A Method for Community Detection in Uncertain Networks

Johan Dahlin and Pontus Svenson
Division of Information Systems
FOI Swedish Defence Research Agency
SE 164 90 Stockholm, Sweden
Email: {johan.dahlin,pontus.svenson}@foi.se

***Abstract*—Social network analysis can be an important help for military and criminal intelligence analysis. In real world applications, there is seldom complete knowledge about the network of interest – we only have partial and incomplete information about the nodes and networks present. Community detection in networks is an important area of current research in social network analysis with many applications. Finding community structures is however a challenging task and despite significant effort no satisfactory method has been found. Here we study the problem of community detection in noisy and uncertain networks with missing and false edges and propose methods for detecting community structures in them. The method is based on sampling from an ensemble of certain networks that are consistent with the available information about the uncertain networks.**

I. Introduction

Social network analysis[1], [2] is a vast and growing research area. Its application areas range from analysis of ecological networks[] to studies of link structures on the web[]. Social network analysis can also be an important help in security and intelligence informatics[3], [4].

The first applications of social network analysis were by sociologists who collected data by questionnaires or direct observations and used graph visualization to gain insight into the communication behavior of small groups. The field has expanded considerably since then, and has also been influence by the advance in computer technology which has enabled collection of network data in enormous quantities. With the increased availability of data, however, comes new challenges: algorithms are needed to handle much larger networks than previously, and some way has to be devised to take account of the inherent uncertainty of the data. In recent years, there has been much development in the social network analysis field of new, more efficient algorithms for handling ever larger amounts of data. So far, however, there has been very little done taking account of the uncertainty of the data.

Uncertainty can arise in many different ways, depending on the way that the network data has been collected. Different *observation models* are needed for different kinds of data collection.

We note that it is important do distinguish between uncertain networks and weighted networks. One way of representing uncertain networks is by simply adding a probability to each edge, which resembles a weighted network. While for some simple network measures, algorithms developed for weighted networks can be applied to uncertain networks, this is not true for for instance the community detection problem.

In this paper, we describe a framework method for how to analyze uncertain networks and describe in detail how the method can be applied to the community detection problem.

This paper is outlined as follows. In sectionII we briefly discuss some of the challenges faced when using social network analysis in intelligence analysis. Section III gives an overview of community detection, while section IV discusses modeling of uncertain networks and briefly introduces the Dempster-Shafer theoretical framework for merging information about network substructures from different sources. Finally, section V presents the method for detecting community structures in uncertain networks, and also presents some results of applying the method to test networks.

II. Network analysis for intelligence applications

The goal of intelligence analysis [5]. is to provide a decision-maker with basic data so that they can make a more informed decision. Most often, the basic data is in the form of an analysis report which summarizes information from many different sources and gives recommendations for future actions. For military intelligence analysis, the report often provides also a description of the most likely future course of events as well as the worst-case course of events. The specific contents of an intelligence report, of course, varies depending on the domain. Social network analysis has emerged as an important tool for producing intelligence reports. In addition to the possibility to visualize relations between people, organizations, events and objects, it is sometimes very useful to be able to do quantitative analysis of a network and determine, for example, the most important actor of it, or the most important substructures/communities.

In intelligence applications, there is always uncertainty present. The first cause of uncertainty is the objective of the analysis itself – while intelligence analysis is goal-driven and directed towards answering a specific information request, this information request can sometimes be very vague and unspecific. Consider for example the different between the requests ”What is the likelihood that the enemy tank battalion will continue to advance tomorrow?” with the more difficult ”What is the current terrorist threat against Sweden?”. For

978-0-7695-4406-9/11 $26.00 © 2011 IEEE
DOI 10.1109/EISIC.2011.58

the first type of question, it is important to have access to good sources of geographical and terrain information as well as information about the supply status and standard doctrinal behavior of the battalion. Information from sensors and other reconnaissance resources can also be used to improve the quality of the answer. For the second type of questions, much more data is needed. The answer will also be inherently more uncertain.

Another source of uncertainty is in the data used to produce the intelligence report. Sensor data can be associated with uncertainties due to, e.g., misclassification probabilities. Data from human resources (HUMINT) is uncertain because we can never know for sure that the human source it not trying to deceive us, and information collected from the web and other open sources (OSINT) can be uncertain both because of deception and because of errors in the collection and processing of the data.

Social network data is particularly uncertain. In order to construct a network to analyze, an intelligence analyst must first determine what type of relations that are of interest, and then map all available data to these types. Data can come from both manual sources ("I saw X talk to Y") and automatic processing of, e.g., signals intelligence. Both carry large amounts of uncertainty.

III. Community detection

Networks in general and social networks in particular often contain some form of group structure known as *communities* (other common terms used are *partitions*, *modules*, and *clusters*). In the context of data clustering, each node inside the community is in some sense similar to its neighbors. For example, we can often find friends, family, and colleagues in the social network of a typical person. These groups are mostly quite isolated and not many friendships exist between these different groups. In this case, these three groups are the communities of the network. They are also similar in regard with their position and social roles in the network. Therefore, it is in general possible to use the obtained community structure for identifying social roles, hierarchies and hidden groups within the network data material.

Communities are a vague and fragile concept found in some networks. It is difficult to find communities and verify their existence and uniqueness but some methods do exist. The most promising is to evaluate the robustness of the clustering. The definition of a community commonly used in social network analysis is that a community is a subset of nodes which have more internal connections than external. In this paper, we take this definition for granted and study the problem of finding such dense subsets when the input data is uncertain. It is important to realize that in the end, the communities detected in the networks are the result of the data which is the only input given. Therefore as in all statistical methods, if the data is flawed the corresponding community structure could be misleading and uncertain. Robustness analysis and similar methods can be used to analyze the significance and stability of the structure found, thereby mimicking hypothesis testing in statistics.

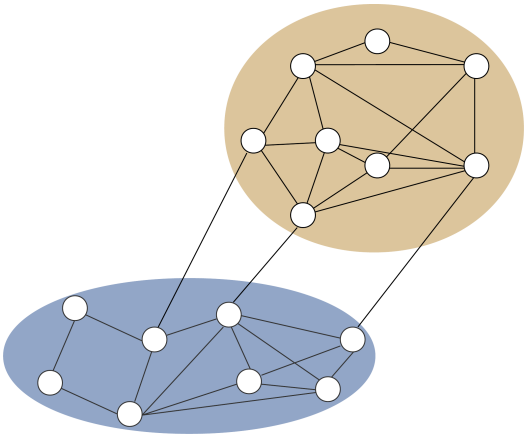

Fig. 1. Two communities in a simple network, the number of edges in each community is much higher than between the two communities.

A. Community detection methods

Historically manual methods have been used to find community structures in collected data. Humans are often good at finding structures in small and sparse networks but manual methods are not practical for larger and denser network. To solve this problem, many algorithmic methods used on computers have been proposed from the fields of physics, computer science, and statistics.

Despite this large effort, no completely satisfactory solution to community detection problem has yet to been devised. The main explanation for this is that community detection (maximization of modularity) is a NP-complete optimization problem. As a consequence from the necessary relaxations of this problem it is often so that different algorithms have different characteristics. Algorithms often have built-in tendencies to find communities of different sizes but also often find different community structures when applied to the same network.

In general, more complex community detection methods are more accurate and robust in comparison with simple fast methods but are limited to small sparse networks. As each method has different properties it is commonplace to apply several different methods to the same problem and compare the results.

The first community detection methods proposed are based on the related problem of graph partitioning. This problem is common in computer sciences and mathematics with many applications. An important everyday application is e.g. to determine the correct division of computational effort on parallel computers[1]. Most graph partitioning methods are only able to divide a network into two parts and often find solutions with a very small cut set[2]. [6]

[1]Often called the load balancing problem in practice, see e.g. some standard work on parallel computation or [6] for more information.

[2]A *cut set* is the set of edges that need to be removed from a graph to generate two disjoint components

There exist many methods for community detection, some of the most promising are e.g. *q-Potts spin glass methods* [7], *label propagation* [8], *Infomaps* [9], *clique percolation* [10], and *synchronization*. Recent developments have also been introduced to improve community detection methods for weighted, directed, and dynamic networks. For more information about this, thorough discussions and comparisons of community detection methods, see Refs. [11], [12].

IV. Uncertain networks

As outlined above, for many real-world applications of social network analysis we do not have complete certain knowledge about the network of interest. Instead, we must make do with partial and incomplete information. In previous work, the data is often considered certain and the uncertainty removed by using one of two alternatives. The first alternative is to include all edges and nodes found, thereby possibly adding false nodes and edges into the network. The second alternative is to remove all uncertain edges and nodes, thereby risking problems with missing edges and networks. It is not difficult to realize that both methods generate different network structures and hence also different detected communities. Another approach to uncertain network analysis is to simply use the probabilities of edges in the network as weights and to use standard methods for analyzing weighted networks. This works very well for simple measures, but gives erroneous results when applied to the community detection problem, since the correlation in communities induced by the presence of higher-order structures in the network is ignored.

In this paper we work under the assumption that a large portion of the data is uncertain and try to utilize the data in the best possible manner. This section explains the observation model of networks and discusses some possible future generalizations of the model to include more interesting difficulties encountered in practical applications. We also introduce a framework to quantify and combine several different sources of information to estimate imperfect networks called *Dempster-Shafer theory*[13].

The basic idea for handling the uncertainty is quite simple [14]: we use the available information to construct an ensemble of certain networks that are consistent with the available information about the network. We then take a sufficiently large number of samples from this ensemble and compute communities in each of these certain networks. This set of community structures is then merged to produce an overall estimation of the community structure of the uncertain network.

A. Observation model

The *observation model* is a formalization of the problem with observation of an underlying network by the use of other related networks. Often it is not possible to observe the network of interest directly and therefore some other *proxy network* has to be used as an approximation. A classical example of this is using the communication network between people as a proxy of different kinds of relations. People with stronger connections and deeper relationships are assumed to communicate more often or in some characteristic pattern.

Formalizing this, we assume that the *real network*, f, is not directly observable, but similar to a proxy network, g, to estimate the underlying network. By using this other network to describe the network of interest several different problems are encountered, e.g. finding edges in the observed network which do not exist in the real network etc. Assume that an *edge existence probability*, $\mathbb{P}(g_{ij})$, i.e. the probability that an edge exist (or does not exist) in the observed network, g, between nodes i and j can be found as,

$$\mathbb{P}(g_{ij}) = FP + TP = \mathbb{P}(g_{ij}|\neg f_{ij}) + \mathbb{P}(g_{ij}|f_{ij}), \quad (1)$$

$$\mathbb{P}(\neg g_{ij}) = TN + FN = \mathbb{P}(\neg g_{ij}|\neg f_{ij}) + \mathbb{P}(\neg g_{ij}|f_{ij}) \quad (2)$$

where FP(N) denote *False Positive* (*Negative*), TP(N) denote *True Positive* (*Negative*), and $\mathbb{P}(\cdot)$ is the observation probability. The probability, $\mathbb{P}(g_{ij})$, should in some aspect indicate the uncertainty of the information regarding the edge, g_{ij}. High probabilities indicate strong evidence for the hypothesis that the edge exist in the real network. Smaller probabilities indicate vague or contradicting evidence. This is an observation model which links the observed with the real network and formalizes the uncertainty in using this approximation. An illustration of this is shown in *Figure 2*.

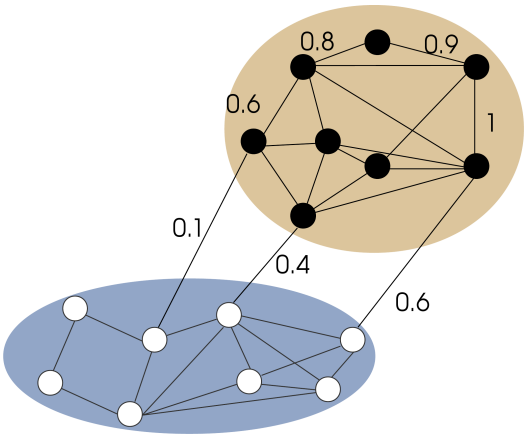

Fig. 2. A small uncertain network with edge existence probabilities.

Definition 1 (Uncertain networks): An *uncertain network* is a graph, $G(V, \mathbf{E})$, where V is a set of nodes and $\mathbf{E} = [E_{ij}]$ is some *edge existence probability matrix* with $E_{ij} = \mathbb{P}(g_{ij})$ as the probability that an edge exists between nodes i and j.

B. Generalizing the observation model

Edges are not the only uncertain and imperfect objects found in imperfect networks, more complex structures and objects may also be uncertain, missing, or falsely included. Nodes can also be modeled in the same manner to edges; i.e. the network may contain uncertain, missing and false nodes. In this case, false nodes could mean that to that two nodes in the observed network are really only one in the real network. The probabilities can also describe different network structures,

e.g. triangles, n-cliques, paths, and trees. Adding evidence regarding these structures can improve the estimated network structure found using observed network data.

Using the framework built in the remaining part of this section, it is possible to combine different sources of information to estimate an imperfect network. The resulting structure from a combination of evidences is an imperfect network with existence probabilities for edges, nodes, and structures as well as missing and false edges.

C. Dempster-Shafer theory

The theory of evidence, or *Dempster-Shafer Theory* (DST) [13], is a generalization of probability theory which relaxes the axiom of additivity and introduces a different method for merging evidence from multiple sources. The theory also allows for the construction of intervals with upper and lower probabilities to include the uncertainty in merged conflicting evidence. DST is popular in some areas of artificial intelligence, decision support, and data fusion. Instead of probability functions, the theory of evidence use belief and plausibility functions, defined in *Definition 2*.

Definition 2 (Belief and plausibility functions): Assume that the *frame of discernment*, Θ, is a finite set and let 2^Θ denote the set of all subsets of Θ. Suppose that the *belief function* $\mathsf{Bel} : 2^\Theta \to [0,1]$ satisfy the following[3],

$$\mathsf{Bel}(\emptyset) = 0, \tag{4}$$

$$\mathsf{Bel}(\Theta) = 1, \tag{5}$$

$$\mathsf{Bel}(A_1 \cup \ldots \cup A_n) \geq \sum_i \mathsf{Bel}(A_i) - \sum_{i<j} \mathsf{Bel}(A_i \cap A_j) + \ldots + (-1)^{n-1} \mathsf{Bel}(A_1 \cap \ldots \cap A_n) \tag{6}$$

where n is some positive integer and every collection, $A_1, \ldots, A_n$, is a subset of 2^Θ. The *plausibility function* $\mathsf{Pl} : 2^\Theta \to [0,1]$ is the dual of the belief function,

$$\mathsf{Pl}(A) = 1 - \mathsf{Bel}(A^c), \tag{7}$$

where A^c denotes the complement of the subset $A \subset 2^\Theta$. The *Belief function*, $\mathsf{Bel}(A)$, is interpreted as the belief that the truth lies in some subset of the set A of all possibilities. The *Plausibility function*, $\mathsf{Pl}(\cdot)$, measure the failure to doubt the truth and note[4] that $\mathsf{Bel}(A) \leq \mathsf{Pl}(A)$ for each $A \subseteq 2^\Theta$. Therefore the probability that the truth lie in A is given by the interval $[\mathsf{Bel}(A), \mathsf{Pl}(A)]$. [13]

Combining probabilities from different sources is the essence of information fusion. Dempster-Shafer theory allow for combining evidence in a more general manner than in probability theory, presented in *Theorem 1*. Although no prior probabilities and likelihood functions are needed, we need to have a *mass function*, $m(\cdot)$, that assign probability mass to the frame of discernment, Θ. [15]

Theorem 1 (Dempster's rule of combination): Let $m_i : 2^\Theta \to [0,1]$, for $i = 1, 2$, be two (different) basic probability assignment functions on some frame of discernment, Θ, satisfying.

$$m(\emptyset) = 0, \quad and, \quad \sum_{A \in 2^\Theta} m(A) = 1. \tag{8}$$

The combined belief of a subset $A \subseteq 2^\Theta$ is,

$$m_{1,2}(A) = (m_1 \oplus m_2)(A) = [1-K]^{-1} \sum_{B \cap C = A \neq \emptyset} m_1(B) m_2(C), \tag{9}$$

where $m_{1,2}(\emptyset) = 0$ and K is the amount of conflict between the two beliefs defined by,

$$K = \sum_{B \cap C = \emptyset} m_1(B) m_2(C). \tag{10}$$

By using the basic probability assignment function, $m(\cdot)$, the belief and plausibility is found by the following expression,

$$\mathsf{Bel}(A) = \sum_{B \subset A} m(B), \qquad \mathsf{Pl}(A) = \sum_{B \cap A \neq \emptyset} m(B), \tag{11}$$

thus allowing for combination of evidence to construct probability intervals. By repeatedly adding evidence using this rule of combination, e.g. four evidences are combined using,

$$m_{1,2,3,4}(A) = (((m_1 \oplus m_2) \oplus m_3) \oplus m_4)(A), \tag{12}$$

any number of evidence can be combined. DST can thus be used to quantify the uncertainty in a network by combining evidence from several different sources.

V. Determining community structures in uncertain networks

In this section, we describe the complete method for detecting community structure in uncertain networks. For an outline of the method, see *Figure 3*. The figure shows the process from the uncertain network data at the top to the estimated community structure at the bottom. In practical applications of this method, we would use the results presented in previous chapters to combine evidence from proxy networks and other forms of collected information to find an estimated network structure with existence probabilities. The result is called an imperfect network and was discussed in the previous section.

The next steps are outlined in detail in this section:

1) sampling candidate networks from the ensemble of consistent networks using (Markov Chain) Monte Carlo-methods
2) detecting candidate communities using standard methods
3) merging candidate communities into the most probable community structure of the uncertain/imperfect network.

[3]The third condition (6) is related to the *inclusion-exclusion principle* in probability theory,

$$\left| \bigcup_{i=1}^{n} a_i \right| = \sum_{i=1}^{n} |A_i| - \sum_{i<j} |A_i \cap A_j| + (-1)^{n-1} |A_1 \cap \cdots A_n|, \tag{3}$$

which is similar to the condition if $\mathsf{Bel}(A_i) = |A_i|$ but has an equality where (6) has an inequality.

[4]This is due to that the belief is *sub-additive*, $\mathsf{Bel}(A) + \mathsf{Bel}(A^c) \leq 1$ and plausibility is *super-additive*, $\mathsf{Pl}(A) + \mathsf{Pl}(A^c) \geq 1$. Probability is additive and does in some situations coincide with the belief and plausibility, when $\mathsf{Bel}(A) + \mathsf{Bel}(A^c) = \mathsf{Pl}(A) + \mathsf{Pl}(A^c) = 1$.

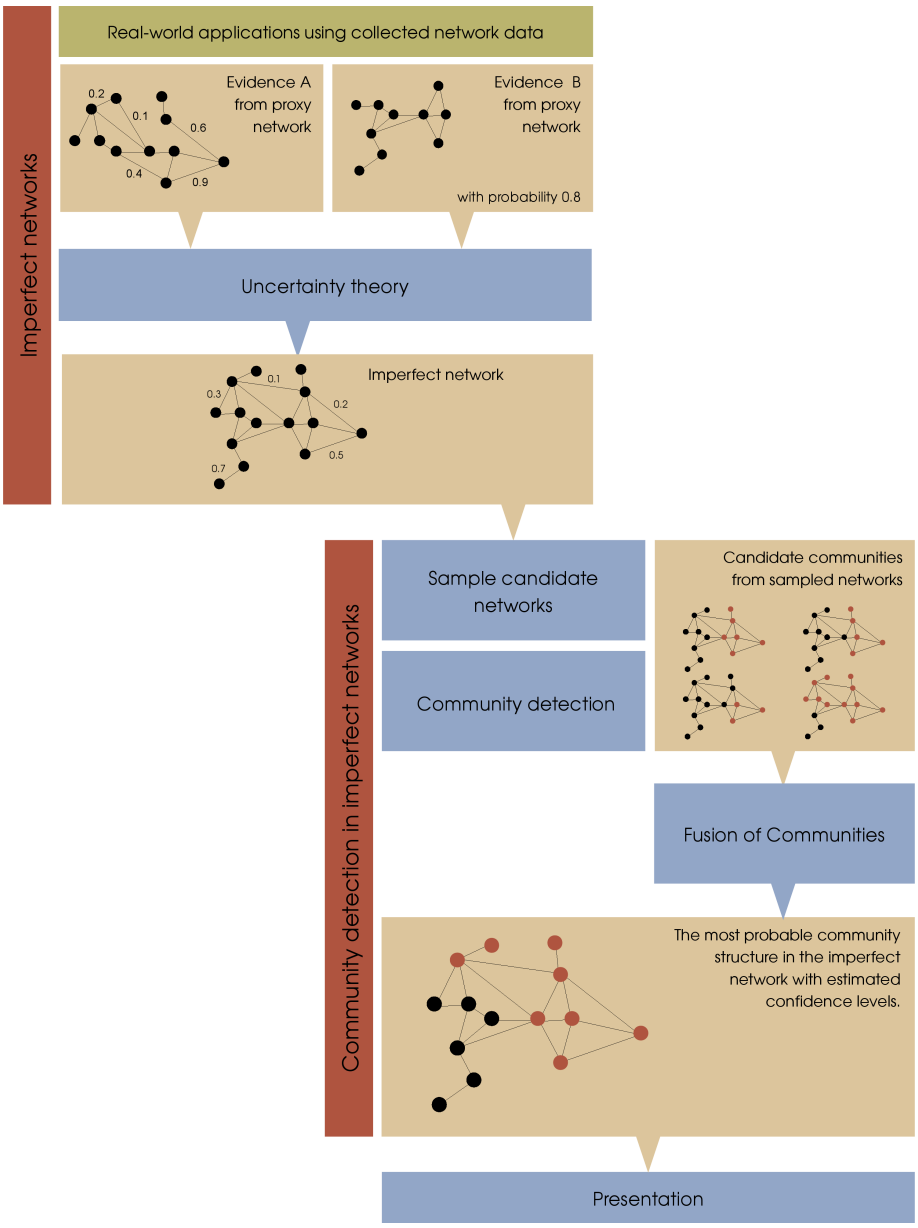

Fig. 3. The proposed method to detect communities in imperfect networks.

A. *Sampling candidate networks*

The introduced methods for quantifying and combining uncertainty in network information generate a probability or a probability interval. This measure corresponds to the degree of uncertainty that an edge, node, or structure exist in the network.

An uncertain network, $G(V, \mathbf{E})$, is constructed from these existence probabilities, $\mathbf{E} = [E_{ij}]$, where e.g. E_{ij} is the probability that an edge exist between nodes i and j. We limit ourselves to probabilities describing uncertain edges in this paper, however there are generalizations for other uncertain objects. Generalization of the method for handling probability intervals (from, e.g., Dempster-Shafer theory) is trivial.

The first step in detecting communities in uncertain networks is to generate an ensemble of networks that are consistent with this network information. Realizations are found using Monte Carlo-sampling with the existence probabilities, E_{ij}. The sampling is performed using uniformly distributed random numbers to generate a matrix, $\mathbf{R} = [R_{ij}]$, where $R_{ij} \sim \mathcal{U}[0, 1]$.

An edge is included in the graph if the random element in the corresponding matrix is less than or equal to the product of the corresponding elements of the edge probability matrix and the adjacency matrix, i.e. $R_{ij} \leq E_{ij}A_{ij}$. By simulating many

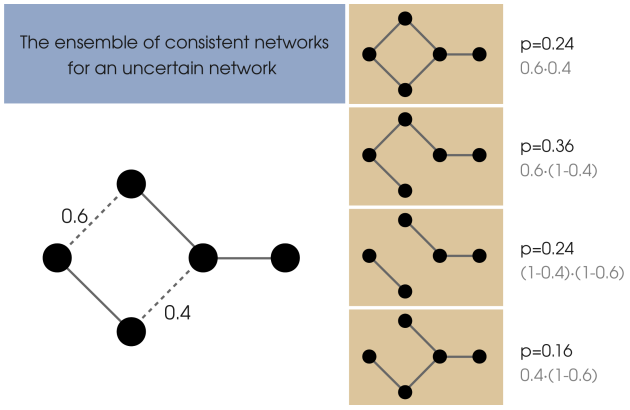

Fig. 4. A small uncertain network and its corresponding ensemble of networks. The proportion of networks of a certain form in the ensemble is determined by the edge existence probabilities.

random matrices, $\mathbf{R}$, a large ensemble of certain networks are generated, each consistent with the information in the uncertain network. The sampled networks are called *candidate networks* or *realizations* of the ensemble of consistent networks.

A simple situation using this sampling method in shown in *Figure 4*, where a small network is sampled. The ensemble of consistent networks consist of the four different possible permutations of the network structure. Each permutation exists with the proportion p and therefore also has the probability p of being sampled.

By sampling many networks from the ensemble, the idea is to mimic the distribution of permutations in the ensemble by the set of sampled networks. Which means that if 50 networks are sampled from the ensemble, then approximately 12 should be of the first type of permutation in which no edge is removed and so on.

To simplify simulations, all nodes of degree one are removed during the community detection and merging steps. After those steps, the nodes with degree one are re-added to the network again after these two steps have completed. The removed nodes (all of degree one) are assigned to the same community as their neighbor. As a further simplification, the community detection is only performed on the largest component of the graph, due to the fact that some algorithms only work on networks with one component.

B. Detecting candidate communities

For each realization of the imperfect network some community detection algorithms is applied. All these methods are used in combination with modularity calculations and assuming that maximizing modularity is equivalent to maximizing the quality of the community clustering.

The community structures of each realization (or candidate network) are referred to as *candidate communities*. The community structures for the n_s candidate networks of the uncertain network are summarized as a *membership matrix*,

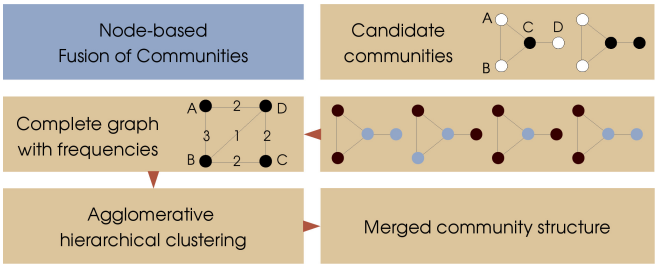

Fig. 5. Node-based Fusion using agglomerative hierarchical clustering with a special linkage calculation.

$\mathbf{M} = [M_{ik}]$, where M_{ik} is the community in which node i is a member in candidate network $k = \{1, 2, \ldots, n_s\}$ and n_s is the number of samplings.

C. Merging candidate communities

The remaining problem is to merge the different community structures found in each candidate network into one community structure. This is accomplished by merging nodes often found in the same cluster or by merging similar candidate communities. Two different methods are proposed to accomplish this: (i) *node-based fusion of communities* and (ii) *community-based fusion of communities*.

1) Node-based Fusion of Communities: The first method, *Node-based Fusion of Communities* (NFC) is an extension of Instance-based Ensemble Clustering presented in Refs. [16], [17]. The NFC-method is outlined in *Figure 5* which begins with the construction of a complete graph $G = (V, \mathbf{F})$ from the candidate communities. In the new graph, nodes correspond to nodes in the old graph and the weighted edges correspond to the frequency of instances when two nodes have been grouped together in the same candidate community.

The nodes are clustered using agglomerative hierarchical clustering with the edge weight as the similarity between two nodes. Thus nodes often found in the same candidate cluster are grouped together by the hierarchical clustering method.

The main difference of this method compared to IBEC is the process in which the frequency, F_{ij}, is recalculated after each merge. The frequency between the merged nodes (cluster) l and the other nodes or clusters, $v_1, v_2, \ldots, v_{n_l}$, is found by

$$F_{k,l} = \left| \bigcap_k M_{kl} \right|, \tag{13}$$

where the *membership matrix*, $\mathbf{M} = [M_{ik}]$, where M_{ik} is the community in which node i is a member in candidate network $k = \{j, i_1, \ldots, i_{n_l}\}$. That is, $F_{k,l}$ is the number of occurrences where all nodes (in both clusters) are in the same candidate cluster.

This incurs some loss of information about individual nodes as they are clustered together and information about the similarity of individual nodes are lost. The result of the hierarchal clustering algorithm is a dendrogram and a list of merges. The clustering corresponding to the maximum modularity is taken as the communities found in the merged candidate networks.

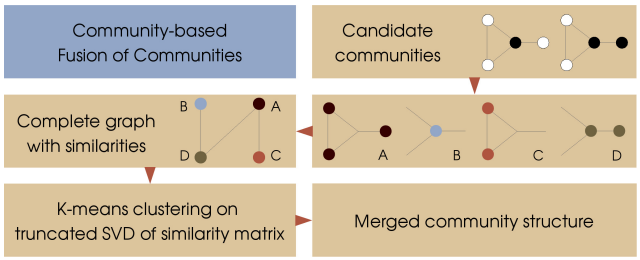

Fig. 6. Community-based Fusion using k-means clustering on the truncated singular value decomposition of the adjacency matrix.

2) Community-based Fusion of Communities: The second method is based on Community-based Ensemble Clustering and by singular decomposition of the similarity matrix, $\mathbf{S} = [S_{ij}] = [\mathsf{sim}(i,j)]$. This method is quite dissimilar to NFC because it is based on merging similar clusters and not similar nodes.

The full method for merging candidate communities using *Community-based Fusion of Communities* (CFC) is outlined in *Figure 6*. The first step is to construct a complete graph $G = (V, \mathbf{S})$, with a set of nodes, $V(G)$, consisting of each candidate cluster, $\mathcal{C}_i$. The similarity matrix, $\mathbf{S} = [S_{ij}]$, is calculated using the the cosine measure,

$$S_{ij} = \frac{|\mathcal{C}_i \cap \mathcal{C}_j|}{\sqrt{|\mathcal{C}_i|\,|\mathcal{C}_j|}}, \tag{14}$$

where $\mathcal{C}_i$ and $\mathcal{C}_j$ are candidate clusters for some k such that $i, j = 1, \ldots, k$ and $i \neq j$. This measure is chosen due to its property of linearity and that it is scaled to unity. The similarity matrix is expanded using singular value decomposition. By using the k first eigenvectors, a low-dimensional representation of the clustering is found.

This approximation is clustered using k-means to find the meta-clustering of candidate clusters. Each node, from the original network, is assigned to the community it is most often a member of. This is repeated for all possible values of $k \leq n$, to find the final community structure of the network, i.e. the one that maximize the modularity.

D. Results

Figure 7 shows the results of the community detection method on three different real networks where there is a known community structure that we can compare with. The networks used are the well-known karate[18], football[19] and dolphin [20]networks. The figure shows the mutual information between the correct community structure and that determined by running the methods as a function of the added uncertainty. The x axis denotes the fraction of edges that have uncertainty associated with them.

The merging method NFC perform as good as or better than CFC when using most community detection algorithms. When using NFC to merge candidate communities, LP is the best choice when observing both NMI and correlation. Using CFC, we get the best results using the SP algorithm (for karate and football networks) and the GA algorithm (for the dolphin network).

VI. Discussion and Conclusion

In this paper, we presented a method for computing community structures of networks where we do not have complete knowledge of nodes and edges. The method is based on generating an ensemble of certain networks that are consistent with the information available about the real network. Community structures are then computed for each such certain network, and the results merged. The method can be used not only when we have knowledge about edge probabilities, but also if there is information about more complicated network substructures and their probabilities. The method for merging the results of the community detection methods can also be used to merge the results of several different community detection algorithms applied to the same certain network.

Results that indicate that it is possible to retrieve community structures in sample networks with added uncertainty were briefly discussed.

We see several possibilities for future work in this area. We are conducting more thorough studies of the robustness of the community detection method when applied to different sample networks with different degrees of uncertainty. The general method for analyzing uncertain networks here can be trivially applied to other network measures. Community structure as a concept is not well-defined, and it would be interesting to investigate this further in the context of partially and incompletely observed networks. The test method whereby a certain network is observed with a fixed degree of uncertainty added could be used for investigating the concept of community in more detail. Another interesting research challenge in this area is how to quantify the uncertainty in the input network data. More research is needed on adequate observation models for uncertain network data.

Acknowledgment

This research was supported by the FOI project "Tools for information management and analysis", which is funded by the R&D programme of the Swedish Armed Forces.

References

[1] S. Wasserman and K. Faust, *Social network analysis : methods and applications*, 1st ed., ser. Structural analysis in the social sciences, 8. Cambridge University Press, Nov. 1994. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0521387078

[2] J. P. Scott, *Social Network Analysis: A Handbook*. SAGE Publications, Jan. 2000. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0761963391

[3] P. Svenson, P. Svensson, and H. Tullberg, "Social network analysis and information fusion for anti-terrorism," in *Proceedings of the Conference on Civil and Military Readiness 2006*, 2006.

[4] L. Ferrara, C. Mårtenson, P. Svenson, P. Svensson, J. Hidalgo, A. Molano, and A. L. Madsen, "Integrating data sources and network analysis tools to support the fight against organized crime," in *In Proceedings of the IEEE ISI 2008 International Workshops: PAISI, PACCF, and SOCO 2008*, 2008, pp. 171–182.

[5] R. M. Clark, *Intelligence Analysis: A Target-Centric Approach*. CQ Press, 2003.

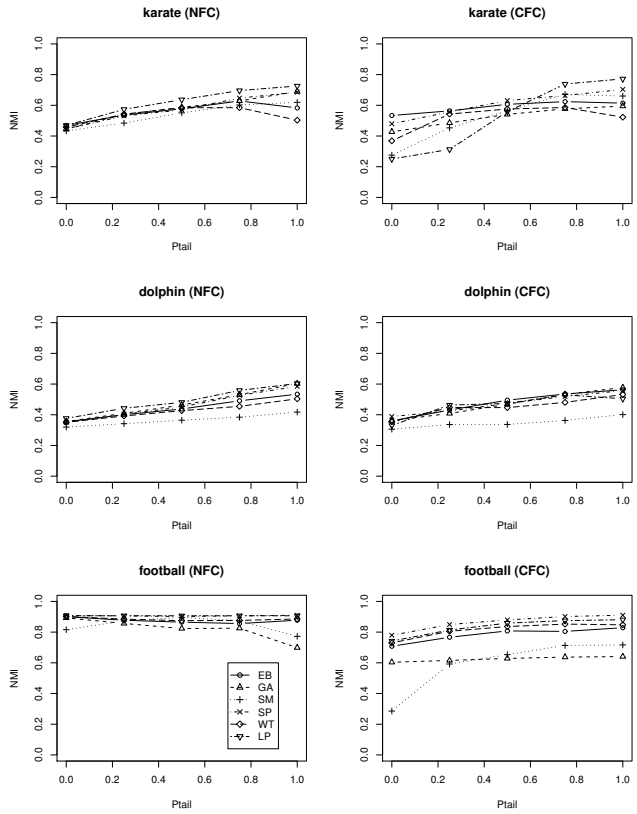

Fig. 7. This figure shows the normalized mutual information between the correct community structure and that determined by the NCF and CFC merging algorithms for the karate, dolphin and football networks, using 6 different community detection methods. The x-axis denotes the fraction of edges in the network that are certain. . The curves are estimated by using non-parametric regression with the Gaussian kernel on the result of 30 runs each using 50 candidate networks.

[6] M. Newman, *Networks: An Introduction*, 1st ed. Oxford University Press, USA, May 2010. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0199206651

[7] J. Reichardt and S. Bornholdt, "Statistical mechanics of community detection." *Phys Rev E Stat Nonlin Soft Matter Phys*, vol. 74, no. 1 Pt 2, Jul. 2006. [Online]. Available: http://dx.doi.org/10.1103/PhysRevE.74.016110

[8] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E*, vol. 76, no. 3, pp. 036 106+, Sep. 2007. [Online]. Available: http://dx.doi.org/10.1103/PhysRevE.76.036106

[9] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences*, vol. 105, no. 4, pp. 1118–1123, Jan. 2008. [Online]. Available: http://dx.doi.org/10.1073/pnas.0706851105

[10] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society." *Nature*, vol. 435, no. 7043, pp. 814–818, Jun. 2005. [Online]. Available: http://dx.doi.org/10.1038/nature03607

[11] M. A. Porter, J.-P. Onnela, and P. J. Mucha, "Communities in Networks," Sep. 2009. [Online]. Available: http://arxiv.org/abs/0902.3788

[12] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, pp. 75–174, Feb. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.physrep.2009.11.002

[13] G. Shafer, *A mathematical theory of evidence*. Princeton university press Princeton, NJ, 1976.

[14] P. Svenson, "Social network analysis of uncertain networks," in *Proceedings of the 2nd Skvde Workshop on Information Fusion Topics*, 2008.

[15] E. Bossé, J. Roy, and S. Wark, *Concepts, Models, and Tools for Information Fusion*, 1st ed. Artech House, Inc., 2007.

[16] S. Monti, P. Tamayo, J. Mesirov, and T. Golub, "Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data," *Machine Learning*, vol. 52, no. 1, pp. 91–118, Jul. 2003. [Online]. Available: http://dx.doi.org/10.1023/A:1023949509487

[17] X. Z. Fern and C. E. Brodley, "Solving cluster ensemble problems by bipartite graph partitioning," in *ICML '04: Proceedings of the twenty-first international conference on Machine learning*. New York, NY, USA: ACM, 2004, pp. 36+. [Online]. Available: http://dx.doi.org/10.1145/1015330.1015414

[18] W. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of Anthropological Research*, vol. 33, pp. 452–473, 1977.

[19] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, pp. 026 113+, Feb. 2004. [Online]. Available: http://dx.doi.org/10.1103/PhysRevE.69.026113

[20] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations," *Behavioral Ecology and Sociobiology*, vol. 54, no. 4, pp. 396–405, 2003. [Online]. Available: http://dx.doi.org/10.1007/s00265-003-0651-y