# Mining Bulletin Board Systems Using Community Generation

Ming Li[1], Zhongfei (Mark) Zhang[2], and Zhi-Hua Zhou[1]

[1] National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210093, China
[2] Computer Science Department, SUNY Binghamton, Binghamton, NY 13902, USA
{lim,zhouzh}@lamda.nju.edu.cn, zhongfei@cs.binghamton.edu

**Abstract.** Bulletin board system (BBS) is popular on the Internet. This paper attempts to identify communities of interest-sharing users on BBS. First, the paper formulates a general model for the BBS data, consisting of a collection of user IDs described by two views to their behavior actions along the timeline, i.e., the topics of the posted messages and the boards to which the messages are posted. Based on this model which contains no explicit link information between users, a uni-party data community generation algorithm called ISGI is proposed, which employs a specifically designed hierarchical similarity function to measure the correlations between two different individual users. Then, the BPUC algorithm is proposed, which uses the generated communities to predict users' behavior actions under certain conditions for situation awareness or personalized services development. For instance, the BPUC predictions may be used to answer questions such as "what will be the likely behavior user $X$ may take if he/she logs into the BBS tomorrow?". Experiments on a large scale, real-world BBS data set demonstrate the effectiveness of the proposed model and algorithms.

## 1 Introduction

Bulletin board system (BBS) is an important information exchanging and sharing platform on the Internet. The analysis of useful patterns from BBS data has drawn much attention in recent years [5,6,8].

A BBS is an electronic "whiteboard" which usually consists of a number of *boards*, the discussion areas relating to some general themes (e.g. *Sports*). On each board, users read and/or post messages on different *topics*, which may be well determined by the titles of the message. In a BBS, one could easily start a discussion on a specific topic or express his/her viewpoint on an existing topic.

Since users with different backgrounds, different interests may access the same BBS, the BBS essentially serves as a mapping to the real world society, such that the relationships between the individual users may be discovered and analyzed through discovering and learning this mapping. Various relationships between users that hold sufficient interestingness to mine through the BBS data include the users with a similar interest or a similar taste, or a similar behavior action, and given what type of users, what specific behavior action may be taken if they share a similar specific interest. For example, two individuals who happen to be both basketball fans are likely to go to the same

boards under a topic related to basketballs of a BBS. Clearly, effective discovery of these relationships between users of a BBS through mining the BBS data is essential and extremely helpful in situation awareness and in the development and delivery of personalized services to users.

*Community generation* is an effective way to identify groups of data items satisfying certain relationship constraints in a large amount of data, where the identified groups are called *communities*. Based on the availability of link information between data items, methods could be divided into two categories [9]. One is *bi-party data community generation* (BDCG), where link information between data items is explicitly provided besides the features that describe the data items. Such link information is important and methods of this category usually generate communities by combining link analysis and clustering techniques (e.g., [1]). Successful applications include [4], [2], [3], etc. The other category, in contrast, is *uni-party data community generation* (UDCG), where the link information is *not* available and must be obtained by further exploring additional information from data items.

In this paper, the BBS data are mined to discover the interest-sharing user groups, or communities. In particular, the topics of the posted messages and the boards the messages are posted to are considered as the two attributes of a user's behavior actions to demonstrate the user's interest, and thus are subsequently considered as the two views to the user's actions. Hence, a formulated BBS data model is proposed in this paper consisting of a collection of the BBS users, whose behaviors or access patterns are described by the history of actions reflected in the two views. Under this model, a UDCG algorithm called ISGI, i.e. Interest-Sharing Group Identification, is proposed to discover the groups of the users with similar interests, where communities are generated by analyzing the correlations between users based on a specially designed hierarchical similarity function. In addition, the users' behaviors are predicted with the help of the interest-sharing groups under certain conditions, which illustrates one of many potential applications using the generated community. Experiments show that the interest-sharing user groups may be effectively discovered by ISGI, and the generated communities are helpful in predicting users' behaviors, which will be useful in situation awareness and personalized services development.

The rest of the paper is organized as follows. Section 2 formulates the BBS data model. Section 3 proposes the ISGI method. Section 4 describes how to use the generated community to predict the behavior of a given user. Section 5 reports on the experiment results. Finally, Section 6 concludes the paper.

## 2   A General Model for Community Generation on BBS

In general, a BBS provides more facilities (e.g., file sharing). To simplify the problem, we only consider the posted messages in a BBS in this paper. For further simplication, the message body is ignored and only the title of a message is used to fully determine the topics of the message. Key words of the tiles are extracted using standard text processing techniques, and mapped to those collected topics through standard statistical analysis (histogramming).

To identify the specific interest-sharing relationships among a BBS users, we explicitly model a user's *access pattern* on BBS using information from two different views. Presumably, a BBS user tends to initiate or join in a discussion on a certain topic in which he or she is interested. Thus, the history of the topics on which the user has posted messages may reflect the interests of the user. Note that the users' interests are time-dependent because the discussions on BBS are usually closely related to the events that happen at the times when the discussions are raised. Consequently, posting messages to the same topic at different times may carry different semantics and meanings. On the other hand, a user's interest level in a specific topic may also be assessed by the *frequency* of messages which this user had posted on this topic within a certain period of time. For example, given a specific time interval, a user posting more messages on a topic presumably shows a greater interest in this topic than another user posting fewer messages on the same topic within the same time interval. Therefore, for the proposed BBS model, in the view of *Topics*, a user's access pattern is explicitly represented as a set of topics and the user access frequencies of the messages posted to different boards by different users along the timeline.

On the other hand, a user's interests may also be revealed by the boards where the messages are posted. In a typical BBS, discussion area is divided into different boards according to a set of categories. When accessing to a BBS, a user usually prefers visiting the boards that have the most interesting categories to this user. After exposing to an interesting topic in these boards, the user may decide to join the discussion on the topic being held in this board. Therefore, for the proposed BBS model, in the view of *Boards*, a user's access pattern is represented as a set of boards and the frequencies of messages posted to the boards along the timeline.

Consequently, the proposed BBS model is represented as a collection of users, each being represented by two timelines of *actions* on the Boards view and Topics view, respectively. Formally, let $ID$ denote the set of all valid users in a BBS. Let $\mathcal{T}$ and $\mathcal{B}$ be the sets of the topics that have been discussed on the BBS and all the boards to which messages are posted, respectively; let $T$ denote the set of time intervals quantified (e.g., a day) for the whole activation period of the BBS. Thus, the proposed BBS model is represented as follows:

$$BBS = \{< id, A_{id}^{\mathcal{T}}, A_{id}^{\mathcal{B}} > | id \in ID, A_{id}^{\mathcal{T}} \subset \mathcal{A}^{\mathcal{T}}, A_{id}^{\mathcal{B}} \subset \mathcal{A}^{\mathcal{B}}\} \qquad (1)$$

$$\mathcal{A}^{\mathcal{T}} = \{< \tau, f_\tau, t > | \tau \in \mathcal{T}, f_\tau \in \mathbb{N}, t \in T\} \qquad (2)$$

$$\mathcal{A}^{\mathcal{B}} = \{< \beta, f_\beta, t > | \beta \in \mathcal{B}, f_\beta \in \mathbb{N}, t \in T\} \qquad (3)$$

where $< \tau, f_\tau, t >$ and $< \beta, f_\beta, t >$ are actions in each view, indicating that at time $t$ posting messages with topic $\tau$ for $f_\tau$ times and to the board $\beta$ for $f_\beta$ times, respectively. Note that the timelines of both views are used together and contribute equally to the representation of the user's access pattern.

## 3 Interest-Sharing Group Identification

Given the BBS model presented above, we can identify the communities of users sharing similar interests. Unfortunately, many widely used methods (e.g., [3,4,7]) rely on
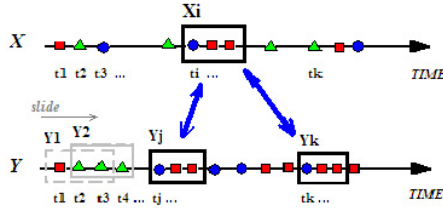
**Fig. 1.** An example of finding similar access patterns between the timelines of users

explicit link information to generate communities. Due to the absence of link information in our problem, we propose ISGI algorithm to identify interest-sharing groups from BBS without provided link information.

Firstly, the links between all the pairs of users are hypothesized, which induces a complete graph $G_h$ on $ID$. And then, the correlation between each pair of users is measured by aggregating the overall similarities in each view of actions of the two users. we hierarchically define a similarity function to determine the correlation between two users access patterns under a given view. Such similarity is measured by combining a set of time-dependent local similarities between all pairs of access patterns in individual time slots along the timeline.

Specifically, given two timelines of actions $X$ and $Y$ (either in the Topic View or in the Boards View) of two users $id_x$ and $id_y$, respectively, we examine similarity between every pair of time slots from different timelines by sliding a time window of size $z$ along both the timelines, as shown in Figure 1. Let $X_i$ and $Y_j$ be sets of the actions in two time slots starting at time $t$ and time $s$ along each timelines, respectively. Note that the order information of actions within a time slot is not considered because users with similar interest may not necessarily take similar actions within a time slot in the same order. A straightforward way to define the similarity between $X_i$ and $Y_j$ is $|X_i \cap Y_j|/|X_i \cup Y_j|$. However, this definition ignores the frequencies of the actions; with this definition, one who takes an action (e.g., posting a message to a board) 100 times would be considered the same as another who takes the action only once. To accommodate the contributions from different action frequencies, the average frequency difference of the actions shared by both $X_i$ and $Y_j$ is defined as

$$fd\left(X_i, Y_j\right) = \frac{1}{|X_i \cap Y_j|} \sum_{a \in X_i \cap Y_j} \left| f_{X_i}(a) - f_{Y_j}(a) \right| \tag{4}$$

where $f_{X_i}(a)$ and $f_{Y_j}(a)$ denote the frequencies of the action $a$ in $X_i$ and $Y_j$, respectively. Then, we define *local similarity* between $X_i$ and $Y_j$ as

$$ls(X_i, Y_j) = \frac{1}{1 + fd(X_i, Y_j)} \cdot \frac{|X_i \cap Y_j|}{|X_i \cup Y_j|} \tag{5}$$

We then construct a global similarity between the two timelines based on the local similarities between all pairs time slots. Firstly, for any time slot $X_i$, we aggregate these local similarities between $X_i$ and all $Y_j \in Y$ into a *hybrid similarity* between $X_i$ and $Y$, which is defined as follows,

**Table 1.** Pseudo-code describing the ISGI algorithm

| | |
|---|---|
| **Algorithm:** | **ISGI** |
| **Input:** | user set $ID$ |
| | correlation threshold $\theta$ |
| **Process:** | Generate a complete graph $G_h(V_h, E_h)$ based on all users in $ID$ |
| | **for each** $id_x \in ID$ **do** |
| |   **for each** $id_y \in ID$ **do** |
| |     Compute the global similarity of $id_x$ and $id_y$ from the Boards view (c.f. Eq. 8) |
| |     Compute the global similarity of $id_x$ and $id_y$ from the Topics view (c.f. Eq. 8) |
| |     Generate the correlation value $c$ on the edge $(id_x, id_y)$ of $G_h$ |
| |   **end** |
| | **end** |
| | Add all the edges whose correlation values are no less than $\theta$ to a new Edge set $E$ |
| | Construct a new Vertex set $V$ with $id_x, id_y$ such that $(id_x, id_y) \in E$ |
| **Output:** | interest-sharing group $G(V, E)$ |

$$hs\,(X_i, Y) = \max_{Y_j \in Y} \{w\,(X_i, Y_j)\, ls\,(X_i, Y_j)\} \tag{6}$$

where

$$w(X_i, Y_j) = \exp\left(-\frac{|i - j|}{M}\right) \tag{7}$$

and $M$ is the number of possible time slot in timeline $Y$.

Note that the local similarities are weighted by Eq. 7, which incorporates regularization that similar actions taken by two users with similar interests should not be too far from each other. The reason has been explained in Section 2.

Then, by using the hybrid similarities with respect to different time slots, we derive the *global similarity* between $X$ and $Y$ as

$$gs\,(X, Y) = \frac{1}{2}\left(\frac{\sum_{X_i \in X, X_i \neq \emptyset} hs\,(X_i, Y)}{\sum_{X_i \in X, X_i \neq \emptyset} 1} + \frac{\sum_{Y_j \in Y, Y_j \neq \emptyset} hs\,(Y_j, X)}{\sum_{Y_j \in Y, Y_j \neq \emptyset} 1}\right) \tag{8}$$

Note that only the hybrid similarities for the non-empty time slots are aggregated in Eq. 8. The reason is that in real world two users with similar interests may differ from each other by the log-in frequency. For instance, user $id_y$ may login BBS everyday, while user $id_x$ may login only once a month but does exactly what $id_y$ does. If we use the hybrid similarities for all the empty time slots, the global similarity between the two users $id_x$ and $id_y$ would be very low.

Since the global similarity in each view reveals the correlation of $id_x$ and $id_y$ in different perspective, the overall correlation between the two users is computed by simply averaging the global similarities in both views.

After correlations between all pairs of users are obtained, all the weak links whose corresponding correlation value is less than a prest threshold $\theta$ is removed from the hypothesized graph $G_h$, and the induced graph is regarded as the interest-sharing groups $G$, where the neighbors of a user $id_i$, i.e., those who are connected to $id_i$ by the links, share similar interests to $id_i$. The pseudo code of ISGI algorithm is shown in Table 1.

## 4   Predicting User Behavior Using Generated Community

In many existing work, the generated communities are only used for identifying correlated entities. Besides such a simple application, we consider another potential application which exploits the communities generated by ISGI on BBS – predicting user behavior under certain conditions.

Given a user $id_i$, now the task is to predict what action $id_i$ may take in the near future, i.e., in a time slot of size $z$ which starts at time $t$. A possible solution to this problem is to learn the probabilistic model directly from the BBS data. Since the actions that have been taken by $id_i$ in current time slot may be closely related to $id_i$'s future actions in the same time slot, the prediction may be made according to Eq. 9, where the posterior probability is estimated by consulting the access history of $id_i$.

$$P\left(a_x|A_i^{obsv};id_i\right) = \frac{\text{\# of } a_x \text{ in a time slot with } a' \in A_i^{obsv}}{\text{\# the time slots contain } a' \in A_i^{obsv}} \qquad (9)$$

where $A_i^{obsv}$ is the set of actions taken by $id_i$ in the current time slot.

In reality, however, such a method fails since $A_i^{obsv}$ is often empty. In this case, the posterior probability cannot be computed directly. This situation is common in a BBS. For instance, in order to provide a discussion recommendation, the prediction is usually required to be made as soon as the user logins to the BBS. Fortunately, with the interest-sharing groups identified by ISGI, this problem can be resolved as follows.

Recall that a community is generated based on the similar access patterns between users. If a user is likely to take an action at a time instant, other users with the similar behavior also tend to take the action at some other time instants. Thus, when the posterior probability of action $a_x$ for user $id_i$ is computed, given that $A_i^{obsv}$ is empty, we consults the neighbors of $id_i$ in the generated community for determine the possible future actions of $id_i$ according to the following equation,

**Table 2.** Pseudo-code describing the BPUC algorithm

| | |
|---|---|
| **Algorithm:** | **BPUC** |
| **Input:** | user to be predicted $id_i$ |
| | view of action to be predicted $V$ |
| | generated community $G$ |
| | time slot $TS_t$ starting at time $t$ |
| **Process:** | Fill the neighbor set $N_i$ with all the neighbors of $id_i$ in $G$ |
| | **for each** action $a_x$ on the view $V$ **do** |
| |   **for each** $id_j$ in $N_i$ **do** |
| |     Record the correlation value $c_{ij}$ between $id_i$ and $id_j$ from $G$ |
| |     Construct $A_j^{obsv}$ of with all the actions taken by $id_j$ in $TS_t$ on the both views |
| |     Estimate the posterior probability $P\left(a_x|A_j^{obsv};id_j\right)$ according to Eq.9 |
| |   **end** |
| |   Approximate the posterior probability using Eq. 10 |
| | **end** |
| **Output:** | predicted user behavior $a^* \leftarrow \arg\max_{a_x} P\left(a_x|A_i^{obsv};id_i\right)$ |

$$P\left(a_x|A_i^{obsv};id_i\right) = \frac{1}{Z} \sum_{id_j \in N_i; A_j^{obsv} \neq \emptyset} c_{ij} P\left(a_x|A_j^{obsv};id_j\right) \qquad (10)$$

where $c_{ij}$ is the correlation value between $id_i$ and $id_j$, and $Z = \sum_{id_j \in N_i; A_j^{obsv} \neq \emptyset} c_{ij}$.

Note that according to Eq.10 the estimation is done by weighting the sum of posterior probabilities of the neighbors instead of filling $A_i^{obsv}$ with the actions in $A_j^{obsv}$ first and then computing the posterior probability $P\left(a_x|A_i^{obsv};id_i\right)$ directly. The reason is that the correlations between users reflect the possibilities that two users may take similar actions at a time instant; hence, the posterior probabilities of the action $a_x$ may be "smoothly" propagated from those similar users to $id_i$. By contrast, propagating the events to $id_i$ assumes that $id_i$ should have also taken the actions that $id_i$'s neighbors have already taken, which is clearly inconsistent with the information conveyed by this community.

Based on Eq. 10, an algorithm called BPUC (Behavior Prediction Using Community), whose pseudo code is shown in Table 2, is proposed to generate the probabilities for user behavior prediction. BPUC may be used to predict what actions a given user may take in the near future. This is extremely useful in situation awareness in which we can foresee any potential event that is likely to happen as well as the likelihood associated with this event. Besides, it is also helpful in the development and the delivery of the personalized services such as discussion recommendation to the BBS users.

## 5   Experiments

### 5.1   Data Set

The data used for the experiments are extracted from the BBS of Nanjing University[1]. Currently, this system is one of the most popular university BBS in mainland China. The daily average number of online users is usually above 5000.

In the experiments, all the messages dated from January 1st, 2003 to December 1st, 2005 on 17 most popular and frequently accessed boards are collected. For each message, all the nouns, verbs and quantities appearing in the title are extracted as a bag of key words to represent a certain topic. Some different topics discussing the same issue are merged together manually for sematic consistency. After that, the topics that have been discussed by less than 5 messages and the users who have posted less than 50 messages are removed from the data set.

After the removal, the data set contains 4512 topics of 17 boards, and there are 1109 users under consideration. For each user, data are organized into two views, i.e. the Boards view and the Topics view. In each view, the sets of actions with their frequencies are ordered along the timeline. Due to the considerations on effectiveness and efficiency, the smallest time unit used in this experiment is *Day*. Thus, there are altogether 1066 time instants along the timeline, and actions taken within a day are regarded as simultaneous events.

---

[1] More information could be found by accessing this BBS at http://bbs.nju.edu.cn

## 5.2   Experiments on Community Generation

In order to evaluate whether ISGI correctly identifies the interest-sharing groups, the ground truth of the data set must be available. However, since this is a real-world BBS, it is not feasible to get all the ground-truth information as this involves the users' privacy. Fortunately, 42 volunteers have joined the experiment and told us their IDs and main interests. Based on this valuable information, an evaluation set $ES$ of 42 users is obtained. According to the main interest of the 42 users, they were roughly divided into 3 groups: 18 users are interested in modern weapons; another 12 users are fond of programming skills; and the rest of the users are fans of various computer games.

   With the availability of part of the ground truth, the performance of the ISGI algorithm is evaluated by the *neighborhood accuracy* and the *component accuracy*, respectively. The neighborhood accuracy describes how accurate the neighbors of a user in the generated community share similar interests to that of the user, while the component accuracy measures how well these generated groups represent certain interests that are common to the individuals of the groups. For instance, considering a generated community shown in Fig. 2, the number of all possible links is 21 ($= \frac{7*(7-1)}{2}$). 7 links between similar users which should be kept in the graph and 10 links between dissimilar users which should be removed are correctly identified from the 21 possible links. Thus, the neighborhood accuracy is $(7+10)/21 = 0.810$. Since 7 pairs of similar users are grouped into the same graph component and no pairs of dissimilar users are split into different group, the component accuracy is $(7+0)/21 = 0.333$.

   In the experiment, the size of the time slot used in ISGI is fixed to 5. Note that many well-known community generation methods (e.g., [1]) are essentially BDCG methods directly working on explicitly provided link information. They are not suitable for our baselines. Here, we only compares ISGI with another recently developed UDCG algorithm CORAL [9], which does not rely on explicit link information either. Due to the large number of users and the long timelines in both views, CORAL fails to generate a community from the experimental data set within a reasonable time interval. In order to report a manageable evaluation comparison between ISGI and CORAL, the original data set is reduced by downsizing the action points along the timelines by a factor of 10 such that each timeline comprises 107 time instants, and all the comparison evaluations with CORAL are reported based on this reduced data set. For simplicity, the original data set and the reduced data set is denoted by *BBS_big* and *BBS_small* respectively.
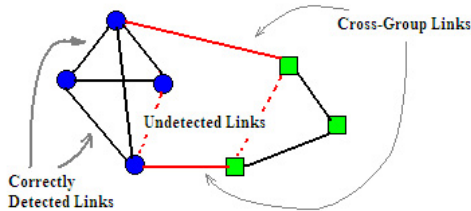


**Fig. 2.** An example of computing neighborhood accuracy and component accuracy

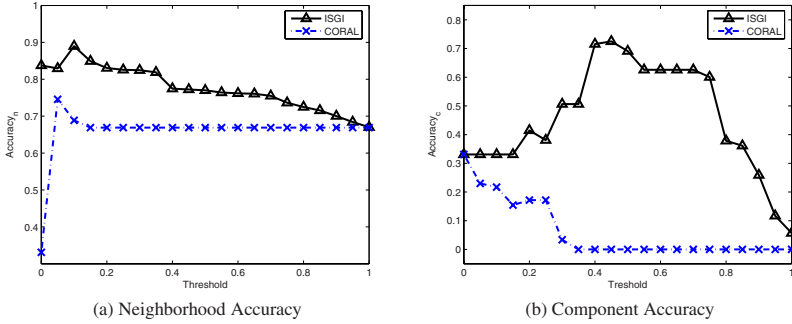(a) Neighborhood Accuracy    (b) Component Accuracy

**Fig. 3.** Accuracies of the communities generated by ISGI and CORAL on *BBS_small*

Also, since CORAL only assumes one timeline for each individual user while in ISGI two timelines are used for the two views, respectively, another version of *BBS_small* is prepared for CORAL by collapsing the two timelines together into one to ensure a fair comparison between the two algorithms.

Recall that the structure of the community is determined by a pre-set minimum correlation threshold $\theta$. In order to see how $\theta$ affects the community generation, in the experiments the value of $\theta$ varies from 0 to 1 with the step length 0.05. For each $\theta$, the correlation values on all the links in communities generated by ISGI and CORAL respectively are normalized into the range $[0, 1]$, and then the accuracy of the communities on $ES$ are measured respectively.

Fig. 3 reports the neighborhood accuracy and the component accuracy versus the threshold $\theta$, respectively. It is clear to observe from the figures that the communities generated by ISGI are always better than those generated by CORAL for different $\theta$ w.r.t. both neighborhood and component accuracies.

Interestingly, when increasing $\theta$ from 0 to 0.05 to remove links from the initial community generated by CORAL, the neighborhood accuracy climbs up from $0.331$ to the highest value $0.746$, while the component accuracy drops at the same time. By
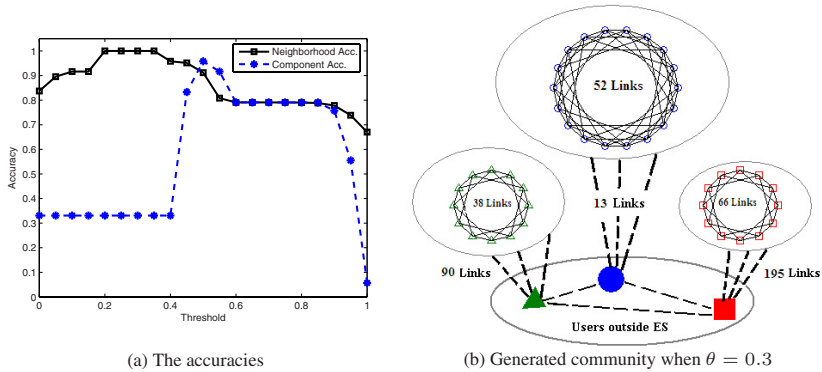


(a) The accuracies    (b) Generated community when $\theta = 0.3$

**Fig. 4.** Results of community generation using ISGI on *BBS_big*

**Table 3.** Time (in hours) taken for community generation

| Data set | ISGI | CORAL |
|----------|------|-------|
| *BBS_small* | 5.5 | 56.1 |
| *BBS_big* | 20.5 | N/A |

investigating the average number of the neighbors of a user and the number of the components when $\theta = 0$ and $\theta = 0.05$, it is found that the average number of a user's neighbors in a generated community drops dramatically from 953.1 to 64.1, and the number of the components in the community increases to 286. Therefore, it is concluded that most of the correlation values between similar users and between dissimilar users are both small such that it is difficult to discriminate links between similar users and those between dissimilar users by increasing $\theta$. In CORAL, only the frequencies of actions can be used. Neither the information on the boards where the messages are posted nor the topics that the messages are addressed are used for deriving the correlations between users. Two users who post 10 messages to $B_1$ and $B_2$ respectively are regarded as similar by CORAL, while two users who post 5 messages and 20 messages to $B_1$ are regarded dissimilar. Therefore, all these facts suggest that CORAL is not suitable for identifying the interest-sharing user groups as ISGI does.

To further illustrate the effectiveness of ISGI on the original data set, ISGI is applied to *BBS_big* to generate communities with respect to different values of $\theta$, and the accuracies of the generated communities are plotted in Fig. 4(a). Similarly, value of $\theta$ varies from 0 to 1 with the step length 0.05. As shown in the figure, ISGI performs even better on this large data set with respect to both the neighborhood accuracy and component accuracy. When $\theta$ ranges from 0.2 to 0.35, the neighborhood accuracy even reaches 1.0. Note that both accuracies of the communities generated by ISGI do not reach their corresponding maxima with the same value of $\theta$. This phenomenon is due to the incomplete evaluation set $ES$. Even if the link between two dissimilar users is removed, the users may still be in the same group since they may still be connected to some other users outside the evaluation set. Moreover, Fig 4(b) gives an insight view of the generated community when $\theta = 0.3$. It is easy to find that the 3 groups of users with different interests are exactly identified by ISGI.

In addition, the evaluations are performed on workstations with 3.0 GHz Pentium 4 hyper-thread CPU. The running time ISGI and CORAL, respectively, on *BBS_small*, and the running time of ISGI on *BBS_big* is shown in Table 3. The CPU time shows that the extensibility of ISGI is better than that of CORAL in that ISGI is able to generate from large data set while CORAL fails.

### 5.3   Experiments on User Behavior Prediction

The community generated by ISGI in Section 6.2 is used to evaluate the BPUC algorithm described in Section 5. Here the task is to predict what actions a given user might take in the near future, i.e., within a time slot of the size $z$.

For each user in the experimental data set, the actions along the timeline in each view, either Boards or Topics, are split into two parts. One part which contains the

actions taken in the first 1056 days are used for training the probability model, while the actions in the last 10 days are kept aside for testing. In the experiment, the length of the time slot, within which the predicted actions may take place, is set to 5 days. Thus, there are altogether 6 different predictive time slots in the last 10 days. Predictions are made for each time slot and the errors are averaged over the 6 time slots. When predicting the most probable action that may be taken by a user within a time slot in the last 10 days, all the actions in the corresponding time slot of the user's neighbors are considered as the observed actions and are available for use.

Two algorithms, PM and Comm, are compared with BPUC. PM is a pure probabilistic model directly learned from the training data without using the generated community. Due to the characteristics of the task specified in Section 4, where a user has taken no actions in the predictive time slot observed, it is unable to compute the posterior probability according to Eq. 10. Instead, the prediction of the most probable action taken by the user is made based on the user's *prior probability* on the action to be predicted. Comm is a method that totally bases its prediction on the generated community. It considers the most frequent action taken by a user's neighbors in the community as the most probable action taken by the user, where the frequency of an action $a_x$ is the correlation-weighted sum of the frequencies of $a_x$ taken by the neighbors.

Leave-one-out test is used. In detail, when making prediction for a user with respect to a certain predictive time slot, the actions of the other users in the corresponding time slot are available for use. The users without neighbors in the community is skipped for prediction. Note that some neighbors of a user in the generated community may take no actions in the predictive time slots. In this case, both BPUC and Comm ignore these neighbors in making the prediction. If all the neighbors are ignored, the prediction for this user is also skipped.

Since a user may take serval actions in a predictive time slot, the prediction is made correctly if the predicted most probable action appears in the given predictive time slot. Thus, the error rate with respect to a predictive time slot is computed by the ratio of the number of users whose predicted actions do not appear in the time slot over the total number of the users included in prediction. The evaluations are repeated for each of the 6 predictive time slots and the error rates are averaged to report the final error rate.

Different communities can be generated using different $\theta$, thus, the experiment is repeated on each generated community. However, as $\theta$ increases, a user may have fewer neighbors in the community. To ensure that the neighborhood size is larger than 2, $\theta$ only ranges from 0 to 0.55 with a step length of 0.05.

For each community determined by $\theta$, PM, Comm, and BPUC are used to predict the most probable boards a user might access. The error rates are tabulated in Table 4. It is obvious that BPUC and Comm outperforms PM. The average error rate of BPUC over different structures reaches 0.231, which improves 17.5% over PM on average. Moreover, even though Comm makes prediction only based on the generated community, it reaches lower error rates than PM. The average performance improvement of Comm over PM is 5.3%. Thus, the generated community is helpful to improve the prediction on the user behavior.

The average performance improvement of BPUC is higher than that of Comm. Although Comm achieves higher improvements for 6 different communities ($0.2 \le \theta \le$

**Table 4.** Error rates of compared algorithms based on the communities specified by $\theta$

| $\theta$ | 0 | 0.05 | 0.1 | 0.15 | 0.2 | 0.25 | 0.3 | 0.35 | 0.4 | 0.45 | 0.5 | 0.55 | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PM | .307 | .307 | .307 | .307 | .307 | .310 | .310 | .305 | .277 | .246 | .199 | .182 | .280 |
| Comm | .392 | .390 | .339 | .261 | .215 | .213 | .220 | .233 | .241 | .230 | .227 | .214 | .265 |
| BPUC | .249 | .249 | .232 | .225 | .226 | .236 | .241 | .260 | .247 | .242 | .197 | .174 | .231 |

0.45), it also performs worse than BPUC for the other 6 communities. By contrast, BPUC performs stably well for different structures of the communities in the experiments. This fact indicates that BPUC benefits from the combination of probabilistic model and the generated community. BPUC is more suitable for this special task than Comm which bases its predictions only on the community.

## 6    Conclusions

Bulletin board system is an important platform for information exchange and sharing. This paper attempts to mine the interest-sharing groups from the BBS data and further applies the identified groups for user behavior prediction under certain condition. The contributions of this paper are as follows:

- We have formulated a general BBS data model for community generation as a collection of BBS users represented by two timelines of actions on different views. One view stands for the boards where the messages are posted, while the other represents the topics of the posted messages.
- We have designed a hierarchical similarity function to measure the relationship between different user IDs under the formulated model. This similarity function exploits time-dependent local similarities between timelines for each view and combines them for use.
- We have proposed a uni-party data community generation method called ISGI to identify the interest-sharing user groups under the formulated BBS data model. We have proposed the algorithm that combines a probabilistic model and the identified interest-sharing groups to predict the user behavior under certain conditions, which may be very useful for applications such as situation awareness and personalized services development.

Note that two users may post a message on the same topic to the same board with totally different actual contents. Consequently, besides the boards and the topics of the posted messages, the content of a message may also be used to describe a user's interest in the future work. Moreover, the user behavior prediction is just one application of the generated communities; identifying other applications using the generated communities will also be investigated in future.

## Acknowledgement

# References

1. Bhattacharya, I., Getoor, L.: Deduplication and group detection using links. In: KDD Workshop on Link Analysis and Group Detection (2004)
2. Cohen, W.W., Fan, W.: Web-collaborative filtering: recommending music by crawling the web. In: WWW 2000, pp. 685–698 (2000)
3. Culotta, A., Bekkerman, R., McCallum, A.: Extracting social networks and contact information from email and the web. In: CEAS 2004 (2004)
4. Gibson, D., Kleinberg, J., Raghavan, P.: Inferring web communities from link topology. In: Hypertext 1998, pp. 225–234 (1998)
5. Kou, Z., Zhang, C.: Reply networks on a bulletin board system. Phys. Rev. E 76 (2003)
6. Pena-Shaff, J.B., Nicholls, C.: Analyzing student interactions and meaning construction in computer bulletin board discussions. Comp. & Edu. 42, 243–265 (2004)
7. Toyoda, M., Kitsuregawa, M.: Creating a Web community chart for navigating related communities. In: Hypertext 2001, pp. 103–112 (2001)
8. Xu, J., Zhu, Y., Li, X.: An article language model for bbs search. In: Lowe, D.G., Gaedke, M. (eds.) ICWE 2005. LNCS, vol. 3579, pp. 152–160. Springer, Heidelberg (2005)
9. Zhang, Z., Salerno, J.J., Yu, P.S.: Applying data mining in investigating money laundering crimes. In: KDD 2003, pp. 747–752 (2003)