DOI 10.1007/s13278-011-0017-9

REVIEW ARTICLE

Identification of leaders, lurkers, associates and spammers in a social network: context-dependent and context-independent approaches

Mohamed Fazeen · Ram Dantu · Parthasarathy Guturu

Received: 4 August 2010 / Accepted: 6 January 2011 / Published online: 15 February 2011

Abstract In this paper, we present two methods for classification of different social network actors (individuals or organizations) such as leaders (e.g., news groups), lurkers, spammers and close associates. The first method is a two-stage process with a fuzzy-set theoretic (FST) approach to evaluation of the strengths of network links (or equivalently, actor-actor relationships) followed by a simple linear classifier to separate the actor classes. Since this method uses a lot of contextual information including actor profiles, actor-actor tweet and reply frequencies, it may be termed as a context-dependent approach. To handle the situation of limited availability of actor data for learning network link strengths, we also present a second method that performs actor classification by matching their short-term (say, roughly 25 days) tweet patterns with the generic tweet patterns of the prototype actors of different classes. Since little contextual information is used here, this can be called a context-independent approach. Our experimentation with over 500 randomly sampled records from a twitter database consists of 441,234 actors, 2,045,804 links, 6,481,900 tweets, and 2,312,927 total reply messages indicates that, in the context-independent analysis, a multilayer perceptron outperforms on both on classification accuracy and a new F-measure for classification performance, the Bayes classifier and Random Forest classifiers. However, as expected, the context-dependent analysis using link strengths evaluated using the FST approach in conjunction with some actor information reveals strong clustering of actor data based on their types, and hence can be considered as a superior approach when data available for training the system is abundant.

Keywords Context dependent and context independent data analysis · Social networks · Twitter · Fuzzy logic · Naive Bayesian classifier · MLP · Random Forest

1 Introduction

Recent years have witnessed a proliferation of social networks with popular applications such as Twitter, Flickr, YouTube, LiveJournal, Orkut, and Facebook. These networks are poised for further growth with emerging applications such as social television (TV) (Klym and Montpetit 2008) to share our thoughts with family and friends while watching TV alone at home. With this rapid pace of growth in social networks (SN), there has also been a growing interest in the Internet research community in the SN analysis to address various aspects of social networking. Wu and Zhou (2009) perform SN analysis using Del.icio.us, a free SN bookmarking web service that permits actors to tag each one of their bookmarks with freely chosen index items. They have shown that the patterns of actors' tagging can be detected by visualizing the actors' tagging behaviors and tag's evolution. They also established that the actors within a

M. Fazeen
Department of Computer Science and Engineering, College of Engineering, University of North Texas, P.O. Box 311366, 1155 Union Circle, Denton, TX 76203, USA
e-mail: MohamedFazeen@my.unt.edu

R. Dantu (✉)
Department of Computer Science and Engineering, College of Engineering, University of North Texas, P.O. Box 311366, Denton, TX 76203, USA
e-mail: rdantu@unt.edu

P. Guturu
Electrical Engineering, College of Engineering, University of North Texas, P.O. Box 310470, Denton, TX 76203-0470, USA
e-mail: guturu@unt.edu

subscription network share more common interests than random pairs of actors in Del.icio.us. Using a large dataset containing 11.3 million actors and 328 million links from multiple online SNs namely Flickr, YouTube, LiveJournal, and Orkut (Mislove et al. 2007) develop a large-scale measurement procedure to analyze the structure of multiple online social networks, and find significant structural differences between the SNs and previously studied networks, particularly the Web. SNs have a much higher fraction of symmetric links and exhibit much higher levels of local clustering. These properties may be used to design effective SN algorithms and applications. For an analytical study on information dissemination in large-scale SNs (Cha et al. 2009) use the data including 11 million photographs from favorite markings of 2.5 million actors on the Flickr to address the following questions: (a) how widely does information propagate in the SN? (b) How quickly does information propagate? (c) What is the role of word-of-mouth exchanges between friends in the overall propagation of information in the network? Results of their analysis indicate that: (a) even in case of popular photographs, information does not spread widely throughout the network, (b) it spreads slowly, and (c) information exchange between friends accounts for over 50% of all favorite markings, and it incurs a significant delay at each hop.

Since the current work is on SN analysis, it is better presented in the context of relevant literature. Hence, we present in Sect. 2 a survey of relevant SN literature with particular emphasis on the past work on ‘social’ dimensions of social networks, the nature of social exchange between humans, measurements of ‘relationships’ (including works on the ‘tie’ strength and prestige measurement), evolution over time of network structures of exchange, and the limitations inherent in large scale SN datasets. Organization of the rest of the paper is as follows. Section 3 presents our approach to context-dependent classification of twitter actors. In Sect. 4, we outline a context-independent classification approach to handle situations where sufficient information regarding twitter actors under consideration is lacking. Details of experimentation and results are presented in Sect. 5. Finally, summary and conclusions are presented in Sect. 6.

2 Related work

SN analysis can be categorized into two groups: (a) qualitative analysis, and (b) quantitative analysis. The early works by Berkowitz (1982) and Scott (1992) on qualitative SN analysis focus on patterns of relations among people, states, organizations, etc. (Garton et al. 1997) define ties as multistranded relationships in an SN, present two views—egocentric and whole network views—of the SNs, and discuss various SN issues such as network characteristics, partitioning, hierarchical organization, and positional analysis. They also discuss how to collect and analyze SN data, but this work may still be classified as a qualitative analysis, because no results of explicit data analysis are presented. In their seminal work on SN ties, McPherson et al. (2001) explore the concept of homophily or the bondage between similar types of people, and present a study on how homophily structures SN ties of every type including marriage, friendship, co-membership, work, information exchange, etc., and limits people’s social worlds with respect the information exchange, mutual interactions, and formation of attitudes. They discuss various sociodemographic dimensions of homophily (e.g. race, ethnicity, sex, age, religion, education, etc.), and indicate using prior research by others that education, occupational prestige, and social classes are roughly at the same level of homophily as religion and sex. They argue for more research on (a) ecological processes for the basic ecological processes linking organizations, associations, cultural groups and many other social forms, (b) the impact of multiple relationships on homophily, and (c) the dynamics of network change over time as the SNs and the social entities co-evolve. Watts et al. (2002), in their influential paper in science, present an SN model based upon plausible social structures, and explain how the quick search of target SN entities can be achieved because of hierarchical partitioning of SNs due to the network ties and identities of individuals in an SN. The identities, as per their definition, are the sets of characteristics attributed to the individuals by themselves and others by virtue of their association in an SN.

The block-modeling method for quantitative SN analysis that inductively uncovers underlying role structures such as similarity of positions in an organization, co-membership of a community or association, etc. by juxtaposing multiple indicators of relationships in analytic matrices was introduced around 35 years ago by White et al. (1976). But it was well developed more recently by Wasserman and Faust (1994). In a later work, Wasserman and Pattison (1996) propose a large class of generalized stochastic block models for investigation of the structure of SNs, and show how approximate model fits are obtained using the estimation strategy of Strauss and Ikeda (1990).

In one of the earlier works on “tie” measurement for quantitative SN analysis, Marsden and Campbell (1984) apply multiple indicator techniques to construct and validate measures of tie strength. They consider two distinct aspects of tie strength (a) time spent in a relationship, and (b) depth of the relationship. They find that a measure of “closeness” or intensity is the best indicator of strength, but they have difficulty in using frequency and duration of contact as indicators of strength. In a recent work, Onnela et al. (2007) propose methods for estimation of structure and tie strengths in mobile social networks.

In a work on SN content analysis, Mathioudakis et al. (2010) employ ISIS, a general stochastic model (with a set of sequential statistical tests) for Interacting Streaming Information Sources, to identify items that gather a higher attention in social media. In a similar application context, Cheong and Lee (2009) identify messages dealing with the trending topics or special events in an SN using visualization techniques and artificial intelligence-based data mining methods.

Evolution of social networks has gained the attention of many SN researchers recently. In their book, Dorein and Sockman (1997) consolidate the research on SN dynamics and evolution scattered over different journals and books. They present four simulational studies, three empirical studies, and two statistical discussions for evolution of SN structure, which is defined as a set of social actors with a social relationship among them. In a recent paper, Kossinets and Watts (2006) present an empirical study of an evolving social network comprising 43,553 students, faculty, and staff at a large university. They indicate that, in the absence of global perturbations, average network properties approach an equilibrium state, whereas individual properties are unstable. In the latest work on SN evolution, Kumar et al. (2010) consider the evolution of structure in large scale online social networks using a series of measurements on two real networks–one with the friend relation within the Flickr photo sharing application and the other from Yahoo!'s 360 social network.

In the context of new challenges in this field related to privacy, background knowledge, and data utility, many researchers address the anonymization (actor identity suppression) problem. Zhou et al. (2009) present a short but systematic review of the existing anonymization techniques for privacy preserving publishing of social network data. Considering trust between actors in a social network as a parameter similar to the reputation of a specified actor rather than a quantification of preference and profile matches between actors, Kim and Han (2009) propose a fuzzy logic-based system to compute the trust values for individual actors in an SN by propagation and aggregation through the network the trust values provided on a scale of 0–10 by the actors about the other directly connected (hence well known) actors. Lin et al. (2007) address the problem of detection of spam among blogs, the SN media similar to a micro blogger like Twitter, but with more capability. The authors basically identify the spammers from the repetitive temporal regularity of contents and consistent linking patterns. The temporal regularity, in turn, is measured using the entropy of the "blog post time" difference distribution. Experimental results indicate that a high accuracy of 90% in spam blog detection can be achieved by their method. Weng et al. (2010) address the problem of identification of influential actors in twitter using an improved page-ranking system called Twitter-Rank based on the concept of homophily, the tendency of individuals to associate and link with similar others, or those with similar topics of interests.

Finally, the problem of large scale data sets in classification of SN actors is actually an inherent problem associated with the pattern recognition (PR) methods. If the dimensionality of the pattern vectors is large, the PR system suffers from what is called by Bellman (1957) as curse of dimensionality. In this case, the pattern samples available for training the system turn out to be very sparse in a higher dimensional space, and hence become insufficient for accurate training of the classifiers. Thus, higher the pattern vector dimensionality necessitates much higher availability of training samples. Large datasets are not only difficult to procure but also pose computational problems and consequently may need some sophisticated parallel and distributed processing algorithms.

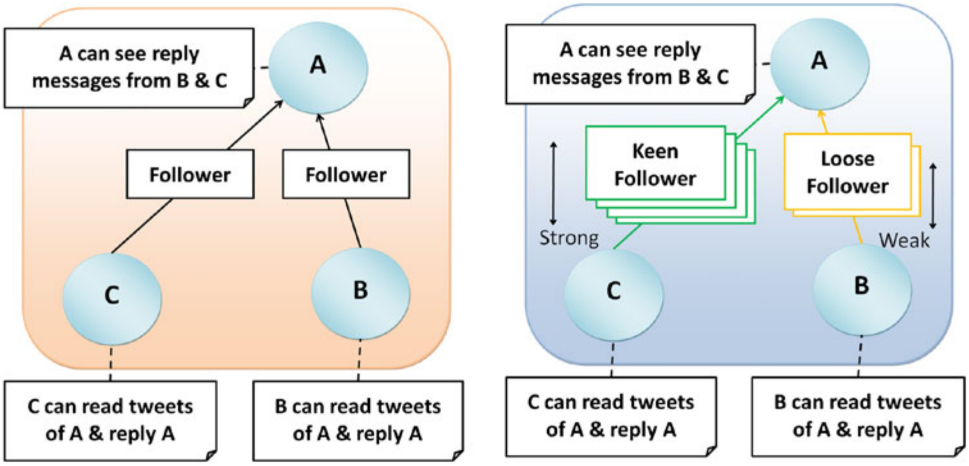
In the context of the above-referenced research, the current work may be considered as a quantitative SN analysis. As a step towards addressing the problem of SN privacy, we present the classification algorithms for the problem of identification of types of actors (individuals or organization) in a twitter network. We categorize the twitter actors into four types: (a) leaders (those like the news groups, who start tweeting, but do not follow any one there after, though they could have many followers), (b) Lurkers, who are generally inactive, but occasionally follow some tweets, (c) spammers, the unwanted tweeters, also called as twammers, and (d) close associates, including friends, family members, relatives, colleagues, etc. Two classification approaches have been proposed here to address this problem: (a) context-dependent classification for situations where an abundant amount of tweet data is available; here, we employ a fuzzy classification scheme in contrast to the stochastic estimation methods in the above-referenced literature, and (b) context-independent classification (based on the actor tweet patterns) that is suitable when enough information is not available about the network context.

3 Context-dependent classification of twitter actors

Social networks are formed by social groups of people that are linked by social bond or relationship. In a group, one can follow another or one can be followed by the other. In the twitter jargon, those two types of individuals are called followers and followees, respectively. Unlike in the case of emails, the mutual relationships between individuals/groups in the twitter network can be tracked by monitoring the tweets. Even though it is possible to compile email message and response pairs, the sparseness of data there makes it difficult to estimate the strength of relationship between the corresponding individuals. In the twitter (generally, SN) domain, on the other hand, it is possible to estimate the SN link (relationship) strengths using the followee–follower message statistics. At a gross level, it can be said that better the mutual communication, higher the link (relationship) strength. Further, if the link between two individuals is stronger, it is unlikely that either of them is a twammer.

Since most of the tweets occur between close associates, the twitter data for this group is generally overabundant compared to the other three groups, and this data imbalance poses problems for an identification of the actors, particularly those belonging to sparsely populated classes. Hence, we follow a two-stage process. In the first stage, we estimate the strength of links in the social network and eliminate a large number of actors with strong social bond (or link strength) because they naturally classify into the group of close associates. Then, in the second stage, we perform a linear classification of the four actor types mentioned in Sect. 1, using number of tweets and the followee–follower ratio as two features considering only those tweeters with weak link strength (less than 15% of maximum strength) between them.

For a good estimation of the link strengths, we make use of the fact that compared to an SN representation as a simple graph with nodes and links as shown in Fig. 1a, its representation as a graph with weighted links as in Fig. 1b provides better insights into not only the twitter spam identification problem but also various other SN problems. However, since the relationship strength depends upon a vague concept such as the "keenness" of the followers, we consider the fuzzy logic approach as the most appropriate method compared to other competing approaches for estimation of the relationship (link) strength. Furthermore, the 100% classification accuracy obtained with a linear classification scheme following this FST link strength estimation approach obviated the need for testing with other competing computational intelligence paradigms. Consequently, in the current experimentation, we focus exclusively on the fuzzy methodology, and do the implementation using jFuzzyLogic (Cingolani 2010), an open source code in the Java language for the fuzzy control language (FCL) defined by the International Electrotechnical commission (IEC)'s standard 1131-7 (Commission 1997). Three parameters "Reply Message Percentage", "Common Follower Percentage", and "Normalized Mean Reply Delay" have been considered as indicators of the keenness with which a tweeter is followed, and hence used to constitute the input set of our system depicted in Fig. 2. Percentage of reply messages among the total messages and the promptness (or inversely the delay) with which a actor responds are obviously indicators of the actor keenness in following a conversation. Similarly, since common followers of the tweeters on either side of a link suggest a sharing of similar ideas or topic of interests between the tweeters, percentage of common followers among the total tweeter population is a good indicator of relationship strength. To bring the "mean reply delay" parameter into the range [0 100] just like the other two, we normalized it with a scaling factor that sets the maximum delay to 100. The fuzzifier module of the system performs fuzzy quantification of each one of the 3 inputs into 5 levels (linguistic terms/values)–very low (VL), low (L), medium (M), high (H), and very high (VH), and determines for each input parameter the membership values in each category (level) based on the parameter distributions for each level that are pre-configured at the time of system setup. The fuzzy logic


A can see reply messages from B & C
A
Follower
Follower
C
B
C can read tweets of A & reply A
B can read tweets of A & reply A
A can see reply messages from B & C
A
Keen Follower
Loose Follower
Strong
Weak
C
B
C can read tweets of A & reply A
B can read tweets of A & reply A

(a) Social network represented as a simple graph

(b) Social network represented as a graph with weighted links

Fig. 1 Two graphical views of a social network

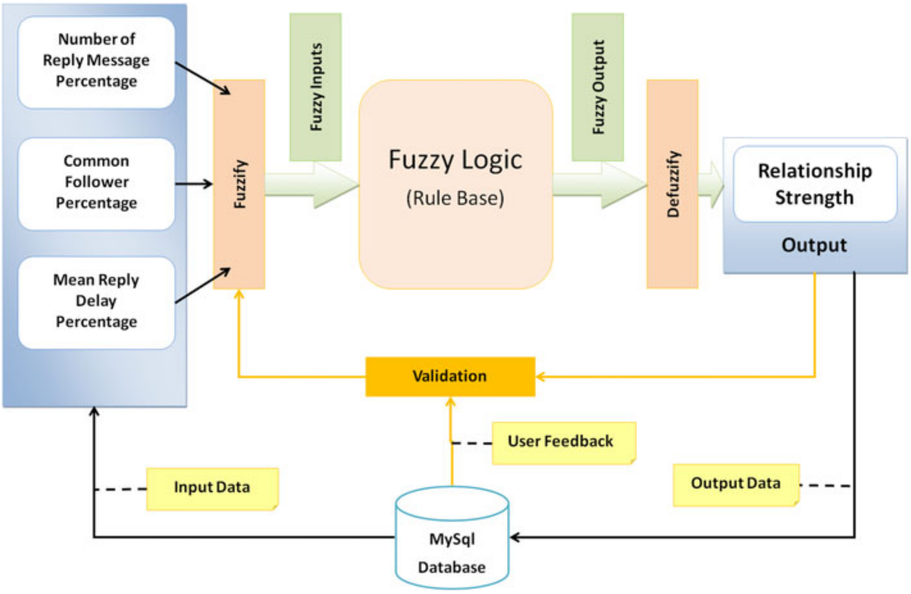


Fig. 2 Proposed fuzzy system architecture for link (relationship) strength evaluation

Table 1 Sample rules of the fuzzy rule base

No.	Rule
1	IF replies IS VL AND common_followers IS VL AND mean_reply_time IS VL THEN rel_strength IS VL
5	IF replies IS VL AND common_followers IS VL AND mean_reply_time IS VH THEN rel_strength IS L
10	IF replies IS VL AND common_followers IS L AND mean_reply_time IS VH THEN rel_strength IS M
25	IF replies IS VL AND common_followers IS VH AND mean_reply_time IS VH THEN rel_strength IS H
75	IF replies IS M AND common_followers IS VH AND mean_reply_time IS VH THEN rel_strength IS VH

(rule base) processes the fuzzy inputs (level-membership tuples for the three input parameters) generated by the fuzzifier and generates a fuzzy output (level-membership tuple) for the relation (link) strength. The rule base itself is generated using a meta-rule (implemented in Java) that assigns integer values 0 through 4 for the linguistic terms VL though VH, respectively, and maps each input term triplet onto an output term. Specifically, the output linguistic variable will assume the values VL, L, M, H, VH for values of S, the sum of the integer values corresponding to the three input linguistic terms, in the ranges $0 \leq S \leq 2$, $2 < S \leq 4$, $4 < S \leq 7$, $7 < S \leq 9$, $9 < S \leq 12$, respectively. In Table 1, we present 5 typical rules among the total 125 rules (for 3 input variable each with 5 possible linguistic values). The output linguistic values depend on the rules that are fired and the membership value of the output is determined by aggregating the membership values from the input level-membership tuples of the corresponding rule. Since the input variables can be members of different categories (levels) with different membership values, multiple rules could fire and yield multiple output level-membership tuples, which are finally mapped onto a real number by the defuzzifier. The defuzzifier is preconfigured to use the well known center of gravity method for generating the crisp output (real number) from the linguistic term-membership tuples.

The jFuzzyLogic software facilitates the actors to define the input/output parameter distributions for each one of the linguistic classes (terms) for determination of the class membership values based on the parameter values. For these distributions, it is possible to use either standard functions such the Gaussian or sigmoid or custom functions. We have used mostly Gaussian functions except for the fringe VL and VH classes for the inputs where we used sigmoids. The Gaussian function for the linguistic classes in the middle (L, M and H) was chosen so as to provide an overlap with the classes on either side. However, for the VL and VH classes, only a function that provides an overlap with either L or H is required, and hence we chose sigmoids, which look like leading or trailing portions of Gaussians. Table 2 gives the functions chosen for each one of the inputs and output, and the five linguistic classes. In this table, G represents the Gaussian distribution, and S, the sigmoid. For the Gaussian function, the two parameters represent mean and for the Sigmoid function, the two parameters are the gain and center, respectively. The distribution selection and parameter tuning is done by repeated trails to get a smooth and gradually increasing output.

Table 2 Membership distribution functions for the input and output variables

Fuzzy sets → crisp inputs/output↓	VL	*L*	*M*	*H*	VH
Input 1: Reply message %	*S* (−2, 1)	*G* (4, 1)	*G* (7, 3)	*G* (12, 3)	*S* (0.1, 55)
Input 2: Com. follower %	*S* (−2, 1)	*G* (2, 1)	*G* (4, 1)	*G* (8, 2)	*S* (0.1, 55)
Input 3: Mean reply time	*S* (−0.3, 25)	*G* (50, 10)	*G* (75, 5)	*G* (95, 2)	*S* (4, 100)
Output: Rel. strength	*G* (0, 15)	*G* (25, 10)	*G* (50, 10)	*G* (75, 10)	*G* (100, 15)

Com. follower common follower, *rel. strength* relational strength

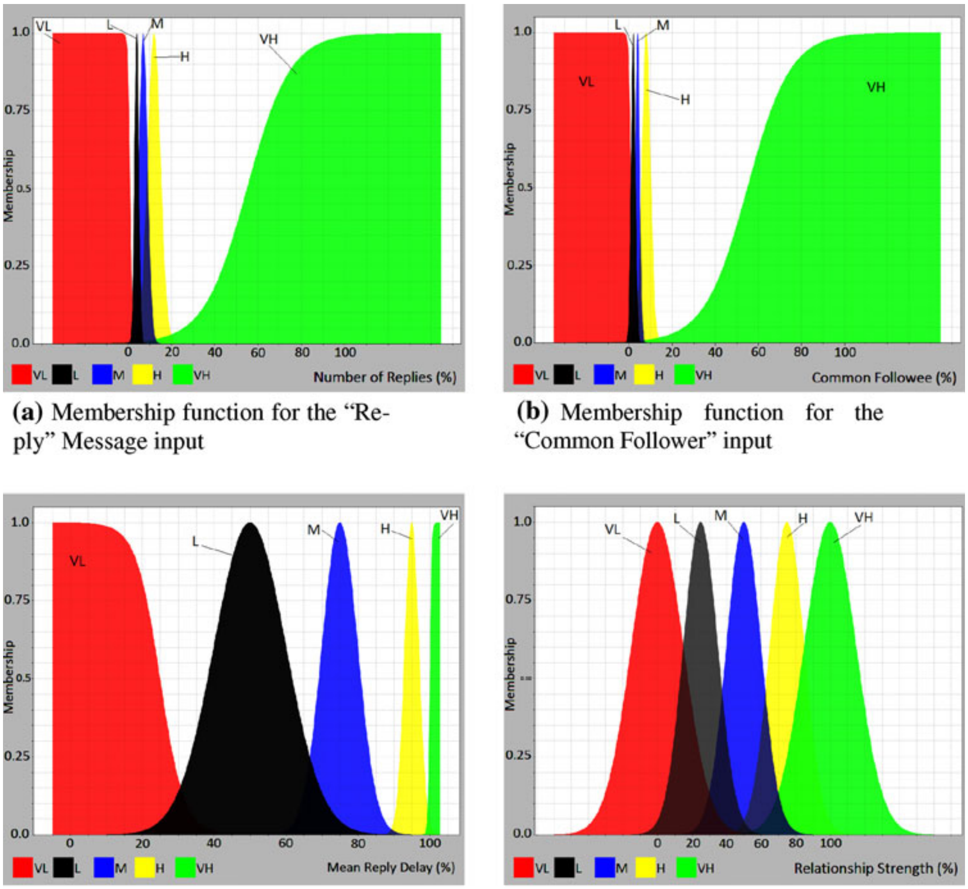


Fig. 3 Membership functions for various linguistic classes of the 3 input and 1 output variable

(a) Membership function for the “Reply” Message input

(b) Membership function for the “Common Follower” input

(c) Membership function for the “Mean Reply Delay” input

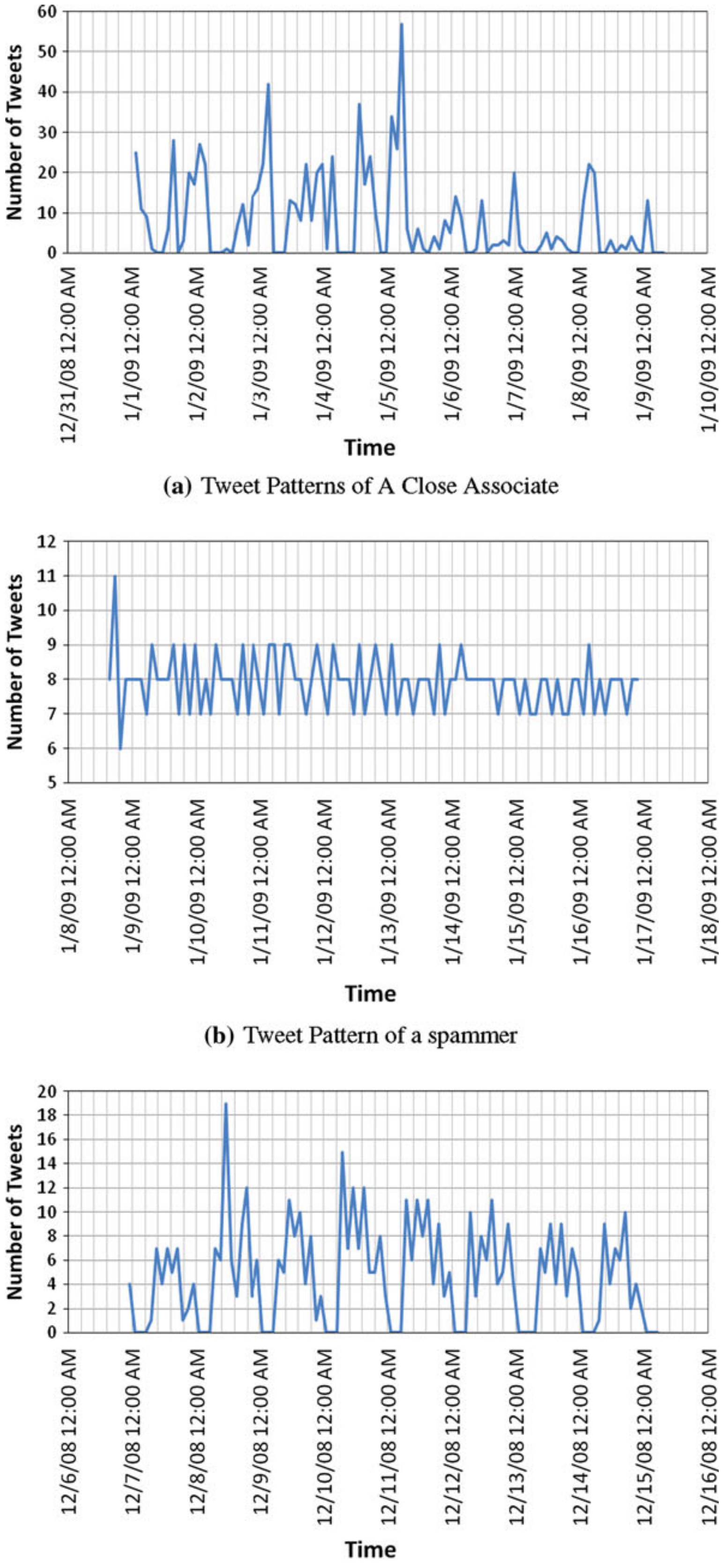
(d) Membership function for the “Relationship Strength” output

Figure 3 depicts the membership functions for the 3 inputs and 1 output. It may noted here that though some graphs extended for the values the parameters beyond the range [0 100], the membership values are computed only for values within the range.

4 Context-independent classification of twitter actors

Oftentimes, extensive contextual information for the estimation of the relationship between two tweeters is not available. Particularly, the twammers (or tweet spammers) catch us by surprise thereby rendering the approach detailed in Sect. 3 useless. Hence, we propose in this section an approach for actor (tweeter) type detection simply based on the generic tweeting patterns of the tweeters belonging to different tweeter classes. This approach derives support from the empirical evidence about the distinctive nature of the tweet patterns collected from different types of tweeters. Figure 4 depicts the sample 10-day tweet patterns from our twitter database that are developed by a procedure described in the following

Fig. 4 Tweet patterns of the three different classes of tweeters (though actual time series patterns extend over a period of 25–28 days, they have been truncated to 10 days here because periodicity is not visible on a compressed scale)



(a) Tweet Patterns of A Close Associate

(b) Tweet Pattern of a spammer

(c) Tweet Pattern of a News Blogger

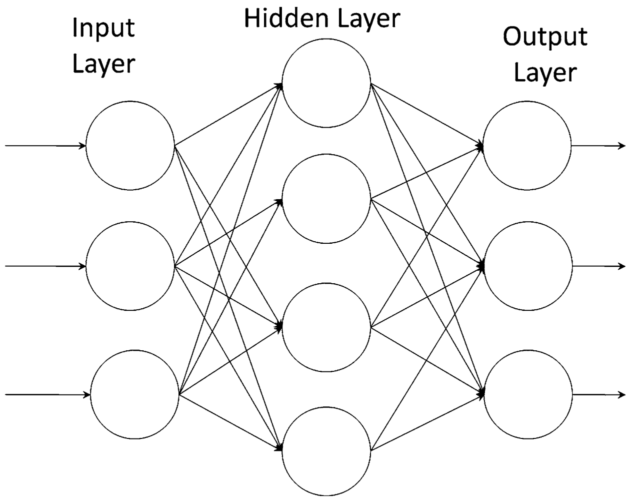


Fig. 5 A typical three-layered feed-forward network

section on experimental results. The tweet pattern shown in Fig. 4a is that of a normal tweeter, who could be a friend, colleague, or family member, and hence does not show much periodicity. The pattern shown in Fig. 4b is that of a spammer with actor name "la_hora," who spams from a bot named "Ya son las 02:50!". It is worth noting the periodic nature of the pattern with no distinction between day time and night time segments. When the tweet messages are analyzed most of them turn out to be similar and meaningless. Tweeting occurs every 10 min, and hence nearly a constant number of messages are generated each day. Finally, the patern in Fig. 4c is that of a news blogger with actor-ID "spitsnet." We verify that this person is a news blogger by following the links. Further evidence comes from our observation that this person tweets in the dutch, and Sp!ts(pronounced Spits), according to Wikipedia (2010), is a tabloid format newspaper freely distributed in trains, trams and buses in the Netherlands. It is interesting to notice that this pattern resembles a harmonic series with a very low number of tweets in the midnight and a high number during the day time. Thus, it is possible to attempt the same tweeter classification problem using a context-independent approach of matching an unknown tweet pattern with the prototypes (or the labeled samples) from different classes. The actor classes we consider here are the leaders (e.g. the news blogger), spammers, and associates. Since it is difficult to identify lurkers with sparsely available data, we consider only a three-class problem that excludes this class in our context-independent approach.

For email spam detection, the statistical Bayesian approach has been very popular, and is found to be effective. In a typical work in the Bayesian framework, Ma et al. (2009) used a technique called "Negative Selection" to detect spam email without any prior knowledge about the spam emails. Yeh and Chiang (2009) carry out a re-evaluation on the Bayesian filter for email spam detection, and find that though it yields high accuracy on plain text email messages, it is not as effective with modern emails with multimedia content and message encoding. They suggest the use of a scheme combining different spam detection strategies. Thus, it may be seen that both the Bayes and MLP classifiers are competitive at least for the email spam detection problem. Hence, these two classifications tools have been considered for our investigations into the problem of actor type detection based on the tweet patterns. We have included in this repertoire of classification tools, the Random Forest method developed by Breiman (2001).

In the Bayes approach, $P(\mathcal{C}_i|\mathcal{X})$, the probability of a given tweet pattern vector $\mathcal{X}$ (formed by the time samples over a specific period of tweet frequencies) belonging to a pattern class $\mathcal{C}_i \in \{Associate, Spammer, Leader\}$ is given by:

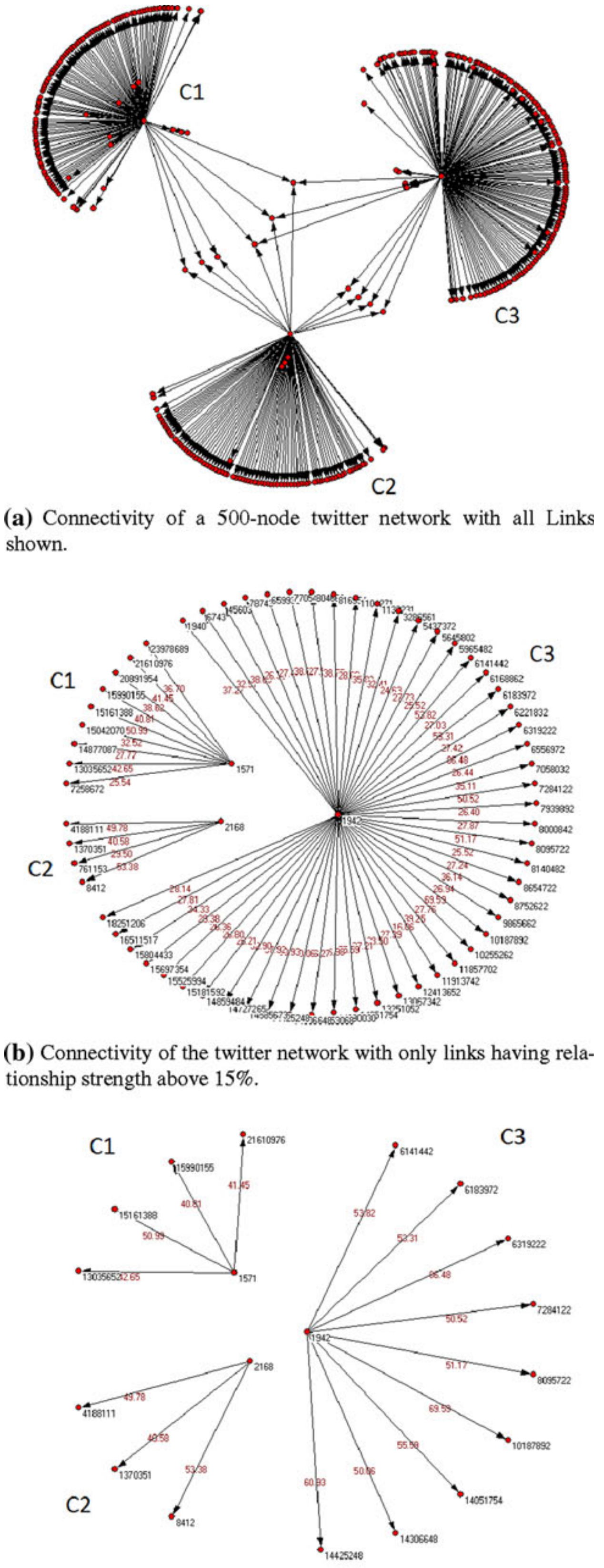
$$\begin{aligned} P(\mathcal{C}_i|\mathcal{X}) &= \frac{P(\mathcal{X}|\mathcal{C}_i).P(\mathcal{C}_i)}{P(\mathcal{X})} \\ &= \frac{P(X_1,X_2,\ldots,X_n|\mathcal{C}_i).P(\mathcal{C}_i)}{P(X_1,X_2,\ldots,X_n)} \\ &= \frac{\prod_{j=1}^{n} P(X_j|\mathcal{C}_i).P(\mathcal{C}_i)}{P(X_1,X_2,\ldots,X_n)} \end{aligned} \qquad (1)$$

In the above $\{X_1, X_2, , X_n\}$ are the components of the pattern vector $\mathcal{X}, P(\mathcal{C}_i)$ is the a priori probability of the class $\mathcal{C}_i$, and $P(\mathcal{X}) = P(X_1, X_2, \ldots, X_n)$ is the probability of the sample vector. The third line in the above equation is a consequence of the assumption that the components of $\mathcal{X}$ are conditionally independent; a Bayes classifier using this assumption is termed as a naive Bayes classifier. Now, since $P(X_1, X_2, \ldots, X_n)$ does not depend upon the chosen class, the Bayes classification algorithm boils down to assignment of the pattern vector $\mathcal{X}$ to the class $\mathcal{C}_i$ for which $P(\mathcal{C}_i) \cdot \prod_{j=1}^{n} P(X_j|\mathcal{C}_i)$ is maximum. For this computation, it may be assumed that $P(X_j|\mathcal{C}_i)$ conforms to a normal distribution as follows:

Table 3 Our twitter database statistics

Total actors	Total links	Total tweets	Total tweet replies	Average links/actor	Average tweets/actor	Average replies/actor
441,234	2,045,804	6,481,900	2,312,927	5	15	5

Fig. 6 Changes in twitter network clustering with thresholding of ▶ links (the numbers at the nodes are the actor IDs and the numbers on the links are their strengths



(a) Connectivity of a 500-node twitter network with all Links shown.

(b) Connectivity of the twitter network with only links having relationship strength above 15%.

(c) Connectivity of the twitter network with only links having relationship strength above 40%

$$P(X_j|\mathcal{C}_i) = \frac{1}{\sqrt{2\sigma_{ji}^2}} e^{-\frac{(X_j-\mu_{ji})^2}{2\sigma_{ji}^2}} \tag{2}$$

Here μ_{ji} and σ_{ji} are the mean and standard deviation, respectively, of the jth component of the pattern vectors belonging to the class $\mathcal{C}_i$.

In the MLP approach, a feed-forward neural network (Rojas 1996) of the form shown in Fig. 5 is used. The circular elements are nonlinear (usually, sigmoid) functions representing neurons, and the lines are weights representing inter neuronal synapses. The leftmost layer of neural net is the input layer, and the rightmost is the output layer. The input layer has as many neurons as the components of the pattern vector. The output neurons are as many as the number (in our case, three) of classes in the classification problem. The neuron layers in between the input and output layers are called hidden layers; they are effective in producing good nonlinear mappings. During training phase, the pattern vectors with known class labels are presented at the input layer, and the outputs are observed. All neurons but one in the output layer are expected to assume zero values; the one representing the input pattern's class should have a high value (one). If that does not happen, the weights connecting the neurons in the penultimate layer to those in the last (output) layer should be so adjusted so as to minimize the following mean square error function:

$$E = \sum_{k=1}^{n} (t_k - O_k)^2, \tag{3}$$

where t_k and O_k are the target (expected) and actually observed output values of the kth output neuron. It can be shown that the local minima of this function can be achieved by gradient descent formulas that adjust the weights of neurons connecting the last layer first, then those connecting into the previous layer, and so on till those feeding into the first layer are modified. Since this weight training takes place in the backward direction from the output to input layers, this algorithm is known as the back-propagation learning algorithm (Rojas 1996). Once the network is trained with all the training patterns, it will be ready to classify the patterns with unknown class labels by producing high output at all but one of the output neurons.

According to Breiman (2001), Random Forest is classifier formed by an ensemble of decision tree classifiers $\{h(\mathbf{X}, \Theta_k), k = 1, \ldots\}$ where the $\{\Theta_k\}$ are independent and identically distributed random vectors, and $\mathbf{X}$ is the input vector. The random forest classifies X into the class with

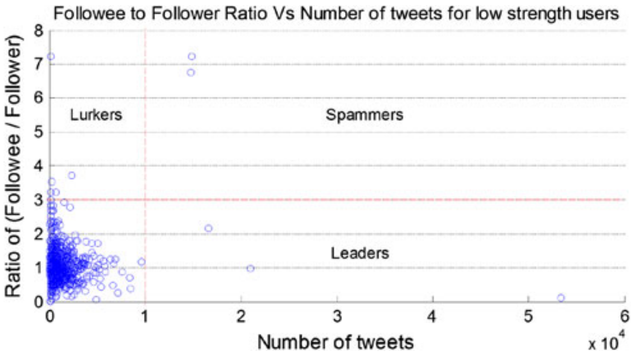


Fig. 7 Linear separation of leaders, lurkers, associates, and spammers

maximum vote considering the votes for the most popular class at **X** by the constituent classifiers.

5 Experimental results

5.1 Data collection procedure

The main requirement for this research is availability of a good dataset that includes details of all the activities in the twitter network such as actor profiles, the number of messages interchanged between the actors, and the interactions between the actors, etc. Since all the online social networks are based on real individuals, privacy settings make it very difficult to acquire a proper dataset with all activities of each individual from the network. Most popular social network web-sites provide Application Programming Interfaces (APIs) for running their profile crawlers, but these are either restricted or need special permissions. However, we circumvented the privacy restrictions problem by simply programming the Twitter crawler to skip over the restricted actors. Since the number of restricted actors is very low compared to the total number of actors in Twitter, this strategy is expected to have little effect on the results of our analysis. Another problem we faced while accessing twitter database is due to Twitter's API Rate Limiting policy which limits the number of requests per hour for the data records made through the API to 150. Luckily, this restriction could be waived to white-listed actors with special permissions. We obtained these permissions from Twitter and could access as many as 20,000 records per hour. Table 3 summarizes the statistics of the twitter database we developed using this process. However, since it is difficult to visualize such a huge network, we show only the results on 500-node subnet in Sect. 5.2. Further the difficulties in hand-labeling the actor types forced us to limit our experimentation only 528 randomly sampled tweeter records in Sect. 5.3.

5.2 Results on link strength determination and context-dependent classification

We determine the link strengths of a 500-node twitter network by applying the fuzzy logic-based classification method discussed in Sect. 3 on the twitter database developed by us. It is interesting to visualize some characteristic features of this network from its graphical depiction in Fig. 6. First, it can be observed that the network nodes form strong clusters, and the cluster structures do not change much when weak links are removed; this indicates that the same set of tweeters communicate frequently with one another though some are more involved than the other in tweeting. Next, since it is easy to infer that the tweeters with strong connectivity are close associates, we need to apply our classification to the tweeters with low relationship strength. From Fig. 6b and c, it is clear that a threshold of 15% (of maximum relational strength) for the link strength is good enough to separate out the tweeters who are unambiguously close associates. Hence, we apply the simple linear classification algorithm using number of tweets and the followee–follower ratio as two components of the pattern vector only to the tweeters with the link strength below this threshold. Clear separation of the four tweeter classes as depicted in Fig. 7 suggests that this method holds promise for an effective identification of leaders, lurkers, spammers, and associates. In view of the enormity of the analysis required to be done for hand-labeling of the records for the purpose of validation of these classification results, we first formed a smaller validation set consisting of: (a) the few records corresponding to the points in the leader, lurker, and spammer quadrants in the plot (Fig. 7), (b) those represented by the points on

Table 4 Confusion matrices of the three classifiers in the TCV test procedure

Class↓→	Naive Bayesian confusion matrix			MLP confusion matrix			Random forest confusion matrix		
	L	*S*	*A*	*L*	*S*	*A*	*L*	*S*	*A*
Leaders (*L*)	224	0	0	220	4	0	223	1	0
Spammers (*S*)	0	163	1	1	161	2	2	162	0
Associates (*A*)	0	0	140	4	2	134	0	0	140
Classification accuracy→	99.81%			97.54%			99.43%		

the boundary of the "close associates" quadrant, and (c) those corresponding to a few randomly selected points in the middle of the "close associates" quadrant. We then hand-labeled each one of these records in the validation set by going through the profile information and the tweets contents. Finally, considering the hand labels of the records as the ground truth, we tally the results of linear classification on the validation set with the ground truth. Validity of our classification approach has been established by a perfect tally.

5.3 Results of context-independent classification

For this experimentation, we used a set of total 528 tweeting frequency patterns that includes 224 leaders (news groups), 164 spammers, and 140 associates. Using the nine equally spaced time samples over a 10-day window as components, ten-dimensional pattern vectors are formed for each one of the 528 data records. For the MLP we have used 9 neurons (corresponding to the components of the pattern vector) in the input layer, 11 neurons in the hidden layer, and 3 neurons (corresponding to the 3 classes) in the output layer.

Since the classification results depend not only on the classifier chosen but also on the training procedure adopted, we have considered three training/test procedures as follows for our experimentation: (a) ten-fold cross validation (TCV) method. Here, the data are split into ten sub-parts, and each sub-part is used in a round robin fashion as the test set with the remaining nine as training sets, (b) R66T method where randomly selected 66% records form the training set and the rest are used for testing, and (c) O66T method where the first (oldest) 66% of the time ordered records form the training set and the rest are used for testing; this method is an implementation of the strategy of applying old experience to new situations. The classifiers used for experimentation, as discussed in Sect. 4 are the naive-Bayes (NB) classifier, multi-layer percetron (MLP), and and random forest (RF) classifier. In the experiments with each classifier-training procedure combination, we obtained the confusion matrices and classification accuracies for each one of the three classifiers under the above three test scenarios. Further, since classification accuracy does not reflect the correct performance when the dataset is imbalanced with far more samples in some groups than the other, we employed the popular F-measure that is used in the database (DB) literature to measure the efficacy of retrieving the correct DB records. In case information retrieval, the records intended for retrieval as one class, and the rest are considered as irrelevant. The relevant records retrieved by an algorithm are considered as true positive (TP). Irrelevant records retrieved are considered as false positive (FP) and the relevant records not retrieved are considered as false negatives (FN). Finally, the irrelevant records not retrieved are denoted as true negatives (TN). The proportion of records retrieved from the total set of relevant records is termed as TP rate or recall (*R*), and the proportion of relevant records among the total set of records correctly processed (retrieved or omitted) by a data retrieval algorithm is called precision (*P*). The F-measure is the harmonic mean of recall and precision. These relations can be stated mathematically as follows:

Table 5 Confusion matrices of the three classifiers in the R66T test procedure

Class↓→	Naive Bayesian confusion matrix			MLP confusion matrix			Random Forest confusion matrix		
	L	*S*	*A*	*L*	*S*	*A*	*L*	*S*	*A*
Leaders (*L*)	75	0	0	73	1	1	75	0	0
Spammers (*S*)	0	56	0	0	55	1	0	56	0
Associates (*A*)	0	0	49	1	2	46	0	1	48
Classification accuracy→	100%			96.67%			99.44%		

Table 6 Confusion matrices of the three classifiers in the O66T test procedure

Class↓→	Naive Bayesian confusion matrix			MLP confusion matrix			Random Forest confusion matrix		
	L	*S*	*A*	*L*	*S*	*A*	*L*	*S*	*A*
Leaders (*L*)	30	6	20	56	0	0	0	0	56
Spammers (*S*)	26	29	1	0	56	0	1	55	0
Associates (*A*)	2	0	66	20	0	48	54	0	14
Classification accuracy→	69.44%			88.89%			38.33%		

$$R = \frac{\mathrm{TP}}{(\mathrm{TP} + \mathrm{FN})}$$

$$P = \frac{\mathrm{TP}}{(\mathrm{TP} + \mathrm{FP})}$$

$$\text{F-measure} = \frac{2}{\left(\frac{1}{R} + \frac{1}{P}\right)} = \frac{2RP}{(R + P)} \tag{4}$$

For adapting the F-measure to measure classification performance, we consider successively each one of the three actor classes (i.e. leaders, spammers, and associates) as the relevant class and treated the remaining two together as a single irrelevant class. For different classifier and test method combinations, the TP, TN, FP, and FN are then be obtained from the corresponding confusion matrices, and the F-measures are computed. Ultimately, the overall F-measures are computed by an weighted average scheme in which individual class F-measures are given weights (in the interval [0 1]) proportional to the number of samples in the corresponding classes.

Tables 4, 5, and 6, respectively, present the confusion matrices for the three classifiers under the TCV, R66T and O66T test procedures described above. These results indicate that MLP outperforms the other two classifiers with respect to classification accuracy when the O66T test procedure is used, though its results are slightly shy of those with the other classifiers when the TCV and R66T test procedures are used.

Tables 7, 8, and 9 summarize the results of F-measure computations for the three classifiers under the TCV, R66T, and O66T procedures, respectively, using the confusion matrix data in Tables 4, 5, and 6. These results include the F-measures for individual classes (lumping together the remaining two classes as an irrelevant class) as well as the weighted average measure computed by applying to the data of each class a weight proportional to its population in the training set. On F-measure also, the MLP exhibits a very good performance compared to the other two classifiers with O66T test procedure, and yields

Table 7 F-measure Computations for the three classifiers in the TCV test procedure

Class↓	TPR	FPR	*P*	*F*
Naive Bayes				
Leaders	1	0	1	1
Spammers	0.994	0	1	0.997
Associates	1	0.003	0.993	0.996
Weighted averages→	0.998	0.001	0.998	0.998
MLT				
Leaders	0.982	0.016	0.978	0.98
Spammers	0.982	0.016	0.964	0.973
Associates	0.957	0.005	0.985	0.971
Weighted averages→	0.975	0.013	0.976	0.975
Random Forest				
Leaders	0.96	0.007	0.991	0.973
Spammers	0.988	0.003	0.994	0.991
Associates	1	0	1	1
Weighted averages→	0.994	0.04	0.994	0.994

TPR true positive rate, *FPR* false positive rate, *P* precision, *F* F-measure

Table 8 F-measure computations for the three classifiers in the R66T test procedure

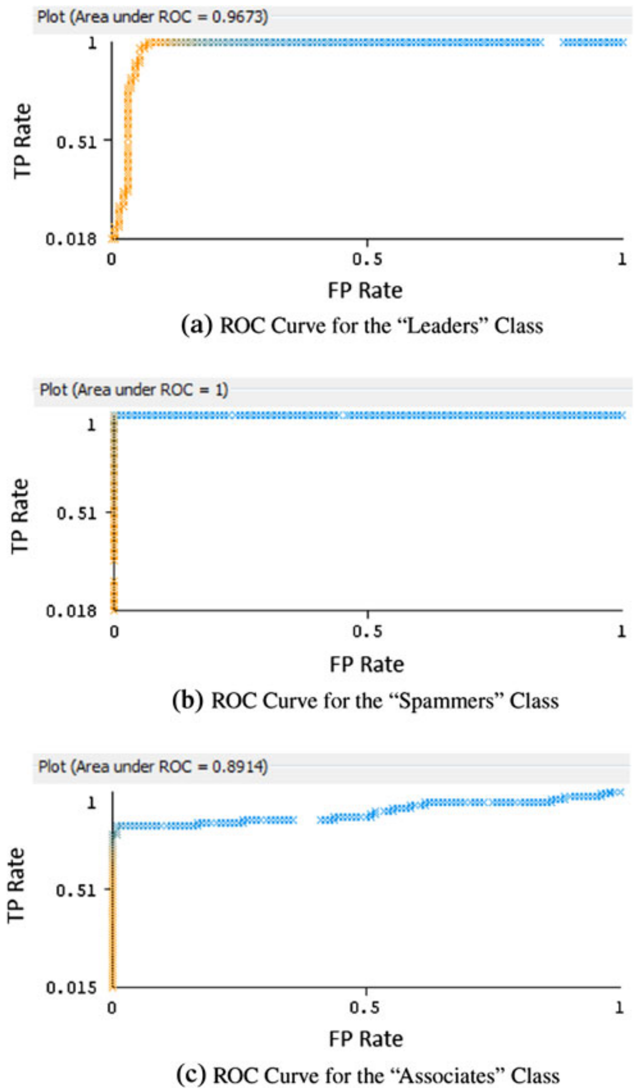
Class↓	TPR	FPR	*P*	*F*
Naive Bayes				
Leaders	1	0	1	1
Spammers	1	0	1	1
Associates	1	0	1	1
Weighted averages→	1	0	1	1
MLT				
Leaders	0.973	0.01	0.986	0.98
Spammers	0.982	0.024	0.948	0.965
Associates	0.939	0.015	0.958	0.948
Weighted averages→	0.967	0.016	0.967	0.967
Random Forest				
Leaders	1	0	1	1
Spammers	1	0.008	0.982	0.991
Associates	0.98	0	1	0.99
Weighted averages→	0.994	0.003	0.995	0.994

TPR true positive rate, *FPR* false positive rate, *P* precision, *F* F-measure

Table 9 F-measure computations for the three classifiers in the O66T test procedure

Class↓	TPR	FPR	*P*	*F*
Naive Bayes				
Leaders	0.536	0.226	0.517	0.526
Spammers	0.518	0.048	0.829	0.637
Associates	0.971	0.188	0.759	0.852
Weighted averages→	0.694	0.156	0.705	0.684
MLT				
Leaders	1	0.161	0.737	0.848
Spammers	1	0	1	1
Associates	0.706	0	1	0.828
Weighted averages→	0.889	0.05	0.918	0.888
Random Forest				
Leaders	0	0.444	0	0
Spammers	0.982	0	1	0.991
Associates	0.206	0.5	0.2	0.203
Weighted averages→	0.383	0.327	0.387	0.385

TPR true positive rate, *FPR* false positive rate, *P* precision, *F* F-measure



(a) ROC Curve for the "Leaders" Class

(b) ROC Curve for the "Spammers" Class

(c) ROC Curve for the "Associates" Class

Fig. 8 ROC curves for the three classes of actors when MLP is used under the O66T procedure

competitive results under the remaining two test procedures. The performance of the Random Forest method under the O66T procedure is below par.

The ROC curves shown in Fig. 8 for the three actor classes for the best classifier (i.e. MLP) under the most challenging test procedure (i.e. O66T) establish the viability of context-independent classification procedure.

6 Summary and conclusion

In this paper, we present two classification methods for twitter network actor identification: (a) context-dependent (data-heavy) method which employs a fuzzy logic approach to estimation inter-actor relationship strengths in the first step and then a linear classifier to separate out the four actor classes, and (b)context-independent method that uses tradition classifiers and generic actor tweet patterns to distinguish between the actors in situations where the availability of actor information is limited.

This research enforces the conventional wisdom that spammers follow a large number of people (followees), but they themselves are followed by very few people. Specifically, as evidenced by the results of Sect. 5.2, spammers are defined by the accounts that make more than 10,000 tweets in a 10-day interval (or equivalently over an average of 1,000 tweets a day) and have a followee-to-follower ratio of 1.5 to 1 or more. The twitter leaders, on the other hand, can be distinguished by their high rate of tweeting, large number of followers, but a few, if any, followees, and hence by a followee-to-follower ratio much below 1. Close associated are marked by strong connectivity to their followers, low to moderate number of tweets (1,000) per day, and small to moderate (less than 3) followee–follow ratio. Finally, lurkers is a rare class of tweeters, who follow many people, but they themselves rarely post or reply any tweets.

Our results on the more challenging problem of classifying actors with limited data, the MLP classifier has been found to outperform the naive Bayesian and Random Forest classifiers when the procedure of classifying the new patterns with the old patterns is adopted. Since that is how new spammers are identified generally, the MLP can be considered as an effective context-independent approach to spam filtering.

Acknowledgments We would like to thank Dr. Zuoming Wang, Assistant Professor, communication studies, University of North Texas, Texas, USA, for helping us in this work. We would also like to thank Dr. Santi Phithakkitnukoon, SENSEable City Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA, for giving us valuable comments on this work. This work is supported by the National Science Foundation under Grants CNS-0627754, CNS-0619871 and CNS-0551694.

References

Bellman RE (1957) Dynamic programming. Rand Corporation

Berkowitz SD (1982) An introduction to structure analysis: the network approach to social research. Butterworth

Breiman L (2001) Random forests. Mach Learning 45(1):5–32

Cha M, Mislove A, Gummadi KP (2009) A measurement-driven analysis of information propagation in the flickr social network. In: Proceedings of 18th international conference on World wide web, pp 721–730

Cheong M, Lee V (2009) Integrating web-based intelligence retrieval and decision-making from the twitter trends knowledge base. In: Proceedings of 2nd ACM workshop on social web search and mining, Hong Kong, pp 1–8

Cingolani P (2010) jfuzzylogic, open source fuzzy logic library and FCL language implementation. http://jfuzzylogicsourceforgenet/html/indexhtml

Commission IE (1997) Technical committee no. 6: Industrial process measurement and control. sub-committee 65 b: Devices.

iec 1131-programmable controllers.http://www.fuzzytechcom/binaries/ieccd1pdf

Dorein P, Sockman FN (1997) Evolution of social networks, vol 1. Overseas Publishers Association, Amsterdam

Garton L, Haythornthwaite C, Wellman B (1997) Studying online social networks. J Comp Mediated Commun 3(1):75–105

Kim S, Han S (2009) The method of inferring trust in web-based social network using fuzzy logic. In: Proceedings of international workshop on machine intelligence research, vol 2, pp 140–144

Klym N, Montpetit MJ (2008) Innovation at the edge: social TV and beyond. MIT Communications Futures Program (CFP). www.http://cfp.mit.edu/publications/CFP_Papers/Social%20TV%20Final%202008.09.01%20for%20distribution.pdf

Kossinets G, Watts DJ (2006) Empirical analysis of an evolving social network. Sci Agric 311(5757):88–90

Kumar R, Novak J, Tomkins A (2010) Structure and evolution of online social networks. In: Link mining: models, algorithms, and applications. Springer, Berlin

Lin Y, Sundaram H, Chi Y, Tatemura J, Tseng B (2007) Splog detection using content, time and link structures. In: Proceedings of IEEE international conference on multimedia and expo, pp 2030–2033

Ma W, Tran D, Sharma D (2009) A novel spam email detection system based on negative selection. In: Proceedings of 4th international conference on computer science and convergence information technology, Seoul, pp 987–992

Marsden PV, Campbell KE (1984) Measuring tie strength. Social Forces 63(2):482–501

Mathioudakis M, Koudas N, Marbach P (2010) Early online identification of attention gathering items in social media. In: Proceedings of 3rd ACM international conference web search and data mining, pp 301–310

McPherson M, Smith-Lovin L, Cook JM (2001) Birds of a feather: Homophily in social networks. Annu Rev Sociol 27:415–444

Mislove A, Marcon M, Gummadi KP, Druschel P, Bhattacharjee B (2007) Measurement and analysis of online social networks. In: Proceedings of 7th ACM SIGCOMM conference on international measurement, pp 29–42

Onnela JP, Saramaki J, Hyvonen J, Szabo G, Lazer D, Kaski K, Kertesz J, Barabasi AL (2007) Structure and tie strengths in mobile communication networks. Proc Natl Acad Sci USA 104(18):7332–7336

Rojas R (1996) Neural networks—a systematic introduction. Springer, Berlin

Scott J (1992) Social network analysis. Sage, Newbury Park

Strauss D, Ikeda M (1990) Pseudolikelihood estimation for social networks. J Am Stat Assoc 85:204–212

Wasserman S, Faust K (1994) Social network analysis. Cambridge University Press, Cambridge

Wasserman S, Pattison P (1996) Logit models and logistic regressions for social networks: I. an introduction to markov graphs and p*. Psychometrika 61(3):401–425

Watts DJ, Dodds PS, Newman MEJ (2002) Identity and search in social networks. Sci Agric 296(17):1302–1305

Weng J, Lim E, Jiang J, He Q (2010) Twitterrank: finding topic-sensitive influential twitterers. In: Proceedings of 3rd ACM international conference on web search and data mining, New York, pp 261–270

White H, Boorman S, Breiger R (1976) Social structure from multiple networks: I. blockmodels of roles and positions. Am J Sociol 81:730–780

Wikipedia (2010) Twitter. http://enwikipediaorg/wiki/Twitter

Wu C, Zhou B (2009) Analysis of tag within online social networks. In: Procedings of ACM 2009 international conference, pp 21–30

Yeh C, Chiang S (2009) Revisit bayesian approaches for spam detection. In: Proceedings of. 9th international conference for young computer scientists, Hunan, pp 659–664

Zhou B, Pei J, Luk W (2009) A brief survey on anonymization techniques for privacy preserving publishing of social network data. In: Proceedings of ACM SIGKDD Explorations Newsletter, vol 10, pp 12–22