IJCSS International Journal of Complex Systems in Science

Int. J. Complex Systems in Science
vol. **1(2)** (2011), pp. 115–118

Systematic learning of edge probabilities in the Domingos-Richardson model

András Bóta[1,†], Miklós Krész[2] and András Pluhár[1]

[1] *Department of Computer Science, University of Szeged*

[2] *Gyula Juhász Faculty of Education, University of Szeged*

Abstract. In this paper we propose a method for estimating the edge infection probabilities in a generalized Domingos-Richardson model. The probabilities are considered as unknown functions of a priori known edge attributes. To handle this inverse infection problem, we divide the past data to learning and test sets. Then we try to assign edge probabilities such that the model results in infection patterns similar to the learning set, while we evaluate the overall process by the test set. Usually not the edge probabilities themselves are estimated, but their dependences on other available information, such as the previous behaviors of nodes, like in [5]. In our case these are vertex or edge attributes. Mathematically we face with various optimization problems, where the objective functions are known only implicitly. We study different measures of goodness, and develop algorithms for the optimization and investigate the possible best estimations given the boundary conditions.

Keywords: Domingos-Richardson model, infection model, data mining
MSC 2000: 05C82, 91D30, 68U20.

† **Corresponding author:** bandras@inf.u-szeged.hu
Received: October 14, 2011
Published: October 24, 2011

1. Background

Studying virus marketing, Domingos and Richardson [1] introduced the so-called Independent Cascade (or IC) model. In fact, Kleinberg, Kempe and Tardos proved in [2] that a certain form of this model is equivalent to Granovetter's Linear Threshold model [3]. In the IC model, we have a directed graph $G = (V, E)$ and for each $e \in E$, a probability $0 < w_e < 1$ is given along with a small set of initial adopters W_0. The members of W_0 are trying to infect neighboring vertices in stages resulting in the set of total infected vertices W. This model has been successfully extended and applied to many real life examples [4].

However, such a model requires the probabilities of infection assigned to the edges to be known *a priori.* This information is neither provided nor known in most applications, i.e. the probabilities must be estimated by using partial data. These estimations are usually done with some intuition-guided *trial and error,* using other vertex or edge functions of the network. While this crude approach improves the earlier models for estimating credit default or churn, it is definitely sub-optimal, so more systematic methods are needed.

To handle this inverse infection problem, we take a generalized approach to the original IC model. Instead of active or inactive nodes, for each $v \in V$ we define a probability p_v, creating a probability distribution indicating the chance of infection. This distribution can be viewed as an *uninduced* infection, that has nothing to do with the network. In the process, due to the infections, this *a priori* distribution changes to an *a posteriori* distribution.

We assume that the *a priori* distribution is known as the input of the method, and we aim to estimate the edge probabilities resulting in an *a posteriori* distribution close to the one computed from the learning set. To express the edge probabilities we use the attributes $a_0, \ldots, a_n$ assigned to the edges or nodes. The infection probabilities will be considered as $w_e = g(f_0(a_0), \ldots, f_n(a_n))$, where g and the f_i's are some naturally selected functions. Our goal is to estimate the parameters of functions $g, f_0, \ldots, f_n$.

In order to solve the inverse infection problem, we have used grid search and various gradient based approaches. For the purpose of measuring the error, we have used the mean squared error.

2. Methods

In this section we will describe the methods, error measurements and test data we have used.

For the parameter estimation, we have tried grid search and several gradient based searches. For the latter, we have implemented both single and multi

n	m	IC it	error	agents	parameters	runtime
34	79	10000	0.000123	8	7	651
34	79	10000	0.0000134	8	6	400
190	1216	10000	0.0000277	8	6	3932
190	1216	1000	0.000103	8	8	888
1000	2377	1000	0.000122	4	3	950

Table 1: Results for the gradient search

agent variations. The gradient itself is estimated numerically.

To measure the precision of this estimation, we have used the mean square error: $\frac{1}{|V|}\sum_{v\in V}(\hat{p}_v - p_v)^2$, where $\hat{p_v}$ denotes the *a posteriori* infection probability of the vertices and p_v denotes the *a priori* infection probability.

We have tested various benchmark graphs with different sizes and density, mostly coming from the social sciences, including Zachary's famous network. The attributes of the edges as well as the function parameters were drawn randomly from an uniform distribution.

3. Results

Tables 1 and 2 show the test results for three different test graphs with differing numbers of parameters. Here n is the number of nodes, while m is the number of edges. *IC it* denotes the number of iterations used to compute the *a posteriori* distribution. *Error* denotes the mean squared error, *parameters* the number of parameters to estimate and *runtime* is the running time of the method in seconds[1]. *Agents* denotes the number of agents in the multi agent search, and the grid search *distribution* and *iteration number* are also included in table 2.

As can be expected, grid search is considerably slower, and does not scale very well with the number of parameters. Acceptable running time can only be reached by lowering the number of IC iterations and grid size, but this results in decreased precision.

The gradient based approach is reasonably fast, and is able to support higher IC iteration numbers resulting in increased precision. Adding additional agents can improve the precision considerably.

Note, that due to the number of attributes and functions used to compute the edge probabilities, the error surface (objective function) of this problem is quite complicated. Since it can have a large number of local optima, finding

[1]We have used an Intel i7 2630QM processor and 8 GB-s of memory.

n	m	IC it	error	grid dist	grid it	parameters	runtime
34	79	100	0.000453	5	5	7	697
34	79	1000	0.0000196	5	5	6	999
190	1216	1000	0.000166	4	3	6	2182
190	1216	100	0.000725	3	3	8	412
1000	2377	1000	0.000110	8	5	3	1312

Table 2: Results for the grid search

the global optimum turned out to be a very difficult problem. It should also be noted that the quality of the local optima can be very good, meaning the difference from the global optimum is small.

Acknowledgements

The authors were supported by the Project named "TÁMOP-4.2.1/B-09/1/ KONV-2010-0005 - Creating the Center of Excellence at the University of Szeged" also supported by the European Union and co-financed by the European Regional Fund.

The second author was partially supported by HSC-DAAD Hungarian-German Research Exchange Program (project P-MÖB/837) and Gyula Juhász Faculty of Education, University of Szeged (project CS-001/2010).

The third author was partially supported by the grant OTKA K76099.

References

[1] P. Domingos and M. Richardson, *Proc. 7th Intl. Conf. on Knowledge Discovery and Data Mining*, 57–66 (2001).

[2] D. Kempe, J. Kleinberg, and É. Tardos, *Proc. 9th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, (2003).

[3] M. Granovetter, *American J. of Sociology* **83**, 1420-â1443 (1978).

[4] A. Csernenszky, Gy. Kovács, M. Krész, A. Pluhár and T. Tóth, *Proc. of the Challenges for Analysis of the Economy, the Business, and Social Progress*, (2009).

[5] K. Saito, R. Nakano and M. Kimura, *Knowledge-Based Intelligent Information*, Part III, LNAI 5179, pp. 67–75, 2008.