Identifying long lived social communities using structural properties

Mark Goldberg, Malik Magdon-Ismail, and James Thompson
Computer Science Department
Renssalear Polytechnic Institute, Troy, NY 12180
Email:{goldberg, magdon, thompja}@cs.rpi.edu

***Abstract*—We present a two step procedure to identify long lasting communities, or evolutions, in social networks. First, we use axiomatic foundations to 'rigorously' establish shorter, strongly-connected evolutions. In the second step, we use heuristics to combine these shorter evolutions to form longer evolutions. We apply the procedure on data generated from two networks - the DBLP co-authorship database and LiveJournal blog data. We visually validate our algorithms by examining the topic evolution of the associated documents. Our results demonstrate that our algorithms, based solely on structural properties of the data (who interacts with whom), are able to track thematic trends in the literature. We then use a machine learning framework to identify the structural features of the early stages of a community's evolution are most useful for predicting the lifetime of the community. We find that (in order) size, intensity and stability are the most important features.**

INTRODUCTION

Social networks constantly change because of the many different interactions individuals of the network participate in. Bloggers constantly comment on different posts; people partake in different activities each day; and proteins interact with different proteins in different situations. Due to advances in technology and the growth of available information, the underlying behaviors of such networks have become an increasingly studied subject. In order to examine these characteristics of a network, it is important to be able to determine community structure and evolutionary dynamics of the network. We focus on social collaboration networks, but our methods are general.

A common way to examine temporal network behavior is to partition time into discrete intervals and study the structural characteristics of the static networks within each interval. Sometimes, communities within the social network are well-defined, such as conferences within a research network or user-defined communities of a social networking site. These situations provide a great opportunity to examine the patterns of membership in communities. Backstrom et al. [1] predict an individual's future membership of a community based on features such as the number of neighbors the individual has who are already members of the community, much like a diffusion model. Tantipathananandh et al. [2] have formulated community membership detection as a graph coloring problem, using greedy heuristics in order to match individuals to communities in any given time interval.

Generally, however, information on community structure is unavailable and can not be determined without using a discovery method. Any of the myriad researched clustering algorithms [3] can be used to detect communities at each time interval individually, but some approaches consider more. Hopcroft et al. [4] use minor perturbations on each of the static graphs in order to find communities that are stable throughout. In [5], Aynaud et al. propose a similar approach, finding static community structure that fits reasonably well with a number of consecutive time steps. Chakrabarti et al. [6] use the communities detected in previous intervals to enforce some level of continuity in newly detected communities.

Many approaches to examining the evolution of social communities have focused on constructing an underlying event framework which defines specific behaviors a community can exhibit, then searching for occurances of these behaviors within the network. Palla et al. [7] utilize six behaviors such as birth, growth, and merging for evolution detection. Asur et al. [8] include events for individual members of the network, such as joining or leaving a community. Both of these approaches require all events to take place between two consecutive time intervals. Greene et al. [9] and Takaffoli et al. [10] utilize events, but allow an evolution to 'skip' time intervals. Chen et al. [11] use a Hidden Markov Model to define group evolution.

Each approach utilizes a rigorous event based on algorithmic framework in order to identify communities and evolutions. This can cause evolutions that do not exactly prescribe to a specific framework to go undetected. [12] develop an axiomatic algorithmic framework for tracking community evolution that is "non-parametric" in that it is not based on communities that have a specific form. Our algorithms are based on this axiomatic framework.

A. Our Contributions

We study community evolution in social networks, in particular social networks that can be obtained through interactions via social media (for example social collaboration media such as blogs or coauthorship data). The first step in studying community evolution is detecting the community evolution. We develop a two step approach toward detecting community evolution. First, based on a sound axiomatic foundation of community evolution described in [12], we build (short) tightly connected evolutions in which there is a strong similarity between the communities at successive time steps in the evolution. We then develop a class of similarity measures to compare *evolutions* and use a particular instance of this

similarity measure to *merge* similar evolutions, thus extending the shorter 'tighter' evolutions into longer evolutions.

The first step is stringent and is able to discover evolutions which represent minor changes in the community as the community's 'goals' more or less remain intact. One can view this step as constructing short, but well-defined, 'evolution stubs'. The second step is less stringent, and is able to discover larger shifts in a community as it (for example) changes the topic of its discourse. Our algorithms are very efficient, nearly linear time in the size of the interaction data, and we demonstrate our algorithms on the DBLP author collaboration database, and LiveJournal blog data. In both cases, we first obtain communities over a set of disjoint time intervals, and using these communities we extract the evolution. We present visual validation of our results by using the associated text data to study the correlation between the community evolution and an observed evolution in the topic of the text. As an illustration of this visual validation we present an evolution of a community in DBLP with a lifespan of 21 years whose topic gradually morphed over those 21 years from functional programming and logic to fuzzy neural systems and learning to asynchronous cellular arrays. Naturally it would be interesting to study the topical trends of all the community evolutions discovered, but that is beyond the scope of the present work.

We then develop a cross-validation machine learning framework in which we build a linear regression system to predict the lifespan of a community based on structural features extracted from the early evolution of the community. These features include the intensity of the community as measured by some structural edge density measure; the size and growth of the community; the stability of the community as measured by its turnover; etc. Within this cross-validation linear regression framework, we are able to extract those features which are useful in predicting the lifespan of a community. In particular we find that the three most useful features for predicting community lifespan (from a set of approximately 80 structural features) are its size and growth-rate, and intensity.

The short story is that we present a general methodology for extracting community evolution that we have visually validated. This is the first step toward understanding of how communities evolve in social media. We illustrate by identifying those features correlated with long lived communities.

BACKGROUND

Notation: We consider networks that can be represented as a graph $G = (V, E)$ where V is the set of vertices with each vertex representing an actor in the network and E is the set of edges representing interactions among actors. Every edge in E is weighted with a value representing the strength of the interaction, and has a timestamp denoting when the interaction took place. There are typically many duplicate edges in this graph with different weights and timestamps.

We discretize time into intervals $t = 1, 2, \ldots, T$, each with a duration of τ, with τ chosen as a reasonable duration considering the nature of the interactions in the specific media. We define $G_t = (V_t, E_t)$ with E_t consisting of all the edges having a timestamp within time interval t, and V_t consisting of the endpoints of the edges in E_t. After the set E_t has been constructed, we drop the timestamps and replace duplicate edges with a single edge with a weight equal to the sum of the weights of the duplicate edges.

Let $\mathcal{C}_t = C_{0,t}, \ldots, C_{n_t,t}$ be the communities detected with G_t. These communities are taken to be the results of some clustering algorithm on G_t. Each community $C_{i,t}$ has a size $|C_{i,t}|$ and a density. We define the density of a community C in a graph to be the ratio between the number of interactions exclusively between members of community and the number of interactions involving at least one member of the community :

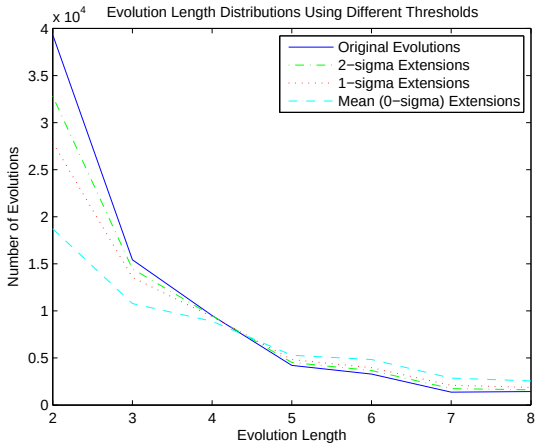
$$D(C) = \frac{W_{in}(C)}{(W_{in}(C) + W_{out}(C, \bar{(C)}))},$$

where we explicitly show that W_{out} consists of the edges between C and its complement.

Starting at a community $X_0 \in \mathcal{C}_t$, a chain $\mathcal{X}$ is a set of communities $X_0, X_1, \ldots, X_n$ where community $X_i \in \mathcal{C}_{t+i}$. The parent of the chain is X_0, and X_n is the leaf. To define a valid chain, or evolution, we use a 'goodness' function, $F(\cdot)$, that takes a chain as input and computes a value representing the viability of the chain as a real evolution. If $F(\mathcal{X})$ has a high value, the communities in $\mathcal{X}$ are very similar to each other in some way. If the value of $F(\mathcal{X})$ is small, the communities of $\mathcal{X}$ are unrelated to each other. We take a chain $\mathcal{X}$ as a valid evolution if $F(\mathcal{X})$ is above a certain threshold, λ. The choice of a value for λ depends on the implementation used for $F(\cdot)$ and user discretion. An axiomatic foundation for the form of F was given in [12]. There, it was proven that an acceptable function is one that satisfies $F(\mathcal{X}) = min(F(X_i, X_{i+1}))$. We choose the Jaccard coefficient for $F(X, Y)(= \frac{|X \cap Y|}{|X \cup Y|})$ and $\lambda = 0.2$ to detect "tight", short evolutions as the first step in community detection.

Our goal is to discover all of the disjoint pairs of evolutions, $(\mathcal{X} = (X_0, \ldots, X_k), \mathcal{Y} = (Y_0, \ldots, Y_l))$, that should be considered a valid evolution even though $F((\mathcal{X}, \mathcal{Y}))$ was not large enough. Similar to $F(\cdot)$, we take a function $M(\cdot, \cdot)$ that calculates a similarity value between two evolutions. Unlike the $F(\cdot)$ value, which is often based on the similarity between consecutive, pairwise communities [8]–[10], the $M(\cdot, \cdot)$ value is based on the membership characteristics of multiple communities from each evolution. This value is a quantitative measure of the plausibility of $\mathcal{X}$ and $\mathcal{Y}$ actually being a single evolution that was mis-regarded as two separate tight evolutions. The chain ($\mathcal{X}$, $\mathcal{Y}$) is considered a valid evolution if $M(\mathcal{X}, \mathcal{Y})$ is above a threshold μ. The value for μ, like λ, depends upon the implementation of $M(\cdot, \cdot)$ and user discretion.

Problem Statement: Given a set of tight evolutions $\mathbb{X}$, determine all of the valid evolutions (under relaxed requirements) that are unions of the given tight evolutions. For a given evolution $\mathcal{X}_0$, a union of evolutions $(\mathcal{X}_0, \ldots, \mathcal{X}_n)$ is maximal if there is no other valid union containing $\mathcal{X}_0$ that spans a larger number of time steps. Using a brute force approach to discovering these maximal chains would be impractical as



	Number of Evolutions			
Length	Original	+ 2σ	+ σ	Avg M value
2	39335	32795	27900	18731
3	15429	14468	13565	10779
4	9509	9483	9375	8901
5	4212	4529	4830	5293
6	3288	3670	3961	4837
7	1372	1747	2096	2861
8	1440	1613	1872	2568
9	509	728	957	1512
10	618	797	1001	1432
11	223	383	510	868
12	292	376	470	826
13	81	176	262	514
14	125	187	292	420
15	36	83	112	252
16	74	100	137	254
17	8	30	72	174
18	40	57	64	98
19	9	25	33	63
20	23	37	40	45

Fig. 1. Distributions of evolution lengths for original evolutions and evolutions found using different merging thresholds. A large number of merges occurs between smaller evolutions, causing the distributions to even out as the threshold is lowered. The figure displays the most drastic changes in evolution length distributions across thresholds and the table shows corresponding values.

there could be a large number of evolutions to consider and an exponential number of possible unions to compute the $M(\cdot,\cdot)$ value for. We use pre-processing of evolutions and dynamic programming techniques in order to avoid long run times.

Determining Communities and Evolutions

The first step to discovering evolutions within a network is to identify short, "tight" evolutions. We consider communities and evolutions discovered by using the process described in [12]. The graphs of each time step, G_t, are clustered independently to find static communities. The static communities are then analyzed to detect evolutionary behavior. Specifically, the communities are discovered using LOS clustering [13] and evolutions with an $F(\cdot)$ value above 0.2 were taken to be valid. Both networks resulted in about 50,000 evolutions.

It is probable that this approach does not detect all of the real evolutions (nor only the real evolutions) within a social network. Too high of a threshold for the value of $F(\cdot)$ may split an evolution into multiple parts where the similarity between two timesteps just wasn't large enough. Lowering the threshold too far results in chains of communities with no real underlying similarity to be considered an evolution. In addition, some communities may not be identified in certain timesteps, either because some information was not gathered or the clustering algorithm did not detect it. This, again, would split an evolution in smaller, disjoint evolutions. The second step of our approach attempts to recover these split evolutions.

Similarity Between Tight Evolutions

The second step to our approach slightly relaxes the definition of an evolution, and merges the short evolutions into longer ones. Let $\mathcal{X} = (X_0, \ldots, X_k)$ and $\mathcal{Y} = (Y_0, \ldots, Y_l)$ be two chains where $X_k \in \mathcal{C}_t$ and $Y_0 \in \mathcal{C}_{t+1}$. We consider the last three communities of $\mathcal{X} - (X_{k-2}, X_{k-1}, X_k)$, and the first three communities of $\mathcal{Y} - (Y_0, Y_1, Y_2)$. Let $A = (A_0, \ldots, A_n)$ be the set of members of the union of the six communities $(X_{k-2} \cup X_{k-1} \cup X_k \cup Y_0 \cup Y_1 \cup Y_2)$. Define the function

$$I(A_i, C_i) = \begin{cases} 1, & \text{if } A_i \in C_i; \\ 0, & \text{if } A_i \notin C_i. \end{cases}$$

We associate two values with every A_i. First, the number of communities out of the last three communities of X that A_i is a member of, and second, the number of communities out of the first three communities of Y that A_i is a member of:

$$A_{i,X} = \sum_{j=k-2}^{k} I(A_i, X_j) \qquad A_{i,Y} = \sum_{j=0}^{2} I(A_i, Y_j).$$

Using these values, we construct two vectors $A_X = [A_{0,X}, \ldots, A_{n,X}]$ and $A_Y = [A_{0,Y}, \ldots, A_{n,Y}]$ representing the community memberships of the end of evolution X and the beginning of evolution Y, and calculate the cosine similarity between the vectors as the M-value between two evolutions.

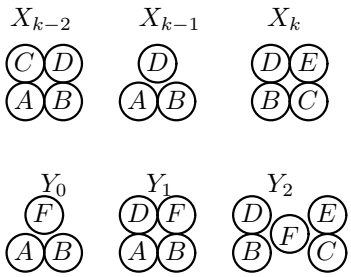
$$M(A_X, A_Y) = \frac{A_X \cdot A_Y}{|| A_X || * || A_Y ||}$$

Figure 2 shows how to calculate the M-value between two evolutions. The more similar the members of the communities at the end of X are to the members to the communities at the beginning of Y, the larger the value of M will be.

More general approaches to constructing the vectors A_X and A_Y may weight the participation of nodes in a community based on the number of time steps separating the community and the evolution split. In other words, the values associated with each A_i could be of the form:

$$A_{i,X} = \sum_{j=k-2}^{k} w_{k-j} I(A_i, X_j) \qquad A_{i,Y} = \sum_{j=0}^{2} w_j I(A_i, Y_j)$$

where $w_j = 2^{-j}$. These vectors would result in an M-value that takes the full membership of the evolutions into account, but places more importance on membership near the end of X



Node Participation						
Node	X_{k-2}	X_{k-1}	X_k	Y_0	Y_1	Y_2
A	1	1	0	1	1	0
B	1	1	1	1	1	1
C	1	0	1	0	0	1
D	1	1	1	0	1	1
E	0	0	1	0	0	1
F	0	0	0	1	1	1

Participation Vectors

$$A_X = \begin{bmatrix} 2 & 3 & 2 & 3 & 1 & 0 \end{bmatrix}$$
$$A_Y = \begin{bmatrix} 2 & 3 & 1 & 2 & 1 & 3 \end{bmatrix}$$

$$M(X,Y) = \frac{(2*2)+(3*3)+(2*1)+(3*2)+(1*1)+(0*3)}{(2+3+2+3+1+0)*(1+3+1+2+1+3)} = \frac{1}{6}$$

Fig. 2. Calculating the M-value between two evolutions. The nodes in the last three communities of evolution X, and the first three evolutions of Y are shown at the top, followed by the specific calculation.

and the beginning of Y. In our choice, w_j is 1 if $j < 3$ and 0 otherwise. If w_j is 1 when $j = 0$ and 0 everywhere else, the M values are equal to the original F-values.

To construct maximal evolutions, the beginning set of evolutions $\mathbb{X}$ is broken into sets $\mathbb{E}_i$ and $\mathbb{B}_{i+1}$. The set $\mathbb{E}_i$ consists of all evolutions that end at time step i and, similarly, $\mathbb{B}_{i+1}$ is the set of all evolutions that begin at time step $i + 1$. Since evolutions require a community to be present in every consecutive time step, evolutions in $\mathbb{E}_i$ can only be extended by evolutions in $\mathbb{B}_{i+1}$. From this point, we emulate the similarity calculation in [12], only calculating an M value for pairs of evolutions that do not have completely disjoint membership, and using dynamic programming to construct the maximal union sets of evolutions.

EXPERIMENTS

We demonstrate results on graphs constructed from two real social networks: the DBLP co-authorship database and LiveJournal (BLOG). Both data sets represent a bipartite graph with users and objects. Users collaborate on creating objects. For example, DBLP consists of authors collaborating to write research papers. In the BLOG data, users collaborate in a conversation in the form of a comment thread. Furthermore, each collaboration has a time stamp associated with it. For papers in DBLP, this time stamp is the paper's publish date. In the BLOG data, an object is a certain comment thread, and the users connected to it are those who authored a comment in the thread together with the original blog poster.

Determining μ: The data is split into groups of collaborations that occured within each time step. Table I shows the length of a single time step and how many time steps are available in each data set. Collaborations in a time step form a bipartite graph with users and objects, but we can infer a graph on the users alone. Two users are connected if they collaborated on an object. The weight of this edge represents the number of such collaborations. In the DBLP network, two authors would be connected by an edge in time step t if they coauthored a paper published in time step t. The weight of this edge is the number of papers coauthored with a publish date in time step t. In the BLOG data, two users who post a comment in the same thread during the same week are connected.

The contribution of a single collaboration to the weight of an edge could be scaled to reflect the quality of the collaboration. For example, if two users, i and j, participate in a collaboration of n total users, the contribution of the collaboration toward the weight of edge $\{i, j\}$, $w_{i,j}$, may be defined as $\frac{1}{n-1}$. For simplicity, we use weight as just the number of collaborations.

The choice of value for μ dictates the extent of relaxation to the axiomatic definition of evolution. Low μ values allow for a larger deviance from the axioms and extends evolutions with unrelated members. High values extend evolutions with closely related members, but may disqualify some valid extensions. We choose a high value for μ in order to examine the viability of combining the most similar evolutions. If $\mathbf{M}$ is the set of all M-values of potential evolution extensions, we set μ to be the average of $\mathbf{M}$ plus two standard deviations ($\bar{\mathbf{M}} + 2 * \sigma(\mathbf{M})$).

Figure 1 shows the evolution length distribution of the original evolutions compared with the distributions of evolutions after combining evolutions using different values for μ: $\bar{\mathbf{M}}$, $\bar{\mathbf{M}} + \sigma(\mathbf{M})$ and $\bar{\mathbf{M}} + 2 * \sigma(\mathbf{M})$. All of the distributions follow an inverse power law, but as μ is reduced, the tails of the distributions grow larger, as expected. This change becomes much more pronounced when μ gets closer to $\bar{\mathbf{M}}$, suggesting that any value of μ somewhat larger than $\bar{\mathbf{M}}$ should yeild well-defined results.

For the rest of the paper, we use $\mu = M + 2\sigma$ to obtain a set of extended evolutions $\mathbb{X}_e$ containing merges from the original set of tight evolutions, $\mathbb{X}$.

Validation

Due largely to the ambiguity concerning the definition of communities within a social network, a chronic problem when working with community detection and evolution is validation of results. We use human judgement based on examining the subject matter of interactions within communities. In DBLP, this requires looking at the titles and abstracts of papers written by authors of a community. In the BLOG data, the content of the posts are used in place of abstracts.

To construct a visual representation of the interactions for a community, we collect all the abstracts or posts associated with the links of the community and create a word cloud. The larger a word appears in the word cloud, the more relevant it is as the main topic of interest within the community. Valid communities will have main interests that are easily

Name	Description	$\|V\|$	$\|E\|$	$\tau\,(N)$	C	ρ	α
DBLP	Authors of academic Computer Science papers are linked if they coauthored a paper	4x10^5	1x10^6	1 yr(19)	3.5x10^5	0.28	2.7
BLOG	A user is linked to users that comment on its blog post or respond to a comment it made	2x10^5	1x10^6	1 wk(66)	2x10^5	0.01	2.22

TABLE I

DESCRIPTION OF THE DATA SETS USED IN OUR STUDY. N IS THE NUMBER OF CONSEQUTIVE TIME STEPS WITH DATA. C IS THE SIZE OF THE LARGEST CONNECTED COMPONENT IN THE USER-USER GRAPH OF THE FULL DATA SET. ρ IS THE CLUSTERING COEFFICIENT OF THE GRAPH, AND α IS THE SCALE-FREE PARAMETER.

gleaned from the word clouds while valid evolutions will have consistency or logical development of ideas throughout its communities.

Figure 5 shows the word clouds constructed for two evolutions that were merged to form a longer evolution in the DBLP data set. The break between evolutions was located between the tenth and eleventh communities. When constructing the original evolutions, we used the Jaccard index to measure similarity and a threshold of 0.2. The similarity between the tenth and eleventh communities in this evolution was 0.167.

Some of the major keywords discovered within the merged evolution are shown in Figure 4. It shows that in the beginning of the first evolution, the authors main focus was on using inductive inference to solve problems. In the first evolution, the focus quickly becomes learning, a more general topic that includes inductive inference. Alongside learning comes the interest in using or studying languages. The end of the first evolution and the beginning of the second share the main common topic of validation, a logical progression from developing techniques for learning. Finally, the continuity of the second evolution is made clear when the focus on learning returns strongly, with the other keywords appearing frequently as well.

Prediction: We consider the first four communities, $X_0 \ldots X_3$, of an evolution and try to predict the length of the evolution (see Figure 3). The same analysis could be used using any number of the beginning communities of an evolution. Using four communities provides a long enough chain to give confidence that the evolution is not random, and also allows for the examination of most of the detected evolutions.

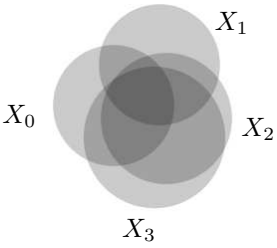


Fig. 3. Early stages of an evolution

Let $s_i = |X_i|$ and $d_i = D(X_i)$ be the size and density of a community, repectively. We let $r_i = |X_i \cap X_{i+1}|$ be the size of the pairwise intersections, define the core size $q_i = |X_i \cap X_{i+1} \cap X_{i+2}|$ as the intersection sizes of the r_i, and define the hypercore size $c_0 = |X_i \cap X_{i+1} \cap X_{i+2} \cap X_{i+3}|$ as the size of the intersection of the cores. These parameters relate to the shaded regions in the figure. In addition, let $d_{r_i} = D(|X_i \cap X_{i+1}|)$ be the density of the nodes in the intersection of two communities in respect to the union of the graphs from each time step. Similarly, let $d_{q_i} = D(|X_i \cap X_{i+1} \cap X_{i+2}|)$ be the core density and $d_{c_i} = D(|\cap_{k=i}^{i+3} X_k|)$ be the hypercore density.

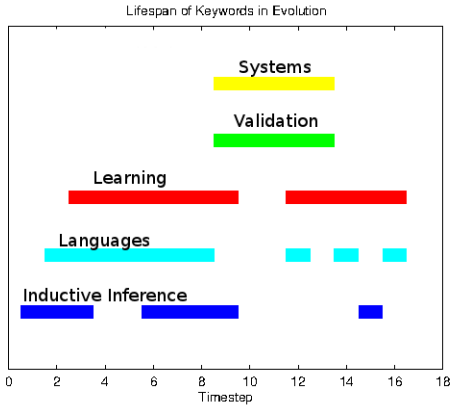


Fig. 4. Span of main keywords found in word clouds for a merged evolution. The overlap of spans shows the succession of topics discussed within the communities of the evolutions and the continuity between the two evolutions that caused the merge.

Using these parameters, we derive 79 features to characterize the early stages of an evolution.

Our goal is to use these features to estimate evolution length. We use a simple linear regression framework with leave-one-out cross validation (LOO-CV) to find the most useful features for predicting evolution length (Table IV).

To construct a parameter's score, we consider the fifteen most useful features. We give the i^{th} most useful feature a score of $15 - i$. A parameter's score is the sum of the scores of all the features that were derived from that characteristic.

The table shows that prediction of the lengths of the evolutions discovered using a stricter framework uses the sizes of communities and combinations of contiguous communities evenly. Once evolutions are combined into longer evolutions with more relaxed definitions, the density of communities becomes much more representative of evolution length while the size of community combinations becomes less.

CONCLUSION

We have developed a two-step process for the identification of evolutions within a network. Our results indicate that the evolutions detected at each step of the process differ from random chains of communities. Furthermore, the results support the need to include the second, merging step of the process in order to augment the quality of the detected evolutions.

The length of evolutions identified by the two step approach used in this paper can be predicted using select features (see Table II and Table III) of the early characteristics of the evolutions. This suggests that the detected evolutions have underlying structures that differentiate an evolution from

ID	Feature Name
8	Average Community Density
7	Slope of Min-Normalized Community Size
14	Average Intesection Size
19	Slope Intersection Size
40	Average First-Intersection-Normalized Core Size
78	Average Density of Core-Normalized HyperCore
34	Average Core Size
5	Slope Community Size
36	Average Min-Normalized Core Size
35	Average First-Normalized Core Size
4	Average Min-Normalized Community Size
3	Average First-Normalized Community Size
2	Average Community Size
18	Average Min-Community-Normalized Intersection Size

TABLE II
MOST DETERMINANT FEATURES OF EVOLUTION PREDICTION IN ORIGINAL EVOLUTIONS IN ORDER. COMMUNITY DENSITY, WITH A POSITIVE CORRELATION TO EVOLUTION LENGTH, IS THE SINGLE MOST PREDICTIVE FEATURE OF EVOLUTION LENGTH. HOWEVER, FEATURES OF COMMUNITY, INTERSECTION, AND CORE SIZES SATURATE THE LIST.

ID	Feature Name
8	Average Community Density
15	Average First-Normalized Intersection Size
7	Slope Min-Normalized Community Size
3	Average First-Normalized Community Size
12	Slope of First-Normalized Community Density
4	Average Min-Normalized Community Size
63	Average HyperCore Size
38	Average Min-Community Normalized Core Size
18	Average Min-Community-Normalized Intersection Size
25	Average Density of Intersection
41	Slope Core Size
19	Slope of Intersection Size
54	Slope of Core Density
11	Slope of Community Density

TABLE III
MOST DETERMINANT FEATURES OF EVOLUTION PREDICTION IN MERGED EVOLUTIONS (WITH MERGE THRESHOLD OF 2σ). COMMUNITY DENSITY REMAINS THE MOST PREDICTIVE FEATURE WITH THE FOLLOWING FEATURES TRENDING TOWARDS DENSITY MEASURES. THE CORE ALSO EXPERIENCES A LOSS IN RELEVANCE, WITH COMMUNITIES AND INTERSECTIONS EXPERIENCING A GAIN.

Evolutions Before			Evolutions After		
Parameter	Score	Wgt	Parameter	Score	Wgt
Core (q_i)	25	-	Size (s_i)	29	
Size (s_i)	24		Growth		+
Growth		+	Average		-
Average		-	Density (d_i)	26	+
Intersection(r_i)	21	+	Intersection(r_i)	19	+
Density (d_i)	13	+	Core(q_i)	9	+

TABLE IV
MOST PREDICTIVE CHARACTERISTICS OF EVOLUTION LENGTH IN THE ORIGINAL AND MERGED EVOLUTIONS. THE WGT ENTRY DENOTES WHETHER THE FEATURES BASED ON THE CHARACTERSTIC GENERALLY HAD A POSITIVE (+) OR NEGATIVE (-) CORRELATION WITH EVOLUTION LENGTH. THE CHARACTERISTIC'S SCORES ARE SIMILAR IN THE ORIGINAL EVOLUTIONS WHILE AFTER MERGING COMMUNITY SIZE AND DENSITY DISTINGUISH THEMSELVES AS THE MAIN CHARACTERISTICS.

random communities. If the detected evolutions were merely chains of random communities, these structures would not be there and length prediction would be practically impossible.

The visualization of evolutions, such as that in Figure 5, depict a logical flow of topics within both the shorter and merged evolutions. The fact that this flow is found within the shorter evolutions supports the belief that these evolutions are valid. It can be argued that the merged evolutions are better in some sense, as they are based on the same communities and exhibit the same type of logical flow in topics, but last longer.

In addition, the features that are most predictive of evolution length in merged evolutions seem more natural than those in the shorter evolutions. As Table IV shows, un-merged evolutions have, somewhat unintuitively, a negative correlation with the core of the first four communities that is highly predictive of the evolution lifespan. In merged evolutions, this changes to a positive correlation which is not as predictive. In addition, it is intuitive to think that the density of communities significantly influences evolution length. Community density is much more predictive of evolution length in merged communities, suggesting that the merged evolutions are 'better'.

ACKNOWLEDGEMENTS

This material is based upon work partially sponsored by: U.S. DHS through ONR grant number N00014-07-1-0150 to Rutgers University and continues under the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053.

REFERENCES

[1] L. Backstrom, D. P. Huttenlocher, J. M. Kleinberg, and X. Lan, "Group formation in large social networks: membership, growth, and evolution," in *KDD*, 2006, pp. 44–54.
[2] C. Tantipathananandh, T. Berger-Wolf, and D. Kempe, "A framework for community identification in dynamic social networks," in *KDD*, New York, NY, USA, 2007, pp. 717–726.
[3] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, pp. 75 – 174, 2010.
[4] J. Hopcroft, O. Khan, B. Kulis, and B. Selman, "Tracking evolving communities in large linked networks," *PNAS*, vol. 101, pp. 5249–5253, 2004.
[5] T. Aynaud and J.-L. Guillaume, "Multi-step community detection and hierarchical time segmentation in evolving networks," in *KDD*, 2011.
[6] D. Chakrabarti, R. Kumar, and A. Tomkins, "Evolutionary clustering," in *KDD*. ACM, 2006, pp. 554–560.
[7] G. Palla, A.-L. Barabási, and T. Vicsek, "Quantifying social group evolution," *Nature*, vol. 446, pp. 664–667, 2007.
[8] S. Asur, S. Parthasarathy, and D. Ucar, "An event-based framework for characterizing the evolutionary behavior of interaction graphs," *SNAKDD*, vol. 3, no. 4, p. 913, 2007.
[9] D. Greene, D. Doyle, and P. Cunningham, "Tracking the evolution of communities in dynamic social networks," UCD, Tech. Rep., 2011.
[10] M. Takaffoli, F. Sangi, J. Fagnan, and O. R. Zaïane, "Modec - modeling and detecting evolutions of communities," in *ICWSM*, L. A. Adamic, R. A. Baeza-Yates, and S. Counts, Eds. The AAAI Press, 2011.
[11] H.-C. Chen, M. Goldberg, M. Magdon-Ismail, and W. A. Wallace, "Reverse engineering an agent-based hidden markov model for complex social systems," in *IDEAL*, 2007, pp. 940–949.
[12] M. Goldberg, M. Magdon-Ismail, S. Nambirajan, and J. Thompson, "Tracking and predicting evolution of social communities," in *SocialCom*, 2011.
[13] M. Goldberg, S. Kelley, M. Magdon-ismail, K. Mertsalov, and A. Wallace, "Finding overlapping communities in social networks," 2010.

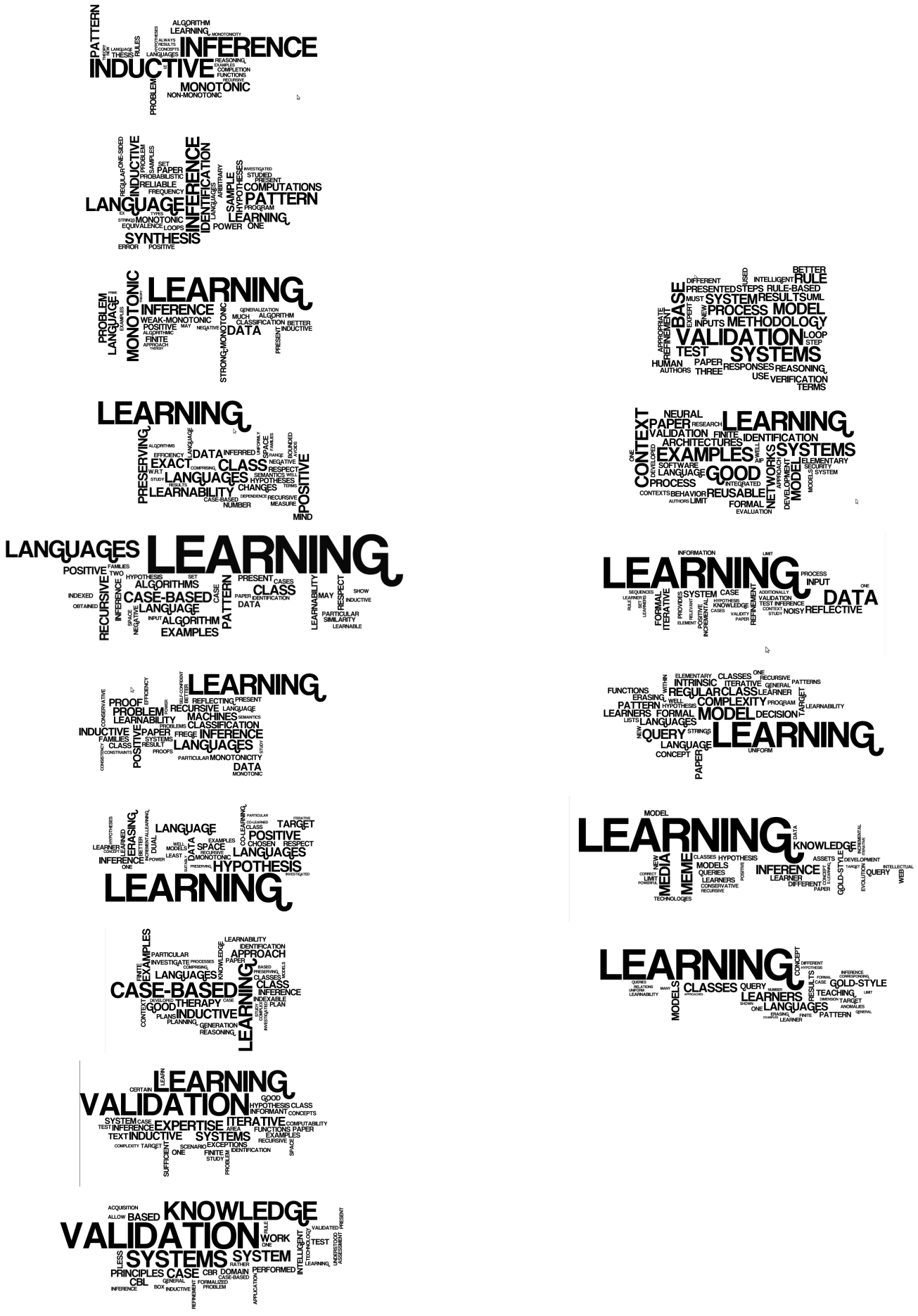


Fig. 5. Word clouds of paper abstracts in communities of two merged evolutions in the DBLP network. Word sizes are determined by frequency in abstracts. Each column represents an original evolution.