Economic Network Analysis based on Infection Models

M. Krész
Institute of Applied Sciences
University of Szeged, Szeged, Hungary
kresz@jgypk.u-szeged.hu
and
A.Pluhár
Institute of Informatics
University of Szeged, Szeged, Hungary
pluhar@inf.u-szeged.hu

Synonyms

Graphs in economy, data mining and knowledge discovery in economic networks, prediction of credit default and churn.

Glossary

(Strong) Component: maximal set of vertices that are reachable from each other by (directed) paths.
Cluster: a class of a partition.
Community: a dense part of a graph (possibly overlapping).
Host graph: the vertices are companies; the edges represent either money transfer or other type of connections.
Intersection graph: here the vertices correspond to sets, and edges are drawn if the sets intersect.
Transaction graph: edges mean some transaction (call, money flow etc.) among vertices (clients).

Definition

There are several ways to associate graphs (host graphs) with economic data, and in fact this process is a crucial step in any application. Different types of effects or information can spread on an economic network, i.e. the bankruptcy of business partners, the customer's churn, patterns of fraud and this spreading can be modeled by different infection models. Hidalgo et al. [16] introduced a classic economic graph model based on global export-import data to build up a graph on the "*product space*" of a country which turned out to be useful to understand its economic development: The vertices of the graph are economic products and an edge *(x, y) represents* that the export of product *x* increases the conditional probability of exporting product *y* as well. In this network representation, the products of a country can be considered as *infected vertices*, while the economic development is the spread of this *disease*.
Another classic example of economic networks is based on data warehouses of companies or organizations. ; a non-exhaustive list of some examples follows. There are different transaction (or call) graphs, based on the interactions taken by the customers of a bank or telephone company [5, 10] where the directed and weighted edges represent e.g., the total sum of money transfers, or the number of calls and messages in a certain time period. Similarly, graphs can be built up from economic data published on the web, like using WWW links, community pages, or blogs. Even in those cases where there are no obvious transactions; graphs might be defined to express indirect relations like the similarity between customers or organizations. For example, a usual company database has fields containing the name of the CEO, the shareholders, the given company's address, and its economic activity that can be considered as discrete sets of properties. Two entities can then be connected based on the similarity of their properties – in the simplest case if they just have a number of common elements [7]; the latter network representation is called an *intersection graph*. The same approach works for individual customers in a bank or in an insurance company. Once the graphs are obtained, models can be developed that might provide deeper understanding. Most of the questions in economics mathematically answerable in these models are formulated as computing (approximating) functions of the vertices, edges and their attributes: Among others, Hidalgo et al. [16] tried to predict the economic development of a country, given that it already has certain industries. The current status is coded in the product space graph with a function $f: V(G) \to \{0,1\}$ on the vertices; 1 if the industry is present, 0 otherwise. The question is then how f changes in time.
Similarly, based on transaction or similarity graphs the spread of events can be simulated: For telephone companies *churn* (chance of losing a costumer), for insurance companies *fraud* (of a contract or client) and for banks both of the above as well as the probability of *credit default* are very important. In this case, the weight of the edges (or the probability of the spread of an event through those) is a crucial issue.
To properly estimate these probabilities, the network structure has to be analyzed together with the information associated with the vertices and edges. In the study of a marketing problem by Domingos and Richardson [12], a simple spreading model is proposed which applies uniform infection probabilities. While this is satisfactory for a viral marketing problem, economic networks in general require a more careful approach. A new model proposed by Csizmadia et al. [11] uses a number of sophisticated algorithms in order to approximate the edge probabilities through application data (e.g., credit default). These algorithms were partly borrowed from network analysis, AI and data mining

or developed to meet the needs. In the next chapter we briefly discuss some of those.

Before going further we must stress that the use of infection models does *not* imply that the process is completely analogous to an epidemic as the infection models predict only the probabilities of various events. For instance, in a bank transaction graph the probability of bankruptcy can be predicted by infection processes starting from identified defaulting companies. Nevertheless, the estimation of infection probabilities requires a sophisticated analysis, as real infections are frequently influenced by information that are not represented in the given network structure: for example, common ownership of companies can contain significant information with respect to defaulting prediction in bank transaction graphs. Intersection graph models provide an alternative for the above problem; they might represent hidden information spread in a more realistic way.

Key Applications

As we mentioned before, the obvious and tested applications of infection processes are based on problems in the banking sector, the insurance sector, and the telecommunication sector. Note that projects of varying subjects or objectives (acquisition, bankruptcy, credit default, costumer churn, fraud detection etc.) start with different host graphs, and the infection processes are also tailored to the specific applications.

We will refer to the credit default model [10] several times, using it as an illustrative example. The experiences of its use originate from a case study of network research in one of the largest corporate banks in Hungary. The objective of the research was risk decisions (performing at the bank), especially bankruptcy forecasting.

Introduction

Infection models.

There are several types of models depending on the speciality of the field; for a brief introduction see Chapter 7 of [18]. The type of events (default, churn, fraud, route of development) does not involve recovery or resistance, so simple percolation models suffice.

SI models. First of all, for all the applications listed above, a version of SI infection models were used. In an SI model each vertex is either *susceptible* or *infected*, and the infected vertices can transmit the disease to the susceptible ones. The rules of the transmission vary in different models. A main categorization differentiates between *deterministic* and *probabilistic infections:*

Deterministic infections.

1. A node will be infected once a given threshold of its neighbors is infected (Bootstrap percolation). This type of infection model has been used in the product space network described above; it is implemented such that, itself. The method is convincing, although it gives only qualitative predictions [16].

2. Another infection model infects all those vertices that are *reachable* from the initially infected set. This simple approach works surprisingly well in fraud detection, when the host graph is an intersection graph of the contracts.

Probabilistic infections. A very flexible random tool is described in [12], as the *Independent Cascade Model.* Unlike deterministic infections, in this model a vertex is infected by its neighbors with some probability. For a network G and probabilities assigned to the edges of G, and a set of *active (infected)* vertices A, in each step the vertices infected in the previous round can then independently infect healthy vertices connected to them with the corresponding probability assigned to the connecting edges. The process continues while there are newly infected vertices, and halts otherwise. For a basically equivalent model, see [14].

Generalized cascade. To suit real applications, the model is generalized in a way that the initial infection comes from a distribution (*a priori infection*) which is transformed by this process to an *a posteriori infection.* See different aspects of the above concept in [6, 10, 11]. In this way the a priori knowledge, e.g. the results of standard data mining models, can be integrated into a network based model. Indeed, banks or insurance companies have computed customer risks for a long period, but those models are based solely on individual client data, e.g. age, salary, credit history etc. These models performed satisfactory in a static environment, and they are still useful in predicting the a priori infection (the risk of a sole client).

Fine tuning

In the above mentioned models it is not trivial how to determine the parameters in the transmission rules. We call this an *inverse infection problem*, since our task is to find those possible values of edge infection probabilities that fit well to some given past data. Note that in the deterministic cases there are also (hidden) parameters; like in Bootstrap percolation a lower bound needs to be defined which determines the minimum number of infected neighbors necessary for infection transmission. Nevertheless, this lower bound may highly depend on the characteristics of the host graphs. More generally, one can say that the parameters of deterministic infections are encoded in

the definition of the host graphs. Still, in that case the number of possible different parameters is small, and one can select a nearly optimal solution manually. For the random case, however, the parameter space of candidate edge probabilities is large, and a systematic analysis of the possible cases is not realizable. Therefore a more careful algorithmic methodology to handle this problem is necessary.

Estimation of infection probabilities. The main idea is that the infection probability (edge weight) of an edge *(x, y)* is an unknown, but deterministic function of the attributes of the vertices *x* and *y* and the edge *(x, y)* itself. The above approach can be justified from theoretical and practical viewpoints as well. The estimation of edge weights in general is an underspecified problem, thus it is not correctly realizable even by a theoretical model. On the other hand, in practice it is reasonable to consider real infections as functions of attributes, and to restrict the edge weight function to the measurable variables. Then ad-hoc trial and error method gives fairly good results for the generalized cascade, but it is definitely suboptimal [10].

Learning infection probabilities. For a systematic solution, a standard AI approach can be used. The past data is divided into *learning* and *test sets.* Then one can try to assign edge probabilities in a way that the model results in infection patterns similar to the learning set, while the overall process is evaluated by the test set. As noted before, not the edge probabilities themselves are estimated, but their dependencies on attributes. Mathematically, one has to solve various optimization problems, where the objective functions are known only implicitly. Furthermore, if $x \in \Omega$, then the value of $f(x)$ can be only approximated, since its exact computation is a *#P*-complete problem [9].To find a sub-optimal solution, variants of *grid search* and *gradient methods* were applied, see [6, 8, 10]. Note that the search space Ω is not a scalar space; it also has discrete dimensions that describe the types of transformations acted on the attributes. E.g. in the credit default problem to estimate the edge infection probabilities the most important variables are the "amount of money sent from *x* to *y*" (continuous), and "whether the edge *(x, y)* is between nodes in the same community" (discrete).

Functions. The attributes of the edges/vertices are real (e.g. sent money) or discrete (e.g. membership in a community) types of functions. A possible way to transform edge weights from those parameters is to transform them individually and aggregate the results. Of course, the functions acting on the parameters are unknown a priori. That is, one has to choose some reasonable functions to approximate them. The type of the approximating function itself is a discrete variable, while the constants of those are handled as variables of the associated optimization problem. Some examples: c , $c_0 + c_1 x$, $c_0 + c_1 x + c_2 x^2$, $c_0 + c_1 \log(x)$, $c_0 + c_1 e^x$, and sigmoid functions are classical functions to test. To sum up the overall effect, one can a sum or product with an affine scaling to the $[0, 1]$ interval. That is, the parameter space Ω consists of the mixture of discrete decisions (which function should be used for a given attribute) and the constant parameters in those functions.

Evaluation of estimations. The most natural measures for the error functions are the mean square error (or L^2 norm), the mean absolute deviation (L^∞ norm), or the L^1 norm. However, in data mining another measure of goodness is also important, the so-called *gain curve*. Here the vertices of the graph are monotonically decreasingly ordered by their infection probabilities computed on the training set. Let $w_1, \dots, w_n$ be the values of the same vertices given in the test set. Then the function $gain(x) = \frac{\prod_{i<x} w_i}{\sum_{i=1}^{i=n} w_i}$, and $\int_{x=1}^{n} gain(x)dx$ should be maximized.

Attributes

The next step in modelling is the selection of the appropriate attributes. This needs a thorough study of the particular problem, although general observations can also be taken. Like in data mining in general, it is not necessarily true that more information is better. For example, in the credit default models most individual attributes are useless. Some exceptions are as follows: the age of a company, whether it is municipal or not, and the sector in general. One must emphasize that these data should be gathered and cleaned carefully, and also need to be transformed to include common sense. An example for the latter is the infection model for the credit default. The weight of an edge (x, y) is *not* simply the function of the amount of money sent on that edge, but it should be normalized with the amount of *all* money sent to *y*. We must note that because of the measurement problems, directed edges might cause paradox phenomena. Since a contractor might default months earlier than its procurer, the observed infection is spreading seemingly *backwards* on an edge.

The network topology also gives rise to some attributes; these values are generated by graph mining algorithms. The principal parameters induced by the graph structure are, for example, generated from cluster and community data.

Clusters and Communities. As we indicated in the Glossary we call the elements of a partition of the vertex set *clusters*, while *communities* are dense subsets of nodes that can overlap. Clustering is a basic technique in data mining, and in particular it is important in graph mining. There are a plethora of concepts and algorithms for it, see [4, 19]. So it is somehow surprising that it has little relevance in finding the best edge weights for an infection method [10]. One possible explanation is that some of the static attributes (e.g. the sector, types of activity of a company) already contain similar information about a vertex: e.g. clustering on the graph structure may reflect the classification of the company with respect to their types of activity.

On the other hand, attributes induced by the community structure of a graph have a central role in infection parameter optimization. Communities might be defined in several ways [1, 11, 15], but if an edge (x, y) is identified as being within a community by one algorithm, than it is usually also identified to be within a community in general [11, 15]. The studies show that a community edge is about three times as significant (i.e. has three times more weight) than an edge with similar attributes that is not within a community. This experiment is an important justification of the use of (overlapping) communities against traditional clusters in network based data mining. In fact, this result experimentally confirms Granovetter's classification of the graph edges into weak and strong ones [13].

All these findings stress the importance of extending the search of communities to *dynamic graphs*, that is if the graphs G_1 and G_2 are given, we need to map their communities to each other. The problem is that the communities may change with time, they may grow, shrink, die, born, split, unite or get involved in even more complicated events. Those kinds of problems were investigated in [2, 5, 20] on different networks. These studies mainly concentrated on the survival of super structures, the connection of the size of the communities and their average life span, and the dynamical equilibrium. The latest (unpublished) experiences show the role of these algorithms in infection processes. As one can expect, fine tuning will assign larger weights to those edges that stay within communities during the whole process.

Proposed Solution and Methodology

A typical application consists of a cyclical execution of the following steps. .

Step 1: Preparation. Identify the problem; what questions should be answered, is data available data to answer this question?.

Step 2: Data retrieval. Retrieve data (from warehouses, or by any other means).

Step 3: Data cleaning. Clean data by unifying the format, handling missing data/duplication, and fix errors if possible.

Step 4: Data selection. Use statistical tools to select the significant data, create fields (attributes), keep the important ones and drop the others.

Step 5: Network representation. Build graph(s) from the preprocessed data.

Step 6: Graph analysis. Process the graph(s). Compute the (strong) components, clusters and communities. Create new attributes based on these, and add to the ones gained in Step 4.

Step 7: A priori probability distribution. Compute an a priori probability distribution on the vertices that corresponds to the examined problem.

Step 8: Parameter learning. Divide past data to learning and test sets, run inverse infection algorithms.

Step 9: Quality assessment. compare the results with the observed data by using an appropriate metric. If the results are not satisfactory, identify the possible place of the problem and start over from there

Future Directions

In the application/project part, it is obvious to extend the methodology in predicting other phenomena (spread of success/growth, innovation). As we noted before, an infection model is nothing more than a mechanism that integrates the effects reaching a vertex in a network. That is, the method should be extended to handle more general functions (not only probabilities) with reasonable running time.

A more theoretical, but still an application driven idea is to correlate graphs of different origin with a common vertex set. As an example, for the analysis of the relationships among companies, one can consider a www link graph based on partnerships and the company ownership graph etc. There are intimate connections among these networks; it is shown that the conditional probability of an (unknown) edge between x and y in G_1 increases provided that their neighbors are well connected in a different

network G_2, see [17]. Still, in order to describe these effects in depth, further research is needed.

An interesting crossroad of ecology and economy is the theory of *mutualistic networks*, see [3, 21]. These are bipartite graphs (e.g. plant-pollinator network, the designers-manufacturers in garment industry, world trade etc.), in which the neighbourhoods of vertices are nested into each other. A similar theory for general (and noisy) transaction graphs is to be developed. This would lead not only to more refined attributes, but give rise to a new class of clustering algorithms.

Cross-references

Clustering Algorithms; Communities Discovery and Analysis in Online and Offline Social Networks; Communities in Social Networks, Evolution of; Community Detection, Current and Future Research Trends; Community Evolution; Data Mining; Extracting Social Networks from Data; Fraud Detection Using Social Network Analysis, Case Study; Game Theory in Social Networks; Learning Networks; Modeling of Business Processes and Crisis Management; Models for Community Dynamics; Viral Marketing/Advertising and Social Media

Acknowledgements

This work was partially supported by the European Union and the European Social Fund through project FuturICT.hu (grant no.: TÁMOP-4.2.2.C-11/1/KONV-2012-0013). The second author also would like to thank to the grant OTKA K76099.

References

1. Adamcsek, B. et al. CFinder: Locating cliques and overlapping modules in biological networks. *Bioinformatics* 22, 1021-1023 (2006).
2. Asur, S. et al. An event-based framework for characterizing the evolutionary behavior of interaction graphs. *ACM Transactions on Knowledge Discovery from Data* Volume **3**, Issue 4, November 2009.
3. Bascompte, J.P & Jordano, P. The structure of plant-animal mutualistic networks. In: Ecological networks: linking structure to dynamics in food webs (Eds. Pascal, M & Dunne, J) pp. 147-159, Oxford University Press, 2006.
4. Blondel, V.D. et al. Fast unfolding of community hierarchies in large networks. *J. Stat. Mech.* (2008) P10008
5. Bóta, A. et al. Dynamic Communities and their Detection. *Acta Cybernetica* Volume **20** (2011) 35-52.
6. Bóta, A. et al. Systematic learning of edge probabilities in the Domingos-Richardson model. *Int. J. Complex Systems in Science* Volume **1(2)** (2011) 115-118.
7. Bóta, A. et al. Models for fully mapping the economic ties in Hungary before and during the recent crisis. In *Proceedings of Crisis Aftermath: Economic policy changes in the EU and its Member States* 8th-9th March, 2012.
8. Bóta, A. et al. Approximations of the Generalized Cascade Model. *Acta Cybernetica*, to appear.
9. Chen, W. et al. Scalable Influence Maximization for Prevalent Viral Marketing in Large-Scale Social Networks. *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM (2010) 1029-1038.
10. Csernenszky, A. et al. The use of infections models in accounting and crediting. *Proceedings of the Challenges for Analysis of the Economy, the Business, and Social Progress, International Scientific Conference (2009) 617-623.*
11. Csizmadia, L. et al. Community detection and its use in Real Graphs. *Proceedings of the 13th International Multiconference INFORMATION SOCIETY – IS* 2010, 393-396.
12. Domingos, P. and Richardson, M. Mining Social Networks for Viral Marketing. *IEEE Intelligent Systems*, 20(1), pp. 80-82, 2005.
13. Granovetter, M. The Strength of Weak Ties. *Amer J of Sociology* **78(6)** 1360-1380, 1973.
14. Granovetter, M. Threshold models of collective behavior. *Amer J of Sociology* **83(6)** 1420-1443, 1978.
15. Griechisch, E. et al. Community detection by using the extended modularity. *Acta Cybernetica* Volume **20** (2011) 69-85.
16. Hidalgo, C. A. et al. The Product Space Conditions the Development of Nations. *Science* (2007) 317: 482-487.
17. Horvát E-Á et al. (2012) One Plus One Makes Three (for Social Networks). PLoS ONE 7(4): e34740. doi:10.1371/journal.pone.0034740
18. Jackson, M.: *Social and economic networks*, Princeton University Press (2008).
19. Newman, M.E.J. The structure and function of complex networks. *SIAM Rev*. **45**:167-256 (2003).
20. Palla, G. et al. Social Group Dynamics in Networks. *Adaptive Networks*, chapter 2, 2009.
21. Uzzi, B. The sources and consequences of embeddedness for the economic performance of organizations: The network effect. *Am. Soc. Rev.* **61**:674-698.

Recommended Reading

Albert, R. & Barabási, A. L. Statistical mechanics of complex networks. *Reviews of Modern Physics*, **74** 2002.

Diekmann, O. & Heesterbeek, J.A.P. Mathematical epidemiology of infectious diseases. Model Building, Analysis and Interpretation. *John Wiley & Sons*, 2000.

Fortunato, S., Castellano C Community Detection in graphs. 2009 *Encyclopedia of Complexity and System Science* ed B Meyers (Heidelberg: Springer)

Kempe, D. et al. Maximizing the Spread of Influence through a Social Network. *Proc. 9th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, 2003.

Népusz, T. et al. Fuzzy communities and the concept of bridgeness in complex networks. arXiv:0707.1646v3, 2007.