

Research Article

Identifying Network Public Opinion Leaders Based on Markov Logic Networks

Weizhe Zhang, Xiaoqiang Li, Hui He, and Xing Wang

School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China

Correspondence should be addressed to Weizhe Zhang; wzzhang@hit.edu.cn

Received 20 January 2014; Accepted 26 February 2014; Published 1 April 2014

Academic Editors: T. Cao and F. Yu

Copyright © 2014 Weizhe Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Public opinion emergencies have important effect on social activities. Recognition of special communities like opinion leaders can contribute to a comprehensive understanding of the development trend of public opinion. In this paper, a network opinion leader recognition method based on relational data was put forward, and an opinion leader recognition system integrating public opinion data acquisition module, data characteristic selection, and fusion module as well as opinion leader discovery module based on Markov Logic Networks was designed. The designed opinion leader recognition system not only can overcome the incomplete data acquisition and isolated task of traditional methods, but also can recognize opinion leaders comprehensively with considerations to multiple problems by using the relational model. Experimental results demonstrated that, compared with the traditional methods, the proposed method can provide a more accurate opinion leader recognition and has good noise immunity.

1. Introduction

As the Internet enters into the We-media era, every individual can be a message sender. However, public opinion emergencies may affect the social activities significantly due to the mixed netizen qualities. To avoid adverse effect, it is necessary to have a comprehensive understanding of the development trend of public opinion and recognize special communities like opinion leaders.

The earliest domestic and foreign researches on the discovery of network opinion leaders focused on using the opinion leader theory in the traditional social sciences and research method transplantation directly to recognize the internet leaders but failed to achieve ideal results. They often determined community opinion leaders based on the quantitative data analysis. These methods [1], regardless of the characteristics of objective fact, only depend on logic reasoning and could not represent the public opinion transmission characteristics on the new media of network. Recently, scholars began to explore the difference between network environment and offline practical society through quantitative measurement. The accuracy of the leader discovery model [2–5] witnessed a continuous improvement from the clustering analysis of statistical properties [6] to social

network analysis based on complex network [7–9] and to the semantic analysis of text content.

1.1. Attributes of Public Opinion Participants. Behaviors of every netizen involved in the public opinion transmission are described by inherent attribute, content attribute, and social network attribute.

Inherent attribute refers to the independent attributes of participants from the concerning public opinion events such as career, position, internet age, logins, community credits, fans population, and concerns.

Content attribute refers to the behavioral description of the participants in a certain public opinion event including posts, replies, comments received, reposts, number of mentions, number of words, and emotional tendency.

Social network attribute refers to the mutual relationship of participants in the network mainly including fans and concerns of the participants.

1.2. Shortcomings of Existing Research Methods. Existing network opinion leader discovery is based on the recognition model involving only one or two attributes of participants. No network opinion leader recognition involving all

three attributes of participants has been reported yet. This will affect the accuracy of the opinion leader recognition method.

Existing network opinion leader discovery views the attributes of participants as independent and identically distributed (IID) data. In the theory of probability statistics, mutually independent sequence of variables or other random variables that have the same probability distribution is called IID. However, attribute data of participants are relational data. Different attributes of participants are mutually correlated instead of being independent from each other. For example, fans' population of inherent attribute often is proportional to comments received of content attribute; participants enjoying high attentions from opinion leader are more likely to be the opinion leader. The incomprehensive understanding of such relations will make some opinion leaders remain unidentified. Furthermore, existing network opinion leader discovery has no modeling solution to the relational data.

As a result, simultaneous application of all three attributes and the exploration of relationships in public opinion data can improve the performance of network opinion leader recognition method.

2. Markov Logic Networks

Markov Logic Networks refer to a learning method of statistical relation that is combining the Markov network and first-order logic together. It was proposed by Singla and Domingos [10] in 2004 and then improved by Domingos and his students.

Markov network [11] is a joint distribution model of random variable set ($X = (x_1, x_2, \dots, x_n)$). It is composed of an undirected graph (G) and a potential function (Φ_k) set. Every random variable occupies a node of the graph and each group has a potential function in the model. Potential function is a nonnegative real function, which represents the state of the corresponding group. The joint distribution of Markov network is as follows:

$$P(X = x) = \frac{1}{Z} \prod_k \Phi_k(x_{\{k\}}), \quad (1)$$

where $x_{\{k\}}$ is the state of random variables in the group and Z is partition function (state sum) that is defined as $\sum_{x \in X} \prod_k \Phi_k(x_{\{k\}})$. Weight all characteristic values of potential-use states of each group in the Markov network. Then sum them and calculate the exponentiation. Finally, a log-linear model can be gained as follows:

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_j \omega_j f_j(x)\right). \quad (2)$$

Characteristic function can be any real function of state. In this paper, characteristic function refers to the dual characteristics value ($\{0, 1\}$). Equation (1) is the most direct expression of potentials, in which every possible state of each group has a corresponding characteristic value and a weight. Equation (1) is related to the power of groups.

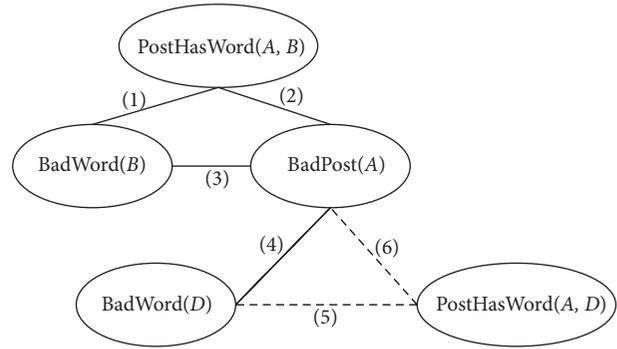


FIGURE 1: Closed Markov Logic Network.

However, the amount of characteristic values can be reduced freely by some methods (e.g., logic function of state), thus enabling the characteristic function to provide a simpler expression under large number of groups compared with the potential functions. This is the principle of the Markov Logic Networks.

Markov Logic Networks are a first-order logic knowledge base where every code has a weight. This first-order logic knowledge base can be viewed as the template of Markov Logic Networks. Viewed from probability, the Markov Logic Networks provide a simple language to define large Markov network as well as a flexible and modularized integration with abundant knowledge. Viewed from the first-order logic, the Markov Logic Networks provide a sound processing to knowledge base with uncertainties, defects, and even contradictions, thus decreasing the vulnerability.

Take the data set of Skyline for example. The simplest situation is as follows: suppose the knowledge base only contains the formula F_1 (weight = 1.5), and the corresponding MLN of the knowledge base is $\{(F_1, 1.5)\}$:

$$F_1 : \text{PostHasWord}(y, x) \rightarrow (\text{BadWord}(x) \rightarrow \text{BadPost}(y)), \quad (3)$$

where x and y are individual variables; $\text{BadWord}(x)$, $\text{BadPost}(y)$, and $\text{PostHasWord}(y, x)$ are predicates, representing whether x has bad word, y is a bad post, and y contains x . F_1 means that if y has the bad word x , y can be deduced as a bad post.

Given an individual constant set $C = \{A, B, D\}$, a closed Markov Logic Network can be generated (Figure 1).

2.1. Markov Logical Reasoning Method. Markov logical reasoning is equal to the probabilistic reasoning of the complex relationship. The basic task of reasoning is to inference the most possible state of world y according to the given evidence x (word set).

There are two basic types of reasoning: First, we search the most possible state satisfying some evidence and probability of computer random condition. Next, lazy reasoning and relieving reasoning are compared in improving the performance of above two types of reasoning in processing more complicated relationships. The lazy reasoning only requires

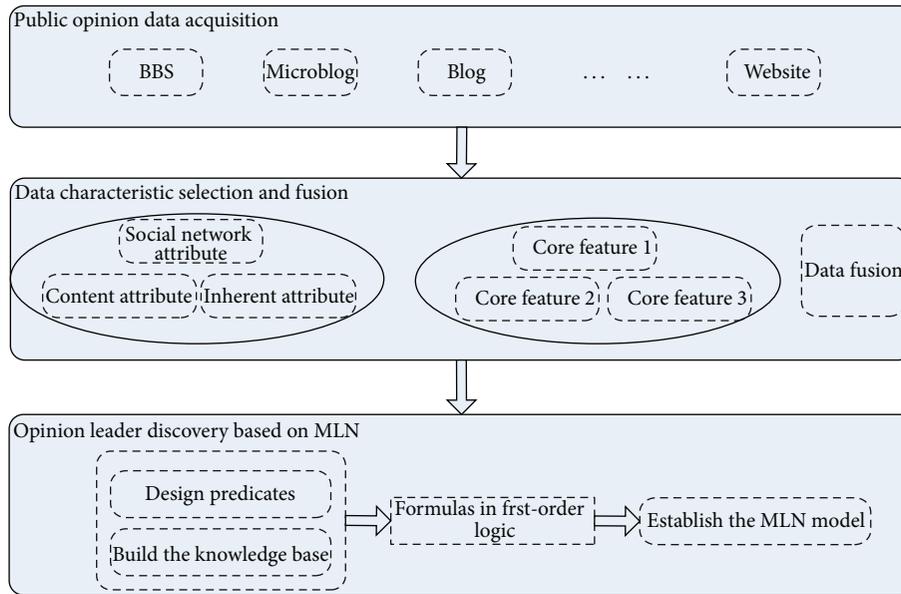


FIGURE 2: Overall structure of the network opinion leader recognition system based on the Markov Logic Networks.

adopting instantiation “default” value to the deviated basic values, while relieving reasoning divides indistinctive atoms into one group and views them as an independent unit.

2.2. Markov Logic Networks Learning. The Markov Logic Networks learning includes structure learning and parameter learning. Structure learning is to learn the model structure (network structure of the Markov Logic Networks) from data. In structure learning, it is more difficult to learn the rules. There are two structure learning methods based on the inductive logic programming (ILP): (1) learn model structure by using CLAUDIEN (an LP system) directly; (2) learn structure of Markov Logic Networks from the relational database by combining LP and feature introduction in Markov network together. Parameter learning is committed to find the satisfying rule weight with maximum likelihood or maximum conditional likelihood of associated database. Parameter learning includes production parameter learning and discriminant parameter learning. Parameter learning is needed under known structure of Markov network to acquire weight of each rule. However, when the structure of Markov network is unknown, structure learning is needed before the parameter learning to get the network structure.

Markov Logic Networks learning includes parameter learning and structure learning. Parameter learning has production parameter learning and discriminant parameter learning as well as their corresponding formulas and approximate algorithms. Structure learning also includes top-down structure learning and bottom-up structure learning.

3. System Framework

The overall structure of the network opinion leader recognition system based on the Markov Logic Networks is shown in Figure 2.

The designed network opinion leader recognition system includes three modules: public opinion data acquisition module, data characteristic selection and fusion module, and opinion leader discovery module based on Markov Logic Networks. The public opinion data acquisition module is for data collection concerning specific public opinion event. Data characteristic selection and fusion module is for processing and analyzing collected data to disclose the relationship between core characteristics and attributes. The opinion leader discovery module based on Markov Logic Networks is to design predicates, build the knowledge base, and establish the MLN model according to the relationship between core characteristics and attributes.

3.1. Opinion Leader Discovery Module Based on Markov Logic Networks. The technical route of opinion leader recognition based on Markov Logic Networks is presented in Figure 3. The top data input, initialization module, and structure learning module are the training module of the model, whereas the bottom data (Leader) verification, verification module, and data output are the verification module of the model.

The primary task of the training module is to design predicate. Predicate design has two stages: (1) original design of initial predicate set according to the characteristic matrix and relational knowledge gained by the data characteristic extraction and fusion module as well as personal priori knowledge; (2) repetitive reasonable adjustment of predicate design according to the learning results of the model and follow-up experimental analysis of learning results until getting satisfying experimental result.

The initialization module will convert contents in the corpus into DB according to existing predicate design. The structure learning module can conduct the structure learning through the available learnstruct program of

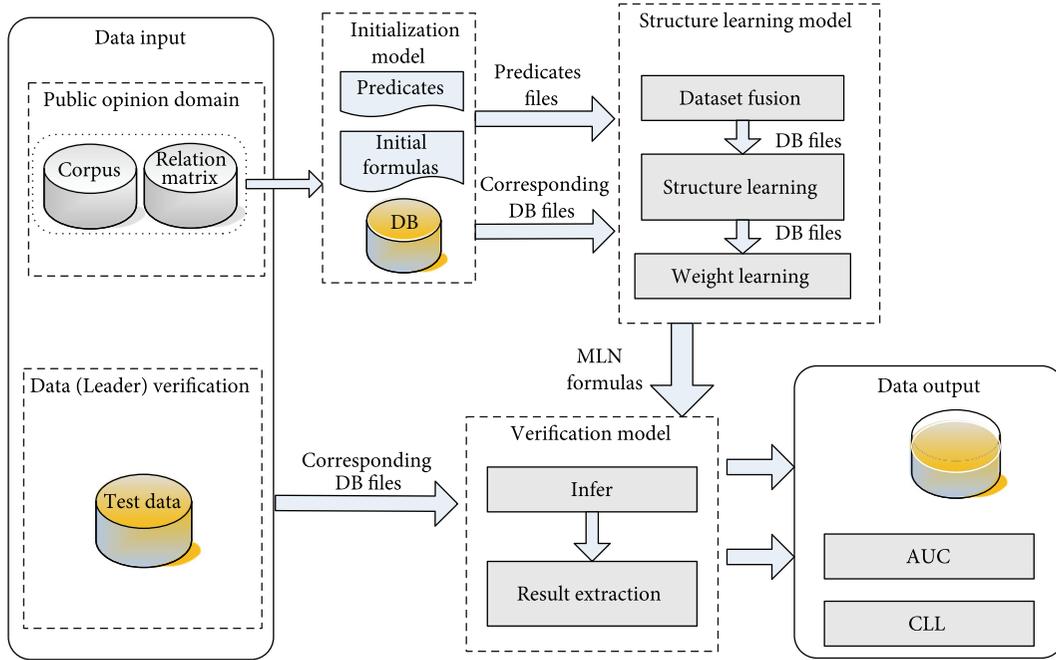


FIGURE 3: Technical route of opinion leader recognition based on Markov Logic Networks.

Alchemy. Beam search is the default structure learning algorithm. Weight learning can be implemented by the available learning program. Weighted MLN clause can be gained through the structure learning and weight learning, which will be used for reasoning. In the designed system, these weighted MLN clauses are used to deduce user's identity.

In the verification module of the model, both verification data and training data are converted into DB according to the predicate design. The verification module is mainly used to deduce user's identity.

Reasoning results contain all possible predicates and their possibilities. Take the predicate of teachby(course, teacher) for instance. Course represents course and teacher represents teacher, meaning that the course is taught by the teacher. If the course is Chinese, possible teacher set is {Tom, Jack, Lily}. Then, the reasoning results may be as follows:

$$\begin{aligned}
 &\text{teachby}(\text{Chinese}, \text{Tom}) \quad 0.9275 \\
 &\text{teachby}(\text{Chinese}, \text{Jack}) \quad 0.7465 \\
 &\text{teachby}(\text{Chinese}, \text{Lily}) \quad 0.3556.
 \end{aligned}
 \tag{4}$$

Result extraction means to select the reasoning result with highest probability as the final result. In the above case, Tom will be selected as the teacher of Chinese. The data output module will calculate AUC and CLL.

4. Experimental Verification

4.1. Recognition of Nonrelational Data Model. Netizens were classified according to the nonrelational data model provided by Weka. First of all, the original data have to be converted

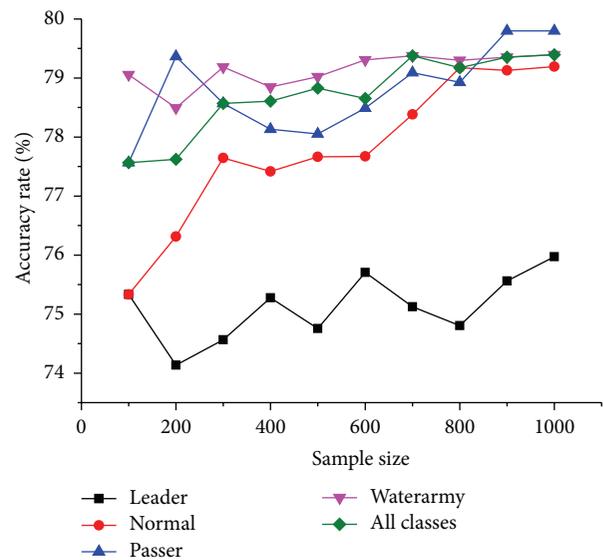


FIGURE 4: Group recognition of nonrelational data model.

into ARFF. ARFF mainly includes attribute assertion and data [12, 13].

We chose SVM to classify netizens involved in the “Xu-Ting Event” on the legal forum of Skyline Gossip as Leader, Normal, Passer, and Waterarmy. In our experiment, every netizen was recognized independently. Experimental results are presented in Figure 4.

It can be seen from Figure 4 that the nonrelational data model achieved a recognition accuracy of about 77%, which presented a continuous growth as the sample size increases.

TABLE 1: Designed predicates of content attribute.

Predicate	Meaning
Post(<i>person_id</i> , <i>post_id</i>)	User who is represented by <i>person_id</i> published a post which is represented by <i>post_id</i> .
ReplyNumOfPost(<i>post_id</i> , <i>level_replynumofpost</i>)	The reply number of the post which is represented by <i>post_id</i> is <i>level_replynumofpost</i> .
ClickNumOfPost(<i>post_id</i> , <i>level_clicknumofpost</i>)	The click number of the post which is represented by <i>post_id</i> is <i>level_clicknumofpost</i> .
TotalPostNum(<i>person_id</i> , <i>level_totalpostnum</i>)	The post number of the user who is represented by <i>person_id</i> is <i>level_totalpostnum</i> .
TotalReplyNum(<i>person_id</i> , <i>level_totalreplynum</i>)	The reply number of the user who is represented by <i>person_id</i> is <i>level_totalreplynum</i> .
TotalBeReplyNum(<i>person_id</i> , <i>level_totalbereplynum</i>)	The number of replies to the user who is represented by <i>person_id</i> is <i>level_totalbereplynum</i> .
Correlation(<i>person_id</i> , <i>post_id</i> , <i>level_correlation</i>)	The correlation level between the user who is represented by <i>person_id</i> published content in the post which is represented by <i>post_id</i> and post topics is <i>level_correlation</i> .
Sentiment(<i>person_id</i> , <i>level_sentiment</i>)	The degree of the emotional tendencies bases on the content published by the user who is represented by <i>person_id</i> is <i>level_sentiment</i> .

TABLE 2: Designed predicates of social network attribute.

Predicates	Meaning
FansNum(<i>person_id</i> , <i>level_fansnum</i>)	The fans number of the user who is represented by <i>person_id</i> is <i>level_fansnum</i> .
FollowNum(<i>person_id</i> , <i>level_follownum</i>)	The follow number of the user who is represented by <i>person_id</i> is <i>level_follownum</i> .
Follow(<i>person_id</i> , <i>person_id</i>)	A user who is represented by the first <i>person_id</i> follows another user who is represented by the second <i>person_id</i> .
Reply(<i>person_id</i> , <i>person_id</i> , <i>post_id</i> , <i>num</i>)	In the post which is represented by the <i>post_id</i> , the number of a user who is represented by the first <i>person_id</i> reply to another user who is represented by the second <i>person_id</i> is <i>num</i> .

TABLE 3: Designed predicates of inherent attribute.

Predicate	Meaning
Gender(<i>person_id</i> , <i>gender</i>)	The gender of the user who is represented by <i>person_id</i> is <i>gender</i> .
Age(<i>person_id</i> , <i>level_age</i>)	The age of the user who is represented by <i>person_id</i> is <i>level_age</i> .
NetworkAge(<i>people</i> , <i>level_networkage</i>)	The network age of the user who is represented by <i>person_id</i> is <i>level_networkage</i> .
LogNum(<i>person_id</i> , <i>level_lognum</i>)	The login number of the user who is represented by <i>person_id</i> is <i>level_lognum</i> .
CommunityCredits(<i>person_id</i> , <i>level_communitycredits</i>)	The community credits of the user who is represented by <i>person_id</i> is <i>level_communitycredits</i> .
HasPosition(<i>person_id</i>)	The user who is represented by <i>person_id</i> has a communities position.
Role(<i>person_id</i> , <i>role</i>)	The role of the user who is represented by <i>person_id</i> is <i>role</i> .

4.2. Recognition of Relational Data Model. Firstly, we have to design predicates. Our designed predicates according to the characteristic selection and personal priori knowledge are listed in Tables 1, 2, and 3, including three classes (social network attribute, content attribute, and inherent

attribute). These three classes of attribute are used to describe different network behaviors and individual characteristics of netizens.

After the predicate design, we have to convert original data into DB.

```

follow(person_id, person_id)
reply(person_id, person_id)
post(person_id, post_id)
replynumofpost(post_id, level_replynumofpost)
clicknumofpost(post_id, level_clicknumofpost)
totalpostnum(person_id, level_totalpostnum)
totalreplynum(person_id, level_totalreplynum)
totalbereplynum(person_id, level_totalbereplynum)
act(person_id, role)
repeat(person_id, level_repeatnum)

```

Box 1

```

10.4475    totalbereplynum(a1,a2)
10.7128    clicknumofpost(a1,a2)
10.4475    totalreplynum(a1,a2)
-6.70724   replynumofpost(a1,a2) v !replynumofpost(a1,a3)
5.77628    !repeat(a1,a2) v !repeat(a1,a3)
5.77628    !totalbereplynum(a1,a2) v !totalbereplynum(a1,a3)
5.83318    !totalpostnum(a1,a2) v !totalpostnum(a1,a3)
5.77628    !totalreplynum(a1,a2) v !totalreplynum(a1,a3)
5.83318    !clicknumofpost(a1,a2) v !clicknumofpost(a1,a3)
4.7222     !post(a1,a2) v !post(a3,a2)
5.83318    !act(a1,a2) v !act(a1,a3)

```

Box 2

TABLE 4: Valuable clauses learned from the “Xu-Ting Event”.

Weight	Formula
2.76133	!fansnum(a1,Level_fansnum_10To49) v !follownum(a1,Level_follownum_Lessthan10) v gender(a1,Male) v !networkage(a1,Level_networkage) v !lognum(a1,Level_log_num_1000To4999) v !communitycredits(a1,Level_communitycredits_1000To9999)
2.63516	gender(a1,Female) v !lognum(a1,Level_log_num_1000To4999)
3.21897	!fansnum(a1,Level_fansnum_10To49) v !follownum(a1,Level_follownum_Lessthan10) v age(a1,Level_realage_Morethan35) v !networkage(a1,Level_networkage) v !lognum(a1,Level_log_num_1000To4999) v !communitycredits(a1,Level_communitycredits_1000To9999)
3.77246	gender(a1,Female) v !lognum(a1,Level_log_num_Lessthan1000)
4.27395	!reply(a1,a2) v act(a2,Leader) v !act(a1,Leader)
5.65332	gender(a1,a2) v !age(a1,a3) v !age(a1,a4) v lognum(a1,a5) v lognum(a1,a6)
6.06442	!communitycredits(a1,a2) v !communitycredits(a1,a3) v !totalreplynum(a1,a4) v !totalreplynum(a1,a5)
6.06101	!networkage(a1,a2) v !networkage(a1,a3)

```

//Waterarmy
repeat(a1,+a2)=>act(a1,+a3)
//Leader
totalreplynum(a1,+a2)^totalpostnum(a1,+a3)^totalbereplynum(a1,+a4)=>act(a1,+a5)
post(a1,a2)^replynumofpost(a2,+a3)^clicknumofpost(a2,+a4)=>act(a1,+a5)
//Try to take advantage of relational data
act(a1,+a2)^follow(a1,a3)=>act(a3,+a4)
act(a1,+a2)^reply(a1,a3)=>act(a3,+a4)
//Passer
//totalreplynum(a1,+a2)^totalpostnum(a1,+a3)^totalbereplynum(a1,+a4)=>act(a1,+a5)

```

Box 3

```

5.8819 act(a1,Waterarmy) v !repeat(a1,Level_repeatnum_20To49)
5.28195 act(a1,Waterarmy) v !repeat(a1,Level_repeatnum_Morethan50)
4.26896 !totalpostnum(a1,Level_totalpostnum_2To2)
      v !totalreplynum(a1,Level_totalreplynum_10To49)
      v !totalbereplynum(a1,Level_totalbereplynum_2To9)
      v act(a1,Normal)
4.93641 !totalpostnum(a1,Level_totalpostnum_Lessthan2)
      v !totalreplynum(a1,Level_totalreplynum_Lessthan2)
      v !totalbereplynum(a1,Level_totalbereplynum_Lessthan2)
      v act(a1,Passer)
3.76921 !totalpostnum(a1,Level_totalpostnum_Lessthan2)
      v !totalreplynum(a1,Level_totalreplynum_Lessthan2)
      v !totalbereplynum(a1,Level_totalbereplynum_Morethan200)
      v act(a1,Leader)
3.74401 !reply(a1,a2) v act(a2,Normal) v !act(a1,Passer)
4.34903 !reply(a1,a2) v act(a2,Leader) v !act(a1,Leader)
    
```

Box 4

act(Person15,Waterarmy) 0.505	act(Person15,Normal) 0.813969
act(Person15,Leader) 0.495	act(Person15,Passer) 0.99995
act(Person30,Waterarmy) 0.99995	act(Person30,Normal) 0.99995
act(Person30,Leader) 0.99995	act(Person30,Passer) 0.894961
act(Person21,Waterarmy) 0.495	act(Person21,Normal) 0.823968
act(Person21,Leader) 0.504	act(Person21,Passer) 0.99995
act(Person55,Waterarmy) 0.99995	act(Person55,Normal) 0.99995
act(Person55,Leader) 0.99995	act(Person55,Passer) 0.893961
act(Person51,Waterarmy) 0.490001	act(Person51,Normal) 0.794971
act(Person51,Leader) 0.513999	act(Person51,Passer) 0.99995

Box 5

Next, we have to implement structure learning and weight learning. The input MLN file (predicate statements) for structure learning is shown as Box 1.

And the structure learning results are shown as Box 2.

The input MLN file (predicate statements and design statements) for weight learning is shown as Box 3.

The design statements are recognition results of four groups.

The weight learning results are represented as Box 4.

Table 4 lists some learned valuable clauses such as the following:

```

4.27395 !reply(a1,a2) v act(a2,Leader) v
!act(a1,Leader).
    
```

This clause means that if a1 is the leader and a1 replies a2, then a2 is a leader too. This is true in real life.

The learned valuable clauses were selected for reasoning of the test set. According to the experimental results (shown as Box 5), Person15, Person30, Person21, Person55, and Person51 were identified as Passer, Normal, Passer, Leader, and Passer, respectively. Four were recognized accurately and 1 was recognized wrongly.

The recognition accuracy comparison results of relational data model and nonrelational data model to different events

are listed in Table 5, finding that the relational data model has higher recognition accuracy compared with the nonrelational data model.

5. Conclusions

This paper firstly summarizes and evaluates the shortcomings of existing opinion leader recognition method, describes the advantages of Markov Logic Networks in opinion leader recognition, and summarizes the associated theories of Markov Logic Networks including basic concepts as well as theoretical models (reasoning and learning). The Markov Logic Networks combine the probability theory and first-order logic perfectly, integrating logic/relation expressions, uncertainty processing, and learning. Secondly, this paper designs and implements a network opinion leader recognition system based on previous theories. This designed system firstly collects some public opinion data as the training set for structure learning of Markov Logic Networks and then uses the learning results to reasoning the control results of corresponding public opinion domain of the test data. The experimental results are compared and analyzed to evaluate their validity. Thirdly, this paper carries out an experimental verification, which verifies

TABLE 5: Recognition accuracy comparison of relational data model and non-relational data model.

Event name	Forum	The accuracy of IID model (%)	The accuracy of relation model (%)
Xu-Ting Event	Legal Forum	79.5	82.5
Xu-Ting Event	Tianya By-talk	77.4	80.6
Three years of great Chinese famine	Discussion about the history	77.8	81.8
Three years of great Chinese famine	Tianya By-talk	76.9	80.8

the superiority of the designed network opinion leader recognition system.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

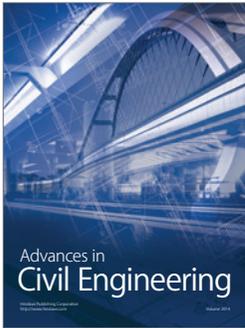
This work was supported by the National Natural Science Foundation of China (NSFC) under Grant no. 61173145, the National Basic Research Program of China under Grant no. G2011CB302605, and the National High Technology Research and Development Program of China under Grant no. 2011AA010705.

References

- [1] Y. Li, S. Ma, Y. Zhang, R. Huang, and Kinshuk, "An improved mix framework for opinion leader identification in online learning communities," *Knowledge-Based Systems*, vol. 43, pp. 43–51, 2013.
- [2] R. Goldsborough, "The influence of active online users," *Black Issues in Higher Education*, vol. 19, no. 5, pp. 30–31, 2002.
- [3] N. Matsumura, Y. Ohsawa, and M. Ishizuka, "Mining and characterizing opinion leaders from threaded online discussions," in *Proceedings of the 6th International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies*, 2002.
- [4] N. Matsumura, "Topic diffusion in a community," in *Chance Discovery*, pp. 84–97, Springer, Berlin, Germany, 2003.
- [5] N. Matsumura, "Collaborative communication strategies in online community," in *The 4th International Workshop on Social Intelligence Design (SID '05)*, 2005.
- [6] Z. Zhai, H. Xu, and P. Jia, "Identifying opinion leaders in BBS," in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT '08)*, vol. 3, pp. 398–401, IEEE, December 2008.
- [7] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social Networks*, vol. 1, no. 3, pp. 215–239, 1978.
- [8] L. Page, S. Brin, R. Motwani, and T. Winograd, *The PageRank Citation Ranking: Bringing Order to the Web*, 1999.
- [9] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM*, vol. 46, no. 5, pp. 604–632, 1999.
- [10] P. Singla and P. Domingos, "Markov logic in infinite domains," In press, <http://arxiv.org/abs/1206.5292>.
- [11] M. Richardson and P. Domingos, "Markov logic networks," *Machine Learning*, vol. 62, no. 1-2, pp. 107–136, 2006.

[12] W. Zhang, H. He, and J. Ye, "A two-level cache for distributed information retrieval in search engines," *The Scientific World Journal*, vol. 2013, Article ID 596724, 6 pages, 2013.

[13] W. Zhang, B. Wang, H. He, and Z. Tan, "Public opinion leader community mining based on the heterogeneous network," *Acta Electronica Sinica*, vol. 40, no. 10, pp. 1927–1932, 2012.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

