

# On the Evolution of Contacts and Communities in Networks of Face-to-Face Proximity

Mark Kibanov   Martin Atzmueller   Christoph Scholz   Gerd Stumme  
Knowledge and Data Engineering Group, University of Kassel  
Wilhelmshöher Allee 73  
D-34121 Kassel, Germany  
Email: {kibanov,atzmueller,scholz,stumme}@cs.uni-kassel.de

**Abstract**—Communities are a central aspect in the formation of social interaction networks. In this paper, we analyze the evolution of communities in networks of face-to-face proximity. As our application context, we consider four scientific conferences. We compare the basic properties of the contact graphs to describe the properties of the contact networks and analyze the resulting community structure using state-of-the-art automatic community detection algorithms. Specifically, we analyze the evolution of contacts and communities over time to consider the stability of the respective communities. In addition, we assess different factors which have an influence on the quality of community prediction. Overall, we provide first important insights into the evolution of contacts and communities in face-to-face contact networks.

## I. INTRODUCTION

Social ties inherent in different events like academic conferences naturally lead to community structures that can be modeled in social interaction networks. Then, the dynamics and evolution in such events provide interesting directions for the understanding and possible explanation of the underlying processes. This concerns especially the evolution *across* similar events of the behavior of single links, i.e., contacts between people, but also of sets of links, i.e., communities. Understanding the evolution of communities can be useful, for example, for the prediction and recommendation services for offline social networks and ubiquitous environments.

In this paper, we consider the evolution of both contacts and communities at academic conferences. Specifically, we consider the LWA 2010, LWA 2011, LWA 2012 and Hypertext 2011 conferences, where the CONFERATOR<sup>1</sup> system [2], [3] was applied. CONFERATOR is a social conference guidance system for supporting social interactions and conference planning. Using RFID technology, it allows us to collect face-to-face contact data [7], which we can utilize for analyzing contacts and communities.

Our contribution is summarized as follows:

- 1) We analyze if the structure of the contact graphs is similar for different conferences.
- 2) We investigate the progress of face-to-face contacts during the respective conferences.
- 3) We consider automatically detected communities, and analyze the quality of the used algorithms.
- 4) Finally, we analyze how communities develop over time during a conference and whether detected communities stay stable and thus predictable.

To the best of the authors' knowledge, this is the first time, that these research questions have been addressed in the context of human face-to-face contact networks.

The rest of the paper is structured as follows: Section II discusses related work. After that, we define basic definitions in Section III. Dataset and the basic properties of the contact network are described in Section IV. Next, we provide a comprehensive analysis of the evolution of contacts and communities in Section V. Finally, Section VI concludes the paper with a summary, discussion of the results and options for future work.

## II. RELATED WORK

In this section we discuss related work concerning the analysis of community structures and human contact patterns. For our experiments, we use a new generation of active RFID tags that are able to detect face-to-face proximity (in the range of about 1-1.5 meters) of other individuals [7], [9]. These so called proximity tags are developed by the SocioPatterns collaboration and were first introduced at ESWC 2009 by Alani et al. [1]. In [8] the authors compared the participants' contact behavior with their co-authorship and activity in social web platforms. Macek et al. [15] explore the dynamics of human communication behavior. Scholz et. al [23] focus on link predictability in the context of human face-to-face contacts.

Fortunato et al. [10], [11] discuss various aspects connected to the concept of community structure in graphs and present a thorough comparison of many different state of the art community detection algorithms in graphs. Using a metric which is purely based on the structure of graphs, Newman presents algorithms for finding communities and assessing community structure in graphs, e.g., [17]. A thorough empirical analysis of different community mining algorithms and their resulting community structures is presented in [14], which is based on the size resolved analysis of community structure in graphs [13]. Atzmueller et al. [4] describe the community structure of scientific conferences. Kumar et al. [12] describe the evolution of connected component structure in graphs by the examples in the blogspace. Backstrom et al. [6] analyze community evolution in online social networks.

In contrast to previous work, we focus on the structural and temporal evolution of contacts and communities, specifically in the context of networks of face-to-face proximity.

---

<sup>1</sup><http://www.conferator.org>

### III. BACKGROUND

In the following, we first describe the basics of graphs and communities, before we summarize common community detection algorithms that are later applied in Section V-B.

#### A. Basic Definitions

An (undirected) graph  $G = (V, E)$  is an ordered pair, consisting of a finite set  $V$  containing the *vertices/nodes*, and a set  $E$  of *edges/connections* between the vertices, with  $n := |V|$ ,  $m := |E|$ . A graph is a mathematical representation of a *network*. A *weighted* graph is a graph  $G = (V, E)$  together with a function  $w : E \rightarrow \mathbb{R}^+$  that assigns a positive weight to each edge. We identify a *community* of nodes as a set of vertices  $C \subseteq V$ . The *adjacency matrix* of a graph is a matrix  $A \in \mathbb{R}^{|V| \times |V|}$  such that  $A_{u,v} = 1$ , iff  $\{u, v\} \in E$  for nodes  $u, v \in V$ . We identify a graph with its according adjacency matrix where appropriate. We consider the following properties for characterizing a graph:

- The *Density* of a graph is the ratio of the number of edges and the number of possible edges, i. e.,  $\frac{m}{\binom{n}{2}}$ .
- The *Clique Number*  $CN$  denotes the number of vertices in the largest graph clique.
- The *Normalized Clique Number*  $NCN$  is given as  $\frac{CN}{n}$ .
- We define the *Radius* as the longest shortest path between two nodes in a graph.
- The *Transitivity* measures the probability that the adjacent vertices of a vertex are connected. Transitivity is also called *clustering coefficient*. Adjacent vertices are vertices connected by an edge.

The concept of a *community* intuitively describes a group  $C$  of individuals out of a population such that members of  $C$  are strongly “related” among each other but sparsely “related” to individuals outside of  $C$ . This notion translates to vertex sets  $C \subseteq V$  of a graph  $G = (V, E)$ .

The *share of inner edges* is a natural and simple measure to determine the quality of a partitioning of a graph with  $k$  communities  $C_1, \dots, C_k$ , where  $C_i \subseteq V$ .

$$\frac{\sum_{i \in 1..k} C_i.in}{\sum_{i \in 1..k} C_i.out + \sum_{i \in 1..k} C_i.in}, \quad (1)$$

where  $C_i.in$  is the number of edges within the community  $i$ , and  $C_i.out$  is the number of edges which connect the community  $i$  with other communities.

Another prominent measure to determine the amount of relatedness is given by the *modularity* MOD [16], [17]. It focuses not only on the number of inner edges *within* a community but also compares that with the *expected* number of edges. The expected number is computed with a help of a null model (i.e., a corresponding random graph where the node degrees of  $G$  are preserved):

$$MOD = \frac{1}{2m} \sum_{u,v \in V} \left( A_{u,v} - \frac{d(u)d(v)}{2m} \right) \delta(C(u), C(v)), \quad (2)$$

where  $C(i)$  denotes the community to which node  $i \in V$  belongs.  $\delta(C(u), C(v))$  is the *Kronecker delta* symbol that equals 1 if  $C(u) = C(v)$ , and 0 otherwise.

#### B. Community Detection Algorithms

For the automatic detection of communities, we used several prominent state-of-the-art algorithms: *InfoMap*, *Label Propagation*, *Leading Eigenvector* and *Walktrap*.

The *InfoMap* algorithm [21], [22] is based on the map of random walks. To describe a random walk, the nodes of the graph should be given unique names using Huffman code. The map equation can tell us how efficient the optimal code would be for any given partition. To find an optimal partition, the algorithms calculates a theoretical limit of how concisely we can specify a network path using a given partition structure and chooses the one with the lowest limit. This partition has the shortest description length.

The *Label Propagation* algorithm [20] uses only the network structure to detect the communities. The label (which denotes the community) of the node is determined by labels of its neighbors: the node chooses to join the community to which the maximum number of its neighbors belong. At the beginning the algorithm initializes each node with unique labels and then propagates labels through the network. Thus the members of each densely connected group are assigned the same label and may start to expand outwards.

Newman’s *Leading Eigenvector* method [18] for community detection is based on the idea of modularity maximization. The maximization process can be written in terms of the eigenspectrum of a special matrix – the modularity matrix. After calculating this matrix, the algorithm finds the eigenvector corresponding to the most positive eigenvalue of the modularity matrix and divides the network into the groups according to the signs of the element of this vector.

The *WalkTrap* algorithm [19] is based on short random walks and utilizes the idea of hierarchical clustering: the algorithm computes the distances between all adjacent vertices. Then, in the each step the algorithm chooses two communities to merge, so that a mean of the squared distances between each vertex and its community is minimized (this criterion is defined as the distance between two communities) and afterwards updates the distances between communities.

### IV. DATASET

In the following, we first describe the setup used for collecting the face-to-face contact data. After that, we describe the datasets collected at four different conferences and analyze basic structural properties of the different conference contact networks.

#### A. RFID Setup

At the LWA 2010, 2011, 2012 and Hypertext 2011 conferences we asked each participant to wear proximity tags, so they could use the CONFERATOR [2], [3] system. These tags can detect close-range face-to-face proximity (1-1.5 meters) of the participants wearing them [9].

As in [24], we record a face-to-face contact when the length of a contact is at least 20 seconds. A contact ends when the proximity tags do not detect each other for more than 60 seconds. For more information about the proximity sensing technology, we refer to the SocioPatterns<sup>2</sup> collaboration [7].

<sup>2</sup><http://www.sociopatterns.org>

Event	No. of Participants		No. of Contacts		No. of Edges in Contact Graph	
	all	$\geq 180s$	all	$\geq 180s$	all	$\geq 180s$
LWA 2010	77	76	5154	1989	1004	562
LWA 2011	42	39	1423	570	300	167
LWA 2012	42	41	2745	1178	478	298
HT 2011	69	64	1902	730	550	269

TABLE I. SUMMARY STATISTICS OF THE APPLIED DATASETS (ALL CONTACTS, CONTACTS WITH DURATION  $\geq 180$  SECONDS).

Parameter	LWA 2010	LWA 2011	LWA 2012	HT 2011
Density	0.34	0.35	0.56	0.23
CN	16	8	14	10
NCN	0.21	0.19	0.33	0.14
Radius	3	3	2	7
Transitivity	0.57	0.52	0.72	0.43

TABLE II. BASIC CHARACTERISTICS OF CONTACT GRAPHS. THE ABBREVIATIONS MEAN: CN – CLIQUE NUMBER, NCN – NORMALIZED CLIQUE NUMBER

### B. Basic Properties of Communication Graphs

As shown in Table I, for LWA 2010 and Hypertext 2011 more participants utilized the system (77 and 69 participants) compared to LWA 2011 and LWA 2012 (42 participants each). The LWA conferences typically exhibit a certain continuity concerning the participants as these conferences serve as an event for professional exchange of the german data mining community.

Overall, comparing the different graphs, see Tables II, III, we observe, that especially LWA 2010 and 2011 exhibit many structural similarities with respect to the parameters density, normalized clique number, radius and transitivity. The contact graph during LWA 2012 is rather dense compared to LWA 2010 and LWA 2011. Furthermore, the participants of Hypertext 2011 seemed to be less active concerning their face-to-face contacts: the density, transitivity and clique number to number of participants are smaller and the radius bigger, compared to the LWA conferences. However, this can be explained by the fact that Hypertext is visited by many scientists from different countries which may not be as well “connected” as the german computer science community at the LWA conferences.

Considering the contact graphs containing only contacts with a duration of at least three minutes, we observe, that these contain approximately half of edges of usual graphs (see Table I) but show the same trends (cf., Tables II, III): LWA 2010 and 2011 exhibit the same structural indicators, except the radius, and the participants of the Hypertext conference are less active. Thus, the comparison of long conversations (180 seconds or longer) shows the same similarities and differences between different conferences despite containing only half of the edges of original graph.

Parameter	LWA 2010	LWA 2011	LWA 2012	HT 2011
Density	0.20	0.23	0.36	0.13
CN	12	6	10	5
NCN	0.16	0.15	0.24	0.08
Radius	7	5	3	8
Transitivity	0.43	0.39	0.56	0.28

TABLE III. BASIC CHARACTERISTICS OF CONTACT GRAPHS, CONSIDERING ONLY CONTACTS WITH A DURATION OF AT LEAST 180 SECONDS. THE ABBREVIATIONS MEAN: CN – CLIQUE NUMBER, NCN – NORMALIZED CLIQUE NUMBER

## V. ANALYSIS

In this section, we analyze the development of the contact network for each conference. After that, we compare automatically detected communities, and consider the evolution of communities in terms of stability for the different conferences.

### A. Evolution of Contacts

For analyzing the evolution of the contact graphs of the conferences, we considered all subsequent time periods ongoing during the conference, i. e., we excluded nights and times when the conference was closed. As shown in Figure 1, the number of edges in contact graph grows nearly linearly during all three LWA conferences. The number of new contacts at the beginning and at the end of these conferences can be explained by the small number of participants who come early or stay longer. An interesting fact for the Hypertext conference is a slow growth of contacts during the second part of the conference. This “tail” is much longer compared to the end of the LWA conferences. We assume that the Hypertext conference has a different “social profile”, so the participants are more focused on “socializing” during the first day. It is rather natural for a conference to reach a point when the contacts become less active. This might be a reason for the “tails” during the last days of the considered conferences. The number of edges in graphs for contacts that are 3 minutes or longer grows also linearly, but twice as slow (cf., Figure 1).

Figure 2 considers the full length of all conversations which grows similar to the growth of the number of edges in the contact graph. The average value of the overall conversation length stays similar or grows slowly throughout the whole conference (cf., Figure 3). Meanwhile the median length stays almost constant and rather low during all conferences. This points to the power law distribution during the whole conference (this distribution was also shown by Macek et. al [15], for the whole contact graph of the Hypertext 2011 conference). The constantly rapidly growing standard deviation of the contact length of two random persons shown in Figure 2 confirms the growing diversity of contacts during the conference: there are more short contacts (growing “tail” if the power law distribution); also the length and number of longer contacts lead to the growth of average conversation length. We observed a decreasing average length of the conversations and its standard deviation during the first ten hours of the Hypertext conference. This time period includes the workshop day and the first hours of the first conference day. This observation confirms the nature of workshops where participants often hardly know each other and thus longer conversations are more seldom. A lot of participants who do not visit the workshops come to the conference and vice versa, so during the first two hours of the first day of conference the average contact length continues to decrease and only starts to increase afterwards.

Another important observation shows that graphs with “long” talks ( $\geq 180$  seconds) have almost half of the number of edges of the graphs with all conversations, but their total length is equal to 80% – 90% of the whole length of the whole graph. This observation also explains why the community detection algorithm were much more efficient when applied on the weighted graphs: about half of the edges in the non-weighted graphs represent “short” contacts ( $< 180$  seconds).

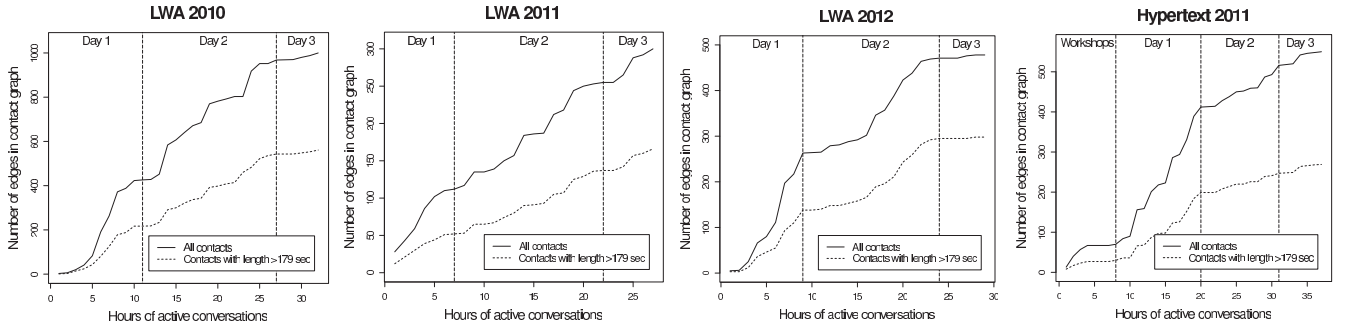


Fig. 1. Temporal evolution of the number of edges in the contact graph of different conferences.

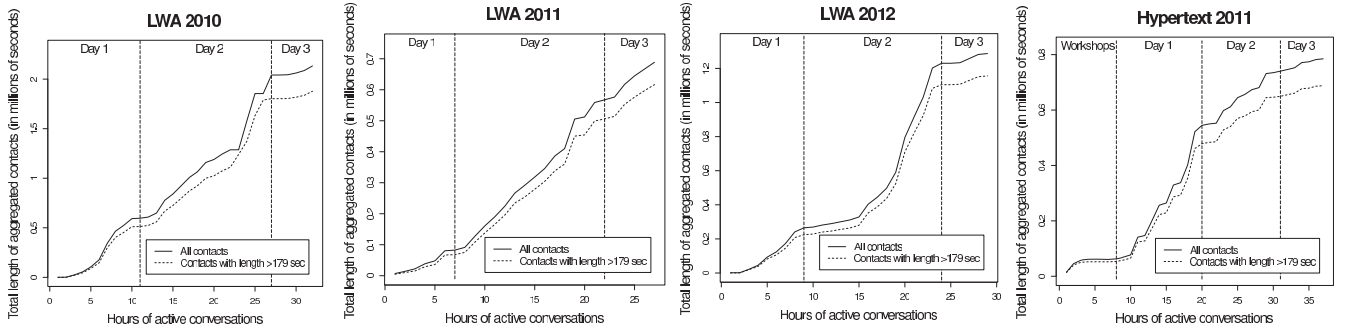


Fig. 2. Temporal evolution of the total length of the aggregated contacts of different conferences.

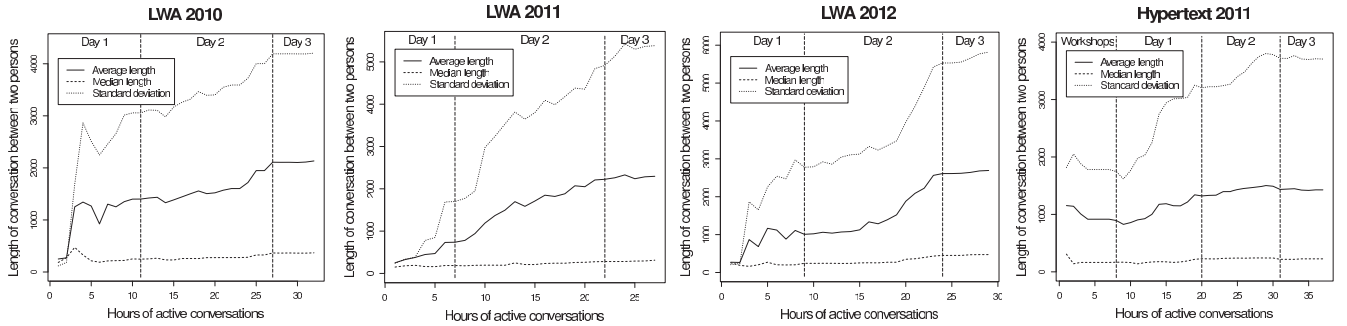


Fig. 3. Temporal evolution of the average, median length of the aggregated contacts and their standard deviation of different conferences.

## B. Community Detection

For automatic community detection, we applied the algorithms briefly summarized in Section III. We utilized two measures to estimate the quality of algorithms for different conferences (as described in Section III), i.e., the share of inner edges, and the modularity. First, we considered the non-weighted contact graph of each conference, for which we assigned a default weight of 1 to each edge. There are no algorithms which determine communities with modularity larger than 0.20 (cf., Table IV). On the other hand, the number of inner edges is rather high. The reason is the small number of detected communities, so the majority of edges “stay” inside the communities.

Applying the same algorithms to the contact graphs weighted with the individual contact lengths, we observed an increase of the number of detected communities and their modularity (cf., Table V). The *Leading Eigenvector* algorithm

does not consider edge weights of the graph and thus delivers the best performance for non-weighted or uniform-graphs (measured by modularity) and the weakest performance for the weighted graphs. Overall the modularity of the communities lies between 0.30 and 0.55 for weighted graphs; the only exception is the *Label Propagation* algorithm applied to LWA 2010 graph).

Three out of four algorithms could detect better communities – in terms of share of inner edges and modularity) – in the weighted graph. As shown before (cf., Subsections IV-B, V-A) half of the edges of these face-to-face networks are short talks, which may create some “noise” in the data. We believe this noise prevents the reliable community detection in the non-weighted and uniform weighted graphs. The *InfoMap* and *Walktrap* algorithms both use random walks for community detection and deliver similar results regarding number of communities and their modularity.



Algorithm	Parameter	LWA 2010	LWA 2011	LWA 2012	Hypertext 2011
<b>InfoMap</b>	No. of Communities	2	2	2	2
	No. of inner nodes	1.00	1.00	1.00	1.00
	Modularity	0.002	0.007	0.004	0.007
<b>Label Propagation</b>	No. of Communities	2	2	2	2
	No. of inner nodes	1.00	1.00	1.00	1.00
	Modularity	0.002	0.007	0.004	0.007
<b>Leading Eigenvector</b>	No. of Communities	3	3	3	2
	No. of inner nodes	0.49	0.52	0.41	0.63
	Modularity	0.10	0.17	0.07	0.12
<b>Walktrap</b>	No. of Communities	11	7	8	17
	No. of inner nodes	0.43	0.53	0.57	0.29
	Modularity	0.06	0.15	0.03	0.09

TABLE IV. PROPERTIES OF GRAPH COMMUNITY STRUCTURES DETECTED IN NON-WEIGHTED CONTACT GRAPHS

Algorithm	Parameter	LWA 2010	LWA 2011	LWA 2012	Hypertext 2011
<b>InfoMap</b>	No. of Communities	11	7	6	10
	No. of inner nodes	0.40	0.37	0.32	0.26
	Modularity	0.29	0.53	0.38	0.53
<b>Label Propagation</b>	No. of Communities	4	8	6	9
	No. of inner nodes	0.88	0.26	0.36	0.31
	Modularity	0.07	0.52	0.35	0.51
<b>Leading Eigenvector</b>	No. of Communities	3	3	3	2
	No. of inner nodes	0.49	0.52	0.41	0.63
	Modularity	0.10	0.17	0.07	0.12
<b>Walktrap</b>	No. of Communities	17	7	7	11
	No. of inner nodes	0.22	0.41	0.37	0.31
	Modularity	0.32	0.52	0.37	0.53

TABLE V. PROPERTIES OF GRAPH COMMUNITY STRUCTURES DETECTED IN WEIGHTED CONTACT GRAPHS

Conference	Day	Part.	(IM)	(LP)	(LEV)	(WT)
LWA 2010	1	63	202	142	637	194
	2	68	394	1839	857	369
	3	52	127	109	214	114
LWA 2011	1	34	156	204	180	189
	2	40	181	215	277	167
	3	36	146	73	163	175
LWA 2012	1	38	97	191	167	115
	2	33	358	190	258	124
	3	24	53	80	53	33
HT 2011	1	61	224	222	444	236
	2	49	119	208	253	223
	3	29	53	62	82	98

TABLE VI. THE NUMBER OF CONFERENCE PARTICIPANTS AND DETECTED C-PAIRS FOR DIFFERENT DAYS OF CONSIDERED CONFERENCES. THE ABBREVIATIONS MEAN: IM – INFOMAP, LP – LABEL PROPAGATION, LE – LEADING EIGENVECTOR, WT – WALKTRAP

### C. Evolution of Communities

We consider the stability of communities in order to investigate the nature of their development during the conferences, and the possibilities to predict future communities. For analyzing the stability of community structure we define a *c-pair* (*Community-pair*) as follows: if two nodes  $u$  and  $v$  belong to the same community,  $cp = (u, v)$  is a *c-pair*.  $CP$  denotes the set of all possible *c-pairs*. The more *c-pairs* stay over time, the more stable is a community structure. We consider only communities of the weighted graph (cf., Subsection V-B), and compute communities for each of the conference days (day 1, 2, 3) for investigating the stability of the respective communities.

It is clear, that the larger the number of participants and the larger the sizes of the communities, the more *c-pairs* there will be. This is reflected by our data: For each of the four applied algorithms, the number of *c-pairs* detected during the different conference days clearly correlates with the number of participants during these days (cf., Table VI): the more persons

take part on conference during the day, the more *c-pairs* are expected. Another observation is a usually higher number of *c-pairs* when we use the *Leading Eigenvector* algorithm. The reason is that this algorithm tends to detect larger communities than the other three algorithms. The largest number of *c-pairs* is usually detected during the second day. As shown in Subsection V-A most of the communication happens during the second day; usually the respective evolution graph shows no “flat” plateaus when the conversations are less active. As an exception, the first day of Hypertext 2011 had similar properties: there are more *c-pairs* found during the Day 1 than during the Days 2 and 3 of the Hypertext.

To estimate and compare the stability of communities during different conferences, we applied a “simple” predictor

$$P : I \times J \rightarrow CP,$$

where  $I \subseteq \mathbb{N}, J \subseteq \mathbb{N}$ . This predictor assumes that all the *c-pairs* that were built during (a) reference day(s) in  $I$  will be also formed during the subsequent day(s) in  $J$ . In the case where  $I$  and  $J$  contain only single elements, we will drop the set notation for simplicity. Let  $CP_i$  be the set of *c-pairs* of day  $i$ :  $CP_i = \{(u, v) \mid u, v \in C_j \subseteq V_i\}$ , where  $V_i$  is the set of the nodes of the contact graph of the day  $i$ . We applied the predictor five times as described below – for each algorithm and each conference. For computing the ‘correct’ predictions, we consider the intersection with a subsequent day, and the respective *c-pairs*. The more *c-pairs* are predicted correctly, the more stable is the computed community structure.

Below, we also define precision and recall for each of the relevant cases. Since some participants missed some days of the conference (or did not attend the conference at all), we eliminated these nodes in the respective networks for the analysis. This was necessary in order not to affect the our stability measures such as *Precision* and *Recall*.

**Day 1 predicts Day 2:** The predictor  $P$  is given by  $P(1,2) = CP_1$ . The correct predictions are  $CP_1 \cap CP_2$ . The recall is then given by

$$Recall = \frac{|CP_1 \cap CP_2|}{|CP_2 \cap \{(u,v) \mid u,v \in V_1\}|}, \quad (3)$$

The precision is defined as

$$Precision = \frac{|CP_1 \cap CP_2|}{|CP_1 \cap \{(u,v) \mid u,v \in V_2\}|}, \quad (4)$$

**Day 1 predicts Day 3:** The predictor  $P$  is given by  $P(1,3) = CP_1$ . The correct predictions are  $CP_1 \cap CP_3$ . Recall and precision are defined analogously as above.

**Day 2 predicts Day 3:** The predictor  $P$  is given by  $P(2,3) = CP_2$ . The correct predictions are  $CP_2 \cap CP_3$ . Recall and precision are defined analogously as above.

**Joint Day 1 and Day 2 predict Day 3:** The predictor  $P$  is given by  $P(\{1,2\},3) = CP_{1,2}$ , where  $CP_{1,2}$  is the set of c-pairs of the aggregated community structure for days 1 and 2. The correct predictions are given by  $CP_{1,2} \cap CP_3$ . Recall and precision are defined as follows:

$$Recall = \frac{|CP_{1,2} \cap CP_3|}{|CP_3 \cap \{(u,v) \mid u,v \in V_1 \cup V_2\}|} \quad (5)$$

$$Precision = \frac{|CP_{1,2} \cap CP_3|}{|CP_{1,2} \cap \{(u,v) \mid u,v \in V_3\}|} \quad (6)$$

**Intersecting Day 1 and Day 2 predict Day 3:** Intuitively, if a c-pair existed during both days, it should also exist during the third day. The predictor  $P$  is given by  $P(\{1,2\},3) = CP_{1,2}$ , where  $CP_{1,2} = CP_1 \cap CP_2$  is the set of c-pairs occurring in both community structures for days 1 and 2. The correct predictions are given by  $(CP_1 \cap CP_2) \cap CP_3$ . Recall and precision are defined as follows:

$$Recall = \frac{|(CP_1 \cap CP_2) \cap CP_3|}{|CP_3 \cap \{(u,v) \mid u,v \in V_1 \cap V_2\}|}, \quad (7)$$

$$Precision = \frac{|(CP_1 \cap CP_2) \cap CP_3|}{|CP_1 \cap CP_2 \cap \{(u,v) \mid u,v \in V_3\}|}, \quad (8)$$

Figure 4 shows the respective recall and precision values. The larger the value of precision, the more c-pairs from the “training”-day tend to appear also during the “result”-day. The larger the value of recall, the less new c-pairs tend to appear during the “result” day. The type of the point defines the applied algorithm and the color of the point defines the conference: e. g., red circles show recall and precision of predictions made by the InfoMap algorithm for the LWA 2010 conference. There are some trends that can be observed: The LWA 2011 data (green points) tend to show a better performance compared to the other conferences and thus we assume the community structure during LWA 2011 is more stable. Similarly, the communities of LWA 2012 are also rather well “predictable”. A potential explanation is given by the significant community structure of the four special interest groups constituting the LWA conferences, see [4].

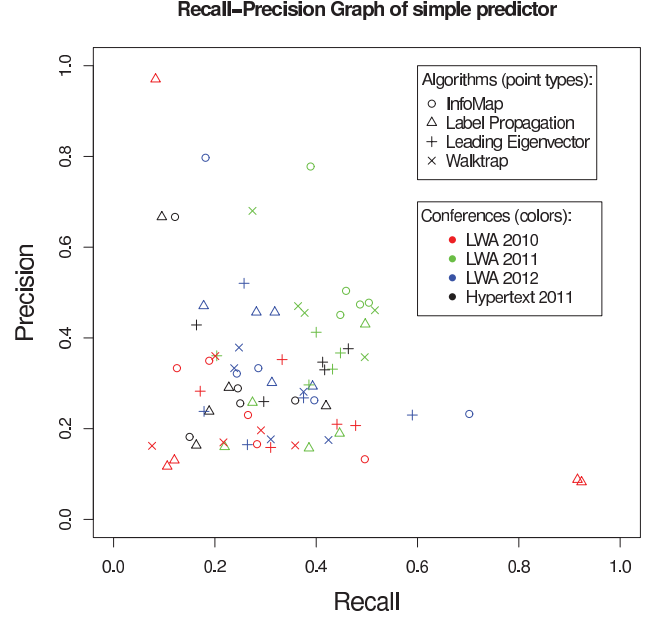


Fig. 4. Recall-Precision Graph of the “simple” predictor of the considered algorithms (marked by point types) and conferences (marked by colors)

#### Boxplot of F1 score for different algorithms and conferences

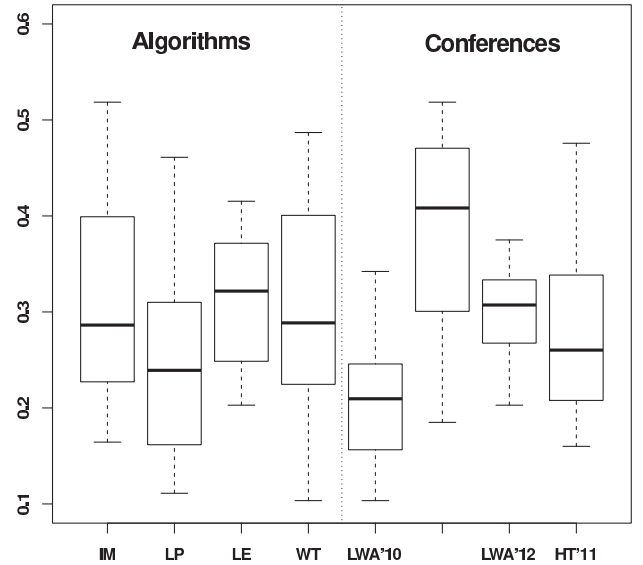


Fig. 5. Boxplots of F1 score of different algorithms (on the left side) and conferences (on the right side). The abbreviations mean: IM – InfoMap, LP – Label Propagation, LE – Leading Eigenvector, WT – Walktrap

For summarizing both precision and recall, we apply the F1 score:

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (9)$$

We calculated the F1 score for each of the 80 predictions we made, i. e., five times for each combination of an algorithm and a conference. Figure 5 shows boxplots of the F1 score for the each applied algorithm and each conference. Each boxplot is based on 20 F1 score values e. g., left boxplot “Algorithm “IM”

consists of F1 values of all predictions made for the InfoMap algorithm (5 predictions for each conference) and the boxplot “Conference “LWA 2010” consists of predictions made for the LWA 2010 conference (5 predictions for each algorithm).

The choice of the community detection algorithm did not have a big impact on the performance of our “simple” algorithm and thus on the obtained communities. The *Label propagation* algorithm has a slightly weaker average performance and thus computes less stable communities. The *Walktrap* and *InfoMap* algorithms show similar performance which corresponds with their similar working methods (cf., Section III-B) and similar detected community structure (cf., Section V-B). The choice of the event has a crucial influence on the stability of the communities: The F1 scores confirm the stability of community structure computed for the LWA 2011 conference (green points in Figure 4). The stability of the community structures detected for the LWA 2012 conference show the smallest deviation (The F1 score lies between 0.2 and 0.4). Thus, all algorithms show similar performance with respect to the community stability for predictors, i.e., for the different days.

As another interesting observation, the active communication does not make communities stable – even vice versa. Comparing the LWA 2011 and LWA 2012 conferences with the similar number of participants, we see that the LWA 2012 communications were less active than those at the LWA 2011 in terms of graph density and the total length of communication cf. Sections IV-B and V-A); overall, we observe more stable communities during LWA 2011. We observed the same phenomenon considering LWA 2010 and HT 2011 – two conferences with the same number of participants but very different dynamics of face-to-face communications (cf. Sections IV-B and V-A). On hypothesis for explaining the negative correlation of community stability and communication is the following: the participants stick to the known persons and tend to have less contacts with new persons which implies both lack of new contacts and stability of the existing communities over the whole conference.

The high number of *c*-pairs (caused by active communication) during one or another day may cause a high precision and recall value and thus a high F1 score.

So far, our proposed measures compare the overall stability of communities of different conferences. However, in order to clarify that these stabilities are significant and not accidental, we need to compare them to some “neutral” standard. Therefore, we compute a null model based on “concentration” of *c*-pairs in the “prediction” graph(s), i.e., the graph(s) used for making the predictions.

The null model  $NM$  can be computed using the following formula:

$$NM = \frac{CP_t}{\frac{n \times (n-1)}{2}} \times CP_{t+1}, \quad (10)$$

where  $CP_i$  is the number of *c*-pairs at day  $i$ , and  $n$  is the number of nodes in the considered graph. As the participants of different days of the conference are not the same, we consider only persons who visited both days, so  $n = |V|$  and  $V = V_i = V_{i+1}$  in this case.  $\frac{n \times (n-1)}{2}$  is the maximal number of possible edges in the network with  $n$  nodes.

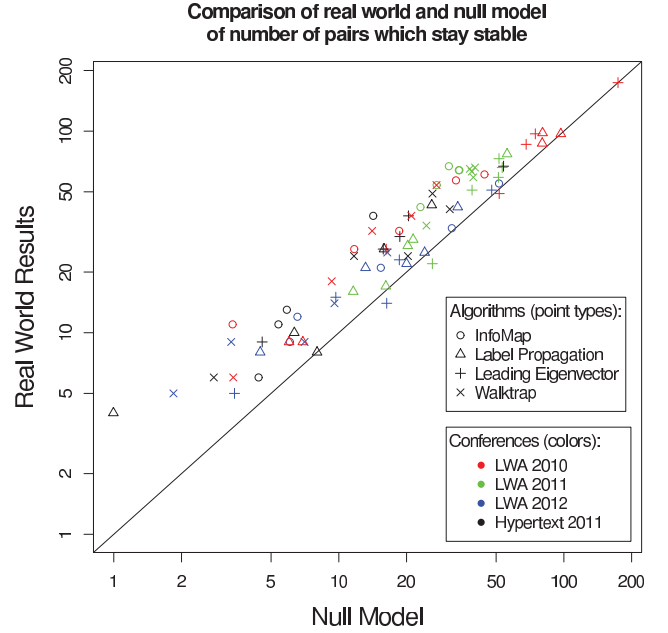


Fig. 6. Comparison of the real community stability with the null model of the considered algorithms (marked by the different point types) and conferences (marked by different colors): The x-axis contains the respective null model values, the y-axis contains the respective real values. Both axes are scaled logarithmically.

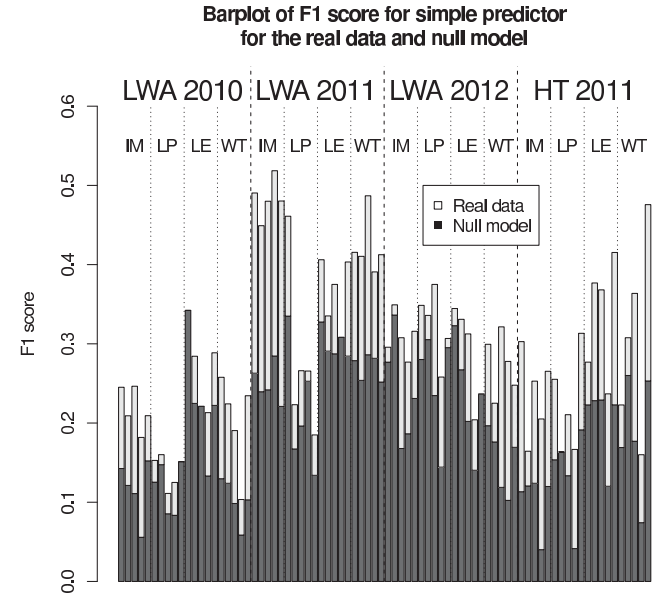


Fig. 7. Barplots of the F1 scores of the predictions compared to the respective null model for different conferences and different algorithms shown on the top. The abbreviations mean: IM – InfoMap, LP – Label Propagation, LE – Leading Eigenvector, WT – Walktrap. The algorithms are grouped by the respective conferences.

Figure 6 shows the comparison of the null model (x-axis) and the real values (y-axis). The majority of points lies above the null model line which means the stability of communities is not a random phenomenon.

Some of the results obtained using the *LeadingEigenvector* algorithm lie below the null model line, while some of the *LabelPropagation* measurements are just placed on the line. These findings would seem to show some randomness of the stability of community structures computed with these algorithms. In order to characterize the stability further, we compare the F1 score of the real data and the null model (see Figure 7). The F1 values computed from the real data are almost in every case larger than from the null model. On average the real world F1 score is 1.65 times larger than the obtained null model F1 score. This shows, that persons tend to stay in the same communities over one conference and the choice of algorithm surprisingly does not effect such stability.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper we presented an in-depth analysis of face-to-face contact graphs and the evolution of communities at four academic conferences. Almost half of the edges in the graphs are formed by conversations that are shorter than three minutes. The conversations tend to develop linearly, while the diversity of communication tend to increase during the different conferences.

We also considered the structure and evolution of local communities during each of the conferences computed by four different community detection algorithms. In our experiments, the identification of these communities using the weighted graphs is much more efficient than in non-weighted graph.

In order to investigate the stability of communities we compared the number of c-pairs – pairs of persons who stay in the same community for different algorithms and conferences. We used a “simple” predictor for inferring the c-pairs of a set subsequent days given a set of reference days, and compared these real-world results with a null model. In almost all cases, the scores using the automatically detected communities outperformed the null model, which shows the significance of the stability evaluation. In these experiments, the choice of the community detection algorithm was surprisingly not extremely significant for our datasets.

For future work, we plan to investigate recommenders based on the observed evolution patterns, including information about the detected communities. In addition, we aim to extend the evaluation by utilizing other community detection approaches including descriptive community detection approaches, e. g., [5]. Another interesting option for future work is given by more elaborate models for deriving the community evolution, e. g., by generative models for capturing community dynamics and evolution.

## ACKNOWLEDGEMENT

This work was supported by the VENUS research cluster at the interdisciplinary Research Center for Information System Design (ITeG) at Kassel University and by the EU-project EveryAware. We thank the SocioPatterns collaboration for providing privileged access to the SocioPatterns sensing platform.

## REFERENCES

- [1] H. Alani, M. Szomszor, C. Cattuto, W. V. den Broeck, G. Correndo, and A. Barrat. Live Social Semantics. In *ISWC*, pages 698–714, 2009.
- [2] M. Atzmueller, M. Becker, S. Doerfel, M. Kibanov, A. Hotho, B.-E. Macek, F. Mitzlaff, J. Mueller, C. Scholz, and G. Stumme. Ubicon: Observing Social and Physical Activities. In *CPSCOM*, 2012.
- [3] M. Atzmueller, D. Benz, S. Doerfel, A. Hotho, R. Jäschke, B. E. Macek, F. Mitzlaff, C. Scholz, and G. Stumme. Enhancing Social Interactions at Conferences. *it - Information Technology*, 53(3):101–107, 2011.
- [4] M. Atzmueller, S. Doerfel, A. Hotho, F. Mitzlaff, and G. Stumme. Face-to-Face Contacts during a Conference: Communities, Roles, and Key Players. In *Proc. MUSE at ECML/PKDD*, 2011.
- [5] M. Atzmueller and F. Mitzlaff. Efficient Descriptive Community Mining. In *Proc. 24th International FLAIRS Conference*, pages 459 – 464. AAAI Press, 2011.
- [6] L. Backstrom, D. P. Huttenlocher, J. M. Kleinberg, and X. Lan. Group formation in large social networks: Membership, growth, and evolution. In *KDD*, pages 44–54. ACM, 2006.
- [7] A. Barrat, C. Cattuto, V. Colizza, J.-F. Pinton, W. V. den Broeck, and A. Vespignani. High Resolution Dynamical Mapping of Social Interactions with Active RFID. *5(7)*, 2010.
- [8] A. Barrat, C. Cattuto, M. Szomszor, W. V. den Broeck, and H. Alani. Social Dynamics in Conferences: Analyses of Data from the Live Social Semantics Application. In *ISWC*, pages 17–33, 2010.
- [9] C. Cattuto, W. Van den Broeck, A. Barrat, V. Colizza, J.-F. Pinton, and A. Vespignani. Dynamics of Person-to-Person Interactions from Distributed RFID Sensor Networks. *PLoS ONE*, 5(7):e11596, 07 2010.
- [10] S. Fortunato and C. Castellano. Community Structure in Graphs, 2007. Chapter of Springer’s Encyclopedia of Complexity and System Science.
- [11] S. Fortunato and A. Lancichinetti. Community Detection Algorithms: a comparative analysis. In *Proc. 4th VALUETOOLS*, pages 27:1–27:2. ICST, 2009.
- [12] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins. On the Bursty Evolution of Blogspace. *World Wide Web*, 8(2):159–178, March 2003.
- [13] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters, 2008. arxiv:0810.1355.
- [14] J. Leskovec, K. J. Lang, and M. Mahoney. Empirical Comparison of Algorithms for Network Community Detection. In *Proc. of WWW*, pages 631–640, 2010.
- [15] B. E. Macek, C. Scholz, M. Atzmueller, and G. Stumme. Anatomy of a Conference. In *Proc. of HT*, pages 245–254. ACM, 2012.
- [16] M. E. Newman and M. Girvan. Finding and Evaluating Community Structure in Networks. *Phys Rev E Stat Nonlin Soft Matter Phys*, 69(2):1–15, 2004.
- [17] M. E. J. Newman. Detecting Community Structure in Networks. *Europ Physical J*, 38, 2004.
- [18] M. E. J. Newman. Finding Community Structure in Networks Using the Eigenvectors of Matrices. *Physical review E*, 74(3), 2006.
- [19] P. Pons and M. Latapy. Computing Communities in Large Networks Using Random Walks. *Computer and Information Sciences-ISCIS*, pages 284–293, 2005.
- [20] U. N. Raghavan, R. Albert, and S. Kumara. Near Linear Time Algorithm to Detect Community Structures in Large-Scale Networks. *Physical Review E*, 76(3):036106, Sept. 2007.
- [21] M. Rosvall, D. Axelsson, and C. T. Bergstrom. The Map Equation. *The European Physical Journal Special Topics*, 178:13–23, 2009.
- [22] M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proc. of the National Academy of Sciences*, 105(4):1118–1123, 2008.
- [23] C. Scholz, M. Atzmueller, and G. Stumme. On the Predictability of Human Contacts: Influence Factors and the Strength of Stronger Ties. In *Proc. of SocialCom. IEEE*, 2012.
- [24] M. Szomszor, C. Cattuto, W. V. den Broeck, A. Barrat, and H. Alani. Semantics, Sensors, and the Social Web: The Live Social Semantics Experiments. In *Proc. of ESWC*, pages 196–210, 2010.