ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/270271528

Applications of the inverse infection problem on bank transaction networks

ARTICLE *in* CENTRAL EUROPEAN JOURNAL OF OPERATIONS RESEARCH · DECEMBER 2014

Impact Factor: 0.83 · DOI: 10.1007/s10100-014-0375-2

CITATIONS
2

READS
35

6 AUTHORS, INCLUDING:

András Bóta
University of New South Wales
12 PUBLICATIONS **35** CITATIONS
SEE PROFILE

András Csernenszky
Santander UK
10 PUBLICATIONS **24** CITATIONS
SEE PROFILE

Miklós Krész
University of Szeged
36 PUBLICATIONS **145** CITATIONS
SEE PROFILE

András Sándor Pluhár
University of Szeged
37 PUBLICATIONS **166** CITATIONS
SEE PROFILE

Available from: András Sándor Pluhár
Retrieved on: 03 April 2016

Noname manuscript No.
(will be inserted by the editor)

Applications of the Inverse Infection Problem on bank transaction networks

András Bóta · András Csernenszky · Lajos Győrffy · Gyula Kovács · Miklós Krész · András Pluhár

Received: date / Accepted: date

Abstract The Domingos-Richardson model, along with several other infection models, has a wide range of applications in prediction. In most of these, a fundamental problem arises: the edge infection probabilities are not known. To provide a systematic method for the estimation of these probabilities, the

A. Bóta
University of Szeged
Institute of Informatics
P. O. Box 652., 6701 Szeged, Hungary
E-mail: bandras@inf.u-szeged.hu

A. Csernenszky
OTP Bank Plc.
Risk Analyses and Regulation Directorate
Babér u. 7, 1131 Budapest, Hungary
E-mail: csernenszkya@otpbank.hu

L. Győrffy
University of Szeged
Bolyai Institute
Aradi Vértanúk tere 1, 6720 Szeged, Hungary
E-mail: lgyorffy@math.u-szeged.hu

Gy. Kovács
Sixtep Ltd.
Máté u. 54, 6771 Szeged, Hungary
E-mail: gyula.kovacs@sixtep.hu

M. Krész
University of Szeged
Gyula Juhász Faculty of Education
Boldogasszony sgt. 6, 6720 Szeged, Hungary
E-mail: kresz@jgypk.u-szeged.hu

A. Pluhár
University of Szeged
Institute of Informatics
Árpád square 2., 6720 Szeged, Hungary
E-mail: pluhar@inf.u-szeged.hu

authors have published the Generalized Cascade Model as a general infection framework, and a learning-based method for the solution of the Inverse Infection Problem.

In this paper, we will present a case-study of the Inverse Infection Problem. Bankruptcy forecasting, more precisely the prediction of company defaults is an important aspect of banking. We will use our model to predict these bankruptcies that can occur within a three months time frame. The network itself is built from the bank's existing clientele for credit monitoring issues.

We have found that using network models for short term prediction, we get much more accurate results than traditional scorecards can provide. We have also improved existing network models by using inverse infection methods for finding the best edge attribute parameters. This improved model was already implemented in August 2013 to OTP Banks credit monitoring process, and since then it has proven its usefulness.

Keywords Graph theory · Information diffusion · Inverse Infection Problem

1 Introduction

The study of infection processes has roots in epidemics and sociology. In the latter, Granovetter [12] created the Linear Threshold model to describe information diffusion processes in social interactions. The first application of infection models in economics was published by P. Domingos and M. Richardson [10]. In their paper, they proposed the Independent Cascade model (IC) for the purpose of modeling virus marketing, and described the influence maximization problem, that is finding the set of individuals yielding the largest expected infection. Kempe et al. [14,15] proved, that the influence maximization problem was NP-hard, proposed a greedy algorithm for it, and also showed that the generalization of the IC model is in fact an equivalent of the Linear Threshold model. A review on infection models in economics can be found in [17].

The computation of the above models requires the edge infection probabilities of the given network to be known beforehand. This information is usually not known in applications. In their place, intuition-guided estimations are used. Recently, several papers have been published, each aiming to develop a systematic approach to estimate or learn the edge infection probabilities. This is usually done by using additional information, like the previous behavior of the network, or some other property. In [11,16] the infection process is known to some degree, while in [4], the authors assume that the result of the infection process can be observed several times.

To answer the demand for the prediction of edge infection probabilities, the authors have published the Generalized Cascade (GC) model [4] and the Inverse Infection Problem (IIP)[3]. The former describes a probabilistic extension of the Independent Cascade model: Both the initial infectors and the result of the infection are represented as probability distributions: a priori and a posteriori distributions respectively. The infection event itself is a transition between these distributions. The formulation of the Inverse Infection Problem builds upon this; it describes the a priori and a posteriori distributions as inputs to compute the edge infection probabilities. We have given a learning method to compute this process in [3]. This method considers the edge infection probabilities as the result of some unknown polynomial function of available attributes. This way only the coefficients of this function have to be estimated. Starting from some initial coefficients randomly chosen within reasonable bounds, the a posteriori distribution is computed and compared with a reference distribution given as an input of our method. The difference between these distribution is minimized with the help of a Particle Swarm Optimization method.

Some of the ideas of these methods, like representing initial and final infections as distributions and the idea of computing infection probabilities from attributes, came from experience from a previous joint work with the OTP Bank of Hungary on a different project. Existing models did not allow us to go deeper into this problem, therefore we decided to develop our own method. During the development of our method we have used various artificial benchmarks for testing purposes [3], but we have always kept in mind the requirements of a real application. Only after properly developing our method were we able to handle real applications. Recently, when an opportunity presented itself to work with the bank again, we have taken it.

In this paper, we will describe an application of the Inverse Infection Problem on a transaction database of the OTP Bank of Hungary. Our goal is to improve current methods for the prediction of credit defaults. We will first describe the methods published in [4,3,2]. Then we will move on to the case-study itself: we will give a detailed description of the bank transaction database and the goals of our paper. Then we will present the application of our method, and its results.

2 Preliminaries and previous works

For the sake of completeness we will give a short introduction to the Generalized Cascade model and its applications, the Inverse Infection model and a Particle Swarm based estimation of the edge infection probabilities in this section. For further reference see [4,3,2].

2.1 Infection models

The process of infection can be represented as a method, which has two inputs and one output. The first input is the network upon which the process takes place. This network is a weighted one; more precisely the weights on the edges are probabilities. Formally consider a graph $G = (V, E)$, where each $e \in E$ has a weight $w_e, 0 \leq w_e \leq 1$. The second input is the set of initially active vertices A_0, and the output is the set of vertices A_f infected during the process.

The process itself takes place in discrete time-steps: in each iteration, newly infected nodes try to infect their inactive neighbors according to the predefined rules of the specific infection model.

In the Independent Cascade model the nodes of the network have three states: infected, active (newly infected) and susceptible. Other infection models have more states than this, and the transitions between them may be more complex. The transition between the states of the IC model is governed by the following process [10,15]:

Starting from the initially active set of nodes $A_0 \subset V(G)$, let $A_i \subseteq V(G)$ be the set of nodes activated in iteration i. In iteration $i+1$, every node $u \in A_i$ has one chance to activate each of its susceptible out-neighbors $v \in V \setminus \cup_{0 \leq j \leq i} A_j$ according to $w_{u,v}$. If the attempt is successful, then v becomes active in iteration $i+1$. If more than one node is trying to activate v in the same iteration, the attempts are made in an arbitrary order and independently of each other still in iteration $i+1$. Vertices infected in iterations $k < i$ are unable to infect other vertices. The process terminates at step t if $A_t = \emptyset$. It is easy to see, that t exists i.e. the process is finite. Finally, $A_f = \bigcup_{i=0}^{t-1} A_i$ contains the set of vertices infected during the process.

We have chosen the Independent Cascade Model as the basis of our works, because it has proven its effectiveness in modeling infection-like processes in economics [10]. There are many other infection methods, for further reference see [9].

2.2 Generalized Cascade Model

We can generalize this framework in the following way. In the Generalized Cascade (GC) model [2] each vertex is assigned a real value p_v between zero and one, that represents the probability of infection before the beginning of the process. We refer to these values as the *a priori distribution*. Vertices are infected independently from each other before the beginning of the process according to their a priori infection probability p_v. This model is capable of summarizing the effect of the a priori infections and the effect of these infections transmitted through the network. Similarly to the input, the output of the model is given as an *a posteriori distribution*, where values p'_v indicate the probability of being infected during the process for all $v \in V$. The actual way a vertex infects another is the same as in the IC model, although it is possible to use other infection models using the terms of the GC model. It should be noted, that in this application the bank is able to give an accurate estimation of the a priori infection probabilities, but they cannot take into consideration the network effect.

Based on these remarks and formulations, we can define the Generalized Cascade model [2]:

The Generalized Cascade Model: *Given an appropriately weighted graph G and the a priori infection distribution p_v, the model computes the a posteriori distribution p'_v for all $v \in V(G)$.*

The computation of the a posteriori infection probabilities in the IC model, and therefore the GC model, is $\#P$-complete [8]. However, there are several existing heuristics to get approximate solutions [7,6]. In [2] three additional heuristics were proposed for the GC model. We have tested these methods and found, that Edge Simulation gives the best performance on this application. All infection estimations in this paper were computed with this heuristic. For a detailed description please see [2].

2.3 The Inverse Infection Problem

Following the framework described above, an infection model computes the a posteriori infections given a weighted graph and the a priori infections as inputs. In the inverse infection problem the a priori and a posteriori distributions (and an unweighted network) are provided as inputs, and we want to assign edge infection probabilities such that the infection model with the input a priori distribution results in the given a posteriori distribution. Based on this, the Inverse Infection Problem can be defined as in [3]:

Inverse Infection Problem: *Given an unweighted graph G, the a priori and the a posteriori probability distributions p_v and p'_v, compute the edge infection probabilities w_e for all $e \in E(G)$.*

Independently estimating all edge weights of a network is both underdetermined and computationally unfeasible, even if the number of edges is small. Instead, we assume the edge probabilities can be expressed as (normalized) functions of some properties of the edges or nodes that are available in the form of attributes[1]. If there is only one attribute, this function can be expressed as $f(a_1(e))$, where f is the attribute function and $a_1(e)$ represents the attribute of edge e. We have used low-degree polynomials or simply a linear functions like $c_0 + c_1 a_1(e)$, where c_0 and c_1 are unknown coefficients. If there are multiple attributes it is necessary to summarize the effect of them, and even in the case of a single one, normalization is required to get valid probability values between zero and one. Therefore it is necessary to use two functions, the attribute function is applied to the individual attributes on each edge, then the results of these functions are summarized and normalized: $w_e = g(f(a_1(e)), f(a_2(e)), ..., f(a_n(e)))$, where w_e is the edge weight of e, g is the summarizer-normalizer function, f denotes the attribute function and $a_i(e)$ represents the i-th attribute of edge e. The attribute and the normalizer function is the same for all edges, this way we only have to estimate the coefficients of these functions, and since the number of attributes and coefficients is limited, the problem becomes tractable. For more details see section 4 or [3].

We proposed a learning method in [3] based on the inputs of the model: the observed a posteriori distribution can be used as a reference or training set. Then the initial coefficients for the edge attribute functions are chosen from reasonable bounds. Given these attribute functions the coefficients and the a

[1] Such attributes are readily available in banking applications.

priori distribution we can compute an a posteriori distribution corresponding to the chosen coefficients. An error function compares the results with the input a posteriori distribution. The process aims to minimize this error function by repeatedly adjusting the coefficients. This is a typical task for global optimization, i. e. finding the minimum of an unknown multidimensional surface, where the points of this surface can be accurately estimated.

During the initial phase of development we tried various optimization strategies including grid search and several gradient based methods. In the end we have settled on an implementation of the Fully Informed Particle Swarm Optimization method of Kennedy et al. [13] using a von Neumann neighborhood with 16 agents. The process gives satisfying results close to the global minimum, and usually terminates with the number of iterations below 20. A further description of the inverse infection problem and the learning method can be found in [3].

3 Case-study: bank transaction networks

The development of the Inverse Infection Problem was heavily influenced by our previous work with the OTP Bank of Hungary[2] [5]. Recently the bank has approached us again with the task of improving current methods for the estimation of credit default of companies. The bank has information about the properties of the companies and is able to give an estimation of the individual probability of default for them. The goal of this project was to improve this estimation by taking into consideration the effect companies have on each other. If one of the companies goes into default, how does this change the probability of default of other companies, especially the ones with financial ties to the original one? We can reach this goal by considering the network of companies and the connections between them, and computing the edge infection probabilities between the vertices with the optimization method proposed in [3].

We have seen in the previous section the requirements of the Inverse Infection Problem. In this section we are going to discuss the details of this application. We will begin with the construction of the transaction network, that represents the companies and their connections. Then we are going to describe how to create the required probability distributions. Finally we will review the available edge and vertex attributes.

It is important to emphasize, that this is an industrial application, therefore many of the details of this case-study are not available to the public either because of privacy reasons or because it is a part of the know-how of the bank. The transaction network itself cannot be published, because the attached attributes often contain information on the neighboring companies that are not public. We were not allowed to release the transaction network even in an anonymous manner. We were also not allowed to present the exact coefficients

[2] We will refer to the OTP Bank of Hungary simply as bank from now on.

representing the significance of the attributes. Only the results presented in section 4 are public.

3.1 Network construction

In order to deeper understand the behavior of purchaser-supplier connections, a transaction database was created by the bank over a large time period. We can construct a network from this database, where vertices represent corporate clients and edges represent financial connections between clients. Since there are many transactions between the clients, we must decide how to define the edges of the graph: a filtering process is required during the construction of the network. We have used three criteria to decide whether two nodes should be connected or not:

- Frequency of the transactions: the average number of transactions in a month.
- Amount of transactions: the average transacted amount in a month.
- Relative amount: average incoming amount from one company divided by the total income from all of the partners.

For each criterion above, we have assigned the transactions between the companies into three categories[3]: high, medium and low. We have added an edge between the companies if the transactions between them belong to the high category for each criterion, that is two companies are connected if the transactions between them have high frequency and they are also in the high total and relative amount part of the database. The transactions also have a natural direction to them: they go from the purchaser to the supplier, therefore the graph is directed. Since the business partners of a company may change dynamically, we have built the network considering edges from a one year period: from April 2012 to March 2013. The resulting transaction graph has approximately 68.000 vertices, and 106.000 edges.

3.2 Probability distributions

We have selected the time period of our estimations according to the currently used practices of the bank. The task was to make short-time predictions. Which corporate client will be in credit default in the near future? We have created the a priori probabilities considering companies from a three month period starting from January 2013 to March 2013. If the client was in default in this period, it was given an a priori probability of 1, otherwise we have used an estimated probability of default. This estimation was performed by the bank with logistic regression based on a 6 months observation period of its behavioral variables. The a posteriori infection or reference values were constructed from April 2013 to June 2013. Like before, the defaulted companies were given a value of 1.

[3] Uniform 33% of the cases in each category.

3.3 Attribute functions

The following attributes are available in the database. Some of these are not edge attributes, but vertex attributes that are related to suppliers[4], like its age and type.

1. the number of transactions
2. the amount of the transaction
3. total incoming transactions of the client (supplier)
4. community information: if the edge belongs to a community[5]
5. relative traffic, that is the transfer of the edge divided by the sum of all incoming transfers
6. the age of the supplier (how old is the company)
7. unpaid items on the accounts of the supplier
8. limit exceeded (for overdrafts)[6]
9. whether the supplier is a company or a municipality

In the inverse infection problem the edge weights are computed from edge attributes by the means of attribute functions. In this application we had the above introduced nine attributes : a_i, $i = 1, \ldots, 9$. We have tried several simple attribute functions including polynomials, but have found little difference between them both in terms of accuracy and computational time. For the sake of simplicity, we have chosen a weighted, normalized sum as the attribute function. More formally:

$$w'_e = f(a_i(e)) = \sum_i c_i a_i(e), i = 1, ..., 9,$$

then the resulting values are normalized according to

$$\text{norm}(\mathbf{e}) = \frac{\mathbf{e} - \min(\mathbf{e})}{(\max(\mathbf{e}) - \min(\mathbf{e}))}, \tag{1}$$

where **e** stands for the unnormalized vector of edge weights. In this formulation the algorithm computes a weighting of the attributes based on their importance in the infection process.

4 Case-study: Evaluation

In the previous section we have discussed the application of the Inverse Infection Problem on a bank transaction network, now we are going to present its results. As we have mentioned before, our goal was to improve currently used methods by the bank. Therefore we will use two of these methods for comparison. One of them is a simple logistic regression based on a 6 months

[4] A vertex v is a supplier if it is at the end of a directed edge.

[5] Here we used a N^{++} algorithm for community detection, see [1].

[6] The actual outstanding is higher then the given credit-limit.

observation period of the companies' behavioral variables. The other one uses the same network we have used for the IIP model, but instead of estimating the edge weights it uses a uniform constant value for each edge. The focus of this paper is the accuracy of the estimations: how well are we able to predict short time default events. For this purpose we will use well-known performance measurements like ROC evaluation, GINI and RMSE. We will discuss these and the accuracy result in the next section. We will also present two other points of interests in subsequent sections: the speed of the estimations and the matter of initial coefficients. We will close the evaluation with some general observations and remarks.

4.1 Accuracy of estimations

We have tried various measurements to evaluate the performance of our method. The optimization algorithm itself uses the root mean squared error (RMSE) function to guide the search,

$$\sqrt{\frac{1}{|V(G)|}\sum_{v\in V(G)}(\hat{\mathbf{p}}'_v-\mathbf{p}'_v)^2},$$

where $\hat{\mathbf{p}}'_v$ denotes the estimated a posteriori infection of vertex v. However this value is not available for the other models.

The ROC curve was also used to measure performance. We can construct it in the following way: We order the vertices of the graph in a monotone decreasing way by their a posteriori infection values computed by the model. Let $t'_1,\ldots,t'_n$ be the binary values of these vertices given in the reference set, and the function

$$\mathrm{roc}(x)=\frac{\sum_{i\le x}t'_i}{\sum_{i=1}^{n}t'_i}.$$

Now the integral

$$AUC=\int_{x=1}^{n}\mathrm{roc}(x)dx$$

should be maximized. We will also use $GINI=(AUC-0.5)/2$ to measure performance.

We have also compared the average new default events in the reference period (excluding the ones defaulted before) with those TOP segments of the ordering described above, where the model predicted high default probabilities: these are companies, where default is the most probable. In banking the highest concern is to identify the riskiest clients, and the ones with lower risks are far less important. Therefore this approach is the most natural in this application.

As a general observation we have found, that the AUC and GINI measurements were less effective, because the infection models are more powerful if we consider only the high influence values, and less powerful when considering the whole portfolio, where traditional methods are better.

Table 1 Performance of the IIP solution compared to existing methods used by the bank. The first column corresponds to logistic regression, the second stands for the network model with constant weights, the third one for IIP.

$\frac{\text{TOP \% default rate}}{\text{Average default rate}}$	Regression	Constant	IIP
TOP 1%	7.27	0.79	7.82
TOP 3%	8.11	2.17	8.77
TOP 5%	7.87	2.89	7.97
TOP 10%	4.97	3.16	4.99
Other measurements			
AUC	72.4 %	65.39 %	72.5 %
AUC (lower bound)	69.9 %	62.97 %	70 %
AUC (upper bound)	74.9 %	67.81 %	75 %
GINI	11.2 %	7.69 %	11.24 %
RMSE			0.1661

In Table 1, we can see the results of the benchmark methods and the IIP based estimation. The measurements TOP 1-2-5-10%, AUC, GINI and RMSE can be seen. For example the TOP measurements, 7.77 in the first row means, that in the highest one percentage the default rate is 7.77 times higher than the average default rate.

We can see in the table that using IIP is far better than the network model with constant edges. Compared to the regression, IIP did not improve the ROC based measurements, but it was able to identify the most risky clients with better precision than the regression. If we take a look at the 1-5% riskiest companies, we can see that our model is able to improve prediction by as much as 10%. Therefore using IIP estimations for short time predictions clearly means an advantage.

4.2 Speed of the estimations

We have measured the running time of the algorithm using different number of attributes. We have ordered the attributes by importance, and we have assigned the top 9, 8, 6, 4, 2, 1 most important ones for different runs. Here we found that the computational time of the algorithm can be reduced by decreasing the number of attributes. The measured squared error remains roughly the same during different runs. In Table 2 we can see the running times and error for different attribute configurations.

4.3 Initial coefficients

The optimization algorithm starts from initially random coefficients, and the bounds of these initial coefficients may affect the performance of our method. We have tried several lower and upper bounds for the coefficients and found,

Table 2 Running time and error compared to the number of used attributes

No attributes	Running time (sec)	RMSE
9	166	0,1661
8	152	0,1661
6	140	0,1661
4	123	0,1662
2	107	0,1668
1	102	0,2324

Table 3 Function bounds

Coefficient bounds	No. iterations	Running time (sec)	RMSE
(-100, 100)	10	323	0.1663
(-10, 10)	9	288	0.1672
(-1, 1)	7	235	0.1674

that the selection only effects the number of iterations the optimization algorithm terminates in, while the measured error at the end of each run remains roughly similar. Results can be seen on Table 3. In subsequent runs we have used the initial bounds $[-1, 1]$ for each coefficient.

4.4 Additional observations

We have experimented with omitting the direction of the edges in the network, because our previous studies [5] showed that Basel II default spreads in both directions. This seemingly paradoxical phenomenon comes from the different stability of companies. When a big buyer has cash flow problems, it delays the transfers causing cash flow problems for its supplier, who might go bankrupt. Still the observer, who has no detailed information on the whole situation, sees only that the suppliers bankruptcy spread to the buyer. In this model however, omitting direction decreases performance. The direction of an edge goes from the purchaser to the supplier, implying that the suppliers depend on the purchasers, therefore the bank should estimate the vulnerability of the supplier.

We have also experimented with increasing the number of edges by reducing the criteria for relevant transactions, again this resulted in decreased performance. A company can have many business partners, filtering out the more relevant ones, where the mutual dependency is higher is very important.

The most significant attributes were community information and relative traffic.

5 Conclusions

We have presented a case-study of the recently published Inverse Infection Problem on a bank transaction network. The goal of this estimation was to improve the efficiency of existing models for the prediction of short-time credit default events. Since the creation of IIP was heavily influenced by banking applications, it was quite suited to handle this task. In this paper we gave an overview to the Inverse Infection Problem and its preliminary, the GC model, and we also presented an optimization method to estimate the edge infection probabilities in these models. These models were previously published in [3, 4, 2]. Then we presented the circumstances of the case-study: the construction of the transaction network, the a priori and a posteriori infection probabilities and the available edge and vertex attributes.

Concerning our results, Inverse Infection estimations allowed us to find new ways to improve the efficiency of existing models for the prediction of short-time credit default events. The new model has better predictive power than traditional methods: our model can identify the companies where default is most probable better than the previously used models of the bank. We were able to predict the bankruptcies of the riskiest 5% clients 10% more accurately than the regression model. Our model was implemented in August 2013 into the OTP Bank of Hungary's credit monitoring process.

Acknowledgements The first and last authors were 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 fifth author was partially supported by the European Union and co-funded by the European Social Fund through project HPC (grant no.: TÁMOP-4.2.2.C-11/1/KONV-2012-0010)

References

1. A. Bóta, L. Csizmadia and A. Pluhár, Community detection and its use in Real Graphs. *Proceedings of the 2010 Mini-Conference on Applied Theoretical Computer Science - MATCOS 10* (2010), 95–99.
2. A. Bóta, M. Krész and A. Pluhár, Approximations of the Generalized Cascade Model. *Acta Cybernetica* **21** (2013) 37–51.
3. A. Bóta, M. Krész and A. Pluhár, The inverse infection problem. *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*, Warsaw (2014).
4. A. Bóta, M. Krész and A. Pluhár, Systematic learning of edge probabilities in the Domingos-Richardson model. *Int. J. Complex Systems in Science*, **1(2)** (2011) 115–118.
5. A. Csernenszky, Gy. Kovács, M. Krész, A. Pluhár, T. Tóth, The use of infection models in accounting and crediting. *Challenges for Analysis of the Economy, the Businesses, and Social Progress*, Szeged (2009), 617–623.
6. T. Cao, X. Wu, T. X. Hu, S. Wang, Active Learning of Model Parameters for Influence Maximization. *Machine Learning and Knowledge Discovery in Databases*, Lecture Notes in Computer Science, Springer Berlin/Heidelberg, (2011) 280–295.
7. W. Chen, Y. Yuan, L. Zhang, Scalable Influence Maximization in Social Networks under the Linear Threshold Model. *Proceedings of the ICDM '10 Proceedings of the 2010 IEEE International Conference on Data Mining*, IEEE Computer Society (2010) 88–97.
8. W. Chen, C. Wang, Y. Wang, 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.
9. O. Diekmann, J. A. P. Heesterbeek, Mathematical epidemiology of infectious diseases. Model Building, Analysis and Interpretation. *John Wiley & Sons,* 2000.
10. P. Domingos, M. Richardson, Mining the Network Value of Costumers. *Proceedings of the 7th International Conference on Knowledge Discovery and Data Mining*, ACM (2001) 57–66.

11. A. Goyal, F. Bonchi, L.V.S. Lakshmanan, Learning influence probabilities in social networks. *Proceedings of the third ACM International Conference on Web search and data mining.* ACM (2010) 241–250.
12. M. Granovetter, Threshold models of collective behavior. *American Journal of Sociology* **83**(6) (1978) 1420–1443.
13. J. Kennedy, R. Mendes, Neighborhood topologies in fully informed and best-of-neighborhood particle swarms. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews.* **36** (4) (2006) 515–519.
14. D. Kempe, J. Kleinberg, E. Tardos, Maximizing the Spread of Influence though a Social Network. *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM (2003) 137–146.
15. D. Kempe, J. Kleinberg, E. Tardos, Influential Nodes in a Diffusion Model for Social Networks. *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP)*, Springer-Verlag (2005) 1127–1138.
16. M. Kimura, K. Saito, Tractable models for information diffusion in social networks. *Knowledge Discovery in Databases*, Lecture Notes in Computer Science Springer Berlin / Heidelberg, (2006), 259–271.
17. M. Krész, A. Pluhár, Prediction of Economic and Social Events by Infection Processes. *Encyclopedia of Social Network Analysis and Mining*, Springer (2013).