UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

BACHELOR THESIS No. 3743

Information propagation in online social networks

Iva Miholić

Zagreb, June 9, 2014

Contents

List of Figures

List of Tables

1. Introduction

A *network* is a set of items with connections between them. The Internet, the World Wide Web, social networks like genealogical trees, networks of friends or co-workers, biological networks like epidemiological networks, networks of citations between papers, distribution systems like postal delivery routes: they all take a form of networks. Most social, biological and technological networks have specific structural properties. Such networks are referred to as *complex networks*. An example of a complex network is represented on the Figure 1.1.

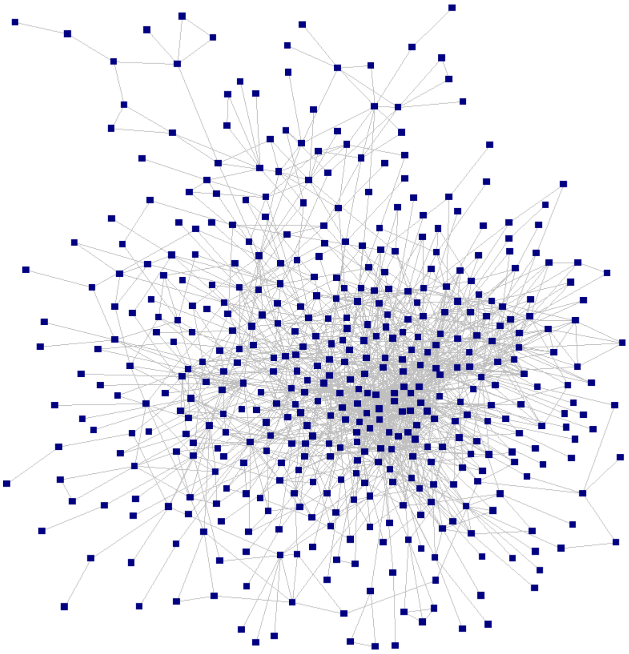


Figure 1.1: A network graph of Paul Erdõs and his collaborators, courtesy of [7]. The nodes represent mathematicians and the edges represent the relationship "wrote a paper with".

A network structure or a topology can be mathematically modeled as a graph with set of vertices (or nodes) representing the items of the network. The network structure can then be analyzed using graph theory. An edge between two nodes represents a connection between the two corresponding items. Edges can be directed or undirected, depending on the nature of the connection. To better mimic the real-world (complex) network structure, it is common to add attributes to nodes and/or edges or to have both directed and undirected edges on the same graph.

For large-scaled complex networks that have millions or billions of vertices, the study in the form of traditional graph theory is not sufficient or sometimes possible. When this is the case, the statistical methods for quantifying large complex networks are used.

The ultimate goal of the study of complex network structure is to understand and explain the workings of systems built upon network such as spreading of diseases or information propagation in social networks.

After statistical properties analysis, the model of a complex network is created. The model can help us understand the meaning of statistical properties - how they came to be as they are and how they relate to the behaviour of a networked system. Based on the statistical properties and using the right model, the behavior of networked systems can be predicted.

The basis of the complex network theory; the structure analysis and the process modeling can be found in [9].

2. Online social networks and information propagation

A complex network of profile pages and connections between them created by users of social networking sites such as Facebook and Twitter is refered to as *online social network*. Through these networks, users can communicate and share information. Since online social networks allow hundreds of millions of Internet users worldwide to produce and consume content, the structure of these social networks represents a huge amount of data that can be used to extract important information about the systems built upon the network.

An online social network's structure is also formally represented by a graph where nodes are users and edges are relationships between them. Direction of edges depends on the social networking site's social model. For example, Facebook's social model of *friendship* connects in bilateral and Twitter's model of *following* connects in unilateral manner.

When there is an edge between two nodes, users corresponding to that nodes may exchange messages. One user can send messages to all of its connections at once or he can send a message to one user at once, depending on the social networking site's social model. Users publish messages to share or forward various kind of information.

2.1. Information propagation

Information item is a news item or an idea that propagates amongst nodes contained in the body of messages. We refer to the nodes that have adopted a particular item as *active* and the nodes that have not adopted the same item as *inactive*. The activated nodes never deactivate. It is important to define when exactly the node became active, *i.e.* has adopted the information item.

When talking about information adoption for a particular node, some authors [8] distinguish between *exposure event* and *infection event*. *Exposure event* occurs when a node gets exposed to the information item, in other words, when the user reads a received message containing the information item. On the other hand, *infection event* occurs when a message is sent. Every exposure may lead to infection event.

With the given network structure and a stream of messages with corresponding content, timestamp, sender and receivers that are sent on the network, the times of all infection events are known. In other words, from such data we can only know exposure has happened sometime before the recorded infection event. Exposures that didn't lead to infection remain unknown. Additionally, one node can be exposed multiple times before the recorded infection. The frequency of those exposures is unknown. For simplicity, we will consider a node as active when it gets infected.

2.2. Social influence and information diffusion

The term *diffusion* in natural sciences refers to the net movement of a substance from a region of high concentration to a region of low concentration. When a message containing some information item is sent, receivers may or may not choose to send a message containing the item to their other connections.

The action of adopting and resharing some particular information item is a decision based on the action of the previous message sender, therefore, individuals are influenced by the actions taken by others. This social phenomenon when actions of a user can induce his connections to behave in a similar way is known as *social influence*.

The stream of messages that carries the same information item among the nodes of the social network driven by *social influence* causes the information to be adopted in a similar way as the molecules of dye diffuse in water. The described spread of information caused by social influence is therefore known as *information diffusion*.

2.3. External influence

When a user adopts particular information item, it is important to ask where did the information item initially came from. If we assume a closed-world model with initially active set of nodes, we can categorize all information adoptions as social influence driven. However, the online social network is not a closed world and exposure to an information item can originate from some other sources.

Due to emergence of mass media, especially online news sites, the information can reach nodes of online social network driven by the influence of exogenous outof-network sources, as well as by the social influence. Depending on the strength of exogenous influence, the process of adopting information on the social network no longer resembles diffusion.

With greater external influence, more and more users whose connections didn't yet adopt the observed information become exposed and infected. This particular kind of adoption can only be explained by the influence of some unobserved exogenous source. However, if it is known the exogenous source exists and the node has neighbours that have already adopted the observed information item, we are not sure whether the adoption of information for that node is due to exogenous or social influence.

The focus of this thesis is to present a model of information propagation with both internal (social) and external influence developed in [2] and use it to classify whether the adoption of a particular item of information for the given network is mainly internal or external influence driven.

3. Referendum Dataset

On 1^{st} of December 2013 referendum on the question "Do you support constitutional amendment about marriage being community of man and woman" was held in Croatia. A week before, starting from 25th of November, Facebook application referendum2013.hr [10] became active. Once the user registered, he/she could answer the same referendum question and see how his/hers Facebook friends have answered.

With users consent, using the application the demographics data, timestamps of registration and vote casting and the votes of 12763 registered users were collected, as well as the network of their Facebook friends. The complete network structure of both registered users and their friends contains roughly 1.695 million nodes with 4.461 million edges.

The degree distribution for the complete network of registered users only fits the function $x^{-\alpha}$ known as the power-law curve which is a common structural property of complex networks [9]. The distribution of degrees is shown on the Figure 3.1.

The majority of users (72.6 %) who have casted their vote (11606 registered users) voted against the support to the definition of marriage as stated in the referendum question. Since that result differs greatly from the official referendum result where 33.5 % of 1.446 million voters voted 'against' the definition of marriage as stated in the referendum question, motivation for performing data analysis and developing models for information propagation arose.

In this case, information propagation refers to information on the existence of the application that spread among the network of users via link sharing through Facebook's social model of *sharing* or directly. The timestamps of registrations indicate the times of adoption of that information item but how the adoption has happened remains unknown.

Along with the network of users and their Facebook friends, major online media coverage was observed and documented in the form of articles along with its timestamps of publishing. The majority of peaks in voting activity dynamics align with the 11 documented article timestamps, but the role of media in engaging the voters has to

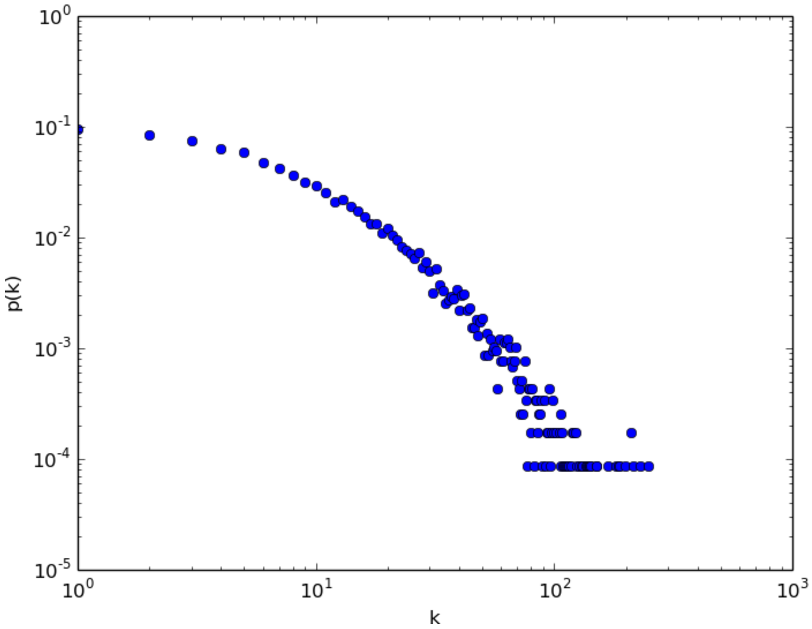


Figure 3.1: The degree distribution for the network of registered users. $p(k)$ is the probability that a vertex chosen uniformly at random has degree k.

be confirmed with a model.

3.1. Datasets

Referendum Dataset represents two network structures we will refer to as *Restricted* and *Complete Referendum Dataset*. The *Restricted Referendum Dataset* consists of two subgraphs. The first subgraph has 12763 nodes representing the registered users. If two users are Facebook friends, there exists a bilateral connection between the two corresponding nodes. Every node also carries the registration timestamp as a node attribute.

The nodes of the second subgraph correspond to 11 recorded mass media articles. These nodes also carry the information about the timestamp of activation as a node attribute (the node is activated when the article got published). Every node in second subgraph has a bilateral edge with every other node in the subgraph. The two subgraphs are connected unilaterally: every node representing the media (from the second subgraph) has edges directed to every node representing a user (but not vice versa).

By extending the first subgraph with nodes representing the Facebook friends of registered users that did not register, network structure we will refer to as *Complete Referendum Dataset* is obtained.

The Complete Referendum Dataset consists of 1695421 nodes representing the users and their friends with 4461556 bilateral edges connecting them. The second subgraph that represents the media is exactly the same as in the *Restricted Referendum Dataset*. Additionally, the set of unilateral edges connecting the nodes of the second subgraph to the nodes of the first subgraph is extended by the set of unilateral edges connecting every node representing the media to every node in the first subgraph that represents an unregistered friend of registered user(s).

The anonymized network of user nodes with corresponding timestamps of activation is stored in GML format [1] while the media nodes and other mentioned connections are defined implicitly.

4. Methods

4.1. Principle of maximal likelihood

A *statistical model* $S_X(\theta)$ is a family of parametrized probability distributions defined for a set of real-valued parameters represented by vector θ and continuous or discrete random variable X. The distribution can be described as a probability mass function $F_X(x;\theta) = P(X < x)$, a probability density function $f_X(x;\theta) = \frac{d}{dx}F_X(x;\theta)$ or discrete probability $P(X = x;\theta)$. With given training set and a statistical model, the principle of maximal likelihood is used to reduce the problem of choosing the distribution from $S_X(\theta)$ for the given training set, what is known as *model fitting*, to the maximization or a minimization problem.

Random sample is a training set of m observations, examples or outcomes x_1, x_2, $\ldots, x_m$ drawn from the fixed but unknown distribution in $S_X(\theta)$. Suppose that we have a random sample drawn. If we assume the examples are independant, the probability of a random sample is the product of probabilities for the individual examples with some fixed parameters θ:

$$f(x_1, x_2, \ldots, x_m; \theta) = \prod_{j=1}^{m} f_X(x_j; \theta). \tag{4.1}$$

The function $L(\theta; x_1, x_2, \ldots, x_m) = f(x_1, x_2, \ldots, x_m; \theta)$ defined like (4.1) but considered as the function of θ depending on m fixed parameters $x_1, x_2, \ldots, x_m$ is a *likelihood function*. The actual value of the likelihood function bears no meaning. It is used to compare how different members of the statistical model fit the training data.

The principle of maximal likelihood says that for the given random sample (training data or a dataset), we should use as a model the distribution from $S_X(\theta)$ that gives the greatest possible probability to the training data. Since the vector $\hat{\theta}$ that maximizes the probability of the traning data also maximizes the likelihood function L, it is obtained using the following equation:

$$\hat{\theta} = \operatorname*{argmax}_{\theta} L(\theta; x_1, x_2, \ldots, x_m). \tag{4.2}$$

The vector $\hat{\theta}$ is refered to as *the maximum likelihood estimator (MLE)* of parameters θ.

The log likelihood function $l(\theta; x_1, x_2, \ldots, x_m)$ is the logarithm of the likelihood function $L(\theta)$. Since logarithm is a monotonic strictly increasing function, maximizing the log likelihood over all possible values of vector θ is equivalent to the maximization problem stated in (4.2). Additionally, minimization of negative log likelihood is also equivalent to (4.2).

When we want to determine $\hat{\theta}$ for the given dataset, we will choose between solving the problem of maximizing the likelihood, maximizing log likelihood or minimizing negative log likelihood.

4.1.1. Conditional likelihood

For two discrete random variables X and Y, conditional probability is formally defined as

$$p(y|x) = P(Y = y|X = x) = \frac{P(X = y, X = x)}{P(X = x)}. \tag{4.3}$$

We can say that (4.3) denotes the probability of a value $y \in Y$ when the value of X is known. If the variables are dependant, Y follows a probability distribution that is different for different values of X. We parametrize all these probability functions with the same vector of parameters θ.

The conditional likelihood of θ given data $x \in X$ and $y \in Y$ is

$$L(\theta; y|x) = P(Y = y|X = x; \theta). \tag{4.4}$$

For finding the conditional likelihood estimator, the random sample consists of m pairs $\{(x^{(i)}, y^{(i)})|1 \leq i \leq m\}$. If all $y^{(i)}$ are independent when each is conditioned on its own $x^{(i)}$, we can define probability of the random sample as a product of probabilities for all pairs $\{(x^{(i)}, y^{(i)})|1 \leq i \leq m\}$ from the random sample, similar to (4.1):

$$L(\theta; y|x_1, x_2, \ldots, x_m) = \prod_{i=1}^{m} P(Y = y|X = x_i; \theta). \tag{4.5}$$

After determining $\hat{\theta}$, we can use it to compute probabilities for the alternative values $y \in Y$ given any specific value of x. The procedure with this feature is called a *classifier*.

4.2. Gradient descent algorithm

The gradient descent algorithm can be used as an iterative optimization method for solving minimization problems that cannot be solved directly.

Let's say we want to find the values of parameter vector θ of n parameters, $\hat{\theta} \in \mathbf{R}^n$, that minimizes function $f(\theta)$. Given the *gradient vector* of n partial derivates $\frac{\partial}{\partial\theta_j} f(\theta)$ with respect to each parameter θ_j, $1 \leq j \leq n$ from θ, the *gradient descent* starts from random θ, and at each step simultaneously updates the values θ_j for all $1 \leq j \leq n$ in the opposite direction of the gradient:

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial\theta_j} f(\theta). \tag{4.6}$$

The coefficient α refers to the *stepsize* or a *learning factor*. It determines how much to move in the direction given by the negative value of the derivative.

When the procedure finds the vector θ that minimizes $f(\theta)$, partial derivatives will evaluate to 0 and the procedure will terminate. Thus, procedure will find the local minimum but there is no guarantee of finding the global minimum if the function $f(\theta)$ is not unimodal. The use for the good value of α is also critical: the algorithm will converge slowly with small α and a large value of α may cause the algorithm not to find the nearest local optimum or it may even cause divergence.

Pseudocode for the described algorithm is given below as Algorithm 1.

Algorithm 1 The Gradient Descent Algorithm

```
for $j = 0$ to $n$ do
    $\theta_j \leftarrow rand(0, 1)$
end for
repeat
    $\theta' \leftarrow \theta$
    for $j = 1$ to $n$ do
        $\hat{\theta}'_j \leftarrow \hat{\theta}'_j - \frac{\partial}{\partial\theta_j} f(\theta)$
    end for
    $\theta \leftarrow \theta'$
until convergence
return $\theta$
```

It is best to initialize θ with random values close to 0. If the initial values of θ are large in magnitude, the procedure may also not find the nearest local optimum.

4.2.1. BFGS

Broyden-Fletcher-Goldfarb-Shanno algorithm (BFGS) is an example of advanced iterative method for solving unconstrained nonlinear optimization problems.

Similarly to the gradient descent algorithm, this method uses both first and second order derivatives of the function for the update. Since BFGS has been proven to have good performance even for non-smooth optimizations, it will be used for optimizations in this paper.

During the optimization, the first order derivative has to be evaluated directly, unlike second order derivatives in the form of Hessian matrix. The Hessian matrix can be approximated using the gradient evaluations. BFGS also finds acceptable stepsize α for every iteration.

4.3. Logistic regression

Suppose Y is a binary (Bernoulli) outcome and that $x \in X$ is a real-valued vector with n values $x_1, x_2, \ldots, x_n$. We model the probability that $Y = 1$ as a nonlinear function of a linear function of $x \in \mathbf{R}^n$ with unknown real-valued vector of parameters $\theta \in \mathbf{R}^{n+1}$ using logistic regression.

Logistic regression is a model with conditional probability functions defined as:

$$P(Y = 1|X = x; \theta) = \sigma(\theta_0 + \sum_{j=1}^{n} \theta_j x_j) = \frac{1}{1 + \exp(-\theta_0 - \sum_{j=1}^{n} \theta_j x_j)} \tag{4.7}$$

with $x \in \mathbf{R}^n$ and $\theta \in \mathbf{R}^{n+1}$.

The function $\sigma(z) = \frac{1}{1+e^{-z}}$ is nonlinear always-increasing function and it is commonly known as *the sigmoid function*. The graph of the sigmoid function is shown on the Figure 4.1.

The logistic regression model is easier to understand in the form of the log odd:

$$\log \frac{p}{1-p} = \theta_0 + \sum_{j=1}^{n} \theta_j x_j \tag{4.8}$$

where p is an abbreviation for $P(Y = 1|X = x; \theta)$.

The odds of event Y = 1 given X = x is exactly the ratio $\frac{p}{1-p}$. Since probabilities range from 0 to 1, odds range from 0 to ∞ and log odds, $\log \frac{p}{1-p}$, range from $-\infty$ to ∞. A linear expression on the right hand side of (4.8) can also take unbounded values so it is reasonable to use a linear expression as a model for log odds.

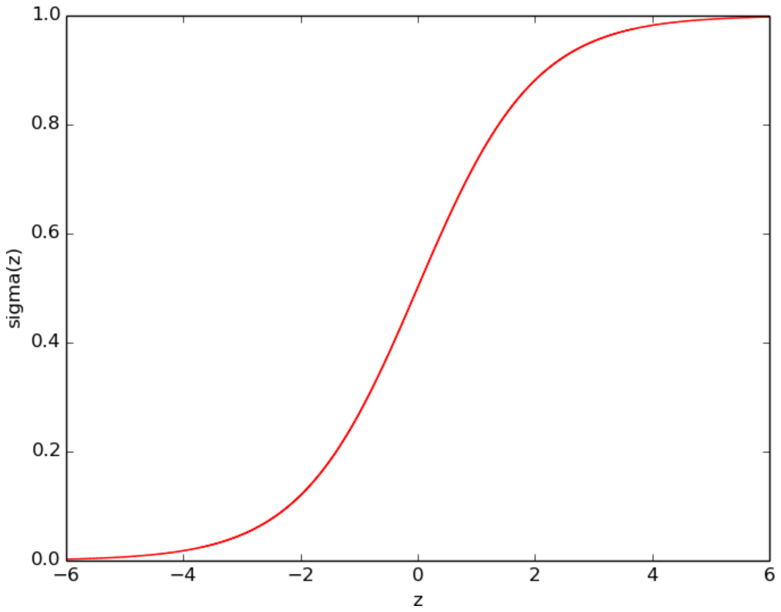


Figure 4.1: Function $\sigma(z) = \frac{1}{1+e^{-z}}$.

Logistic regression is the simplest reasonable model for a random binary outcome whose probability depends linearly on n features represented by the feature vector $x \in \mathbf{R}^n$. The limitation of the basic logistic regression model is that the probability must either increase monotonically or decrease monotonically as a function of each feature $x_j \in x, 1 \leq j \leq n$. To extend the model for polynomial dependencies, feature vector can be extended with higher-order polynomial factors of the original features.

4.3.1. Logistic regression classifier

For the given vector of features $x \in \mathbf{R}^n$ drawn from the distribution of a multidimensional random variable X, we want to predict a binary response from the binary predictor Y dependent on X. The dependency (the conditional probability) is modeled as a logistic regression. To learn a logistic regression classifier we must choose parameter values $\hat{\theta}$ that best fit the training set data. Once training is complete and $\hat{\theta}$ is obtained, we use thus defined conditional probability to make further predictions: the value of Y is predicted as 1 if the probability of $P(Y = 1|X = x) \geq 0.5$ for the input feature vector x, otherwise it is predicted as 0.

Given the training set of m training example pairs $\{(x^{(i)}, y^{(i)}) \mid 1 \leq i \leq m\}$, the logistic regression classifier is learned by minimizing the negative conditional log likelihood.

Taking the logarithm of (4.5) and negating gives the expression for the negative conditional log likelihood, $NCL(\theta)$:

$$NCL(\theta) = \sum_{i=1}^{m} -\log L(\theta; y_i|x_i) = \sum_{i=1}^{m} -\log p(y_i|x_i;\theta) \tag{4.9}$$

If we denote $P(Y = 1|X = x_i;\theta)$ as p_i, the probability of the training example pair $(x^{(i)}, 1)$ is exactly p_i, while the probability of the example $(x^{(i)}, 0)$ is $1 - p_i$. Now we can simplify the above equation as

$$NCL(\theta) = \sum_{i;y=1} -\log p_i + \sum_{i;y=0} -\log(1 - p_i) \tag{4.10}$$

The problem of minimizing $NCL(\theta)$ can now be defined similarly to 4.2:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}}(-\sum_{i;y=1} \log p_i - \sum_{i;y=0} \log(1 - p_i)) \tag{4.11}$$

Because of the nonlinearity of the sigmoid function, the minimization problem stated in (4.11) cannot be solved directly, therefore optimization techniques are used.

Derivatives of the $NCL(\theta)$ with respect to every parameter in θ have to be determened in order to use previously described optimization techniques.

The partial derivative of NCL (4.9) with respect to parameter θ_j is

$$\frac{\partial}{\partial\theta_j} NCL(\theta) = -\sum_{i;y=1} \frac{1}{p_i} \frac{\partial}{\partial\theta_j} p_i - \sum_{i;y=0} \frac{1}{1 - p_i} (-\frac{\partial}{\partial\theta_j} p_i)$$

With $p_i = \frac{1}{1+e^{-z}}$ where $z = \theta_0 + \sum_{j=1}^{n} \theta_j$, we have

$$\frac{\partial}{\partial\theta_j} p_i = \frac{e^{-z}}{(1 + e^{-z})^2} \frac{\partial}{\partial\theta_j} z = p_i(1 - p_i)x_j \tag{4.12}$$

for $0 \leq j \leq n$ and for $j = 0$,

$$\frac{\partial}{\partial\theta_0} p_i = p_i(1 - p_i).$$

Hence, $\frac{\partial}{\partial\theta_j} \log p_i = (1 - p_i)x_j$ and $\frac{\partial}{\partial\theta_j} \log(1 - p_i) = -p_i x_j$ with additional extension of vector x with $x_0 = 1$.

The derivation for the entire dataset is easily derived as the sum of the corresponding partial derivatives of the examples:

$$\frac{\partial}{\partial\theta_j} NCL = -\sum_{i;y=1} (1 - p_i)x_j^{(i)} - \sum_{i;y=0} -p_i x_j^{(i)} = -\sum_{i} (y_i - p_i)x_j^{(i)} \tag{4.13}$$

Notation $x_j^{(i)}$ refers to the j-th feature of the i-th feature vector $x^{(i)}$ in the random sample.

5. Modeling information propagation with internal and external influence

5.1. Modeling the structure of the network

Formally, information propagation network can be represented as a graph $G = (V, E)$. We refer to elements of set V as vertices, nodes or *peers*. Every edge $(u, v) \in E$ from node u to node v denotes that node u can (socially) influence node v. In other words, every neighbour of node u can be influenced by u. We call graph G the *peer influence graph*.

We extend graph G with a set A of *external nodes* or *authorities*. Every authority $a \in A$ can potentially influence every node in V but not vice versa. That is, every authority $a \in A$ is connected to every node $v \in V$ and can influence it. This influence is not social and it was previously referenced to as external influence. Additionally, the subgraph A is fully connected, *i.e.* every node $a \in A$ is connected bilaterally with all nodes $b \in A$ if $a \neq b$. If we use F to represent the directed edges from authorities to peers, we can define new a graph $H = (V \cup A, E \cup F)$, the *extended influence graph*.

5.2. Modeling the process of information propagation

Information item propagates between both peer and external nodes. If a particular node (peer or authority) has adopted an item, we refer to it as *active*. We call the nodes that have not adopted the same item as *inactive*.

The actual propagation process can be observed as successive activation of nodes throughout the network that we refer to as the *activation sequence*. Structure representing the activation sequence will be a set of nodes along with their activation timestamp. To describe the propagation process, we construct a new set of nodes D. A node $u \in V \cup A$ is in D if its activation time is recorded. Every node in D has an attribute t — the activation timestamp.

We assume the propagation of a particular information item happens in discrete time steps, from timestamp 1 to timestamp T. At every point in time, $t \in \{1, 2, \ldots, T\}$, each inactive node can become active with probability $P(x, y)$. This probability is a function of x — the number of neighbours of the observed node that became active before time t and y — the number of external nodes that became active before time t.

For every timestamp $t \in \{1, 2, \ldots, T\}$ and every currently inactive node u, an event of u adopting the information is observed. The success of this event is a Bernoulli random variable. If an event is successful, the node becomes active at that time. All successful events are listed in the activation sequence. We will use the logistic regression to model the probability P and determine the parameters of this distribution based on the activation sequence and the network H. That is, the probability P is defined as:

$$P(x, y; \alpha, \beta, \gamma) = \frac{1}{1 + e^{-\alpha \ln(x+1) - \beta \ln(y+1) - \gamma}}, \tag{5.1}$$

where α, β and γ are the parameters of logistic regression. The parameter γ takes the role of θ_0 in (4.7).

The arguments (features) x and y come into linear transformation in the argument of the sigmoid function transformed with the $\ln$ function. That will help in further analysis of retrieved values of α and β since $\ln(x + 1)$ and $\ln(y + 1)$ have a similar range of values.

The value of α in (5.1) captures the strength of social influence in the propagation of an information item, while the value of β captures the strength of external (authority) pressure. We refer to α and β respectively as *peer coefficient* and *authority coefficient*. Parameter γ models the impact of other factors in the propagation of an information item, such as daytime dynamic, effect of random chance and other noises. We call γ the *externality coefficient*.

5.2.1. Information propagation model training

We estimate α, β, and γ by learning the logistic regression classifier using negative log likelihood function.

To calculate the likelihood, we first define number of nodes that at the beginning of time t had x active influencing neighbors and they themselves became active at time t when y authorities were active as $A(x, y, t)$. Similarly, let $N(x, y, t)$ be the number of users who at the beginning of time t had x active influencing neighbors and they did not became active at time t when y authorities were active. Let $A(x,y) = \sum_{t=1}^{T} A(x,y,t)$ and $N(x,y) = \sum_{t=1}^{T} N(x,y,t)$.

The maximum-likelihood estimation of parameters α, β and γ are those that maximize the likelihood of the data at time t. If every inactive node u is characterized by two metrics, x and y, $L(\alpha,\beta,\gamma)$ is defined as

$$L(\alpha,\beta,\gamma) = \prod_{x,y} P(x,y)^{A(x,y)} (1 - P(x,y))^{N(x,y)}. \tag{5.2}$$

In the next step, the problem of finding the arguments α, β, γ that maximize (5.2) is reduced to minimization of negative log likelihood, just like in the general steps of learning the logistic regression classifier:

$$\begin{aligned} -\ln L(\alpha,\beta,\gamma) &= -\sum_{x,y} \ln P(x,y)^{A(x,y)} - \sum_{x,y} \ln(1 - P(x,y))^{N(x,y)} \\ &= -\sum_{x,y} A(x,y) \ln P(x,y) - \sum_{x,y} N(x,y) \ln(1 - P(x,y)). \end{aligned} \tag{5.3}$$

The derivatives with respect to α, β and γ of the same function were derived from (4.12) and (4.3.1) and are as follows:

$$\frac{\partial}{\partial \alpha}(-\ln L) = \sum_{x,y} [-A(x,y)(1 - P(x,y)) + N(x,y)P(x,y)] \ln(x+1), \tag{5.4}$$

$$\frac{\partial}{\partial \beta}(-\ln L) = \sum_{x,y} [-A(x,y)(1 - P(x,y)) + N(x,y)P(x,y)] \ln(y+1), \tag{5.5}$$

$$\frac{\partial}{\partial \gamma}(-\ln L) = \sum_{x,y} [-A(x,y)(1 - P(x,y)) + N(x,y)P(x,y)] . \tag{5.6}$$

The only problem remaining is the calculation of $A(x,y)$ and $N(x,y)$ that has to be efficient for a network with million nodes (number of nodes in the Complete Referendum Dataset).

5.2.2. Computing $A(x,y)$ and $N(x,y)$

For the given extended influence graph $H = (V \cup A, E \cup F)$ and the set of activated nodes D, $A(x,y)$ and $N(x,y)$ are computed using the following dynamic data structure: There are $M + 1$ buckets where M refers to the maximum number of active neighbours one node could have. The x^{th} bucket (with notation $bucket[x]$) will during the procedure contain all nodes that at the time had x active neighbours. At the beginning, all nodes are in the $bucket[0]$.

Every iteration of the algorithm tracks the dynamic of one time step. Values of $N(x, y)$ and $A(x, y)$ are updated for the current y and all currently possible values of x at every iteration. To correctly count the number of neighbours that became active exactly before time t for every node that didn't activate before time t, two passes through the corresponding nodes in D are needed.

Before the first pass, values of N are updated. Value of $N(x, y)$ is incremented for every node that currently has x active neighbours. Since the $bucket[x]$ stores all nodes with x active neighbours at that time, $N(x, y)$ is incremented by the size of this bucket. This step is correct since the following invariants hold:

- at every iteration, one node is stored in at most one bucket and
- the activated nodes are not in any bucket.

With this update, the nodes that will become active at currently observed time t are also counted in the values of N. This will be corrected in the first pass through the nodes in D. When the value of $A(x, y)$ updates for every activating node, corresponding value of $N(x, y)$ decrements.

In the second pass of nodes in D, the data structure of buckets is updated for recently activated nodes and their neighbours. That is, every activated node is removed from the structure and its neighbours are moved to the bucket that corresponds to one activated neighbour more than they previously had.

Lastly, for every iteration, *i.e.* time step, value of y is updated according to the activation timestamps of the external nodes.

The pseudocode of the described algorithm is given as Algorithm 2. This algorithm assumes the existence of the function $getBucket(node\ u)$ that returns the index of the bucket where node u is currently stored in $\mathcal{O}(\log(|V|))$ time.

Assuming all other functions take $\mathcal{O}(1)$ to complete, the time complexity of the algorithm is derived as follows:

$$\begin{aligned}
\mathcal{O}(|V| &\qquad \text{from line } 2 \\
+ TM &\qquad \text{from lines } 4-6 \\
+ T\frac{|D|}{T}\ln(|V|) &\qquad \text{from lines } 7-11 \\
+ T\frac{|D|}{T}\left[\ln(|V|) + M\ln(|V|)\right] &\qquad \text{from lines } 12-22 \\
+ T|A|) &\qquad \text{from lines } 23-27 \\
\Rightarrow \mathcal{O}(|V| + M|D|\ln(|V|) + MT + |A|).&
\end{aligned}$$

Algorithm 2 Computing $A(x, y)$ and $N(x, y)$

```
$y \leftarrow 0$
$bucket[0] \leftarrow V$
for $t = 0$ to $T$ do
    for $x = 0$ to $M$ do
        $N(x, y) \leftarrow N(x, y) + size(bucket[x])$
    end for
    for $u \in D, u.t = t$ do
        $x \leftarrow getBucket(u)$
        $N(x, y) \leftarrow N(x, y) - 1$
        $A(x, y) \leftarrow A(x, y) + 1$
    end for
    for $u \in D, u.t = t$ do
        $x \leftarrow getBucket(u)$
        $bucket[x].remove(u)$
        for $n \in u.neighbours()$ do
            if $notActive(n)$ then
                $x' \leftarrow getBucket(n)$
                $bucket[x'].remove(n)$
                $bucket[x' + 1].add(n)$
            end if
        end for
    end for
    for $a \in A$ do
        if $a.t = t$ then
            $y \leftarrow y + 1$
        end if
    end for
end for
return $A, N$
```

5.3. Assumptions of the presented model

The presented model assumes that the probability of activation for a node grows monotonically with number of activated neighbours as well as with number of active authorities. If the correlation between x and probability of activation were to be negative, the coefficient α would be estimated to some negative value. Similarly, if the correlation between y and probability of activation were to be negative, the coefficient β would be estimated to a negative value.

It is reasonable to assume that social influence of an active node fades away after certain time. That is also true for the external influence in the form of articles at online news sites since the articles aren't always as accessible as at the time of the publishment. This model assumes the influence of an active node does not fade away with time, *i.e.* the influence of a node, peer or authority, is constant after the activation.

Additionally, this model assumes the time progress in discrete time steps.

5.4. Randomization Test

We choose to infer whether information propagation of an item is better explained due to peer or external influence by obtaining the maximum-likelihood estimates of peer and authority coefficients α and β.

It is reasonable to say: if $\alpha > \beta$ than information is peer-propagated, otherwise, if $\alpha < \beta$, the information item is authority propagated. The question remains, how much larger should the value of α (or β) be in order to characterize information item as peer- (or authority-) propagated and how to verify that this result is due to strong evidence in the data.

In order to verify that this kind of conclusion is based on strong evidence in the data, [3] and [2] proposed randomization test called *time-shuffle test*.

Let D be the subset of $V \cup A$ that contains all nodes that eventually become active. The time-shuffle test permutes the activation times of the nodes in D. The randomized version of D made in this way contains the same nodes as D (those that eventually become active) with different attributes t. Denote by D' randomized version of D with permuted timestamp attributes t.

The input for the estimation algorithm consist of network H and set of nodes D, denoted by $\langle H, D \rangle$. Let $\alpha(D')$ and $\beta(D')$ be the estimates of peer and authority coefficients for input $\langle H, D' \rangle$. Let α and β denote the maximum-likelihood estimation of the peer and authority coefficients for the original input $\langle H, D \rangle$. Let $\mathcal{D}$ be the set of all possible randomized versions of set D via the time-shuffle test.

The *strength of peer influence*s S_α is defined as the fraction of randomized datasets $D' \in \mathcal{D}$ for which $\alpha > \alpha(D')$, thus

$$S_\alpha = P_{D' \in \mathcal{D}}(\alpha > \alpha(D')). \tag{5.7}$$

The *strength of authority influence* S_β is defined similarly, as the fraction of randomized datasets $D' \in \mathcal{D}$ for which $\beta > \beta(D')$, thus

$$S_\beta = P_{D' \in \mathcal{D}}(\beta > \beta(D')). \tag{5.8}$$

Both S_α and S_β take values in $[0, 1]$. The large value of strength of peer influence indicates stronger evidence of peer influence in the data. Similarly, larger value of β indicates stronger authority influence in the data. The proof for correctnes of decision making based on thus defined strengths for the similar model parametrized with 2 parameters can be found in [3].

6. Experimental results and analysis

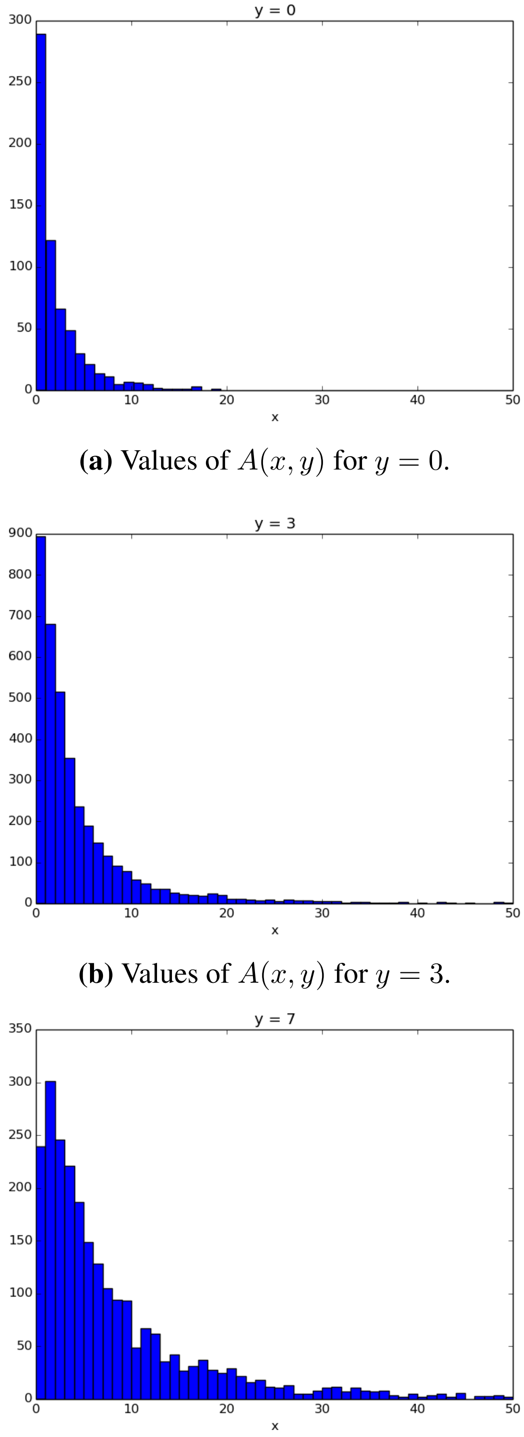
The model estimation was performed on the Referendum Dataset. Firstly, the estimation was performed on the restricted graph of registered users without the nodes that never become active. After that, the estimation was preformed on the complete graph of both registered users and their friends.

The complete procedure was implemented in Python. Although the Referendum Dataset cannot be provided, reusable parts of the implementation can be found in [5]. For the work with GML [1] format and the graph structure, *igraph* library [4] was used and for the minimization step, implementation of BFGS algorithm from the *SciPy* library [6] was used. The 1^{st} order derivatives as derived in (5.4), (5.5), (5.6) were given to the procedure directly while the 2^{nd} order derivatives in the form of Hessian matrix were approximated internally. The initial values of α, β and γ were taken uniformly at random from the range $[0, 1]$. For all experiments, a step size of 1 minute was taken and the data from the first 5 days of application being online was analyzed.

6.1. About the values of $A(x, y)$

Values of $A(x, y)$ for the fixed value of y may indicate the presence of external influence in the dataset. For a fixed value of y, the value of $A(x, y)$ stands for the frequency of activation for nodes with x active neighbours at the time, counted between the activation of y^{th} and $(y + 1)^{th}$ authority. Value of $A(x, 0)$ is thus calculated between time step 0 and the activation of the first authority. The graphs of $A(x, y)$ for the fixed value of $y > 0$ on the Referendum Dataset have the shape similar to Figure 6.1b. However, there is one exception and it is shown on the Figure 6.1c. For the reference, $A(x, 0)$, frequency of activation before authorities were present is shown on the Figure 6.1a.

In the case of information propagation where only social influence is present, graph of frequency of activation with respect to number of activated neighbours x would look similar to the Figure 6.1a. That is, frequency of activation for nodes that have at least one activated neighbour will be greater than the number of activations for nodes with



(a) Values of $A(x, y)$ for $y = 0$.

(b) Values of $A(x, y)$ for $y = 3$.

(c) Values of $A(x, y)$ for $y = 7$.

Figure 6.1: Values of $A(x, y)$ for fixed $y \in \{0, 3, 7\}$ with respect to number of activations with x neighbours active calculated for the Referendum Dataset.

0 neighbours active. A certain sign of authority presence is unexpected frequency of activation for nodes that have 0 activated neighbours. The Figure 6.1b is an example of that kind of activity where frequency of activation with 0 neighbours active exceed the frequency of activation with 1 friend active.

The information propagation dynamic after activation of authority $y = 7$ is suspected to be primarily pressured by internal (social influence) what is further approved with Figure 6.1c. It is important to notice that, as time passes and more nodes become active, the probability of a node having 0 active neighbours decreases, even for the nodes that become active pressured by external influence. Although the authority $y = 7$ becomes active relatively late, the frequency of activation related to authorities that activate after $y = 7$ form a shape similar to 6.1b. The suspicion of information about the application being propagated mostly pressured by external influence will be confirmed with a model.

6.2. Estimating peer and authority coefficients

Estimation of α, β and γ for the Restricted Referendum dataset gives $\alpha = 0.16432$, $\beta = 0.41826$ and $\gamma = -8.84820$. The strength of peer and authority influence was obtained by the time-shuffle test based on 1000 randomized instances of the input graph and the strengths $S_\alpha = 0.996$, $S_\beta = 0.633$ were obtained. Frequency histograms of estimated $\alpha(D')$ and $\beta(D')$ for randomized datasets $D' \in \mathcal{D}$ used in this test are shown on the Figure 6.2.

The estimation algorithm was ran for the Complete Referendum Dataset and the values $\alpha = 1.44990$, $\beta = -1.02360$ and $\gamma = -13.52627$ were obtained. The negative value of β may lead to conclusion the model, along with its assumptions, is not suitable for the Complete Referendum Dataset.

The maximum likelihood function for the model was constructed by examining the events of an inactive node 'trying' to activate independently for all time steps and all inactive nodes at that time. To reason about the negative value of β, the average frequency of activation and average number of trials (observed events) for the duration of time intervals while y authorities were active, along with the corresponding success rate is given in the Table 6.1a for the Restricted and in Table 6.1b for the Complete Referendum Dataset. Since the frequencies of activation and trials depend on the length of the observed time interval, they were given in the tables as average frequency per time step for every observed time interval. The success rate is observed, since the maximum likelihood estimator tries to fit the distribution model to similarly defined success rate.

y	0	1	2	3	4	5	6	7	8	9	10	11
successes	0.77791	9.00000	7.34400	3.91023	4.52381	3.74797	1.26634	0.89345	0.67893	1.41053	3.40826	1.62857
trials	12549.02	11792.91	11058.83	7636.39	6764.81	6443.38	5156.96	3001.87	1882.61	1664.66	1434.86	879.45
success rate (10^{-3})	0.062	0.763	0.664	0.512	0.669	0.582	0.246	0.298	0.361	0.847	2.375	1.852

(a) Restricted Referendum Dataset.

y	0	1	2	3	4	5	6	7	8	9	10	11
successes	0.77791	9.00000	7.34400	3.91023	4.52381	3.74797	1.26634	0.89345	0.67893	1.41053	3.40826	1.62857
trials	1697340	1694519	1693785	1690362	1689491	1689169	1687883	1685728	1684609	1684391	1684161	1639898
success rate (10^{-6})	0.458	5.311	4.336	2.313	2.678	2.219	0.750	0.530	0.403	0.837	2.024	0.993

(b) Complete referendum dataset.

Table 6.1: Average frequencies of observed successful activations per time step, average frequencies of trials per time step and success rates for the time period when y authorities were active on the Restricted and Complete Referendum Datasets.

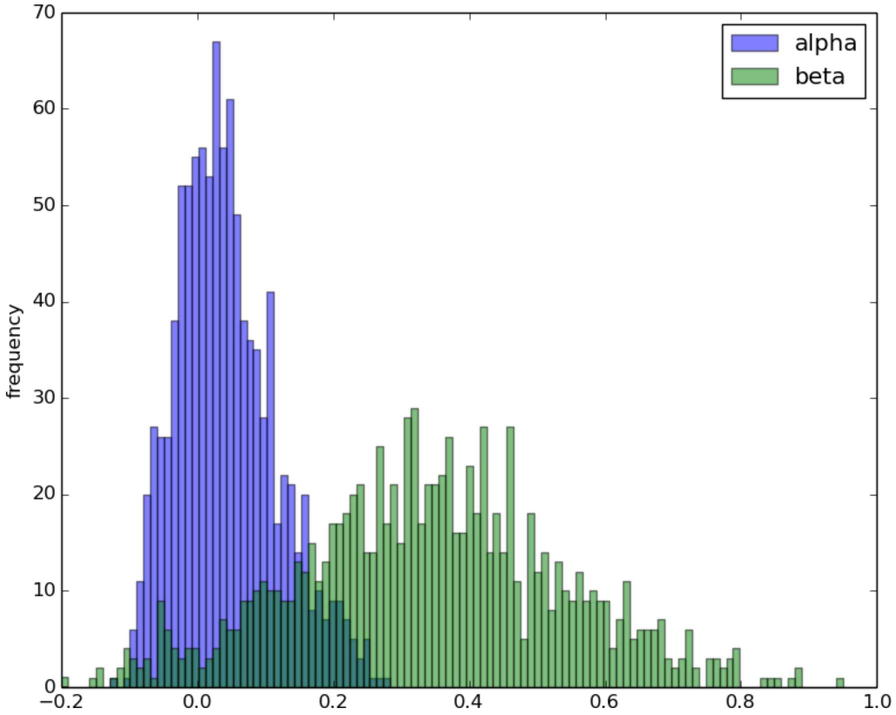


Figure 6.2: Frequency histograms of estimated values of α (blue) and β (green) obtained with original version of the model for the Restricted Referendum Dataset with time-shuffle test based on 1000 randomized instances.

It is important to notice that average frequency of successful events for each time interval stays the same for both Restricted and Complete Referendum Dataset. Since at each time step only inactive nodes are observed and conducted in trials, it is expected for the averaged frequency of trials to decrease with time as the observed set of nodes shrinks in size. The smaller averaged number of trials for the events observed in late time intervals for the Restricted Referendum Dataset, further induces the growth of success rate as the number of active authorities increases.

Since there are 99.25% of nodes in the Complete Referendum Dataset that never become active, the number of trials stays roughly the same for the observed time intervals, as seen in the Table 6.1b. Therefore the success rate is roughly proportional to the number of successful activations for every interval, *i.e.* it decreases, thus explaining the negative value of β.

Since the conducted analysis has shown the proposed model is not suitable for the given Complete Referendum Dataset, the model will be redefined.

6.3. Redefining the model of external influence

The activation of a node without any previously activated neighbours is a pattern attributable to external influence. To make the proposed model suitable for our purpose, the new definition of attribute y will be provided.

If y in (5.1) were to be defined as the frequency of activated nodes at the observed time step whose neighbours haven't been activated before, the estimated probability distribution will be expected to increase with y. Additionally, thus defined attribute doesn't necessarily have positive correlation with time, unlike the initial definition of y. Hence, the errors caused by decreasing number of trials at the end of the observed time for the Restricted Referendum Dataset and the error caused by too many observed nodes that finally don't activate even though maximum number of authorities are active for the Complete Referendum Dataset, have less impact on the results.

Estimation of α, β and γ for the Restricted Referendum Dataset with newly defined model of external influence gives $\alpha = 0.16697$, $\beta = 0.92495$ and $\gamma = -8.60957$. The strength of peer and authority influence was again obtained by the time-shuffle test based on 1000 randomized instances of the input graph and strengths $S_\alpha = 0.018$, $S_\beta = 0.959$ were obtained. Frequency histograms of estimated $\alpha(D')$ and $\beta(D')$ of randomized datasets $D' \in \mathcal{D}$ used in this test are shown on the Figure 6.3.

The estimation for the Complete Referendum dataset returns $\alpha = 0.89215$, $\beta = 1.36409$, $\gamma = -15.7230$. The strength of peer and authority influence was again obtained by the time-shuffle test based on 500 randomized instances of the complete input graph and strengths $S_\alpha = 1.00$, $S_\beta = 0.99803$ were obtained. Frequency histograms of estimated $\alpha(D')$ and $\beta(D')$ of randomized datasets $D' \in \mathcal{D}$ used in this test are shown on the Figure 6.4.

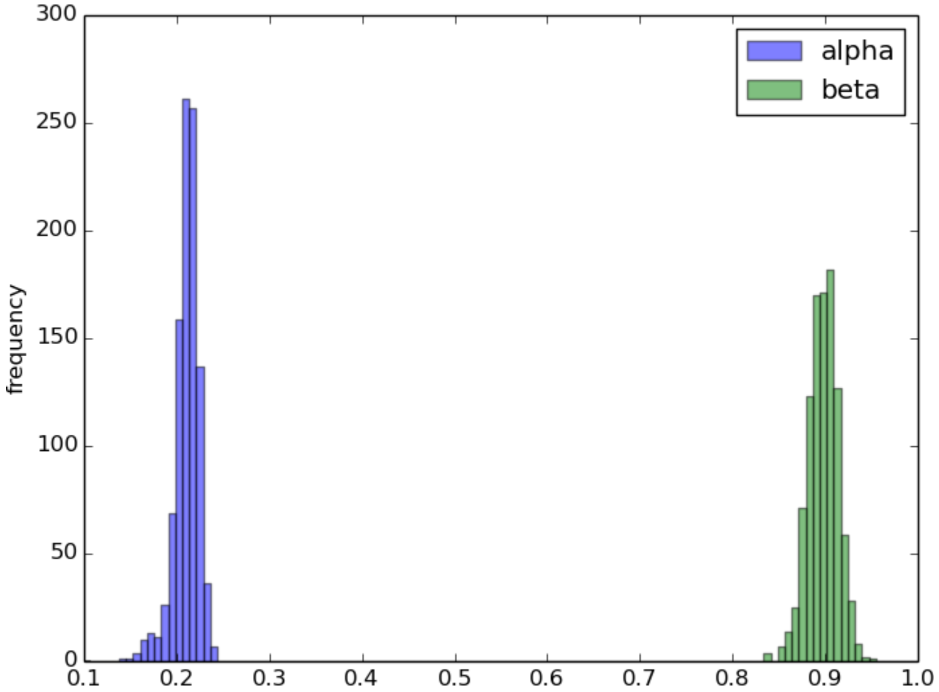


Figure 6.3: Frequency histograms of estimated values of α (blue) and β (green) obtained with updated version of the model for the Restricted Referendum Dataset with time-shuffle test based on 1000 randomized instances.

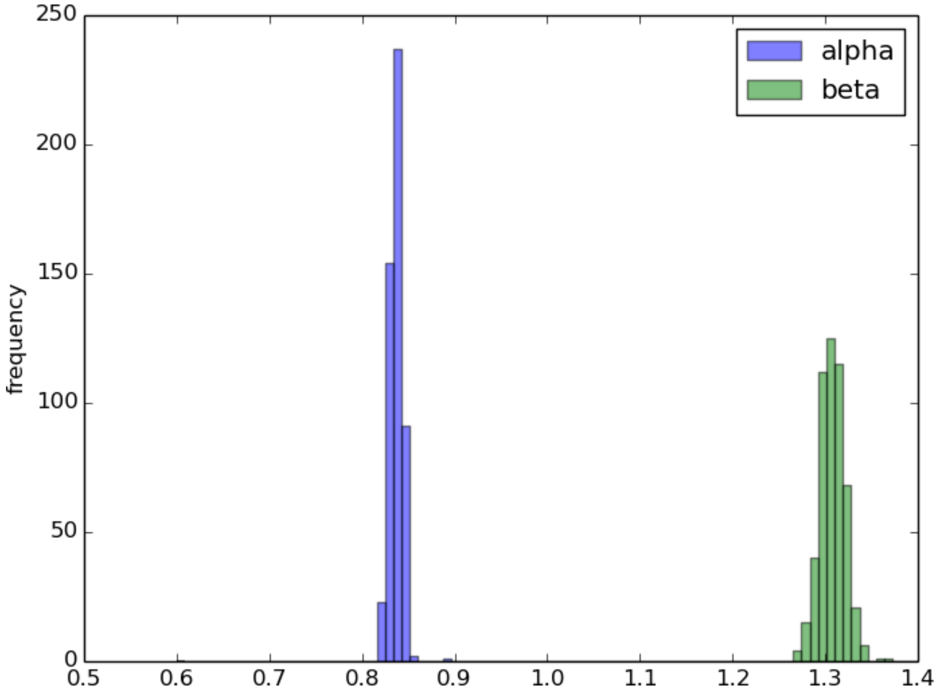


Figure 6.4: Frequency histograms of estimated values of α (blue) and β (green) obtained with updated version of the model for the Complete Referendum Dataset with time-shuffle test based on 500 randomized instances.

7. Conclusion

The model for information propagation given in [2] takes into account that information propagation in social networks can be driven by both social (internal) and authority (external) influence. The proposed model is used for determining whether the information item is mainly propagated driven by internal or external influence. The probability of activation for a node that has x active neighbours when y authorities were active is modeled as logistic regression with 3 parameters, peer influence coefficient α, authority influence coefficient β and externality coefficient γ. The logistic regression model is then estimated using the maximal likelihood estimator. Based on the estimated values of α and β and taking into account the results of the randomized test called the time-shuffle test, a decision is made.

The information propagation analysis for the Referendum Dataset based on the presented model of peer and authority influence has returned estimations for peer and authority coefficients that indicate the model is not suitable for the Referendum Dataset. That is expected since the model assumes the influence of an active node (peer or authority) does not fade away with time which can be easily disproven for the Referendum Dataset.

If the external influence for an inactive node at time t is measured by the number of nodes that activate at that time and do not have any active neighbours, for estimated peer coefficient α and authority coefficient β, $0 < \alpha < \beta$ holds on both Restricted and Complete Referendum Dataset. The time shuffle test for the Restricted Referendum Dataset has returned a small value of peer strength and a large value of authority strength. Hence, the information propagation is mainly pressured by external influence. The estimation and the shuffle test for the Complete Referendum Dataset reassures the conclusion is correct.

BIBLIOGRAPHY

[1] Sheng Ye A, Feng Xuezhi A, Shaotao Yuan C, and Ruan Renzong D. Gml – – an open-standard geospatial data format.

[2] Aris Anagnostopoulos, George Brova, and Evimaria Terzi. Peer and authority pressure in information-propagation models. 2011.

[3] Aris Anagnostopoulos, Ravi Kumar, and Mohammad Mahdian. Influence and correlation in social networks. In *Proc. of the 14th ACM International Conference on Knowledge Discovery and Data Mining (KDD 2008)*, pages 7–15. ACM Press, 2008.

[4] Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006.

[5] Miholić Iva. Implementation of the peer and authority model. `https://github.com/imih/information-propagation`, 2014.

[6] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python. `http://www.scipy.org/`, 2001–.

[7] Valdis Krebs. The social graph of a facmous mathematician. `http://www.orgnet.com/Erdos.html`.

[8] Seth A. Myers, Chenguang Zhu, and Jure Leskovec. Information diffusion and external influence in networks. *CoRR*, abs/1206.1331, 2012.

[9] M. E. J. Newman. The structure and function of complex networks. *SIAM REVIEW*, 45:167–256, 2003.

[10] Nino Antulov-Fantulin, Tomislav Lipić, Iva Miholić, Matija Piškorec, Bruno Rahle, and Mile Šikić. referendum2013.hr. `http://referendum2013.hr/`, November 2013.

Information propagation in online social networks

Abstract

In this paper the information propagation model that takes into account both internal (social) and external (authority) influence is presented and revisited. Based on the estimated parameters for the model and the randomization test called time-shuffle test, a decision whether propagation of an information item is mainly peer or authority influence driven can be made. The proposed model and its updated version were used to describe an information item - the existence of the Facebook application referendum2013.hr - as mainly propagated by external (authority) influence.

Keywords: complex networks, social networks, information propagation,, internal and external influence, peer and authority model,time-shuffle test