engagement of their users is concerned; this is partially due to the inability of identifying and meaningfully exploiting a set of important relationships among community members and collaboration-related assets. To address this problem, this paper introduces a framework that enables one to reveal meaningful relationships, as well as other valuable information, about the community members' roles and competencies. The proposed framework exploits and integrates features originally found in the Data Mining and Social Networking disciplines, and is intended to be used towards strengthening a community's integrity.

2. RELATED WORK

Contemporary approaches to web-based collaboration environments build on diverse user profiling mechanisms (Fink and Kobsa, 2000). These approaches usually distinguish between static (user defined) and dynamic (system updated) profiles. Dynamic attributes derive by tracking down user actions and aim at providing a more personalized environment. Personalization of the environment may include user interface adaptation by making most usable actions or information more easily accessible. Moreover, by taking into account a user's profile, these approaches aim at filtering information that generally resides in a big collection of documents. Information filtering is achieved by reading the content of these documents and combining this content with the user's profile. The main goal of these approaches is to provide individualized recommendations to the users about the system items.

Social network analysis (SNA) is a tool that allows the examination of social relationships in a group of users, which is able to reveal hidden relationships (Wasserman and Faust, 1994). In business, SNA can be a useful tool to reveal relationships and organizational structure other than those formally defined. These relationships are extracted by examining the communication level among employees, usually resulting to a collaboration graph. The outcome of this graph's analysis is likely to result to a flow chart that does not necessarily follow the formal organizational structure. Traditionally, social relationships are revealed by acquiring data manually (through questionnaires or interviews). SNA has been applied with success in business and is regarded as a useful tool to analyze how a corporate operates.

In summary, the need for extracting and analyzing relationships among users has been identified for a long time. The topic of this paper is addressed by two different but complementary approaches. One of them applies Data Mining techniques that results to the identification of groups of people with similar preferences. The other adopts SNA for gathering the required knowledge with regard to these groups. This approach is applied to an online CoP aiming at revealing issues like management or human resource vulnerabilities. Integrating these approaches, this paper introduces a framework that aims at identifying, measuring and exploiting hidden relationships in a CoP. The overall approach followed in this paper, as described in the next section, takes into account the particularities of these communities by customizing practices from Data Mining and Collaborative Filtering and unites them with Social Network Analysis.

3. MINING HIDDEN KNOWLEDGE

Let us consider a collaboration support system, which adopts common Web 2.0 features and functionalities. The system allows members of a CoP to easily express pieces of their knowledge through various types of documents. These documents are uploaded on virtual working environments, usually known as workspaces. Let us also assume that the system supports document rating by its users. For such a setting, we introduce a framework for mining hidden knowledge. The proposed framework exploits a set of metrics that are discussed below.

First, we define user similarity *sim(i, j)* between users *i* and *j*. Similarity is calculated by exploiting the *User Nearest Neighbor algorithm* (McLaughlin and Herlocker, 2004), which takes as input the items' ratings provided by users. More specifically, the algorithm for measuring user similarity is based on the Pearson correlation, with respect to the commonly rated items. We define:

$$sim(i,j) = \frac{\sum_{a \in A_i \cap A_j} (Rating_i(a) - \overline{Rating_i})(Rating_j(a) - \overline{Rating_j})}{\sigma_i \sigma_j}$$

where σ_i and σ_j represent the standard deviations of the common item ratings for users *i* and *j*, respectively, A_i is the set of items that user *i* has rated, $Rating_i(a)$ is the rating of user *i* on item *a*, and $\overline{Rating_i}$ is the average rating of user *i* on the set of commonly rated items.

To avoid overestimating the similarity of users with small corpus of documents that have been rating by both users i and j, the above equation is amended as follows:

$$sim'(i,j) = \frac{\max\{|A_i \cap A_j|, \gamma\}}{\gamma} \times sim(i,j),$$

where γ is a system-defined threshold.

In our case, the users of the system are the authors of all documents uploaded on the system's workspaces. To weigh account ratings of user *i* on documents created by user *j,* a modification of the above measurement is defined. It is:

$$sim''(i,j) = \frac{\overline{Rating_{ij}}}{\overline{Rating_i}} \times sim'(i,j),$$

where $\overline{Rating_{ij}}$ is the average rating of user *i* on documents created by user *j*.

The above equation takes into account how user *i* evaluates documents of user *j* with regards to the rest of their common documents. As derives from the above, similarity between users *i* and *j* is not *reciprocal*. This remark is of big importance. Naturally, depending on a CoP's activity and nature of data, there are cases where the number of user's *j* documents that are rated by user *i* is small and therefore the information is limited. Thus, depending on the amount and nature of our data, either *sim'(i,j)* or *sim''(i,j)* can be selected (see more in Section 3.1).

Secondly, we define $relationship_{ij}$ between users *i* and *j* as a boolean measurement of direct communication between these users as follows: for each couple *(i, j)* of members *i* and *j*, we create a matrix *A* where a_{ij} is the number of document ratings that user *i* has made on documents created by user *j*. From this matrix *A*, we construct a new symmetric matrix *A'* where $a'_{ij} = max\{a_{ij}, a_{ji}\}$ (weak cohesion) or $a'_{ij} = min\{a_{ij}, a_{ji}\}$ (strong cohesion). Assuming that n_i is the overall number of ratings of user *i*, we define a symmetric measurement, called communication degree d_{ij}, which represents the communication flow between users *i* and *j*, as follows:

$$d_{ij} = \frac{a'_{ij}}{n_i + n_j - a'_{ij}} \ .$$

It is noted that d_{ij} is 1 in case where users *i* and *j* rate exclusively each other's documents, and 0 if none of the users has rated any of the other's documents (weak cohesion) or at least one has not rated all of the other's documents (strong cohesion). We define the binary function $relationship_{ij}$ to indicate the existence or not of adequate direct information exchange between users *i* and *j*. It is:

$$relationship_{ij} = \begin{cases} 1 & if\ d_{ij} \geq t, \\ 0 & if\ d_{ij} < t \end{cases}$$

where *t* is a threshold depending on the nature and needs of the community under consideration. The function $relationship_{ij}$ is the fundamental relationship that is used to construct the social network in the community and will be used for the required analysis.

Other metrics adopted within our approach are:

– *Clusters:* they refer to groups of entities (users in our case), in a way that entities in one cluster are very similar, while entities in different clusters are quite distinct. Each cluster can combine various plausible criteria.

– *Degree:* it expresses the number of people a CoP member is connected to. Members with high degree may be considered as of major importance in the network hub, since they keep the CoP tightly connected.

– *Betweenness:* While someone may be tightly connected with someone else, it might be the case that some CoP members express the CoP's integrity "better". People with high betweenness value are considered to better express a collective consensus. More formally, betweenness of a member can be expressed as the total number of shortest paths between all pairs of members that pass through this member (Freeman, 1977).

– *Closeness:* it expresses the total number of links that a member must go through in order to reach everyone else in the network (Sabidussi, 1966).

3.1 Clustering

Clustering can be defined as the process of organizing objects in a database into groups (clusters), such that objects within the same cluster have a high degree of similarity (while objects belonging to different clusters have a high degree of dissimilarity) (Anderberg, 1973; Jain and Dubes, 1988; Kaufman and Rousseeuw,

1990). Generally speaking, clustering methods about numerical data have been viewed in opposition to conceptual clustering methods developed in Artificial Intelligence.

Referring to a specific CoP, a cluster is a collection of users that share similar ratings on items of the same workspace. For example, let SP_1, SP_2,..., SP_k be the k workspaces used by a community A. We build an array X of size $n \times n$ (n is the number of users), where the cell X_{ij} denotes the correlation between user i and user j. Correlation can be either $sim'(ij)$ or $sim''(ij)$, which will result to symmetric undirected or directed arrays, respectively. After the construction of these arrays, a unified array can be built for the whole set of workspaces by calculating the average value of each cell.

Regarding the clustering procedure and depending on the nature of the data gathered, two different approaches with respect to the nature of the arrays (one concerning *symmetric undirected arrays* and one concerning *directed arrays*) are followed.

In the first case (symmetric undirected arrays), an algorithm for hierarchical clustering is applied. In hierarchical clustering, there is a partitioning procedure of objects into optimally homogeneous groups (Johnson, 1967). There are two different categories of hierarchical algorithms: these that repeatedly merge two smaller clusters into a larger one, and those that split a larger cluster into smaller ones. In *MinMax Cut algorithm* (Ding *et al.*, 2001), given n data objects and the pair similarity matrix $W=(w_{i,j})$ (where $w_{i,j}$ is the similarity weight between i and j), the main scope is to partition data into two clusters A and B. The principle of this algorithm is to minimize similarity between clusters and maximize similarity within a cluster. The similarity between clusters A and B is defined as the cutsize

$$s(A,B) = \sum_{i \in A, j \in B} w_{i,j} .$$

Similarity (self-similarity) within a cluster A is the sum of all similarity weights within A: $s(A,A)$. Consequently, the algorithm requires to minimize $s(A,B)$ and maximize $s(A,A)$ and $s(B,B)$, which is formulated by the min-max cut function $MMcut(A,B)$:

$$MMcut(A,B) = \frac{s(A,B)}{s(A,A)} + \frac{s(A,B)}{s(B,B)}.$$

Linkage $l(A,B)$ is a closeness or similarity measure between clusters A and B; it quantifies cluster similarity more efficiently than weight, since it normalizes cluster weight:

$$l(A,B) = \frac{s(A,B)}{s(A,A) \times s(B,B)}$$

For a single user i, his linkage to cluster A is: $l(A,i)=S(A,i)/S(A,A)$, where $S(A,i)=S(A,B)$, $B=\{i\}$. According to this equation, users close to the cut can be found. If a user i belongs to a cluster, his linkage with this cluster will be high. When a user is near the cut, then the linkage difference can be used: $\Delta l(i) = l($ A user with small Δl is near the cut and is a possible candidate to be moved to the other cluster.

In the second case (directed arrays), the clustering algorithm presented in (Chakrabarti *et al.*, 1998), which is based on Kleinberg's link analysis algorithm (Kleinberg, 1999), is adopted. Initially, this analysis was applied to documents related to each other through directed relationships (like in the World Wide Web); for every document, authority and hub scores are calculated as the sum of hub and authority scores pointed to and from this document, respectively. More specifically, having set up the user similarity matrix X of size $n \times n$ as before, a weighted directed graph of users is allocated and the non-principal eigenvectors of $X^T \times X$ are computed. The components of each non-principal eigenvector are assigned to each user. By ordering in increasing order the values of each eigenvector, a partition is declared among users at the largest gap between them. The entries of $X^T \times X$ represent authority scores and those of $X \times X^T$ refer to hub scores. The result is groups of users that are close to each another under the authority or hub score value. The algorithm can create up to $2 \times n$ different groups of users, but experimental results have shown that groups with users that have large coordinates in the first few eigenvectors tend to recur and therefore a first set of collections of users can satisfy the clustering procedure (e.g. (Kleinberg, 1999)).

3.2 Social Network Analysis

After the clustering procedure, groups of users that share the same or closely related preferences with regards to their documents' ratings are revealed. More formally, there is a classification of members of every CoP into a set of clusters $C_1, C_2, \ldots, C_m$. For each cluster, the proposed framework computes the values of each of

the SNA measures (i.e. degree, betweenness and closeness). More specifically, contrary to clustering where the measurement was correlation (i.e. user similarity), SNA exploits the $relationship_{ij}$ measurement, which corresponds to communication among members. This analysis can provide useful findings; for example, a specific group of users considered to have high similarity will appear as a cluster. More generally, the combination of clustering and SNA highlights properties in groups of users of particular interest; this extracted knowledge can be provided to the users through notification or recommendations mechanisms.

4. CONCLUSIONS

This paper has introduced a framework that can be applied to a wide range of software platforms aiming at facilitating collaboration and learning among users. Having described the basic characteristics of the settings under consideration, we presented an approach that integrates techniques from the Data Mining and Social Network Analysis disciplines. More precisely, we formulated two different clustering approaches in order to find the values of some meaningful metrics. Moreover, we combined the outcomes of the proposed clustering methodology with SNA metrics. The result of this effort is to reveal valuable knowledge residing in a CoP.

REFERENCES

Ackerman, M. (1998). Augmenting organizational memory: a field study of answer garden. *ACM Transactions on Information Systems,* Vol. 16, No. 3, pp. 203-224.

Anderberg, M.R. (1973), *Cluster Analysis for Applications*. Academic Press, New York, NY.

Chakrabarti, S., Dom, B., Gibson, D., Kumar, R., Raghavan, P., Rajagopalan, S. and Tomkins, A. (1998). Spectral filtering for resource discovery. *Proceedings of the ACM SIGIR Workshop on Hypertext Information Retrieval on the Web*. ACM Press, New York, NY, pp. 13-21.

Ding, C., He, X., Zha, H., Gu, M. and Simon, H. (2001). A min-max cut algorithm for graph partitioning and data clustering. *Proceedings of the 2001 IEEE International Conference on Data Mining*, Washington, DC, USA, pp. 107-114.

Fink, J. and Kobsa, A. (2000). A review and analysis of commercial user modeling servers for personalization on the world wide web. *User Modeling and User-Adapted Interaction,* Vol. 10, No. 2, pp. 209-249.

Freeman, L. (1977). A set of measures of centrality based on betweenness. *Sociometry,* Vol. 40, No. 1, pp. 35-41.

Jain, A. and Dubes, R. (1988). *Algorithms for clustering data*. Prentice-Hall, Englewood Cliffs, NJ.

Johnson, S. (1967). Hierarchical clustering schemes. *Psychometrika*, Vol. 32, No. 3, pp. 241-254.

Kaufman, L. and Rousseeuw, P. (1990). *Finding groups in data: an introduction to cluster analysis*. Wiley-Interscience, New York, NY.

Kleinberg, J. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, Vol. 46, No. 5, pp. 604-632.

McLaughlin, M. and Herlocker, J. (2004). A collaborative filtering algorithm and evaluation metric that accurately model the user experience. *Proceedings of the 27th annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM New York, NY, USA, pp. 329-336.

Millen, D.R., Fontaine, M.A., Muller, M.J. (2002). Understanding the benefit and costs of communities of practice. *Communications of ACM*, Vol. 45, No. 4, pp. 69-73.

Rosenberg, M.J. (2000). *E-Learning: Strategies for Delivering Knowledge in the Digital Age*. McGraw-Hill, New York, NY.

Sabidussi, G. (1966). The centrality index of a graph. *Psychometrika*, Vol. 31, No. 4, pp. 581-603.

Wasserman, S. and Faust, K. (1994). *Social network analysis: Methods and applications*. Cambridge University Press, New York, NY.