Data Mining for Community Dynamics

Tanja Falkowski, Myra Spiliopoulou

Online platforms are virtual places where people interact for different purposes, thus forming online communities. So far several mining methods have been developed to detect these community structures as densely connected subgraphs in networks. The evolution of these networks is typically analyzed by observing changes in the interaction behavior of single members only. However, we observe that communities are highly dynamic social networks, whose members change over time. People leave gradually while new ones join but the community is perceived as still active. We therefore consider methods for modeling and discovering evolving communities and present our approach CoDyM for the analysis of community dynamics. Furthermore, we report on our first series of experiments on real data from an online community.

1 Introduction

In organizations as well as in the 'virtual world' people communicate with each other for different purposes. Groups of people inside organizations who share a concern or a passion about a topic, and who interact to extend and exchange their knowledge and their expertise are called *communities of practice* [31]. These communities of practice can often be found in organizations where people usually know each other personally and face-to-face communication is the predominant (and desirable) form of knowledge exchange. However, besides meetings in person, communities of practice usually make use of existing technological infrastructure for computer-mediated communication or computer-supported cooperative work, to coordinate, collaborate or communicate. These technologies are also used by employees who are not organized in 'official' communities to share information and to solve problems. These relationships - so-called *informal networks* [4] - are often critical to an organization and their detection is of high interest for the organization. Communities that interact via a (Web-based) community platform are referred to as *online communities* [23] or *virtual communities* [24].

Organizations are interested to support community building for many reasons, e.g. to facilitate knowledge sharing or to improve customer retention. The success of communities depends on internal factors (e.g., infrastructure, community leadership) as well as external factors (e.g., publicity in mass media). To learn more about the factors and events that influence a community's evolution we must be able to detect points in time where structural changes occurred so that in a next step the trigger for the development can be determined.

We observe that communities - in the real or virtual world - are more than just the sum of the participating individuals. At each point in time a community is defined by its current members. However, communities are highly dynamic social networks whose structure changes over time. Some communities consist of a considerable amount of members who participate over a long time and a small amount of fluctuating members but others show a higher fluctuation in its membership. We observe communities where members leave while new ones join and the community is still there even if all old members have left. Let us for example look at the 'KDD community'. We might say that the community established itself in 1989 at the first KDD workshop at IJCAI in Detroit. Several workshops followed and then evolved into an annual international conference in 1995. Since then the community obviously grew, and many of the participants back then are no longer contributing nowadays. Thus, even though the overlap of participants in 1989 and 2005 is rather small, we still consider it as the 'KDD community'. (Out of 78 authors at KDD workshop in 1989 only one person had a paper in the research track of KDD'05. Assuming that the sources do not contain any misspellings and that no one changed his or her name meanwhile). Examples such as this can be found in many areas.

A community over time must therefore not necessarily consist of the same participants who interact in a similar way. Rather, the members and interactions are fluctuating. We therefore propose to define a community as a persistent structure in a graph of interactions among fluctuating members. We model a community as an evolving structure and define a community as a set of similar community instances, i.e. dense subgraphs of interactions (subgroups) among the users under observation. Two community instances are similar if they share a minimum number of members in common. This notion of similarity allows for a continuity inside the evolving community, even if the interaction behavior changes up to a certain extend or if some actors no longer participate. If the similarity between past and current community instances becomes marginal, though, this signals that the old community is dissolved and a new one is formed.

In this article, we discuss methods for modeling and discovering evolving online communities. We furthermore present the Community Dynamics Miner *CoDyM* for the analysis and visualization of community dynamics. We discuss results of experiments on an online student community. This work has been presented in a more detailed form in [7] and [8].

2 Modeling Community Interactions

The communication or more general the interaction inside communities can be described as a graph of vertices (actors) and edges (interactions). Each community member is modeled by a vertex. Interactions are modeled as weighted edges among the interacting members. Networks as sets of nodes and edges are studied in many research fields. Examples are friendship networks [30], citation and author collaboration networks in social sciences [12, 20], genetic networks [32] and food webs [6] in biology or user networks in the Web [15].

According to Desikan and Srivastava, the levels of description of networks range from the vertex level where network measures of individual nodes are observed, to the whole graph level where basic properties of the graph are analyzed. A third level, which lies in between these two levels allows for the fact that nodes in real networks tend to form subgroups in which individuals are closer connected to each other compared to the rest of the network. This is the subgraph or community level [5].

This concept of densely connected subgroups that are only loosely connected to the rest of the graph is known in social network analysis as a *cohesive subgroup*. Cohesiveness can be quantified using several different network properties (mutuality of ties, closeness of subgroup members, frequency of ties among members, relative frequency of ties among subgroup members compared to non-members) (see, e.g. [27]). Cohesiveness is known in social sciences as an important concept, to enhance efficiency and durability of social systems.

Furthermore, in biological networks such as food webs (so-called predator-prey-relationship networks) the concept is known as *compartments*. Compartments are subgroups of species in which many strong interactions occur within the subgroup and few weak interactions occur between the subgroups. Since studied food webs consist of a rather small amount of actors (the largest studied food web consists of 181 species), detection of compartments is done manually [16].

In the field of Web communities as a set of hyperlinked Web pages, research is strongly influenced by the seminal work of Kleinberg [14] on the discovery of *hubs* and *authorities* by studying link topology. The roles of 'hub' and 'authority' are of major importance for the understanding of groups of semantically related documents. In [9], Gibson et al. describe a Web community as a core of authoritative pages linked together by hub pages. However, user communities can take more elaborate forms than captured by these two roles.

Thus, research advances encompass (a) a variety of definitions for the concept 'community', taking the interaction medium and the form of the interaction graph into account and (b) mining algorithms for the discovery of static, densely connected subgraphs, i.e. of participants that interact with each other more intensively than with the rest of the network.

So far methods have been developed to find densely connected substructures in static graphs [21] or to detect similar subgraphs [28]. But not only is the community detection an important aspect of a systematic analysis of communication behavior of participants also the tracking and especially the prediction of community development are important challenges. A projection of an ongoing development would enable organizations to implement appropriate actions in sufficient time to realize a positive impact on the community. This is particularly relevant as the environment in which communities act has a strong impact on how community support solutions can be designed and to what extent the organization can influence the interaction and output of a community.

The dynamic perspective of networks is studied by Leskovec et al., who study evolution in graphs [17]. Their methodology delivers insights to changes and trends concerning the graph properties, such as average vertex degree, with some emphasis on large networks (e.g. the Internet).

Barabási et al. [2] proposed a model for the temporal evolution of networks called preferential attachment. The hypothesis is that strongly connected nodes increase their connectivity faster than less connected actors. In [13] they showed in experiments the existence of preferential attachment for some real evolving networks.

Cortes et al. [3] propose a data structure that captures a dynamic graph over time: The authors introduce the notion of 'communities of interest' upon this structure, where a community is the neighborhood of a chosen vertex, subject to the vertex's connectivity to other vertices.

A method for the discovery of human communities in data streams has been proposed by Aggarwal and Yu [1]: They focus on indicators of a given community's evolution and use them to detect subgroups with the highest increase, resp. decrease of interaction within a graph.

Exploratory research on communities poses large demands on visualization. Software tools like SoNIA [19] and TeCFlow [11] support visualization of social networks across time by creating movies of graphs. They both work on the vertex and edge level and thus visualize changing behavior between single actors rather than the evolution of the community they belong to. It is thus not possible to explore the dynamics of subgroups. However, even though the individuals and the interactions between individuals change over time, the community might be considered by an observer as the same. In contrast, we propose to observe the temporal changes on the community level to allow for an exploration of community dynamics. We propose a model and approach to define, discover and track communities over time.

3 Tracing Evolving Communities

We first describe our approach of [8] for modeling and discovering communities as clusters of community instances. The process is illustrated in Figure 1. In Step 1 we partition the time axis into equidistant time windows. For each time window t we build a graph G_t of interactions. In Step 2, we discover community instances in each time window by clustering the graph G_t. Afterwards, we link community instances in different time windows based on similarity across the time axis (Step 3), thus building a *community survival graph* (Step 4). On this graph, a *community* is a dense subgraph of linked instances. We detect them by clustering the community survival graph to find clusters of similar community instances (Step 5). The five steps are described in the following.

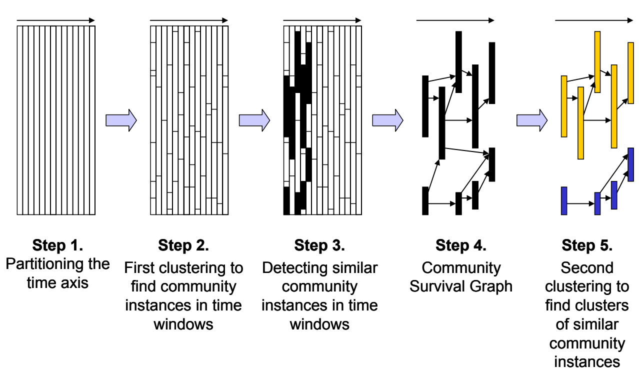


Figure 1: Community Discovery Roadmap

3.1 Clustering Users into Community Instances

We observe user interactions over time and partition the set of interactions in the first step into time windows (cf. Step 1 in Figure 1). Then we model the network of interactions in a way suitable to find communities. We do so by defining a graph $G = (V, E)$ in which V denotes the set of vertices and E the set of edges (i, j), with $(i, j) \in V$. The graph G is represented by an adjacency matrix. Each community member i is denoted a distinct vertex and an interaction between two members i and j, e.g., an e-mail exchange, is represented as an edge (i, j). We quantify the interaction between two members by assigning a weight $w(i, j)$ to the edge (i, j). Appropriate weights are 'number of messages exchanged' or the 'total length of all messages measured in characters'. An aggregation of the weights over time would favor 'old' members of the community over newcomers. Thus, we build the graph and assign the weights for each time window by updating the edge weight $w(i, j)$ in $t + 1$ as the sum of the weights over the last n periods. The edge weight is updated in $t + 1$ by adding the number of interactions that occurred in interval $t + 1$ to the current edge weight.

The transactions that occur in each time window are aggregated and considered as a static representation of the graph in the chosen interval. Next, the obtained graph G_t (Graph G in time window t) is decomposed into community instances of closely connected participants (cf. Step 2 in Figure 1). These clusters are defined as subsets of vertices within a graph with a high degree of interaction among the participants. We apply a hierarchical divisive clustering approach that partitions the graph by the iterative removal of edges into clusters. The algorithm partitions the graph iteratively into denser subgraphs by deleting edges that do not belong to the dense subgraphs but rather serve to separate them. To find these inter-community edges, Watts and Strogatz [29] have proposed a method which is based on counting short loops of edges. The motivation behind this method is that edges between communities belong to less short loops than intra-community edges as the completion of short loops requires a third actor who also interacts with a member from another community. To detect these inter-community edges one should look for edges with a low number of loops. This algorithm is suitable for dense networks with many triangles which social networks usually are [23]. However, for sparse networks or nonsocial networks with a lower transitivity the algorithm might not work well. Therefore, we apply a method proposed by [10] to detect the edges to be deleted which is based on the edge betweenness:

Definition 3.1 *The edge betweenness of an edge e in a graph $G(V, E)$ is defined as the number of shortest paths along it.*

The output of the hierarchical divisive algorithm is a dendrogram; the root is the whole graph, the leaf nodes are individual vertices. A cluster $C \subseteq V$ is a subset of vertices. A *clustering* $\zeta = \{C_1, \ldots, C_t\}$ of G is a partition of all vertices into clusters. We select the clustering with the highest modularity $Q(\zeta)$ as proposed by Newman and Girvan [22]:

$$Q(\zeta) = \sum_{C \in \zeta} \left[\frac{|E(C)|}{m} - \left(\frac{\sum_{v \in C} deg(v)}{2m} \right)^2 \right] \qquad (1)$$

$m = |E|$ and the degree $deg(v)$ of a vertex v is the number of directly connected vertices. The function Q compares the fraction of edges within a community to those that lead to vertices outside the community. Q favors graph partitionings that consist of dense subgraphs and is thus a measure of graph modularity: Values close to $Q = 1$ indicate strong community structure. According to [22], values greater than 0.3 indicate significant community structure. We thus set 0.3 as threshold for Q, to detect community instances composed of intensively interacting individuals.

There is several ways to partition the time axis into time windows. The window size is not computed automatically but we allow the user to choose an appropriate partitioning. We assume that the intervals are known and provide a mechanism to choose the interval size. To ensure that all periods show some interactions but no concentrations (see e.g. [25]), we run several experiments with different time window lengths and analyzed statistics based on indices such as the density of the graph, the average degree and the degree variance. We found that for our data set a period length of 14-days exhibits a degree distribution with a rather low standard deviation.

3.2 Evolution of Community Instances

Given a community instance found at period t, we are interested in its evolution in subsequent periods. For the conceptual definition of 'survival', we assume that a community instance is characterized by the people participating in it. However, we tolerate a fluctuation of the community members [8].

Following our MONIC framework for cluster evolution over a data stream [26], we juxtapose each cluster/community instance C discovered at period t_i with each candidate cluster of the next period t_j (cf. Step 3 in Figure 1): We define similarity among community instances that have been discovered at different times as the overlap of members between the two community instances. Thus, two instances are similar if their overlap exceeds a given threshold. In particular, let t_i, t_j be two distinct time periods, let G_i, G_j denote the corresponding graphs of interactions and let $\zeta(G_i)$, $\zeta(G_j)$ be the corresponding clusterings. Further, let $x \in \zeta(G_i)$ and $y \in \zeta(G_j)$ be two community instances. We define the overlap between two community instances as:

$$overlap(x, y) = \frac{|x \cap y|}{min\,(|x|\,, |y|)} \qquad (2)$$

where $|\cdot|$ is the number of vertices in a community instance or intersection.

The overlap threshold $\tau_{overlap}$ captures the tolerance to member fluctuation: It can be observed in many communities that individuals do not necessarily participate over time on a regular basis. Thus, former members might not show up in each interval or they participate temporarily in another community instance. Thus, the formation of the community instances is fluctuating over time and finding the same community instances over a longer period is rather unlikely. But even though the membership basis fluctuates over time, we still see the same community. This problem can be handled by demanding only for a low overlap of participants between community instances. Therefore it is not necessary that all individuals participate in each period in the same cluster instances.

In [26], we have limited the notion of cluster survival to adjacent time periods. For community evolution, we allow for matching between community instances that are two, three or as many as $\tau_{periods}$ periods apart from each other. This ensures (a) that similar community instances are connected to each other independently of their temporal proximity; subject only to $\tau_{periods}$ and (b) that communities that experience little change over time correspond to highly connected groups of community instances, i.e. to tight clusters.

Definition 3.2 *Using this notion of overlap and $\tau_{periods}$, we define a similarity function $sim(x,y)$ as follows:*

$$sim(x,y) = \begin{cases} 1 & (overlap(x,y) \geq \tau_{overlap}) \\ & \wedge(t_j - t_i < \tau_{periods}) \\ 0 & otherwise. \end{cases} \quad (3)$$

3.3 Communities in the Survival Graph

To detect communities and to determine e.g. at which point in time a community dies, the development of a single community instance is rather irrelevant. One community can show different formations along the time axis. Tracking a specific community instance does not reveal information about the community structure. We therefore do not care about the development of a single community instance but for the persistent structure of similar community instances. Those structures can be revealed by detecting clusters of similar community instances. The borders of these clusters indicate e.g. that a community died or merged with another community.

Within each time window of $\tau_{periods}$, we compare community instances and connect matching instances with an edge, thus establishing a *community survival graph* (cf. Step 4 in Figure 1). This graph is similar to the 'evolution graph' of [18]: In both cases, the nodes are clusters found at different time periods, while edges denote that the nodes match. However, Mei and Zhai perform soft clustering upon documents and study matching cluster labels rather than the clusters themselves; the graph is used to predict label evolution. We are rather interested in discovering superordinate community structures in the survival graph and relate them to their members.

Formally, let $G=(V,E)$ be the *community survival graph*. A vertex v has the form (C,t), where C is a community instance discovered at period t. An edge $e=(v,v')=((C,t),(C',t'))$ denotes that cluster C has survived into C', i.e. $t<t'$ and $overlap(C,C')>\tau_{overlap}$. The threshold value $\tau_{periods}$ ensures that $|t'-t|<\tau_{periods}$ for all pairs (t,t') that appear in edges of E. An edge is further associated with a weight, defined as the edge betweenness of the connected vertices (cf. Definition 3.1). To discover groups of community instances upon the survival graph, we again use hierarchical divisive clustering: It uses the edge betweenness of the edges to eliminate edges which separate subgraphs that are denser than their surroundings (see Step 5 in Figure 1). The number of clustering iterations is given by k.

3.4 Data Set

We conducted first experiments with real data from an online student community. The interactions that we observed on this platform are guestbook entries. Each member is represented as a vertex and two vertices are connected with an edge if at least one bilateral message exchange toke place (both members have written at least one message in the other member's guestbook). The data set contains approx. 1,000 members and 250,000 guestbook entries over a period of 18 months (June 2004 - November 2005, 75 weeks). Members are made anonymous so that it is not possible to trace back the results of the experiments to a certain individual. The results from the experiments are presented in the following in combination with the description of the visualizations.

4 Visualizing Community Dynamics

In the following we present an excerpt from [8], describing our software environment *CoDyM* that supports the above presented community discovering approach and the analysis of community evolution by visualizing the data mining results.

4.1 Visualizing the Evolution of Community Instances

To observe community transitions we track a detected subgroup over time by measuring the structural equivalence. The development of one community instance can be described and assessed by measuring and interpreting different measures which are: Stability, density and cohesion, Euclidean distance, correlation coefficient and group activity. The user can choose a community instance that has been detected in a certain time window and the resulting comparisons are shown using different visualization techniques (for a more detailed description see [7]). The measures and their interpretation are briefly described in the following.

Stability: The *stability* shows how stable the composition of the group is. The *fixed stability* indicates how many of the original members in the chosen time window are active in all other time windows. If the value is one, all members of the chosen community instance are active. For the *periodic stability* the set of members is compared to the previous time window. A low curve indicates high changes in the membership of the community instance.

Density and Cohesion: The *density* indicates the connectivity inside the group. It is the proportion of the number of edges inside the group to the number of possible edges. The higher the density the more connected the group members are. The *cohesion* indicates how connected the group is to non-instance-members. It is obtained by dividing the

Auszug aus: Künstliche Intelligenz, Heft 3/2007, ISSN 0933-1875, BöttcherIT Verlag, Bremen, **www.kuenstliche-intelligenz.de/order**

average strength of the interaction inside the community instance by the average interaction strength to actors outside the subgroup. If the cohesion increases greater than the density, the connectivity to members outside the subgroup grows faster, resulting in a less stable subgroup.

Euclidean Distance: The structural equivalence of two subgroups is measured by their *Euclidean distance*. The larger the distance, the less equivalent the groups are.

Correlation Coefficient: The *correlation coefficient* is obtained by dividing the covariance of the vector representation of the subgroups interaction graph by the product of the standard deviations. The correlation coefficient takes a value between -1 and +1. If two subgroups are structurally equivalent, the correlation is +1.

Group Activity: The actual activity of the group members is measured in number of interactions in each time window. Shown are for example the *Internal Group Activity* and the *External Group Activity*. The *Min Internal Group Activity* illustrates the number of reciprocal interactions inside the group. Thus, by comparing the *Internal Group Activity* with the *Min Internal Group Activity* the reciprocity inside the group can be determined. The *Min External Group Activity* in comparison with the *External Group Activity* can be interpreted analogously.

In our experiment with the above presented data set, we obtained 1181 community instances with more than three members. Ordered by the number of group members, the plot shows a Zipf distribution. The largest group has 45 members and the average group size is 6.4. Thus, the hierarchical edge betweenness clustering algorithm detects in our data set many small community instances. By comparing the interaction behavior of all subgroup members in different periods, we observe that according to the Fixed Correlation Coefficient, the Fixed Euclidean Distance, the Cohesion and the Density, many subgroups show up only once in the observation period. The Group Activity and the Periodic Correlation indicate, that some of the members are interacting in earlier and later time windows. Thus, most subgroups are only active in one period.

However, in this data set, no long-term stable subgroups could be found; therefore tracking a subgroup over time was not possible. The second approach in which similar subgroups are clustered solves this problem by merging similar instances to traceable communities.

4.2 Visualizing the Evolution of Communities

The specifications for the cluster analysis settings (the observation period, the similarity threshold ($\tau_{overlap}$), the maximal number of periods between similar community instances ($\tau_{periods}$) and the number of clustering iterations (k)) are done in a control panel. The outcome of the analysis process is represented in visualizations which are described in more detail in the following.

Community instances that are detected in a time window are visualized as rectangles (see Figure 2). The height of the rectangle represents the number of community members. Two community instances that are detected as similar according to $\tau_{overlap}$ and $\tau_{periods}$ are connected with an edge. The result of the second clustering is visualized by different colors of the nodes. That is, at first all community instances have the same color. Groups of similar community instances that are detected during the clustering process are visualized by different colors.

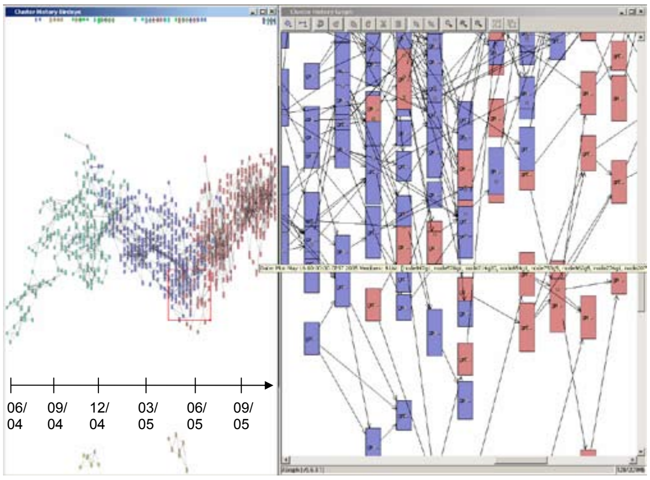


Figure 2: Clustering community instances over time (left), cutout view (right)

The x-coordinate of the vertex is determined according to the period it appears in. Thus, community instances displayed more left are detected in earlier periods. To the right, more recent community instances are displayed. All communities that are considered as similar according to the actual setting of $\tau_{overlap}$ and $\tau_{periods}$ are connected by edges. Clusters of similar community instances are displayed in the same color. In the example in Figure 2 three communities are shown that have been detected after 38 clustering iterations. One community is shown in red, another in blue and one in green.

By moving a rectangle (see red rectangle on left side in Figure 2) a cutout view of the graph (zoom in) is displayed which shows the structure of the community instances in more detail (see Figure 2 right side). The edges between rectangles represent similar community instances. If for example a community instance has edges to more than one rectangle in a later period it shows that members have separated. The horizontal box in Figure 2 on the right provides further information about a marked community instance. It shows the time period when the instance was detected and the list of participating members. The information is shown by marking a community instance.

The transformation of the survival graph to the temporal view in Figure 2 allows observing the evolution along the time axis because each community instance is displayed according to the period when it was observed. Now, the colors of the community instances indicate the borders of the clusters. The break between two differently colored clusters may show a period when a change in the community structure occurred. In our example we see two breaks that separate three communities. The changes in the structure can have different reasons, e.g. the set of participants strongly fluctuated, the interaction behavior of the participants changed, or both. This visualization reveals periods that exhibit structural changes and thus offers a starting point for users to analyze the triggers for such developments.

5 Communities of Fluctuating Members

In the following we show that the community monitor can track evolving communities and detect breaks in this evolution. We used the data set presented above. In the experiments, the similarity threshold $\tau_{overlap}$ is kept stable at 0.5 and $\tau_{periods}$ is set to 6. Thus, we obtain a graph of 1025 similar community instances in which only those community instances are included that are separated by maximal 6 periods.

The given graph of similar community instances is iteratively clustered. The first change in the evolution is detected after 27 clustering iterations, the second after 38 iterations and the third after 48 iterations. The results of the graph clustering are shown in Figure 3. The figure consists of four screenshots that display the temporal graph after a number of clustering iterations. The leftmost screenshot shows the respective graph of similar cluster instances without clustering. The second screenshot (b) shows the graph when the first break in the community evolution was detected. The third screenshot (c) shows the second break and the fourth screenshot (d) the third. The revealed communities are displayed in different colors. To improve the readability, vertical bars that indicate the period when the break occurred are added.

The obtained clustering results can be verified qualitatively in a way that ‘global events' can be related to the breaks between community clusters. The student community is especially oriented towards the integration of international students. Many international students stay at the University for one or two semesters only and this is usually the period when they participate in the community. Thus, the community members highly fluctuate over time and the highest fluctuation can be observed after the end of a semester and at the beginning of a new one.

This observation about fluctuating members corresponds to the results of our experiments. One break in the community structure can be observed during the summer break in 2005. This change can be attributed to the fact that at the end of the summer term many students gradually leave the community because their semester abroad ends. Just a few students stay in touch for a longer time after leaving Germany. Thus, the set of participants as well as the interaction behavior change at this time.

A second structural change can be observed at the beginning of the winter term 2005/06. The start of a new semester is characterized by many new members and this is especially true for the winter term as most studies start in winter. We can observe that those new members form many smaller communities very fast. These new groups sometimes grow but many students break up and others join existing or other new communities. This period thus exhibits a high fluctuation of members and interactions.

The third break can be attributed to the changing interaction behavior during the Christmas break in 2004. It could be observed that people who were active on the community platform during the Christmas break contacted students that were online too, even though they had not been interacting before. This resulted in a rapidly changing community structure, as many new edges appeared in a very short time.

6 Conclusion

The analysis of the temporal evolution of online communities gives raise to a huge amount of interesting research issues. Most of them require the cooperation between rather diverse research areas such as computer science, sociology, psychology and business administration. The presented approach defines a community as an evolving structure and offers a method - the community dynamics miner *CoDyM* - that can provide insights into the evolution of online communities and furthermore to identify triggers for structural changes. These insights are valuable not only for research fields dealing with network dynamics but also for community providers to manage community building, for example to foster intra-organizational knowledge sharing.

7 Acknowledgment

We are grateful to Jörg Bartelheimer who has implemented the mining and visualization tool and collected the data for the analysis during his diploma thesis at the Otto-von-Guericke-University Magdeburg.

References

[1] C. C. Aggarwal, and P. S. Yu, Online Analysis of Community Evolution in Data Streams, In: Proceedings of SIAM International Data Mining Conference, 2005.

[2] A.-L. Barabási, and R. Albert, Emergence of Scaling in Random Networks, Science, 286, pp. 509-512, 1999.

[3] C. Cortes, D. Pregibon, and C. Volinsky, Communities of Interest, In: Proceedings of the 4th International Conference on Advances in Intelligent Data Analysis, pp. 105-114, 2001.

[4] R. Cross, S. P. Borgatti, and A. Parker, Making Invisible Work Visible: Using Social Network Analysis to Support Strategic Collaboration, California Management Review, 44(2), pp. 25-46, 2002.

[5] P. Desikan, and J. Srivastava, Mining Temporally Evolving Graphs, In: B. Mobasher, B. Liu, B. Masand, and O. Nasraoui (eds.), Webmining and Web Usage Analysis (WebKDD'04), 2004.

[6] J. A. Dunne, R. J. Williams, and N. D. Martinez, Food-web structure and network theory: The role of connectance and size, PNAS, 99(20), pp. 12917-12922, 2002.

[7] T. Falkowski, J. Bartelheimer, and M. Spiliopoulou, Community Dynamics Mining, In: Proc. of 14th European Conference on Information Systems, pp. (on CD-ROM), 2006.

[8] T. Falkowski, J. Bartelheimer, and M. Spiliopoulou, Mining and Visualizing the Evolution of Subgroups in Social Networks, In: Proc. of 2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI-06), 2006.

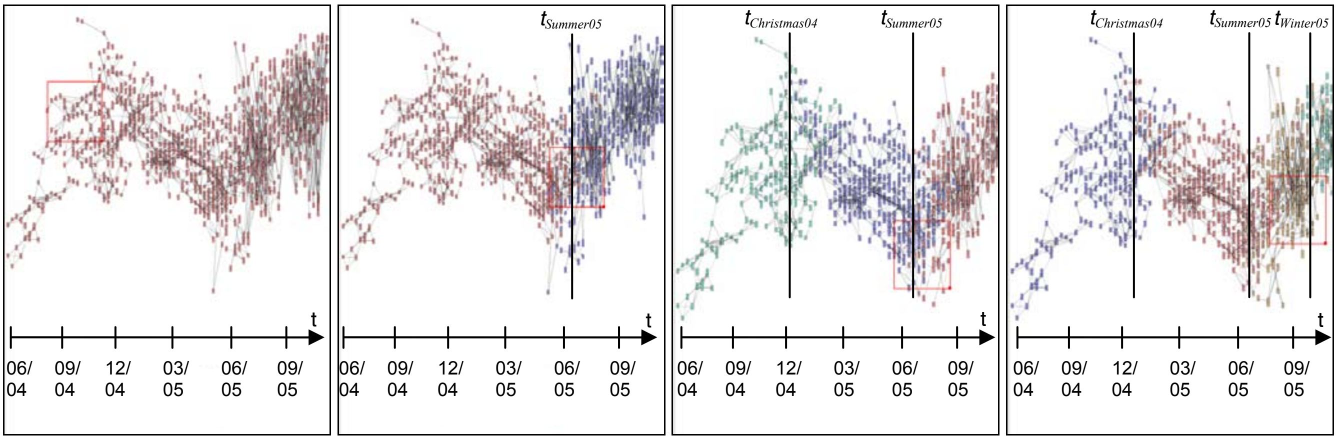
[9] D. Gibson, J. Kleinberg, and P. Raghavan, Inferring Web Communities from Link Topology, In: Proceedings of the 9th ACM Conference on Hypertext and Hypermedia, 1998.

[10] M. Girvan, and M. E. J. Newman, Community structure in social and biological networks, PNAS, 99(12), pp. 7821-7826, 2002.

[11] P. A. Gloor, and Y. Zhao, TeCFlow - A Temporal Communication Flow Visualizer for Social Networks Analysis, In: CSCW'04 Workshop on Social Networks, ACM, 2004.

[12] J. Hopcroft, O. Khan, B. Kulis, and B. Selman, Natural Communities in Large Linked Networks, In: Proc. of KDD'03, 2003.

[13] H. Jeong, Z. Néda, and A.-L. Barabási, Measuring preferential attachment in evolving networks, Europhysics Letters, 61(4), pp. 567-572, 2003.



(a) (b) (c) (d)

Figure 3: Structural breaks in community evolution for $\tau_{periods} = 6$, $\tau_{overlap} = 0.5$ indicated by vertical lines. Vertices mapped as described in Sect. 4.2. The graphs represent: a) no edges removed (red community), b) after 27 clustering iterations (red and blue communities), c) after 38 iterations (red, green and blue communities), d) after 48 iterations (red, green, brown and blue communities) - Cutout views as in Figure 2 are not shown due to space restrictions

[14] J. Kleinberg, Authoritative sources in a hyperlinked environment, In: Proc. 9th ACM-SIAM Symposium on Discrete Algorithms, 1998.

[15] J. Kleinberg, and S. Lawrence, The Structure of the Web, Science, 294, pp. 1849-1850, 2001.

[16] A. E. Krause, K. A. Frank, D. M. Mason, R. E. Ulanowicz, and W. W. Taylor, Compartments revealed in food-web structure, Nature, pp. 282-285, 2003.

[17] J. Leskovec, J. Kleinberg, and C. Faloutsos, Graphs over Time: Densification Laws, Shrinking Diameters and Possible Explanations, In: Proc. of KDD'05, 2005.

[18] Q. Mei, and C. Zhai, Discovering evolutionary theme patterns from text: An exploration of temporal text mining, In: Proc. of KDD'05, pp. 198-207, 2005.

[19] J. Moody, D. Mc Farland, and S. Bender-deMoll, Dynamic Network Visualization, American Journal of Sociology, 110(4), pp. 1206-1241, 2005.

[20] M. E. J. Newman, The structure of scientific collaboration networks, PNAS, 98(2), pp. 404-409, 2001.

[21] M. E. J. Newman, Detecting community structure in networks, European Physical Journal, B 38, pp. 321-330, 2004.

[22] M. E. J. Newman, and M. Girvan, Finding and evaluating community structure in networks, Physical Review, E 69(026113), 2004.

[23] J. Preece, Online Communities: Designing Usability, Supporting Sociability, John Wiley & Sons, Chichester, UK, 2000.

[24] H. Rheingold, The Virtual Community, Addison-Wesley, Reading, MA, USA, 1993.

[25] T. Snijders, Models for Longitudinal Network Data, In: P. Carrington, J. Scott, and S. Wasserman (eds.), Models and Methods in Social Network Analysis, Cambridge University Press, 2005.

[26] M. Spiliopoulou, I. Ntoutsi, Y. Theodoridis, and R. Schult, MONIC - Modeling and Monitoring Cluster Transitions, In: Proc. of KDD'06, pp. 706-711, 2006.

[27] S. Wasserman, and K. Faust, Social Network Analysis: Methods and Applications, Cambridge University Press, 1994.

[28] T. Washio, and H. Motoda, State of the Art of Graph-based Data Mining, ACM SIGKDD Explorations Newsletter, 5(1), pp. 59-68, 2003.

[29] D. J. Watts, and S. H. Strogatz, Collective dynamics of "small-world" networks, Nature, 393, pp. 440-442, 1998.

[30] B. Wellman, and S. Berkowitz, Social Structures: A Network Approach, Cambridge University Press, 1988.

[31] E. Wenger, R. McDermott, and W. M. Snyder, Cultivating Communities of Practice, Harvard Business School Press, Cambridge, 2002.

[32] D. M. Wilkinson, and B. A. Huberman, A method for finding communities of related genes, Proc. National Academy of Sciences U.S.A., 10(1073), 2004.

Contact

Tanja Falkowski and Myra Spiliopoulou
Otto-von-Guericke-Universität Magdeburg
Falkultät für Informatik, ITI-KMD
Universitätsplatz 2, 39106 Magdeburg
Email: {falkowski, myra}@iti.cs.uni-magdeburg.de

Tanja Falkowski studied Business Information Systems at the Technical University of Braunschweig. She is currently working as a Research Associate at the Faculty of Computer Science, Otto-von-Guericke-University Magdeburg towards her PhD on methods for the detection of subgroups in graphs and the analysis of the temporal dynamics of interaction behavior in online communities.

Myra Spiliopoulou is Professor of Business Informatics in the Faculty of Computer Science, Otto-von-Guericke-University Magdeburg. She is working on topics associated to Web mining since several years. Her recent interests include (a) detection and understanding of communities and (b) models and methods that capture change in populations and derived models/patterns.