Ranking the Influence of Micro-blog Users Based on Activation Forwarding Relationship

Yiwei Yang[1,2], Wenbin Yao[1,2(✉)], and Dongbin Wang[3]

[1] Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications, Beijing, China
yiwei_yang@163.com, yaowenbin_cdc@163.com
[2] National Engineering Laboratory for Mobile Network Security, Beijing University of Posts and Telecommunications, Beijing, China
[3] Key Laboratory of Ministry of Education for Trustworthy Distributed Computing and Service, Beijing University of Posts and Telecommunications, Beijing, China

Abstract. How to predict the influence of users in micro-blog is a challenging task. Although numerous attempts have been made for this topic, few of them analyze the influence of users from the perspective filtration mechanism. In this paper, we propose a novel Activation Forwarding Relationship Independent Cascade algorithm for analyzing the influence of users. The algorithm mainly consists of two parts: forwarding prediction and activation process. We predict the forwarding relationship by Random Forest (RF) and improve the Independent Cascade algorithm to construct an activation network. The algorithm can filter non influence users during the construction of the activation network, thus reducing the amount of ranking time. By calculating the user's activation capability, we rank user's influence. The experimental results show that our algorithm can achieve 95% accuracy in predicting forwarding relationships. Besides, our algorithm not only saves computing time, but also shows that the Top-10 users in the ranking list have better ability to spread information than the existing ranking algorithms.

Keywords: Activation forwarding · Random forest
Independent cascade model · Micro-blog

1 Introduction

As a large social network platform, Micro-blog has attracted the attention of many researchers on how to excavate the influential users. There are two problems in the mining process: how to define the influence and how to deal with a large number of user relationships.

At present, most of the research on the influence of user identification is based on the PageRank [1] or HITS algorithm [2]. The improvement of ranking algorithm can be divided into three categories. The first is user influence ranking in the different themes or different areas [3, 4]. Weng et al. proposed the Twitterrank algorithm [3] to measure the influence of users in Twitter. The algorithm takes both the topical similarity

I. Romdhani et al. (Eds.): CollaborateCom 2017, LNICST 252, pp. 383–393, 2018.
https://doi.org/10.1007/978-3-030-00916-8_36

between users and the link structure into count when measure the influence of user. Besides, Ding et al. [4] also improved the PageRank algorithm from the view of topic. In addition, some other researches measure user influence by taking both user interaction and network structure into consideration. One such work is TunkRank [5], a variant of PageRank. This approach reflects differences between users, more in line with the actual situation of network relations. However, complex interactions lead to high time complexity and space complexity. Finally, there is a way to improve the ranking algorithm is constantly updated the influence ranking according to the time [6, 7]. Hu et al. [6], consider three temporal factors that are BTF, FF and SF, and adopted them to PageRank algorithm. Then, they propose a novel algorithm T-PR. Ma et al. [7] focus on user behavioral characteristics and predict the probability that user will respond using logistic regression (*LR*). However, when the data dimension is high, the algorithm of *LR* is not very applicable.

These ranking algorithms have made some improvements from different aspects. However, they have the same problem that a large number of users without influence in the network to participate in the interactive ranking take up much computing time. So it is important to filter out the inactive users in the network. Compared with the above mentioned work, the main contribution of this paper is to filter the non influence users in the network by predicting the forwarding relationship between users, thus reducing the ranking time of the influential users.

The contributions of this paper can be summarized as follows:

- We focus on how to filter non influential users to reduce computation time. For this purpose, an algorithm is proposed, namely Activation Forwarding Relationship Independent Cascade (AFRIC) algorithm.
- We introduce the Random Forest (RF) algorithm to predict the forwarding relationship between users according to the user's recent behavior and attribute data.
- In addition, we combine the results of activation into the improved Independent Cascade (IC) algorithm and construct an activation forwarding network for ranking the influence of users.

The rest of the paper is structured as follows. In Sect. 2, our approach is proposed. Section 3 shows the experimental results. Finally, Sect. 4 concludes this paper.

2 Our Approach

2.1 Factor Selection

Factor selection is the first step in predicting user activation relationships. In this section, ten factors that influence forwarding are explored, and the importance sequencing of these factors is trained. Then, several important factors are chosen as the influencing factors of forecasting information forwarding. Suppose that the user u is the superior user and the user v is the subordinate user. The ten factors in the Table 1 include the individual features of user u or user v and the interaction characteristics of the user pair (u, v). Besides, some factors are direct and some factors are recessive.

Table 1. Factors affecting forwarding

Factors	Means
F_F	Followers number of *u*/friends number of *u*
ST_ratio	Tweets number of *v* forward *u*/number of *u* tweets
ST_topic	*jaccard* of $v's$ topic set and $u's$ topic set
T_activity	Number of *u* tweets over a period of time
S_positive	Number of *v* forward over a period of time
T_created	Registration time of *u*
T_lists	Lists number of *u* involved in
S_friends	Friends number of *v*
T_forward	Average forwarding ratio of $u's$ tweets
T_time	Time period of *u* tweets (0–23)

2.2 Forwarding Prediction

Random Forest (RF) is a kind of classifier which is composed of multiple Classification and Regression Tree (CART). The training set used by each tree is sampled from the total training set. In the training of each node of the tree, the factors are derived from the random sampling of all factors in a certain proportion. The Random forest training process is as follows. Given a training set N, test set T, random sampling from N to form a new sub sample data. Each training sample $(F_{uv}, \sigma(u,v))$ contains two pats, which are the factors $F_{uv} = \{f_{m,uv} | m = 1, 2, \ldots M\}$ and the classes of the user pair $\sigma(u,v) = 0 : No\ forwarding\ or\ \sigma(u,v) = 1 : Forwarding$, where u and v are users. For all the factors, we randomly selected $m \leq M$ factors to construct a complete decision tree. Repeat the above steps and we can get k decision trees $(h_i(i = 1, 2, \ldots k))$. Finally, each decision tree is used to select the optimal classification. For a test sample $F_{u'v'} \in T$, the classification result is got by the way of vote. The formula is as follows.

$$\sigma(u', v') = majority\ vote\{(h_i(F_{u'v'}))\}_{i=1}^{k} \tag{1}$$

For each user pair $(u,v) \in E$, using the Random Forest model to train the factors of (u,v), we can predict whether the user v will forward the user $u's$ information. If $\sigma(u,v) = 1$, the user u can activate user v and $\sigma(v) = 1$, else $\sigma(u,v) = 0$ means the user u can't activate user v and $\sigma(v) = 0$.

The number of the decision trees and the number of factors selected in the node splitting is two important parameters of the Random Forest algorithm. The training set includes N user pairs and M factors. Each decision tree generates new training set by *Boostrap* sampling. When each tree node is split, m input factors are selected from the M factors, and then the best factor is selected to split from the m factors by *Gini*. Every tree splits all the way, until all the training examples of the node belong to the same class.

2.3 Independent Cascade Model

The Independent Cascade (IC) model is based on the theory of probability and the Interaction Particle System (IPS). Given the network $G(V, E)$, for each directed edge $e = (u, v) \in E$, we predict a value $\sigma(u, v)$. Here $\sigma(u, v) = 0$ *or* 1 is the state of the edge $e = (u, v)$. If $\sigma(u, v) = 1$, v will be activated by user u. In the IC model, once a user u is activated in step t, it will activate its neighbor user in the $t+1$ step. Besides, each active user has only one chance to activate its neighbor. The diffusion rules of the IC model are as follows. In the $t-1$ step, the active collection of nodes is defined as S_{t-1}. In the $t+1$ step, each active user $u \in d^i(v) \cap (S_t - S_{t-1})$ will activate it's out of the neighbor, where $d^i(v)$ denotes the input users of v. If successful,$v \in S_{t+1}$, otherwise user u will no longer attempt to activate the user v. Repeat the above steps until no user is activated in the network.

Influence is defined as the ability of users to drive other users to forward their information. This capability includes direct activation and indirect activation of the number of users. Suppose that the users activated by u are distributed in $layer_u$ layers, the number of users in the $1 \leq j \leq layer_u$ layer is marked as $Num(u, j)$. The influence of u is defined as the weight sum of users activated at different layers.

2.4 Activation Process

The main part of the AFRIC algorithm is the activation process. For $G(V, E)$, V is the set of user nodes, and E is the set of directed edges. The direction of the directed edge is the opposite direction of following. The output of the node u is represented by $d^o(u)$, and the input of the node u is represented by $d^i(u)$. As shown in formula (2), all nodes and edges are inactive at the beginning.

$$\begin{cases} \forall \sigma(v) = 0, & v \in V \\ \forall \sigma(u, v) = 0, & (u, v) \in E \end{cases} \tag{2}$$

In the graph, those whose $d^o(v) \neq 0$ are selected as seeds. As shown in Fig. 1, $S_0 = \{v_1, v_7, v_{10}, v_{16}\}$. For the seed $u \in S_0$, $\sigma(u) = 1$. Each seed will attempt to activate the users directly connected to it only once. According to the selected factors, if the Random Forest algorithm predicts that the connected user v will forward his

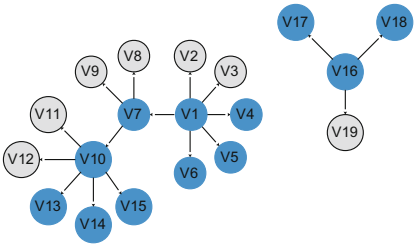

Fig. 1. The activation forwarding graph.

Table 2. Activate relationship and quantity.

Pre_v	v	Num_v
Null	v_1	4
v_1	v_7	1
v_7	v_{10}	3
Null	v_{16}	2

Algorithm 1. (AFRIC) Activate Forwarding Relationship Independent Cascade

Input: Social graph $G(V,E)$, F_{uv}, $(u,v) \in E$
Output: The forecast influential user list $L = (R,V)$
1: select seed $u \in S_0$, $d^o(u) \neq 0$
2: init each Num_u=0
3: **for** every seed $u \in S_0$ **do**
4: **for** every edge $(u,v) \in d^o(u)$ **do**
5: $\sigma(u,v) = RF(F_{uv})$
6: **if** $\sigma(u,v) = 1$ **then**
7: $\sigma(v) = 1$
8: $v \in S_1$
9: $Num_u + 1$
10: $Pre_v = u$
11: **end if**
12: **end for**
13:**end for**
14: **for** every $\sigma(v) = 1$ **do**
15: link edge (pre_v, v)
16: $layer_{pre_v} = layer_v + 1$
17: **end for**
18: **for** every $v \in S_0 \cup S_1$ **do**
19: $Influence_v = \sum_{j=1}^{layer_v} e^{-(j-1)} \log\left(Num_{v,j}\right)$, $Num_{v,j} \geq 1$
20: **end for**
21: $L = Rank(Influence_v)$
22: **return** L

information, the user can be activated by u, and $\sigma(v) = 1$. In the graph, the blue nodes are active users, and the gray nodes are not activated.

In the activation, we need to record each active connection and the number of direct activation per seed user. At the beginning, the precursors of the seeds are null. Once a seed is activated by another seed, we need to change its precursor. For example, v_7 can be activated by v_1, and $Pre_{v_7} = v_1$. After activation, the record table is shown in the Table 2.

2.5 Influence Calculation

The Table 2 records the users who activated more than zero, its precursors (Pre_v) and direct activation number (Num_v). Assuming that the directly activated users by v belong to the first layer, and the users activated by his followers belong to his second layer. If keep on going, we can then calculate the number of layers and the number of active users per layer. For example, the activation list of user v_1 is shown as follow.

$$List_{v_1} = \{(layer_1, 4), (layer_2, 1), (layer_3, 3)\} \tag{3}$$

The activation layer of user v is marked as $layer_v$, and the total number of layer $j(1 \leq j \leq layer_v)$ activations is marked as $Num_{v,j}$. However, the number of users in different layers contributes differently to the influence of user v_1. In fact, the greater of the layer, the smaller its contribution to user's influence. So we count the user's influence as the formula (4).

$$Influence_v = \sum_{j=1}^{j=layer_v} e^{-(j-1)} log\left(Num_{v,j}\right), \quad Num_{v,j} \geq 1 \tag{4}$$

3 Experiments

3.1 Experimental Setup

Our data set [8] contains four parts: user information, micro-blog information, user relation and micro-blog relation. In order to verify the changing trend of the running time of the algorithms with the increase of the amount of user data, we need to divide the whole data set. The main principle of segmentation is to make the relationship between users to focus as much as possible. So we divide the data according to the time sequence of user registration. The data sets are shown in Table 3. Set8 is the complete data set. The Set1 to Set7 is the segmentations of Set8 in accordance with the user registration time sequence. As time goes on, the amount of data increases gradually. The aim is to compare the running time of the AFRIC to the ranking algorithms in different data volumes.

Table 3. Data set list.

Datasets	#User	#Follow	#Tweet	#Retweet
Set1	935	21,813	1,659	458
Set2	5,212	110,041	8,802	2,371
Set3	10,934	219,478	17,425	4,979
Set4	20,205	431,241	31,756	9,713
Set5	30,957	658,147	49,014	14,857
Set6	40,137	875,945	64,970	19,987
Set7	50,925	1102,390	80,126	23,901
Set8	63,642	1,391,719	84,169	27,760

3.2 Forward Prediction

We use 70% user pairs from Set8 to train the Random forest algorithm and 30% user pairs to test the accuracy of the prediction. In the process of training, we mainly adjust and optimize the Random Forest algorithm from two aspects: the factors selection and the number of trees.

The ten factors listed in Table 1 are not all high impact factors, so in order to reduce the complexity of the algorithm we can choose several important factors to predict. In this paper, we use the *MeanDecreaseAccuracy* and *MeanDecreaseGini* as the basis for weighing the importance of the factors. *MeanDecreaseAccuracy* is the average accuracy reduction of the independent variables before and after the perturbation, and the *MeanDecreaseGini* is the reduction in the total number of nodes for all the tree variables. In the case of the two indicators, the importance ranking of the ten factors is shown in Fig. 2.

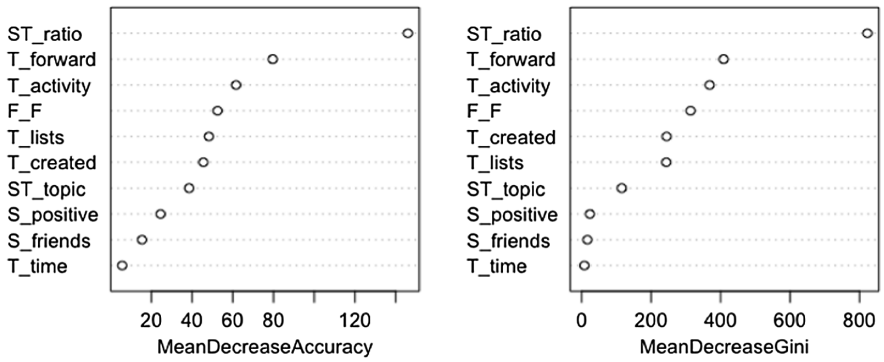

Fig. 2. Factors importance ranking, the horizontal axis is the index reduction quantity and the vertical axis is the factors' name.

As shown in Fig. 2, the importance rankings of factors in the two indicators are consistent and the influence of *T_time*, *S_friends* and *S_positive* are relatively weak. In the importance ranking chart of *MeanDecreaseGini*, it can be seen that the above three factors are close to zero. Therefore, taking the top seven factors can reduce the complexity of the algorithm on the basis of ensuring the accuracy of prediction. Besides, the number of trees in the Random Forest algorithm is an important factor

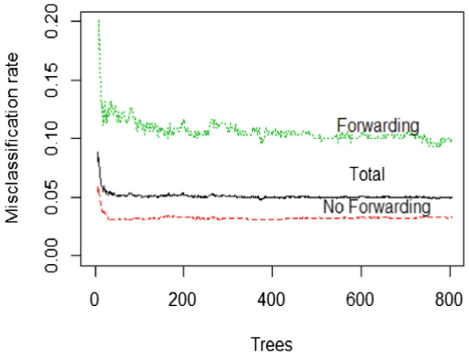

Fig. 3. The misclassification rate of prediction.

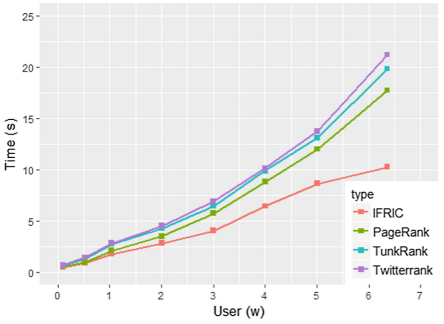

Fig. 4. Time comparison of different algorithms.

affecting the accuracy. In the different number of trees, the misclassification rate is shown in Fig. 3.

As shown in Fig. 3, the green line represents the forwarding misclassification rate, the red line represents the no forwarding misclassification rate and the black line represents the total misclassification rate. With the increase of the tree number, the misclassification rate of Random Forest algorithm decreases. When the number of trees reaches 400, the misclassification rate is almost stable. Therefore, this paper constructs with 400 trees, and the accuracy rate can reach 95%.

3.3 Comparison with Ranking Algorithms

In order to verify the effectiveness of the AFRIC algorithm, we compare it with PageRank, TunkRank and Twitterrank on the data set Set1–Set8. The running time (unit is second) of the three algorithms in different amounts of data is shown in Fig. 4.

It can be seen that the running time of TunkRank algorithm and Twitterrank algorithm is basically the same. When the number of users is less than 20,205, the running time of our algorithm is not distinct from that of the ranking algorithms. While with the increase in the amount of data, the speed of the advantages of AFRIC gradually revealed. When the number of users reached 63,642, the running time of AFRIC is reduced by 42.1% compared with PageRank algorithm, is reduced by 48.3% compared with TunkRank and is reduced by 51.6% compared with Twitterrank algorithm.

On the other hand, we need to verify the accuracy of the AFRIC algorithm on the user influence ranking. We introduce from three aspects of the list's similarity, consistency and the ability of Top-10 users to spread.

We first verify the similarity of the Top-k users in list by the similarity index *Osim*. It determines the repeatability between the Top-k users of the two ranking lists. *Osim* is defined as follows, where l_{k1} and l_{k2} are the Top-k user lists of L_1 and L_2. Here we choose Set8 as the data set and the experimental results are as shown in Fig. 5

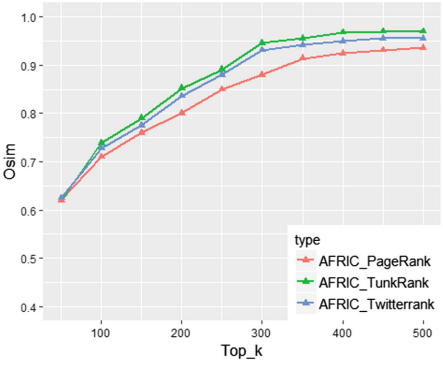

Fig. 5. *Osim* of AFRIC with other algorithm under different k.

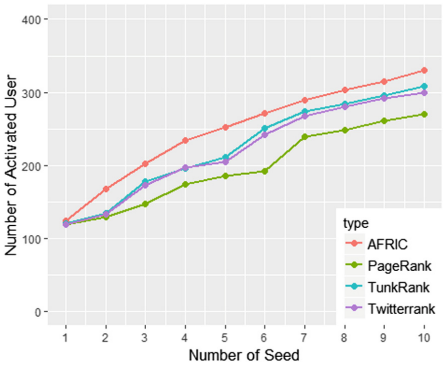

Fig. 6. Contrast the number of activated users by different methods.

$$Osim(L_1, L_2)_k = \frac{|l_{k1} \cap l_{k2}|}{k} \tag{5}$$

As can be seen from Fig. 5, with the increase of k, the similarity increases gradually. When k grows to 500, the $Osim(L_{AFRIC}, L_{PageRank})$ reaches 93.6%, the $Osim(L_{AFRIC}, L_{TunkRank})$ reaches 97.2%, and the $Osim(L_{AFRIC}, L_{Twitterrank})$ reaches 95%. Therefore, the AFRIC algorithm has credibility in identifying high influence users.

To compare the consistency of the ranking lists, we propose index *Kendll tau*. It determines the rank consistency of two lists containing the same users. It is defined as follows.

$$Kendall\ tau(L_1, L_2) = \frac{1 - 2|(u', v') : u', v'\ is\ reverse\ order\ in\ L_1, L_2|}{|L_1|(|L_1| - 1)} \tag{6}$$

where $|L_1|$ is the length of L_1, and L_2 has the same length with L_1. However, the AFRIC algorithm doesn't rank the inactive users. In order to ensure that the ranking lists contain the same users, we remove the inactive users from the PageRank list, TunkRank list and Twitterrank list. In addition, the contrast is implemented on the data Set8. The *Kendll tau* results of AFRIC contrast other three ranking algorithms are shown in Table 4. As can be seen from Table 4, when the number of users reached 63,642, the *Kendall tau* of AFRIC is 0.56 compared with PageRank algorithm, 0.65 compared with TunkRank algorithm and 0.62 compared with Twitterrank algorithm.

Table 4. *Kendall tau* ranking coefficient.

Pairs	Kendall tau
AFRIC vs PageRank	0.56
AFRIC vs TunkRank	0.65
AFRIC vs Twitterrank	0.62

In the last step, we compare the ability of Top-10 users to spread information in the ranking lists of the four algorithms on data Set8. Similar to influence maximization [9], we regard the Top-10 users as the seed set and simulate information spread in the network to get the number of finally activated users. Then we make a contrast on which method activates most users. The seed set size is from 1 to 10, according to the ranking lists of different algorithms. The Fig. 6 shows us that although all the trends are ascending, each time the number of activated users by AFRIC is greater than that by other three algorithms. From the overall activation trend, the number of activated users by AFRIC algorithm is more stable. It verifies the top users selected by our method are really most influential.

4 Conclusions and Future Work

We present an AFRIC algorithm to rank the influence of users in the micro-blog. It is better at running time and spreading information than existing ranking algorithms. The success of our algorithm is from (1) it filtered out inactive users in the network by activating forwarding and (2) it calculated the user's influence through the user's activation capability, which significantly improved the speed of ranking. The experiment results show that when the user amount reaches 63,642, the running time of AFRIC is reduced by 42.1% compared with PageRank algorithm, is reduced by 48.3% compared with TunkRank algorithm, and is reduced by 51.6% compared with Twitterrank algorithm. Besides, the similarity and consistency of AFRIC and ranking algorithms are credible. Finally, we verify the Top-10 users of our list are better at spreading information than that of other three lists. So when the greater the amount of data generated by micro-blog, the advantages of the AFRIC algorithm will be better displayed. In future work, we will take advantage of the user relationship to predict the information dissemination.

Acknowledgements. This work was partly supported by the NSFC-Guangdong Joint Found (U1501254) and the Co-construction Program with the Beijing Municipal Commission of Education and the Ministry of Science and Technology of China (2012BAH45B01) and National key research and development program (2016YFB0800302) the Director's Project Fund of Key Laboratory of Trustworthy Distributed Computing and Service (BUPT), Ministry of Education (Grant No. 2017ZR01) and the Fundamental Research Funds for the Central Universities (BUPT2011RCZJ16, 2014ZD03-03) and China Information Security Special Fund (NDRC).

References

1. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: bringing order to the web. Technical report, Stanford InfoLab, November 1999. http://ilpubs.stanford.edu:8090/422
2. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. J. ACM **46**(5), 604–632 (1999)
3. Weng, J., Lim, E.P., Jiang, J., He, Q.: Twitterrank: finding topic-sensitive influential twitterers. In: Third International Conference on Web Search and Data Mining, pp. 261–270. ACM (2010)
4. Ding, Z., Zhou, B., Jia, Y., et al.: Topical influence analysis based on the multi-relational network in microblogs. J. Comput. Res. Dev. **50**(10), 2155–2175 (2013)
5. Page, L.: The PageRank citation ranking: bringing order to the web. Stanf. Digit. Libr. Working Pap. **9**(1), 1–14 (1999)
6. Hu, W., Zou, H., Gong, Z.: Temporal PageRank on social networks. In: Wang, J., et al. (eds.) WISE 2015. LNCS, vol. 9418, pp. 262–276. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26190-4_18
7. Ma, X., Li, C., Bailey, J., Wijewickrema, S.: Finding influentials in Twitter: a temporal influence ranking model. arXiv preprint arXiv:1703.01468 (2017)

8. Data tang: 63641 sina micro-blog data set. http://www.datatang.com/data/46758. Accessed 20 Nov 2016
9. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 137–146. ACM (2003)