Fuzzy Descriptions to Identify Temporal Substructure Changes of Cooccurrence Graphs

Matthias Steinbrecher[1] Rudolf Kruse[1]

1. Department of Knowledge Processing and Language Engineering
Otto-von-Guericke University of Magdeburg
Universitätsplatz 2, 39106 Magdeburg, Germany
E-mail: {msteinbr, kruse}@ovgu.de

Abstract— *Cooccurrence graphs easily grow very dense when applied to represent binary association patterns of large amounts of data. Therefore, postprocessing is needed to extract valuable information from them. We propose an approach to identify subgraphs of cooccurrence graphs that show a certain temporal behavior. This behavior is described with linguistic variables and fuzzy connectives defined over the change rate domains of certain graph measures. These measures assess graph properties whose change over time the user is interested in. To justify our proposed method, we are going to present evidence from a real-world dataset.*

Keywords— Cooccurrence graphs, Temporal change, Fuzzy description

1 Introduction

Frequent pattern mining has become a prominent method for identifying patterns in large volumes of data and led to algorithms for postprocessing these patterns [1] or transfer the underlying ideas to other data structures such as graphs [2, 3, 4]. Since the search for frequent patterns necessarily has to deal with subsets of input data, one easily runs into the problem of combinatorial explosion which is reflected by the common problem of finding more patterns than there are input data. We have addressed this issue in previous work [5], arguing that the user requires tools that allow to filter the results in order to identify only those results that meet his criteria (such as interestingness, novelty, etc.). In addition to that we also provided arguments and empirical evidence [6] that patterns usually do not arise all of a sudden but evolve or disappear rather slowly as time passes. In consequence we proposed a method that allows the user to specify linguistically (in terms of fuzzy variables) the temporal behavior of the values of association rules' evaluation measures that he is interested in. The presented algorithm thinned out the entire rule set retaining just those rules that matched the users' concepts (to some degree).

In this paper we develop this idea further while keeping the way of describing temporal behavior (explained in the background section) but transferring it to a different area of application.

This area of application comprises the identification of interesting substructures in cooccurrence graphs. These graphs arise quite naturally wherever, theoretically speaking, one is interested in the fact that two entities share some property with respect to a so-called location (which not necessarily has to be a spatial artifact but often is). Two authors being cited by the same paper [7], two persons having visited the same website [8, 9, 10], or two crimes being committed at the same location are just three examples of cooccurrences. We will elaborate the possible applications later in the future work section. All these examples can be represented by an undirected graph with the node set comprising all possible locations and the weighted edges representing the cooccurrences. A typical graph of a real-world application is depicted in Fig. 1. It represents 100 locations in a 3D gaming environment. Whenever two locations have been visited by the same players, an edge is inserted (which also gets assigned a weight drawn as the width of the edge representing the number of common visitors, but this is omitted here). It is pretty obvious that a user needs some assistance tools that allow him to identify interesting substructures (edge combinations). This becomes even more important if we have multiple such graphs representing the cooccurrences of different time frames (e. g. weeks or months). The focus of this paper is to present a straightforward yet powerful approach how to identify common substructures in a collection of cooccurrence graphs by means of linguistics expressions that address the temporal change in these patterns.

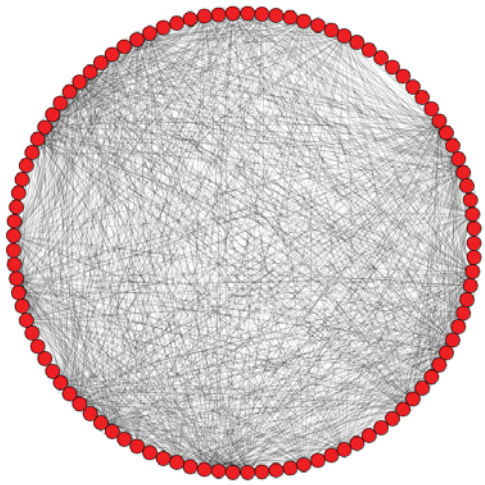


Figure 1: A typical cooccurrence graph with 100 nodes as it arose in a small-sized application (online gaming players that having visited the same locations in a 3D world). It is clear that this representation calls for some means of postprocessing in order to extract usable information from it.

The remainder of this paper is organized as follows: Section 2 introduces the notation used throughout the paper and revisits the linguistic filtering introduced in [6]. Section 3 mo-

ISBN: 978-989-95079-6-8

tivates and presents the method of extracting substructures of cooccurrence graphs that exhibit a certain user-specified temporal behavior. To underpin our proposal, we use a real-world dataset to illustrate the different stages of analysis in Section 4. Since this dataset reveals substructures that could not have been better generated manually, we refrain from handcrafting an artificial dataset and use selected subsets instead. We conclude our paper in Section 5 and propose other applications as well as possible promising extensions.

2 Background and Nomenclature

2.1 Graph Notations

In this paper we are going to deal exclusively with undirected graphs which we model as a tuple $G = (V, E)$ with vertices V and edge set E with

$$E \subseteq V \times V \setminus \{(v, v) \mid v \in V\},$$

and the constraint

$$(u, v) \in E \;\Rightarrow\; (v, u) \in E$$

to emphasize the undirected character. We will interpret the graphs as cooccurrence graphs where edges determine the number of cooccurrences (of whatever kind). This is taken into account with an edge weight function for every edge $e = (u, v)$:

$$w : E \to \mathbb{N}_0 \;\text{ with }\; w(e) = w(u, v) = w(v, u).$$

In the figures, this weight is represented as the edge width, thus we use the notion *width* and *weight* interchangeably. Given a subset $W \subseteq V$, we can induce a subgraph $G_W = (W, E_W)$ with

$$E \supseteq E_W = \{(u, v) \mid u, v \in W \wedge (u, v) \in E\}.$$

In the remainder we will sometimes use such a subset W in the context of a graph; it is G_W that we then refer to. A threshold θ defines the subgraph $G_\theta = (V, E_\theta)$ with

$$E_\theta = \{(u, v) \mid (u, v) \in E \wedge w(u, v) \geq \theta\},$$

i. e., as the graph containing only edges with a weight greater or equal to θ. Both operations can of course be combined, i. e., $G_{W,\theta}$ represents the subgraph of G induced by the node set W after having removed all edges with weight less than θ.

Since we will deal with sequences of graphs, we denote the temporal index as a superscript. All graphs share the same node set V and differ only in their edge sets or edge weights or both. Given a sequence $G^{(1)}, \ldots, G^{(n)}$ of graphs, we define the sum of these graphs as follows: $G_\Sigma = (V, E_\Sigma)$ with

$$E_\Sigma = \bigcup_{i=1}^{n} E^{(i)} \;\text{ and }\; w_\Sigma(u, v) = \sum_{i=1}^{n} w^{(i)}(u, v).$$

2.2 Linguistic Filtering Revisited

As described in [6] it is not only important to find patterns that meet some predefined constraints (such as minimum support or confidence) but also interpret these patterns in terms of temporal change. If a pattern describes a problem in some domain or an interesting customer behavior, it might be of interest to find such evolving patterns early. The underlying idea is as follows: given a pattern, we devise a set of evaluation measures that characterize the particular pattern (in [6] we used association rule measures such as support, lift, confidence, etc.). Next, the time series of a user-selected subset of these measures is calculated for every pattern. Each time series in turn is aggregated to a single value representing the overall trend, if any. The domain of this aggregate (i. e., the change rate domain) is equipped with an adequate fuzzy partition. Given a fuzzy rule antecedent (representing the user's intention of what temporal behavior of which measure(s) he is interested in), for every rule a membership degree to this concept is computed and an ordered list of rules according to these degrees is returned.

3 Spatio-temporal Filtering

3.1 Motivation

The objective of our approach is to answer questions of the following type (given a sequence of cooccurrence graphs):

> "First, what are interesting candidates for subgraphs that it would be worth looking at over time?"

and

> "Second, given a (still intractable large) set of subgraphs, which graphs become more sparse and less balanced over time?"

Before we turn to the algorithmic part of our approach, we need to negotiate which types of substructures within the graphs are most interesting to users. We will exploit the edge weights for this purpose. Several measures are needed to quantify for every subgraph aspects such as size, completeness, edge balance, etc.

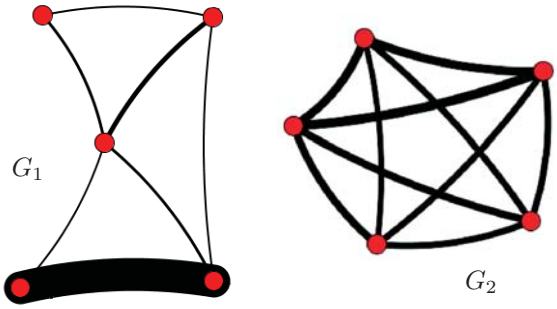
If the cooccurrence graphs represent visits of different web pages within the same online shop portal, then it might be desirable to know whether customers are able to use the web portal as intended by the owners. Are there dead ends where users are stuck? What are the "hot spot" sites, i. e., the pages that attract the most users and are visitors able to find the recently introduced shortcut to related pages? How do the accesses to the support area of the site change after renewing the navigational aids, etc.

We explicitly stress that subgraphs that are heavily interconnected with large edge weights only provide us with a *hint* that there may be an interesting visiting pattern. However, we can never conclude transitivity just from the cooccurrence graph! This is due to the fact that it only represents *binary* cooccurrences. Even a fully connected graph does not tell us anything about individual events. The sets of cooccurring events whose cardinality is represented by the edge weights even might be mutually disjoint. However, these subgraphs are found to be valuable hints that are worth being investigated.

3.2 Graph Measures

Focussing on the before-mentioned type of aspects one can identify highly connected subgraphs with large edge weights to be one type of substructures that are most interesting to users. Another type may be single edges just connecting two nodes or substructures that are highly interconnected but with

ISBN: 978-989-95079-6-8



Measure		G_1	G_2
size	Size	5.0	5.0
comp	Completeness	0.7	1.0
wght	Total Weight	5082.0	7906.0
avg	Average Weight	726.0	790.6
med	Median Weight	257.0	770.0
dev	Std. Dev.	1191.1	129.3

Figure 2: Two graphs with the same number of nodes. G_1 lacks three edges to be complete, therefore the completeness is just 0.7, whereas the clique G_2 yields 1.0. The rather large difference between average weight and median weight for G_1 (in contrast to G_2) indicate an imbalanced edge widths distribution which is strengthened by the large standard deviation value. The two graphs obviously justify this finding. The layout only acts as a visual cue and thus does not have any influence on the graph measures.

a large imbalance in the edge weights. The latter might represent two active sets of websites (large edge weights) between which users are able to navigate back and forth (numerous edges in between but with small weights since not every user is likely to use the offered navigational freedom).

The last arguments call for measures that on the one hand capture the mentioned properties of subgraphs and on the other hand allow to build a fuzzy partition on their domains since we are not going to ask for subgraphs *with 9 nodes and edges with weights greater than 100* but for *large* subgraphs with *moderately sized* edges. We use the following set of (sub)graph measures to quantify different aspects:

$$\begin{aligned}
&\text{Size} && \operatorname{size}(G_W) = |W| \\
&\text{Completeness} && \operatorname{comp}(G_W) = \frac{2\,|E_W|}{|W|^2 - |W|} \\
&\text{Edge Weight} && \operatorname{wght}(G_W) = \sum_{e \in E_W} w(e)
\end{aligned}$$

The size simply represents the number of nodes of the subgraph, whereas completeness refers to the relative number of edges compared to the maximal number. Zero represents an isolated graph (no edges) while a value of 1 designates a clique. Finally, the edge weight simply returns the sum of all edge weights without giving any clue about the distribution of these weights among the edges. Therefore, three additional measures are used: $\operatorname{avg}(G_W)$ calculates the arithmetic mean of all edge weights, $\operatorname{med}(G_W)$ returns the median of the weights and $\operatorname{dev}(G_W)$ represents the standard deviation of the weights. Fig. 2 illustrates these intentions with two graphs of the same size. Note, that these are subgraphs from real-world data and no artificial graphs that were crafted to meet the requirements.

3.3 *Candidate Graph Generation*

As we are now equipped with measures to assess certain aspects of subgraphs that we would like to track over time, the remaining question is how to determine such candidate graphs? It is clear that a brute-force approach (testing all subsets of nodes as potential subgraph node sets) fails immediately due to runtime problems, even for small node sets. We therefore promote the following heuristic: The graphs of all time frames are added as shown in Section 2 to arrive at the sum graph G_Σ (or simply the cooccurrence graph if we ignore the time frames). Next, a threshold θ is chosen and the graph's components (disconnected subgraphs) $\mathcal{C}_{G_\Sigma} = \{C_1, \ldots, C_j, \ldots, C_m\}$ of $G_{\Sigma,\theta}$ are taken as the candidate subgraphs. The choice of θ can be entirely left to the user (e. g. by offering a graphical preview tool that shows the components instantly whenever the user selects a new threshold via a slider) or θ may be determined in such a way to limit either the number of components or the (average) size of the components.

3.4 *Matching Against Linguistic Concepts*

Whatever way of determining the granularity of components is chosen, we are left with a set of mutual disjoint node sets $\mathcal{C}_{G_\Sigma}$ that are used to create a sequence of subgraphs $\left\langle G^{(i)}_{C_j} \right\rangle$, $i = 1, \ldots, n$, $j = 1, \ldots, m$ (one sequence for every subgraph induced by the node set) that are evaluated against the user-specified temporal behavior description. A time series is generated for every measure referenced in this user description. The temporal change within this time series is computed and the degree of membership to the user description is calculated. We will employ a simple regression approach, i. e., we fit a regression line into the time series and interpret its slope as an indicator of decrease, stability and increase.

The example concept from the motivation of this section is repeated here:

"Completeness is decreasing and std. deviation is increasing"

Translated into a linguistic concept, the user may specify

$$\langle\, \Delta_{\text{comp}} \texttt{ is decr} \,\wedge\, \Delta_{\text{dev}} \texttt{ is incr} \,\rangle\,,$$

which is evaluated to

$$\top\left(\mu^{(\text{decr})}_{\Delta_{\text{comp}}}(C_j), \mu^{(\text{incr})}_{\Delta_{\text{dev}}}(C_j)\right),$$

where $\top$ represents a t-norm modelling the fuzzy conjunction (we use $\top_{\min}(a,b) = \min\{a,b\}$ in this paper). Fig. 3 depicts an example subgraph consisting of 9 nodes. Five time frames are shown with the respective edge weights. The graph is obviously becoming less dense with time, i. e., the completeness is decreasing. The chart for this measure is depicted in Fig. 4. In analogy to this, Fig. 5 shows the increasing deviation of the edge weights which is attributed to the emergence of the strong cooccurrence (the sudden appearance in this case can be explained with time frames that were too large to appropriately cover the short period during which this strong cooccurrence emerged). If we equipped the change rate domains

ISBN: 978-989-95079-6-8

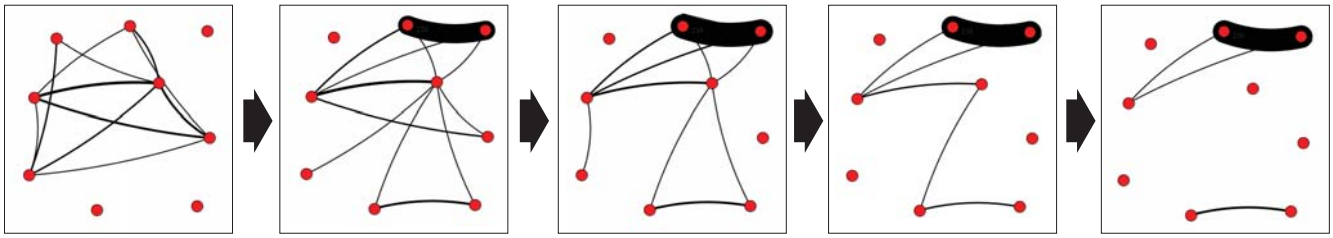


Figure 3: The temporal evolution of the graphs induced by a set of nine nodes. The number of edges is decreasing with time resulting in an almost isolated graph. Simultaneously, the edges that are remaining grow more and more unbalanced, i. e., the deviation of the edge weights is increasing. Both time series of the corresponding measures comp and dev are shown in Fig. 4 and Fig. 5, respectively.

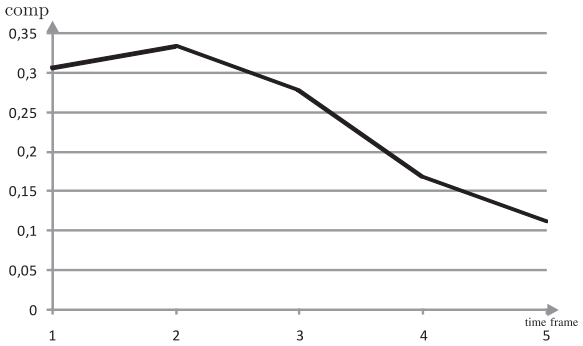


Figure 4: Time series with decreasing trend for the completeness of the edge weights for the five graphs of the time frames depicted in Fig. 3.

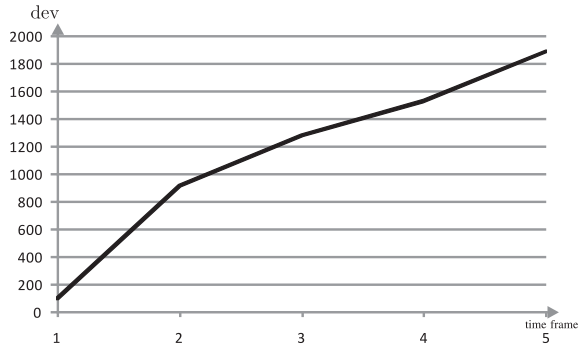


Figure 5: Time series with increasing trend for the standard deviation of the edge weights for the five graphs of the time frames depicted in Fig. 3.

(i. e., domains of the slopes of the regressions lines of the two time series) with appropriate fuzzy partitions (as we will do it in the experiments section) we could calculate the membership degree of this node set to the above-mentioned linguistic concept.

Summarizing, we state the following procedure:

1. Given a sequence $G^{(0)}, \ldots, G^{(n)}$ of cooccurrence graphs with their sum graph being G_Σ.
2. Based on an appropriate value θ, we calculate the candidate graph node sets $\{C_1, \ldots, C_m\}$ which are the vertices of the components of $G_{\Sigma,\theta}$.
3. The user provides a set of linguistic descriptions (fuzzy rule antecedents) that refer to the temporal change of the graph measures.
4. Provide fuzzy partitions for every domain of the change rate of the measures used in the descriptions of step 3.
5. Evaluate for every graph G_{C_j} the degree of membership to the linguistic concepts of step 3.
6. For every linguistic concept sort the graphs in descending order with respect to their membership degrees.

4 Experiments

Now that we have a tool at hand that allows us to determine the structural changes of subgraphs over time, we demonstrate the applicability with a real-world dataset that was already used to illustrate the examples above.

This dataset contains player contacts at certain locations within a 3D environment over a time period of six months. We carefully selected a subset of 100 such locations and discretized month-wise in order to result in a dataset large enough to justify the need for filtering while simultaneously being able to extract subgraphs that exhibit a structure and behavior that could not have been crafted more exemplary by hand. Therefore, we refrain from creating an artificial dataset with just the same structures.

Fig. 6 shows the sum graph of the dataset, i. e., the cooccurrences of six months among 100 locations. The edge weights indicate some subgraphs that are worth looking closer at. The threshold θ has been chosen to be 1000 to induce the candidate node sets.

We will match two linguistic concepts against these graph candidates: first, we are interested in decay, i. e., in graphs that show kind of a dissolving behavior, translating into a decreasing completeness and decreasing total weight. In the sample data this might indicate locations whose attractiveness is diminishing. A second concept that we would like to assess is that of an establishing pattern. An increasing average edge weight and deviation (of edge weight) might point out a phase of initial apparent random visiting of multiple locations which accumulates into a strong favored visiting pattern.

4.1 Concept 1: Decreasing Completeness and Weight

In order to evaluate the membership degrees to the linguistic concept

$$\langle\, \Delta_{\mathrm{comp}}\ \texttt{is decr} \wedge \Delta_{\mathrm{wght}}\ \texttt{is decr}\,\rangle\,,$$

ISBN: 978-989-95079-6-8

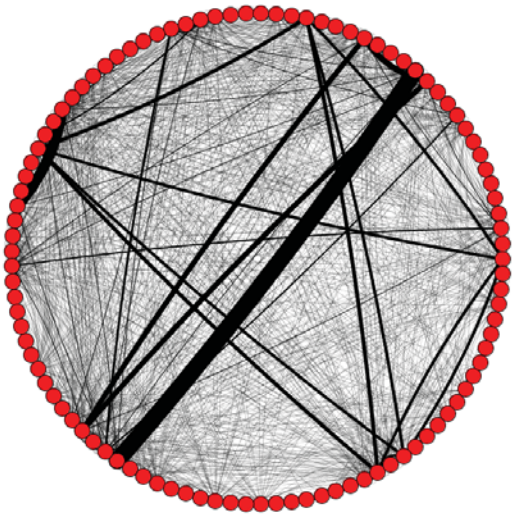


Figure 6: The sum graph of six months of visiting history of players in a 3D environment. We will match the major components (extracted via a user-specified threshold) against two linguistic concepts.

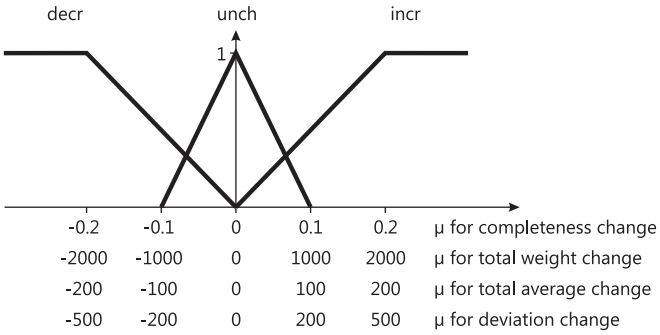


Figure 7: Fuzzy partitions for the four graph measures used in the two experiments.

we need to declare a fuzzy partition on the change rate domains of comp and wght. Fig. 7 displays all used fuzzy partitions. We apply three fuzzy sets. Note the asymmetric slopes of the borders. This setup has proven to be useful in this context since "unchanged" has a more strict semantic to users than the adjectives "decreasing" and "increasing". The respective values that determine the particular fuzzy partition can be read from the four different horizontal scales. These values have been selected with respect to the dataset since the quantity that renders a slope to be highly decreasing or increasing differs, of course, from dataset to dataset.

If we apply the linguistic concept to the candidate graphs and select the one with the highest membership (ignoring the remaining ones here for brevity), the subgraph whose history is depicted in the upper row in Fig. 8 scores 0.71. Most of the high degree can be attributed to the rapid loss of visits in the last two months. The membership degree was evaluated via

$$\min\{\mu^{(\text{decr})}_{\Delta_{\text{comp}}}(C_1), \mu^{(\text{decr})}_{\Delta_{\text{wght}}}(C_1)\} = \min\{0.71, 0.84\} = 0.71,$$

with C_1 being the set containing the five nodes. Data inspection revealed a newly set up structure which was heavily frequented shortly after opening followed by abating excitement.

4.2 Concept 2: Increasing Average and Deviation

We follow the same procedure to find the subgraph that scores best on the concept

$$\langle\, \Delta_{\text{avg}} \texttt{ is incr} \,\wedge\, \Delta_{\text{dev}} \texttt{ is incr} \,\rangle\,.$$

The lower part of Fig. 8 shows the resulting graph with a score of

$$\min\{\mu^{(\text{incr})}_{\Delta_{\text{avg}}}(C_2), \mu^{(\text{incr})}_{\Delta_{\text{dev}}}(C_2)\} = \min\{0.83, 0.89\} = 0.83,$$

The graph shows an establishing link between two nodes in parallel with a weakening in the remaining edges thus rendering the graph history becoming more unbalanced.

5 Conclusion and Future Work

In this paper we discussed the need to postprocess cooccurrence graphs if they grow very dense in practical use. We put forward a heuristic that allows to restrict the candidate node sets, and transferred the fuzzy concept matching approach from [6] to the graphical setting. We provided empirical evidence of the applicability by analyzing a real-world dataset containing six months of game player visits to 100 locations in a 3D environment.

As indicated in the introductory section, there are many other scenarios for which such an analysis might be interesting. We are most interested in applying our method to a web click stream analysis.

Since the used real-world dataset contained time stamps for every event, it is possible to generate a directed graph that also indicates which nodes were the source and target nodes for different users. This requires, of course, some heuristic with which to decide what should be the maximal period in which a user has to commit two visits to different nodes in order to assume that it was a transition and not just two independent visits.

A shortcoming of the presented approach is the heuristic to generate the candidate node sets. It worked well on the underlying datasets but one can imagine that it will return fewer and more interconnected components the more the threshold θ is reduced. Such subgraphs are also referred to as *giant connected components* [11]. To address this problem, we intend to phrase the problem in the setting of the emerging area of graph mining, more specific finding common substructures in a *single* graph [2, 12]. However, we will need a more specific subgraph definition since we have to account for the edge weights and not only the edge presence. This in turn automatically leads to another area of investigation: devising more sophisticated graph measures such as the edge betweenness centrality [13].

References

[1] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. Mining Association Rules between Sets of Items in Large Databases. In Peter Buneman and Sushil Jajodia, editors, *Proc. ACM SIGMOD Int. Conf. on Management of Data, Washington, D.C., May 26-28, 1993*, pages 207–216. ACM Press, 1993.

[2] Mathias Fiedler and Christian Borgelt. Subgraph support in a single graph. In *Proc. IEEE Int. Workshop on Mining Graphs and Complex Data (MGCS 2007 at ICDM 2007, Omaha, NE)*, pages 399–404. IEEE Press, Piscataway, NJ, USA, 2007.

ISBN: 978-989-95079-6-8

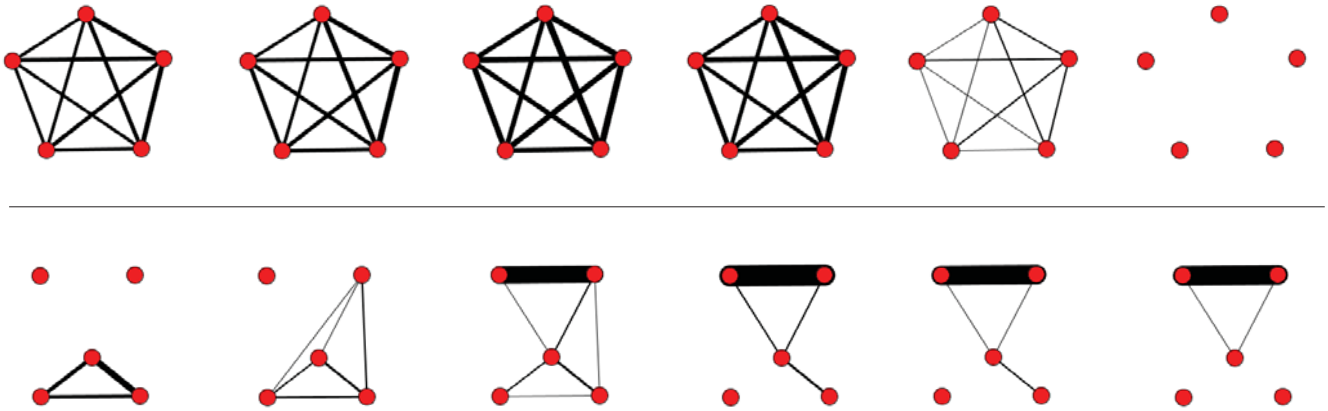


Figure 8: The histories of the two subgraphs that scored highest in the two experiments. The upper row shows a six-month development of five locations that were heavily visited but declined rather rapidly towards the end. It yielded a membership degree of 71% to the concept "completeness is decreasing and total weight is decreasing". The lower row depicts the best-scoring subgraph of the second experiment resulting in a membership degree of 83% to the concept "average is increasing and deviation is increasing."

[3] Christian Borgelt. On canonical forms for frequent graph mining. In *Workshop on Mining Graphs, Trees, and Sequences (MGTS'05 at PKDD'05, Porto, Portugal)*, pages 1–12, Porto, Portugal, 2005. ECML/PKDD'05 Organization Committee.

[4] Tobias Werth, Alexander Dreweke, Marc Wörlein, Ingrid Fischer, and Michael Philippsen. Dagma: Mining directed acyclic graphs. In IADIS, editor, *Proceedings of the ECDM 2008 (IADIS European Conference on Data Mining (ECDM))*, pages 11–17. IADIS PRESS, 2008.

[5] Rudolf Kruse, Christian Borgelt, Detlef D. Nauck, Nees Jan van Eck, and Matthias Steinbrecher. The role of soft computing in intelligent data analysis. In *Proc. 16th IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'07, London, UK)*, pages 9–17. IEEE Press, Piscataway, NJ, USA, 2007.

[6] Matthias Steinbrecher and Rudolf Kruse. Identifying temporal trajectories of association rules with fuzzy descriptions. In *Proc. Conf. North American Fuzzy Information Processing Society (NAFIPS 2008)*, pages 1–6, New York, USA, May 2008.

[7] Levent Bolelli, Seyda Ertekin, and C. Lee Giles. *Knowledge Discovery in Databases: PKDD 2006*, volume 4213/2006, chapter Clustering Scientific Literature Using Sparse Citation Graph Analysis, pages 30–41. Springer Berlin / Heidelberg, 2006.

[8] Prasanna Desikan and Jaideep Srivastava. *Advances in Web Mining and Web Usage Analysis*, volume 3932/2006 of *Lecture Notes in Computer Science*, chapter Mining Temporally Changing Web Usage Graphs, pages 1–17. Springer Berlin / Heidelberg, 2006.

[9] M. E. J. Newman. Modularity and community structure in networks. *PROC.NATL.ACAD.SCI.USA*, 103:8577, 2006.

[10] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical Review E*, 70:066111, 2004.

[11] Béla Bollobás. *Random Graphs*, chapter The Evolution of Random Graphs — the Giant Component, pages 130–159. Cambridge Univ Press, 2nd edition, 2001.

[12] Björn Bringmann and Siegfried Nijssen. *Advances in Knowledge Discovery and Data Mining*, volume 5012/2008 of *Lecture Notes in Computer Science*, chapter What Is Frequent in a Single Graph?, pages 858–863. Springer Berlin / Heidelberg, 2008.

[13] U. Brandes and C. Pich. Centrality estimation in large networks. *International Journal of Bifurcation and Chaos 17*, 7:2303–2318, 2007.

ISBN: 978-989-95079-6-8