Data Mining in Complex Networks: Missing Link Prediction and Fuzzy Communities

Ph.D. thesis booklet

Tamás Nepusz

Advisors:
Dr. Fülöp Bázsó (MTA KFKI-RMKI)
Dr. György Strausz (BME MIT)

Budapest University of Technology and Economics, Department of Measurement and Information Systems

2008
Overview

This dissertation is about networks: complex systems consisting of unique elements connected by binary relations arranged in seemingly random but intrinsically structured patterns. Network theory was successfully applied to model various real-world phenomena ranging from the interactions of proteins in living organisms to the large-scale organization of human society or to the structure of man-made technological networks like the Internet. Since networks are built from binary relations among entities, they can naturally be transformed into graphs, allowing one to study network properties by the tools of a well-established field of mathematics, namely graph theory.

My dissertation consists of two major parts, and this is also reflected in the contents of this thesis booklet. In the first part, I describe and examine a stochastic graph model where vertices of the graph are assigned to vertex types, and the connection probability of any two vertices depends solely on the types of the vertices involved. My primary aim was to apply this model to the problem of predicting unknown connections in a network whose connectional structure is known only partially. The second part of my dissertation investigates a method that finds dense subgraphs (modules, communities, clusters) in a network under the assumption that these network modules are not well-separated and vertices of the network may belong to multiple communities at the same time.

Predicting the probability of unknown connections in complex networks

Introduction

Most of the state-of-the-art tools in network science assume that the connections of the network being studied are either completely known, or even if they are not, the uncertainties in the
dataset are negligible. However, this assumption does not hold in many cases, especially when our knowledge about the connections stems from experiments. This is a common scenario in biology and sociology. A peculiar example is the graph model of neural connections in the cortex, since the existence of a given connection between two cortical areas can only be proved or disproved by expensive and complicated experiments with sometimes ambiguous results. It is therefore of primary importance to estimate the probability of the existence of yet uncharted neural connections based on the known ones in order to help experimenters to concentrate on those that are likely to exist.

There are multiple ways to tackle the link prediction problem, which can roughly be classified as follows:

**Methods based on local similarity indices.** These methods calculate a similarity measure between all pairs of vertices in the graph based on simple local properties of the vertices (and possibly some additional *a priori* information). The common underlying assumption of these methods is that uncharted connections are likely to exist between vertex pairs with high similarity scores. These scores are usually derived from the set of neighbours of the vertices and the amount of overlap between these sets. Some exemplars of these methods are the Jaccard similarity score or the inverse log-weighted similarity index of Adamic and Adar [1].

**Methods based on paths and random walks.** These methods assess the similarity of two given vertices from the set of paths or random walks between them. Most probably one of the oldest method from this family is the Katz similarity score [10], which considers all the possible paths between vertex pairs. Since in general there are infinitely many paths between any two of the vertices (allowing vertex repetitions in paths), the weight of longer paths is damped exponentially to keep the sum finite. When $P_{x,y}^{(k)}$
denotes the set of paths of length $k$ between vertices $x$ and $y$, and $0 < \beta < 1$ is the damping factor, the Katz similarity of $x$ and $y$ is given by:

$$\sum_{k=1}^{\infty} \beta^k |P_{x,y}^{(k)}|$$ (1)

Another frequently used path-based similarity measure is SimRank [3]. Similarly to the well-known Google PageRank measure, SimRank is defined by a self-consistent recursive equation. The basic idea is that two vertices are similar if they have incoming edges from similar vertices. Let $\Gamma^+(x)$ be the set of predecessor vertices of $x$ (i.e., for every vertex in $\Gamma^+(x)$, there exists at least a single edge that originates from this vertex and terminates in $x$), and let $0 < \gamma < 1$ an appropriate damping factor (a common choice is 0.8 [3]). The definition of SimRank is then as follows:

$$\text{SimRank}(x, x) = 1$$
$$\text{SimRank}(x, y) = \gamma \frac{\sum_{a \in \Gamma^+(x)} \sum_{b \in \Gamma^+(y)} \text{SimRank}(a, b)}{|\Gamma^+(x)| \ |\Gamma^+(y)|}$$ (2)

It can be shown that the solution of the equations above is unique and can be determined iteratively [3]. At the same time, SimRank is the expected value of $\gamma^L$, where $L$ is a random variable describing the number of steps required for two random walks started from $x$ and $y$ to meet for the first time.

A common disadvantage of the methods described above is that unknown or uncertain connections are treated as nonexistent, thus they are more appropriate for prediction problems where one has to extrapolate to the future behaviour of the network.
based on the present state, which is assumed to be known completely \[11\].

There is also a third, relatively new approach of the link prediction problem: let us first construct an appropriate random graph model that is sophisticated enough to describe the network being studied, then find the parameterisation of the model that reproduces the given network with the highest probability, and use the probability of the existence of our unknown connections in the model to estimate their probability in the real network. If the original random graph model is able to handle known non-existing and unknown connections differently, one can expect more accurate predictions from these methods than from the ones that do not distinguish between them. The method described in my dissertation utilises these ideas.

**Research goals**

My goal was to conceive a method that is able to estimate the probability of unknown connections in a given static network under the following assumptions:

1. All the vertices in the network are known, the possibility of adding new vertices or merging existing ones should not be considered.
2. Vertex pairs are either connected (and confirmed), disconnected (and confirmed), or uncertain (their connectedness is unknown, but we may have an associated *a priori* degree of belief regarding its existence).
3. Vertex pairs are ordered; in other words, the connections are directed, and the existence of a connection from $A$ to $B$ does not imply the existence of a connection in the opposite direction.
Research methods

I defined a stochastic graph model (called the **preference model**), where the connection probabilities of vertex pairs are governed by vertex types assigned to the vertices involved. The model uses \( k \) types, and every vertex has two types at the same time. Let us call these types **in-types** and **out-types**. The in- and out-types of the vertices are encoded by integers between 1 and \( k \) in vectors \( \vec{u} = [u_i] \) and \( \vec{v} = [v_i] \). The model also contains a **preference matrix** \( \mathbf{P} = [p_{ij}] \) of size \( k \times k \), where \( p_{ij} \) describes the probability that a vertex with out-type \( i \) connects to a vertex with in-type \( j \). Therefore, the regularity in the structure of the network is described by the type assignments of the vertices and the preference matrix. Fitting this stochastic graph model to the network being studied is the basis of the probability estimation outlined in my research goals.

Since the network can contain uncertain connections, I could not have used traditional graph descriptions such as adjacency matrices or adjacency lists. I had to find a description that extends one of these descriptions in a way that enables us to take into consideration the additional information regarding the uncertainty and the degree of belief associated to each possible connection. The network being studied is described by a \( b_{ij} \) belief value for every \((i, j)\) ordered vertex pair: \( b_{ij} = 1 \) in the case of a confirmed existing connection, \( b_{ij} = 0 \) for a confirmed missing connection and \( 0 < b_{ij} < 1 \) for uncertain connections, with higher values corresponding to higher degrees of belief – in other words, to higher a priori probabilities based on our additional domain-specific knowledge. The matrix \( \mathbf{B} = [b_{ij}] \) equals the adjacency matrix of the original graph if it is completely known, thus effectively demonstrating that the adjacency matrix representation is simply a special case of the belief matrix. The belief matrix can be stored efficiently as a sparse matrix if the network is sparse and most missing connections are confirmed.

The preference model can be fitted to an arbitrary network by
finding the parameterisation of the model with the maximum likelihood, given the belief matrix $B$ of the network. The estimated \textit{a posteriori} probabilities can then be determined by the fitted vertex types and their corresponding entries in the calculated optimal preference matrix.

Let $G_0$ denote the graph being analysed and let $n$ be the number of vertices in $G_0$. The likelihood of a given parameterisation $\theta = (k, \vec{u}, \vec{v}, P)$ is then as follows:

$$L(\theta|G_0) = \prod_{i=1}^{n} \prod_{j=1}^{n} (b_{ij}p_{u_i,v_j} + (1 - b_{ij})(1 - p_{u_i,v_j}))$$  \hspace{1cm} (3)

Practical applications use the logarithm of the likelihood in order to avoid rounding errors and numerical instabilities caused by floating point calculating involving very small probabilities. The log-likelihood can be maximised by one of the following methods or their combination:

\textbf{Expectation-maximization (EM) method.} Starting from a random type assignment, one can find a local optimum by repeatedly applying two steps. One of them is the E-step (based on the first letter of \textit{expectation}), while the other one is called the M-step (denoting \textit{maximisation}). By temporarily assuming $\vec{u}$ and $\vec{v}$ (the type assignments) to be constant, the E-step determines $E(\log L(\theta|G_0))$ and then estimates $E\rho_{ij}$ for every possible type pair. The M-step uses the estimated $P$ matrix resulting from the E-step to modify the vertex types in a way that maximises the local contribution of every vertex to the log-likelihood under the assumption that no other vertices will change their group assignments. The algorithm stops when no modification was performed in the M-step (since this implies that nothing will change in the E-step as well).

\textbf{Markov chain Monte Carlo (MCMC) method.} This algorithm performs a random walk in the space of all possible
parameterisations of the preference model. Every step in the random walk involves changing the type assignment of a single vertex chosen randomly. Elements of the \( P \) preference matrix are then re-estimated based on the new configuration similarly to the E-step in the EM algorithm. The new state is accepted unconditionally as the next state in the random walk if its likelihood is higher than the likelihood of the old state. If the new likelihood is lower than the old one, the ratio of the new and the old likelihoods gives the probability of acceptance. This scheme is the application of the Metropolis–Hastings algorithm \([5]\) for this specific problem, therefore the state probabilities in the stationary distribution of the resulting Markov chain are proportional to their likelihoods. By taking a large number of samples from the chain after a sufficiently long burn-in period (which lets the residual effects of the starting state diminish), we can find a parameterisation with high likelihood.

The shortcoming of the EM algorithm is that it can get stuck in a local maximum, but this is counterbalanced by the fact that it converges fast. The MCMC algorithm is free from this shortcoming, since we occasionally allow steps towards worse states as well. In practice, the advantages of the methods can be combined by replacing the burn-in stage of the MCMC process by EM iterations. The Markov chain is then started from the local maximum found by the EM process.

**Results**

**T 1/1.** I showed that vertex degrees in the networks generated by the preference model are described by the weighted sum of Poisson-distributed random variables. I also proved a sufficient condition for the existence of a giant component in these networks.
T 1/2. I devised and implemented algorithms to fit the parameters of the model to a given network, taking into account the degrees of belief associated to the possible connections in the network. I tested the validity of these algorithms on computer-generated test graphs.

T 1/3. I showed that the Akaike information criterion \[2\] is able to choose the most appropriate number of vertex groups of the model in an unsupervised manner.

Publications related to theses T 1/1., T 1/2., T 1/3.:


**Applications**

The applicability of the model and the algorithms is demonstrated in the field of biology, since network datasets in biology usually originate from experiments, therefore they frequently
contain missing data. The dataset I studied was the graph model of the visual and sensorimotor cortices of the macaque monkey as described in [12]. This network incorporated 45 brain areas and 463 confirmed existing neural connections between them. 360 pairs of areas were known to be disconnected, and no information was available regarding the remaining 1157 pairs. Such uncertainty poses a challenge to even the state-of-the-art link prediction approaches.

The preference model was able to reconstruct the known part of the cortical network with high confidence (92.7% of known existent and 83.1% of known nonexistent connections were predicted correctly). Results pertaining to the visual areas of the network describe the most exact reconstruction published in the literature so far, and the predictions regarding the unknown connections also seem plausible in line with earlier reconstruction attempts [3, 9]. ROC curves were also used to compare the method to other generic link prediction methods (see Fig. II).

Conclusion

I presented a method that is able to estimate the probability of unknown connections in a static, directed complex network, taking domain-specific information into account by the means of \textit{a priori} connection probabilities (belief values) and I shown the applicability of the method on a real prediction problem. The method is relevant not only in the field of biology, but in all problems where researchers are confronted with networks which are known to be incomplete.

Fuzzy communities in undirected networks

Introduction

A common feature of networks modeling natural phenomena is sparseness: the vast majority of possible connections are miss-
Figure 1: Comparison of the preference model with other generic link prediction methods on the cortical network dataset. AUC = area under curve, attaining its maximum at 1 when the reconstruction is perfect.

...ing, thus the number of actual edges grows linearly in the number of vertices as the network size tends to infinity. In spite of their sparsity, these networks frequently contain dense submodules, which tend to coincide with larger functional units of the network. For instance, dense subgraphs of a social network usually correspond to circles of friends, groups of coworkers and so on. One of the most studied problems of network theory is the efficient identification of such dense subgraphs [4], also called modules, communities or clusters. It can also be demonstrated that these communities can overlap with each other [4], but most community detection algorithms assume that every vertex belongs to one and only one of the communities. The difference between the overlapping and the nonoverlapping approach is illustrated on Fig. 2.
Figure 2: Nonoverlapping and overlapping clusters of a simple example graph. Left: nonoverlapping clustering with two clusters according to the algorithm of Clauset et al. [4]. Right: overlapping clustering with two clusters. The bridge-like position of the central vertex is not revealed by the nonoverlapping approach.

**Research goals**

Research on algorithms that are able to detect overlapping communities is a relatively new problem in network science. At the time when I started my own investigations, there was no algorithm that was able to quantify how much does a given vertex belong to a given community; the algorithms available were only able to decide whether a given vertex belongs to a given community or not. Therefore, my aim was to develop an algorithm that is able to identify overlapping communities in complex networks and characterise the membership degrees of the vertices with respect to the detected communities. I assumed that edges in the network are undirected and the network topology is known exactly.
Research methods

My research was inspired by the fuzzy c-means clustering [3, 6]. Fuzzy c-means clustering was proven to be useful and efficient in problems when the points to be clustered were embedded in an n-dimensional space with an appropriate distance function. However, there is no single straightforward embedding and distance function for graphs, so I had to take a different approach. Formally, the output of a fuzzy clustering is a fuzzy partition matrix, denoted by $U = [u_{ki}]$ from now on. $u_{ki}$ denotes the membership degree of vertex $i$ in cluster $k$. The following constraints are imposed on the elements of the matrix:

1. $0 \leq u_{ki} \leq 1$ for all $i$ and $k$.
2. $0 < \sum_{i=1}^{n} u_{ki} < n$ for all $k$, where $n$ is the number of vertices in the graph. Informally: clusters can not be empty and no cluster can contain all the vertices to the greatest possible extent.
3. $\sum_{k=1}^{c} u_{ki} = 1$ for all $i$, where $c$ is the number of clusters. Informally: the sum of all membership degrees pertaining to a given vertex is 1, therefore we are not interested in outlier vertices that do not belong to any of the clusters.

My algorithm is based on a similarity function defined over pairs of vertices. If we think about the membership degrees as probabilities ($u_{ki}$ is the probability of the event that vertex $i$ is in cluster $k$), the probability of the event that vertices $i$ and $j$ are in the same cluster equals the dot product of their respective membership vectors: $s_{ij} = \sum_{k=1}^{c} u_{ki} u_{kj}$. It follows that the similarity matrix $S = [s_{ij}]$ based on these probabilities is simply $S = U^T U$. The key assumption of the algorithm is that the presence of an edge between two vertices relates to their similarity, while the absence of an edge implies dissimilarity. Therefore, one should try to find a matrix $U$ that makes connected vertices similar and disconnected vertices dissimilar. We can define
a goal function that quantifies the goodness of fit for a given $U$ based on the sum of squared differences between the expected and the actual similarity:

$$f(U) = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} (\tilde{s}_{ij} - s_{ij})^2,$$

(4)

where $w_{ij}$ is an arbitrary weighing term and $\tilde{s}_{ij} = 1$ if and only if $i$ are $j$ connected or $i = j$, zero otherwise. This goal function has to be optimised with respect to the constraints defined above.

The constraint on the sum of membership degrees of a given vertex can be incorporated into the goal function by Lagrange multipliers, leading to a constrained nonlinear optimisation problem where the individual variables ($u_{ki}$) can take values from the range $[0; 1]$. Starting from a random configuration, the value of the goal function can be optimised by standard gradient-based optimisation methods (e.g., steepest descent or the method of conjugate gradients).

One of the advantages of fuzzy clustering compared to nonoverlapping clustering is that it is able to quantify the sharedness of a vertex between groups. I introduced several measures to achieve that goal:

**Bridgeness.** Intuitively, a vertex is a bridge between communities to the greatest possible extent if it belongs to all the clusters with the same membership degrees. This state is characterised by a membership vector whose coordinates are $1/c$. The other extreme is when the vertex belongs to only one of the communities, resulting in a membership vector with a single element of 1 (all other elements are zeros). Note that the variance of the vector components is zero in the former case and the maximal variance of $(c - 1)/c$ is attained in the latter case. The bridgeness measure can therefore be derived from the variance of the
membership vector after appropriate renormalisation:

\[ b_i = 1 - \sqrt{\frac{c}{c-1} \sum_{j=1}^{c} \left( u_{ji} - \frac{1}{c} \right)^2} \]  \hspace{1cm} (5)

Bridgeness can also be weighted by the centrality of the vertex, allowing one to filter vertices that are outliers (having large bridgeness with small centrality).

**Exponentiated entropy.** Another possible approach is to consider the membership vector of vertex \( i \) as the probability mass function of a discrete random variable \( U_i \) and calculate the entropy of the variable. The entropy of \( U_i \) will be lower if vertex \( i \) is a nonoverlapping vertex and higher if \( i \) is a significant overlap. By using base-2 logarithm in \( H(U_i) \) (the entropy of \( U_i \)), the number of significant communities can be obtained by \( \chi_i = 2^{H(U_i)} \):

\[ \chi_i = 2^{-\sum_{k=1}^{c} u_{ki} \log_2 u_{ki}} = \prod_{k=1}^{c} u_{ki}^{-u_{ki}} \]  \hspace{1cm} (6)

**Results**

**T 2/1.** I devised and implemented an algorithm to find fuzzy communities in undirected networks. The algorithm is based on the maximisation of a global goal function derived from vertex similarities. I tested the validity of the algorithm on computer-generated test graphs.

**T 2/2.** I extended the modularity measure of Newman \[13\] to account for the fuzziness of the obtained partitions. I showed how one can employ the fuzzified modularity to choose the optimal number of communities.
T 2/3. *I quantified the sharedness of vertices between fuzzy communities by introducing the bridgeness, the weighted bridgeness and the exponentiated entropy measures of the membership vectors.*

Publications related to theses T 2/1., T 2/2., T 2/3.:


**Applications**

The applicability of the method is demonstrated again on the cortical network dataset described in the previous part. Vertices of the network can be classified as brain areas related to either visual or tactile input processing. Visual areas can also be subdivided based on anatomical considerations. My expectations were that fuzzy clustering should be able to find the bisection between visual and tactile input processing areas and should identify the areas related to the integration of visual and tactile information as bridges (since the integration task requires strong connections to both clusters).

The most appropriate fuzzy clustering of the cortex was obtained with four clusters. Two of these four clusters included mostly visual areas, the remaining two contained mostly tactile input processing areas. Only two areas were misclassified and the known anatomical subdivision of the visual cortex was also recognisable. There were five areas that were identified as bridges based on the centrality-weighted bridgeness measure; scores for these areas were significantly higher than the average.
bridgeness in the network. All of these areas can be considered as higher level integratory areas according to our present understanding of the visual and sensorimotor cortices. The two misclassified areas were also among these five bridges, suggesting that the classification error is caused by the bridge-like position of these areas.

**Conclusion**

The methodology described in this part of the dissertation is suitable for detecting communities in complex networks even when these communities overlap or their boundaries are not well defined. Some more illustrations of the results of the method on real datasets are also provided. The detected bridges deserve further attention, since these vertices may play a crucial role in the system modeled by the network structure. Further research directions include (but are not limited to) the extension of the method to directed and weighted graphs, outlier vertices and alternative similarity measures.

**List of publications**

**Publications related to the Ph.D. theses**


**Further related publications**


Other publications


Bibliography


