The Trust Value Calculating for Social Network Based on Machine Learning

Wang Yuji
College of Engineering
Cornell University
Ithaca, New York, USA
wangyuji_2017@163.com

***Abstract*—In this paper, a social network model is built for the social network information in the social network, and the machine learning method is used to calculate the node trust value. First, the results calculated by the traditional node trust value calculation method and some auxiliary information are used as the training feature of the machine learning, and the measurement whether there is edge between nodes as label information. Second, the node logistic regression model is used as the training model to calculate the node trust value. Then, recommendation algorithm which is analogous to the user collaborative filtering algorithm is used to calculate node trust value. At last, the simulation is used to verify the performance of the improved method, and the results show that the prediction accuracy of node trust value computing by improved algorithm is significantly higher than that of node trust value computing by formula.**

Keywords—trust value; machine learning; social network

I. INTRODUCTION

With the development of the Internet, more and more information is provided for us. And it also brings the problem of information overload [1]. Information overload makes it difficult for users to obtain effective information, and the information utilization is low. In order to effectively solve the problem, recommended system technology has been widely used. This system links users and items automatically by the recommendation algorithm [2].

Recommendation algorithm is the core of recommendation system, and it has been deeply studied by scholars. A robust collaborative recommendation algorithm based on kernel function and Welsch reweighted M-estimator was proposed to solve the poor robustness against shilling attacks of recommendation algorithms based on matrix factorization [3]. The authors proposed a message-passing based social recommendation algorithm that exploits the social relations between users in the social network to generate top-N recommendations, using only implicit user preference data [4]. A novel trajectory-pattern-based top-k similar users discovery method was proposed, and a two-layer location-type description scheme was suggested to identify the location trajectory on geographical space and describe the type trajectory on semantic space [5]. A new matching recommendation algorithm which established of binary relation between users and information products and mined each user potential object of interest by using selection process or similarity relation was proposed to help an enterprise to find one or several proper celebrities as their product endorsement [6]. Motivated by the observation that related items often have similar ratings, a framework integrating items' relations, users' social graph and user-item rating matrix for recommendation was proposed [7]. The authors described a new model-based algorithm which is based on a generalization of probabilistic latent semantic analysis to continuous-valued response variables [8].

Collaborative filtering recommendation algorithm is one of the most widely used recommendation algorithm. Based on the user evaluation behavior, it finds the user whose interest is similar to that of the target user. However, it ignores the social relationship between users which implies the potential interests of users. Thus, the calculated user trust score based on the social relationships is very valuable. The most critical step of obtaining user trust score is to calculate the node trust value in the social network. The biggest drawback of traditional calculating methods based on formula is the fixed model, thus they are not adapt to different social networks. However, if the results of traditional methods are used as the training feature of the machine learning, the prediction results will not only be more accurate, but also adapt to different social networks.

II. IMPROVED TRUST VALUE CALCULATING ALGORITHM

Traditional formula-based social network node trust value calculation methods have the advantages of simple and low time complexity, however they are not adapt to social networks with different topologies. Thus, it is best to have a specific model used to calculate the node trust value for different network topologies. The machine learning can meet the demand. It can train different models for different network topology, and further calculate and predict using the models in the corresponding network. On the other hand, if the results of some formula-based methods are used as the training feature of the machine learning method, the machine learning method can automatically train the weights. By combining the results of these formula-based methods, the result of the machine learning method will be more accurate.

978-1-5090-5795-5/17/$31.00 ©2017 IEEE

The improved node trust value calculation method of social network node firstly selects the features, then constructs the training set and builds training model, and finally uses the trained model to calculate the node trust value.

A. The Feature Selection

Only valid features can better describe the data and make the prediction model more consistent with the overall data, thus selecting effective features is a key step in machine learning [9]. In this section, the feature selection for social network is introduced.

Suppose the trust value between node x and node y in the social network is to be analyzed. We use $\Gamma(\mathrm{x})$ as the set of neighbor nodes of node x, $\Gamma(\mathrm{y})$ as the set of neighbor nodes of node y, k_x as the number of neighbor nodes of node *x*, k_y as the number of neighbor nodes of node *y*. The features selected in this paper are as below.

Feature 1: The number of neighbor nodes, that is k_x and k_y. It is the degree of the node, and it is the most important attribute of the node.

Feature 2: The sum and difference between the number of neighbor nodes, that is $k_x + k_y$ and $k_x - k_y$. It is the simplest relations among the nodes. Although it can't directly reflect the trust value between the nodes, the auxiliary effect is very obvious when the sum and the difference are used in training model of machine learning.

Feature 3: Number of common neighbor nodes. The definition is $s_{xy}^{CN} = \left|\Gamma(\mathrm{x}) \cap \Gamma(\mathrm{y})\right|$. It is the most obvious measure of node trust value. In general, the larger the number of common neighbor nodes, the more similar the nodes. This is not difficult to understand. Such as if two people know more common friends in the real social network, the two people has the greater possibility to be acquaintances.

Feature 4: Jaccard Index. The definition is $s_{xy}^{Jaccard} = \dfrac{\left|\Gamma(\mathrm{x}) \cap \Gamma(\mathrm{y})\right|}{\left|\Gamma(\mathrm{x}) \cup \Gamma(\mathrm{y})\right|}$. It is one of the node trust value indicator. Actually, it is a way to amend the common neighbor index. In some cases the number of common neighbors may not necessarily represent the node trust value. For example, if node x and node y themselves have a large number of neighbor nodes, then they are not necessarily very similar even if node x and node y have more common neighbors. Thus, the Jaccard index uses the number of common neighbors divided by the number of union of their neighbors to modify the number of common neighbors.

Feature 5: Salton Index. The definition is $s_{xy}^{Salton} = \dfrac{\left|\Gamma(\mathrm{x}) \cap \Gamma(\mathrm{y})\right|}{\sqrt{k_x \times k_y}}$. It is similar to that of cosine trust value for space vector, and is also an effective way to measure the node trust value in social network.

Feature 6: Priority Linking Index. The definition is $s_{xy}^{PA} = k_x \times k_y$. It is widely used to identify the connections function of dynamic network, such as filtering, synchronization and transmission. In particular, this index doesn't need the information of every neighbor node, thus it has the advantage of low computational complexity.

Feature 7: Adamic-Adar Index. The definition is $s_{xy}^{AA} = \sum_{z \in \Gamma(\mathrm{x}) \cap \Gamma(\mathrm{y})} \dfrac{1}{\log k_z}$.

Feature 8: Resource Allocation Index. The definition is $s_{xy}^{AA} = \sum_{z \in \Gamma(\mathrm{x}) \cap \Gamma(\mathrm{y})} \dfrac{1}{\log k_z}$. It is assumed that each node in the social network has one resource to allocate, and the resource is evenly distributed to their neighbors. By calculating the resource allocation index, we can compute the resources allocated by nodes x and y from their common neighbors. In additional, the number of the allocated resources is proportional to the node trust value.

B. Logistic Regression

Logistic regression is a linear regression model which solves the weight of each eigenvalue using the maximum likelihood theory based on the observed values of the sample. For sample x (vector x), a function whose value is between 0 and 1is introduced, and the function value in the logistic regression can represents the probability of x. The function is given as follows:

$$h_\theta(\mathrm{x}) = \mathrm{g}(\theta^T \mathrm{x}) = \frac{1}{1 + e^{-\theta^T x}} \tag{1}$$

where $\mathrm{g}(z) = \dfrac{1}{1 + e^{-z}}$, and the curve of g(z) is shown in Fig. 1.

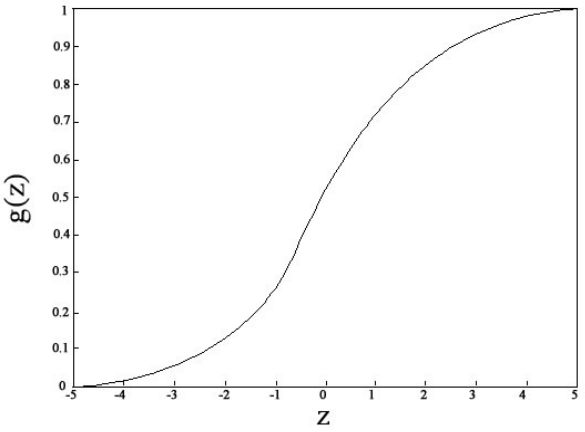

Fig. 1. The curve of g(z).

For all samples, the observed values are obtained, and the weight vector of the features $\theta = \{\theta_1, \theta_2, ..., \theta_n\}$ is solved

using the maximum likelihood estimation and stochastic gradient descent.

C. The Model Training Using Logistic Regression

After selecting the feature, we have to construct the training set. There is a problem to be solved, that is, how to get label information. In this paper, the trust value between the nodes is replaced by the edge between the nodes. That is, if there is edge between the two nodes, the trust value of the two nodes is considered as 1. Otherwise, the trust value of the two nodes is zero. Considering that the number of node pair which haven't edge is much larger than that of node pair which have edge, and the trust value of node pairs which is far away in the social network is relatively small, thus only the node pairs whose distances is less than or equal to 3 are considered.

For each node pair, after obtaining the feature information $f = \{f_1, f_2, ..., f_n\}$ and label information L, we can use these information as a training sample of the training set, and use logistic regression to train the set. For Logistic regression, the model is actually a vector $\theta = \{\theta_1, \theta_2, ..., \theta_n\}$ which represents the weight of the features. For sample (θ, L), the logistic regression function has the following form:

$$h_\theta(f) = g(\theta^T \mathrm{f}) = \mathrm{g}(\theta_1 f_1 + \theta_2 f_2 + ... + \theta_n f_n) = \mathrm{L} \tag{2}$$

where $\mathrm{g}(x) = \frac{1}{1+e^{-x}}$.

There are two points to choose logistic regression as the training algorithm. First, logistic regression is a linear model, and the training speed is fast. Second, the output of logistic regression is between 0 and 1, which can correspond well with node trust value.

The steps for model training with Logistic regression are as follows:

(a) The user social relationship is constructed as a graph structure, and a group of node pairs with distance less than or equal to 3 are sampled from the graph. Then, the feature information of each node pair is calculated. Simultaneously, if there is edge between the node pair, the label information is set to 1, otherwise set to zero. After that, the feature value and label information constitute a training sample, and all the training sample constitute the training set.

(b) The training set is trained using the logistic regression to generate the prediction model.

(c) The prediction model is used to calculate the trust value of all node pairs with distance less than or equal to 3.

For example, the user *u* has three neighbor users denoted by *v*, *m* and *n*, and the distance between user *u* and them are 2, 5, 6 respectively. Base on the above step, only the user *v* is selected from the neighbor users. The prediction model based on the training set is $F(f_u, f_v, L)$, where f_u and f_v are the feature information of user *u* and *v*. Then the trust value between user *u* and *v* can be calculated as follows:

$$sim(u, v) = F(f_u, f_v, L) \tag{3}$$

D. Generate the User Trust Score

After calculating the trust value using logistic regression, a method which is analogous to collaborative filtering recommendation algorithm is used to calculate the user trust score.

Using $\Gamma(\mathrm{u})$ denotes the neighbor user of user *u*, *sim(u,v)* denotes the trust value between the user *u* and its neighbor user *v*, and r_{vi} denotes the score of the user *v* for the item *i*, the score of the user *u* for the item *i* can be calculated by the following formula:

$$r_{ui} = \sum_{v \in \Gamma(\mathrm{u})} sim(\mathrm{u}, \mathrm{v}) r_{vi} \tag{4}$$

III. SIMULATION

A. Experimental Data Set

In this paper, we adopted the data set Track1 of Tencent Weibo as our experimental data set. The data set is as follows:

(a) The basic training data is consisted by 73209277 user's history concerns logs, and every log fomat is {user; recommended stars; concerned or not; timestamp}. These logs include a total of 1392873 users and 4710 stars.

(b) The users' personal information data includes 2320895 users' personal information, and every log format is {user; birth date; sex; number of Weibo; personality label}.

(c) The users' social network data includes 50655143 logs, and every log format is {user; the user who is concerned}.

B. Prediction Accuracy

Suppose there is a sorted recommended list with *m* items. The user may select one, multiple or none of them. Considering the top-*n* recommendations, the average accuracy is as follows:

$$map@\mathrm{n} = \sum_{k=1}^{n} P(K)/c \tag{5}$$

where *c* is the number of user selection in the *m* recommended items, and *P(K)* is the accuracy up to the *k*-th item in the recommended list. In other words, *P(K)* is the number of user selection up to the *k*-th item to the number *k*.

For example, if the number of recommended items is 5, and the user selects #1, #3, #4, then $map@3 = (1/1 + 2/3)/3 \approx 0.56$; if the number of recommended items is 4, and the user selects #1, #2, #4, then $map@3 = (1/1 + 2/2)/3 \approx 0.67$; if the number of recommended items is 3 , and the user selects #1, #3, then $map@3 = (1/1 + 2/3)/2 \approx 0.83$.

The average accuracy for N users can be written as follows:

$$map@\mathrm{n} = \sum_{k=1}^{N} \mathrm{map@n}_i / \mathrm{N} \quad (6)$$

C. Experiments

Dividing the Track1 into training set and testing set, of which 80% is training set and 20% is testing set. The experiment is completed by the testing set.

In the simulation, we compare trust value of machine learning method to that that of traditional method. After constructing the graphic model according to the social network relation of all users and extracting positive and negative samples by sampling, the prediction accuracy of different methods is compared.

D. Results and Analysis

After constructing the graphic model, we extracted 100,000 positive samples and 100,000 negative samples from the nodes whose distance is less than or equal to 3. Then, the node trust value is calculated by different methods. Fig. 2 shows the prediction accuracy of these methods.

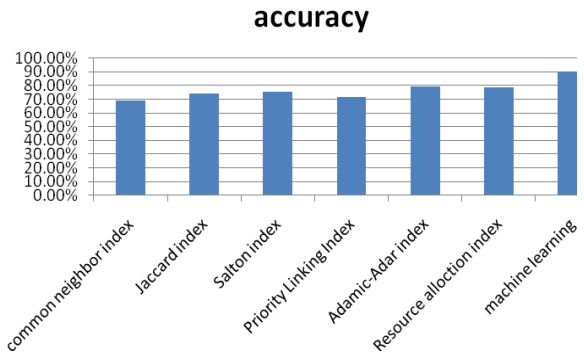

Fig. 2. The prediction accuracy of different methods.

As shown in Fig. 2, the first seven algorithms calculate the node trust value based on traditional formulas and the last computes the node trust value based on machine learning. The training features of our machine learning algorithm include the information of node itself and the calculation results of the above seven algorithms. It is obvious that the prediction accuracy of node trust value computing by machine learning algorithm is significantly higher than that of node trust value computing by formula.

IV. Conclusions

An improved node trust value calculating algorithm for social network nodes is proposed, and the node trust value calculated by this algorithm is used to predict the user trust score. This algorithm uses the logistic regression in machine learning as the training algorithm, and uses some information of node itself and results calculated by the traditional node trust value calculation method as the training features. By using the method of machine learning, the improved algorithm merged the traditional algorithm, and the results are more accurate and stable.

References

[1] D. Bawden, C. Holtham and N. Courtney, “Perspectives on information overload”, Aslib Proceedings, MCB UP Ltd, vol. 51, no. 8, pp. 249-255, 1999.

[2] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, “Recommender systems survey”, Knowl.-Based Syst., vol. 46, pp. 109-132, Jul. 2013.

[3] F. Z. Zhang, S. X. Sun, and H. W. Yi, “Robust collaborative recommendation algorithm based on kernel function and Welsch reweighted M-estimator”, IET Information Security, vol. 9, no. 5, pp. 257-265, 2015.

[4] Z. Jun, and F. Fekri, “On top-N recommendation using implicit user preference propagation over social networks”, IEEE International Conference on Communications (ICC), pp. 3919-3924, 2014.

[5] Z. Liang, et al. “Finding top-k similar users based on Trajectory-Pattern model for personalized service recommendation”, IEEE International Conference on Communications Workshops (ICC), pp. 553-558, 2016.

[6] Hai-xia, Lv, Yu Guang, and Tian Xian-yun. “A matching recommendation algorithm for celebrity endorsement on social network.” International Conference on Management Science and Engineering (ICMSE), pp. 72-77, 2013

[7] Guo L, Ma J, Chen Z, et al, “Learning to recommend with social relation ensemble.” Proceedings of the 21st ACM international conference on Information and knowledge management, pp. 2599-2602, 2012.

[8] Hofmann T, “Collaborative filtering via gaussian probabilistic latent semantic analysis.” Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, pp. 259-266, 2003

[9] LiuH,Yu L. Toward integrating feature selection algorithms for classification and clustering[J]. Knowledge and Data Engineering, vol.17, no.4, pp.491-502, 2005