Followee recommendation in Twitter using fuzzy link prediction

Fernando M. Rodríguez[†], Luis M. Torres[‡], Sara E. Garza[*]
School of Mechanical and Electrical Engineering (FIME)
Universidad Autónoma de Nuevo León (UANL)
San Nicolás de los Garza, NL, Mexico
[†]fernando.aldape@gmail.com, [‡]luis.torres.ciidit@gmail.com,
[*]sara.garzavl@uanl.edu.mx

Abstract

In social networking sites, it is useful to receive recommendations about whom to contact or follow. These recommendations not only allow to establish connections with people one might already know in real life, but also with people or users that have similar interests or are potentially interesting. We propose an approach that tackles contact (followee) recommendation in Twitter by means of fuzzy logic. This fuzzy approach handles recommendation as a link prediction problem and uses three types of similarity between a pair of users: tweet similarity, followee id similarity, and followee tweet similarity. These similarities are calculated by extracting user profiles. These profiles are, in turn, obtained by considering Twitter as a heterogeneous information network. To test our approach, we crawled a repository of 6,000 users and 2 million tweets, and we measured accuracy by comparing our results with the actual followee lists of the users. These results, which are also compared against the results given by state-of-the-art methods, show a high accuracy. Other advantages of the fuzzy system include a self-explanatory capability and the ability to produce a non-binary friendship value.

Keywords: *fuzzy systems, recommender systems, Twitter, link prediction, expert systems, artificial intelligence*

1 Introduction

Social networking sites (SNS's) play a key role in our daily lives. These sites not only enable users to publish and receive information they consider to be of interest, but also allow them to keep in contact with friends, or even to establish new relationships. Social networking sites, ultimately, serve as a platform for personal or social expression and organization — hence their importance for areas such as marketing, human-computer interaction, and research.

Due to the increasingly intensive use of SNS's, it is common for users to be overloaded with information (Benito-Ruiz, 2009; Gantz and Reinsel, 2010), e.g. status updates to read or possible new friends to contact. For this reason, *recommender systems* have been adopted to filter content, activities, products, and social connections of actual interest for the user (Park et al., 2012).

An important representative of SNS's nowadays — with 190 million users, 60,000 daily comments, and ranking ten in Alexa's Top Sites[1] — is Twitter[2]. This site offers a *microblogging service* where users are able to publish short messages (140 characters maximum per post) called "tweets" and receive the posts of other users that they decide to *follow*, where following a user is similar to subscribing to an RSS news feed; the ones who follow a user are called the user's *followers* and the ones followed by a user are called the user's *followees* (in this work, we will use "contacts" and "friends" as synonyms of this term). Other distinctive features of Twitter include post forwarding, where the forwarded message is called *retweet*, user mentions in messages (where each user account starts with "@"), and a special form of message labeling, where the label is called *hashtag* and usually starts with "#".

Twitter, of course, is not exempt of the previously mentioned information overload and crowding. Recommendation approaches for this SNS so far include automatic suggestions of tweets, URL's, hashtags, mentions, retweets, and followees (Kywe et al., 2012). We focus on this last recommendation item, since helpful followee recommendations aid users to receive actual content of interest, to get acquainted with people of interest, and to build stronger communities — either in Twitter or real life.

Because the information in Twitter tends to be imprecise, incomplete, and could highly depend on the context where it is employed, we propose a *fuzzy hybrid recommendation system*. The rules of this system create recommendations by predicting follower-followee relationships (which we shall refer to as "friendships") between pairs of users given the similarity of their tweets, their followees, and the tweets of their followees. These similarities are derived from user profiles whose information is retrieved by considering Twitter as a heterogeneous information network. Our contributions thus include:

- A fuzzy expert system that acts as a hybrid recommender system.

[1] Information available at: `http://www.alexa.com/topsites`. This list of Top Sites was retrieved in March 2014.

[2] Available at: `http://twitter.com`.

This fuzzy system predicts the degree of friendship between a pair of Twitter users and uses this degree as the basis for recommendation.

- A formal conceptual model for extracting user profiles by considering Twitter as a heterogeneous network.
- Evidence for the effectiveness of the approach in different contexts (Twitter, the Cit-HepTh citation network).
- A comparison with state-of-the-art methods.

The rest of this document is organized as follows: Section 2 reviews necessary background and Section 3 discusses related work; Section 4 describes our approach and Section 5 provides experiments and results; Section 6, finally, presents conclusions and future work.

2 Background

This section introduces vocabulary and notation that will be used. It covers basic notions for recommender systems, graph theory, information retrieval, and fuzzy logic.

2.1 Recommender systems

In the midst of overwhelming amounts of information, the aim of a *recommender system* is to suggest users those items that might be of *interest*, such as movies, records, books, and other users. Most recommender systems are *personalized* and build a *profile* to make suggestions based on the information and particular preferences of the *target user* (i.e., the recipient of the recommendations). Common approaches for this kind of recommendation include *content-based* and *collaborative filtering* systems; while the former find items that are similar to the ones highly rated by the user in the past, the latter find items that were highly rated by users that are similar to the target user (Ricci et al., 2011). In that sense, content-based systems are item-oriented and collaborative filtering systems are user-oriented.

The usual output of a recommender system is a *ranked list* of items, where the items at the top — ideally — are the most likely to achieve high ratings by the target user. To build this list, several recommendation algorithms start off with a set of *candidate items* and gradually refine this set. When talking about friend or contact recommendations, the term “item” is usually replaced by “user”. Consequently, it is common to refer to *candidate users* instead of candidate items and to *recommended users* instead of recommended items; let us note that the latter are the ones included in the (final) recommendation list. Another important aspect of friend recommendation is that this task is sometimes stated as a *link prediction* problem, i.e. the problem of determining if a connection will appear between a pair of entities in a network (Getoor and Diehl, 2005).

2.2 Networks

Networks depict related entities and are formally represented with *graphs*. A graph $G = (V, E)$ consists of a set $V = \{v_1, v_2, \ldots v_n\}$ of *vertices* and a set E of *edges*. While the former set represents the entities of the network, the latter stands for the relationships or connections among these entities; $E \subseteq V \times V$, typically[3]. If E is symmetric, i.e. the existence of an edge (u, v) implies the existence of (v, u) as well, then all edges run in both directions and the graph is said to be *undirected*; being this the case, an edge can be formally written as a 2-subset $\{u, v\}$. However, if the property of symmetry does not hold, the graph is *directed* and $(u, v) \in E$ depicts an edge that goes out from u into v. In a directed graph, edges are commonly referred to as *arcs* and are drawn as arrows. When the vertices or edges have labels, the graph is said to be *labeled*, and an edge-labeled graph where the labels are numerical values is said to be *weighted* (it is otherwise known as an *unweighted graph* whose edges have indistinct values). A *subgraph* $G_s = (V_s, E_s)$ is a portion of a graph, such that $V_s \subseteq V$ and $E_s = \{\{u, v\} : u, v \in V_s\}$.

The *neighborhood* of a vertex v (denoted by $\Gamma(v)$) is the set of vertices which share an edge with v, i.e. $\Gamma(v) = \{u : \{u, v\} \in E\}$. For a directed graph, $\Gamma(v) = \Gamma^+(v) \cup \Gamma^-(v)$, where $\Gamma^+(v)$ is the *out-neighborhood* and $\Gamma^-(v)$ is the *in-neighborhood*. While the former consists of the vertices whose arcs go out from v, the latter includes vertices whose arcs go into v:

$$\Gamma^+(v) = \{u : (v, u) \in E\}\,, \tag{1}$$

$$\Gamma^-(v) = \{u : (u, v) \in E\}\,. \tag{2}$$

In several contexts, such as scientific coauthorship, transportation, and recommendation (Newman, 2010), V contains vertices of different *types* (papers and authors / buyers and products / users, resources, and tags). This can be formally represented with a *k-partite* graph, where V is composed of k disjoint subsets and the edges in E join pairs of vertices that belong to different subsets. When $k = 2$, the graph is said to be *bipartite* and thus

[3] Edges usually consist of vertex pairs, but three or more vertices can be simultaneously joined by a single edge. In this case, the edge is referred to as *hyperedge* and the graph is called *hypergraph*.

$$\begin{aligned} V &= V_1 \cup V_2, && (3) \\ V_1 \cap V_2 &= \emptyset, \text{ and} && (4) \\ \forall e = \{u, v\}, e \in E &\Rightarrow (u \in V_i) \wedge (v \in V_j), i \neq j, && (5) \end{aligned}$$

where $V_i, V_j \in \{V_1, V_2\}$. Examples of bipartite graphs include author-paper, actor-movie, metabolite-reaction, and buyer-product networks.

More complex situations can be modeled with *heterogeneous information networks* (Sun et al., 2012, 2009; Wang et al., 2012; Ji et al., 2011), which not only take into account different vertex types but also consider different edge types (directed, undirected, weighted, ...) and connections between arbitrary pairs of vertices (including the same type). Formally, for any heterogeneous graph

$$V = V_1 \cup \ldots \cup V_n = \bigcup_{i=1}^{n} V_i \text{ and} \tag{6}$$

$$E \subseteq \{(u, v) : (u \in V_i) \wedge (v \in V_j), i, j \in \{1, \ldots n\}\}. \tag{7}$$

An example of a heterogeneous graph is the *roll call data network* presented by Wang et al. (2012), which consists of two vertex types and three edge types; while the vertices either stand for legislators or bills, the edges either join legislator pairs (cosponsorship), bill pairs (semantic similarity), or legislator-bill pairs (this last edge type is directed and represents votes).

2.3 The Vector Space Model

The *vector space model* is, so far, one of the central models for information retrieval (Liu, 2007). This model views each document as a *bag of words* (a representation where order is not important) and extracts a *weight vector* from this bag. Each vector's length is equal to the vocabulary (i.e. unique words) of the whole document collection. The weight $w_{i,j}$ for a given word k_i in a particular document d_j is usually awarded by a *text frequency–inverse document frequency* or *tf-idf* score; this score indicates the importance of k_i in d_j:

$$\begin{aligned} w_{i,j} &= \mathrm{tf}_{i,j} \times \mathrm{idf}_i \\ \mathrm{idf}_i &= \log \frac{N}{n_i} \\ \mathrm{tf}_{i,j} &= \frac{f_{i,j}}{\sum_{x=1}^{M} f_{x,j}}, \end{aligned} \tag{8}$$

where $f_{i,j}$ is the number of times that k_i appears in d_j, M is the amount of unique words in d_j, N is the amount of documents in the collection, and n_i is the number of documents that k_i appears in.

A common metric for calculating similarity between document vectors is the *cosine similarity* (Baeza-Yates and Ribeiro-Neto, 1999), which is based on the dot product of the vectors:

$$\mathrm{cosim}(d_a, d_b) = \frac{\mathbf{d_a} \cdot \mathbf{d_b}}{|\mathbf{d_a}| \times |\mathbf{d_b}|}, \tag{9}$$

where d_a and d_b are the documents, and $\mathbf{d_a}$ and $\mathbf{d_b}$ are the vectors. A similarity of 0 indicates that the documents have no common words and a similarity of 1 indicates that the documents are identical.

2.4 Fuzzy logic systems as linguistic processors

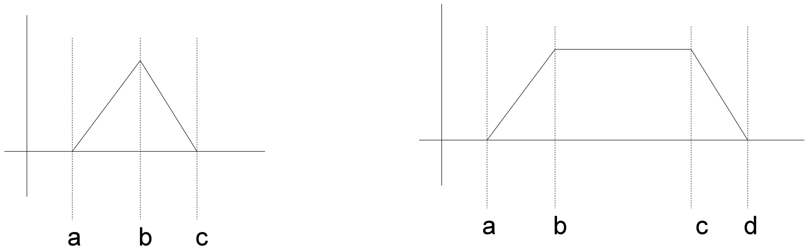
Fuzzy systems encode knowledge, which is commonly expressed as linguistic information. These systems are based on fuzzy logic and were originally proposed by Zadeh (1965). A *fuzzy system* maps a set of given inputs to an output by means of an *inference engine* that uses a fuzzy *rule base*. To perform inferences with this rule base, the inputs have to be *fuzzified*, and the fuzzy result is *defuzzified* to obtain the corresponding output. This process is illustrated in Figure 3.

The core of fuzzy system design is given by fuzzy sets, linguistic variables, and membership functions. A *fuzzy set* assigns a *membership value* to every element, as opposed to a conventional set where elements are either present or absent; e.g., in the set Old $= \{(45, 0.1), (65, 0.5), \ldots (90, 0.9)\}$, the element 90 has a membership value of 0.9. A membership value — whose range is between 0 and 1, inclusive — is assigned by a *membership function*. There are several forms of membership functions; however, the triangular and trapezoidal forms are commonly used:

$$\mathrm{triangular}(x, a, b, c) = \max\left(\min\left(\frac{(x-a)}{(b-a+\epsilon)}, \frac{(c-x)}{(c-b+\epsilon)}\right), 0\right), \tag{10}$$

$$\mathrm{trapezoidal}(x, a, b, c, d) = \max\left(\min\left(\min\left(\frac{(x-a)}{(b-a+\epsilon)}, 1\right), \frac{(d-x)}{(d-c+\epsilon)}\right), 0\right), \tag{11}$$

where x is a given element of the fuzzy set and $\epsilon = 1 \times 10^{-6}$ or another suitable constant. Parameters $a, b, c,$ and d are additionally illustrated in Figures 1a and 1b.



(a) *Triangular*

(b) *Trapezoidal*

Figure 1: Membership functions and parameters.

A *linguistic variable* is a variable associated with a numeric variable x and does not take numbers as values, but instead takes *linguistic values.* For example:

$$\text{Volume} = \{\text{“Low”}, \text{“High”}, \text{“Very high”}, \ldots\}.$$

Each linguistic value is related to a fuzzy set or membership function. A range of operation is defined for every linguistic variable and it is *partitioned* considering the linguistic values used. Usually, an odd number of linguistic partitions is used; three and five are the most common (see Figure 2).

As we previously mentioned, a fuzzy system is an integration of the logic operations shown in Figure 3. Because we can use different membership functions, fuzzifiers, inference engines, and defuzzifiers, there is a variety of fuzzy systems. In this paper, as we will see later, we used a fuzzy system with a singleton fuzzifier, trapezoidal and triangular membership functions, a Mamdani max-min engine, and a centroid defuzzifier. Let us explain each of these.

The *singleton fuzzifier* maps a value a into a fuzzy set A' by setting membership $\mu_{A'}(x)$ to 1 at $x = a$ and 0 at any other value:

$$\mu_{A'}(x) = \begin{cases} 1 & \text{if } x = a \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

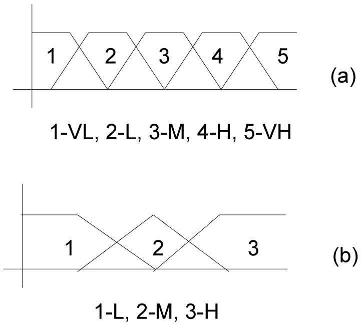


Figure 2: (a) Five fuzzy sets or linguistic values: 1="Very Low", 2="Low", 3="Medium", 4="High", 5="Very High", (b) Three fuzzy sets or linguistic values: 1="Low", 2="Medium", 3="High"

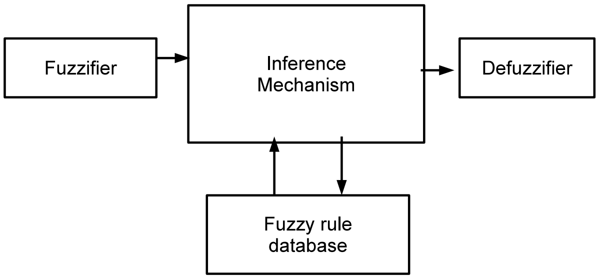


Figure 3: Components of a fuzzy system.

With respect to the inference engine and the fuzzy rule base, in first place, the fuzzy rule base is a matrix (which we shall denote by DB) where every row is a rule and every column is an integer that represents the linguistic value associated with that rule from antecedents to consequents. In the case of three variables and five linguistic values, a maximum of $5^3 = 125$ rules or rows and 4 columns (three inputs and one output) build up the matrix. Second, considering a normalized interval for every linguistic variable, a type1(x, n) function can be defined as follows:

$$\mathrm{type1}(x,n) = \begin{cases} \mathrm{trapezoidal}(x, 0, 0, 0.16, 0.33) & \text{if } n = 1 \\ \mathrm{triangular}(x, 0.16, 0.33, 0.5) & \text{if } n = 2 \\ \mathrm{triangular}(x, 0.33, 0.5, 0.66) & \text{if } n = 3 \\ \mathrm{triangular}(x, 0.5, 0.66, 0.83) & \text{if } n = 4 \\ \mathrm{trapezoidal}(x, 0.66, 0.83, 1, 1) & \text{if } n = 5, \end{cases} \quad (13)$$

where x is an input value and n corresponds to a specific membership function of a linguistic variable. Considering a fuzzy system of three inputs, a *max-min* inference engine is defined as follows:

$$I(r) = \min(\text{type1}(x_1, DB(r)), \text{type1}(x_2, DB(r)), \text{type1}(x_3, DB(r))) \quad (14)$$

where r is the fuzzy rule number, DB is the fuzzy rule base (as already stated), and I is the inference calculated. Finally, the output is defuzzified as follows:

$$y = \frac{\sum_{r=1}^{R}(I(r) \times y_m(DB(r)))}{\sum_{r=1}^{R} I(r)}, \quad (15)$$

where y_m is a precalculated center of mass of every membership function of the output (the predefined values are y_m =[0, 0.25, 0.5, 0.75, 1]) and R is the number of rules. $DB(r)$ establishes the correct value depending of the fuzzy rule.

3 Related Work

As we will see throughout this section, related work is centered on followee recommendation in Twitter and fuzzy recommendation. Twitter, in general, has been a subject of study for some years now. For example, several works analyze the “Twittersphere” as a complex network, including topologic and geographic properties (Java et al., 2007); the impact of retweets, user influence, and the longevity of trending topics (Kwak et al., 2010); and how users with similar interests are connected — a phenomenon known as *homophily* (Huberman et al., 2008). Other works include outcome prediction for the American presidential election (Tumasjan et al., 2010), stock market prediction (Sakaki et al., 2010), rumor propagation (Liu and Chen, 2011), and epidemics detection (Culotta, 2010).

Regarding followee recommendations in Twitter, Hannon et al. (2010, 2011) introduce the *Twittommender*, a real-time system where users can search for followees and receive followee recommendations as well. To generate these recommendations, a user profile of tf-idf weights is extracted and presented as a query to the system. Nine profiling strategies are proposed; five of these use a single information source and four combine two or several sources — either of the same or different types (content vs. structure). The strategies consist of: (1) user tweets, (2) followee tweets, (3) follower tweets, (4) followee id’s, (5) follower id’s, (6) a combination of user, followee, and follower tweets, (7) a combination of follower and followee id's, (8) a combination of user tweets and follower id's, and (9) an ensemble of strategies 1-7 where users which frequently have a high position in recommendations are preferred over users whose frequencies or positions are low. These profiling strategies were evaluated by measuring precision (overlap between the lists of recommended users and the actual followee lists of the target users) and ranking effectiveness (if relevant recommendations were placed in high positions of the list); a live trial where target users were asked if they would follow the recommended users was also carried out. Collaborative filtering strategies (4,5,7) gave, in general, better results than content-based strategies (1-3,6), although the latter tended to place relevant recommendations in higher positions.

Similar results are presented by Armentano et al. (2011a,b), who propose two recommenders: purely based on content and purely based on structure. For the content-based recommender, a set of users is randomly selected from Twitter's *public timeline* (a stream where the tweets of public accounts are added) and represented with term-vector profiles; cosine similarity between these users and the target user (whose profile is also a vector) is calculated and users exceeding a given threshold are added into the recommendation list. For the topology-based (structural) recommender, the followees of the target user's *co-followers* (i.e. the group of users who share followees with the target user) are obtained and ranked according to a score that attempts to balance popularity and resemblance to user preferences; this score takes the product of a candidate recommendation's proportion of repetitions (a higher score is awarded to users recommended several times), proportion of followers vs. followees, and proportion of mentions. Results were evaluated using discounted cumulative gain, precision, and mean reciprocal rank; even when no statistically significant differences were found between the two recommenders, the content-based approach showed to position relevant recommendations at the top of the list. Other recommenders that make use of structural information are given in the works by Hsu et al. (2006) and Silva et al. (2010). The former describes a hybrid recommender that employs graph proximity and logistic regression to suggest friends in the LiveJournal weblog, and the latter uses complex network theory and genetic algorithms to recommend friends of friends in the Oro-Aro SNS.

While several works treat recommendation as an information retrieval problem, Tsourougianni and Ampazis (2013) present a followee hybrid recommender based on classification. The approach, which receives the target user (or target user and followee) tweets as a query, locates the target user's friends-of-friends (*FoF's*) and classifies their tweets as either positive or negative for recommendation; the users with the highest percentage of positives are added to the recommendation list. The features for the classifier (a Markovian classifier) consist of Orthogonal Sparse Bigrams, which are extracted per tweet. As in other cases, the approach was evaluated using precision. While most of the results yielded a high precision, the authors imply that the presence of *bots* in Twitter could cause outlier results.

Followee recommendations in Twitter are also tackled by Gavilanes et al. (2013) with a system that, unlike others, is built on top of *human-generated* recommendations. To show that these influence users' behavior towards following new people, the authors study a 24-week corpus created with manual recommendations from the *#followfriday* (or *#ff*) trend, which consists of users recommending their followees other users to follow (celebrities, authorities in a certain topic, and the like); these recommendations are made on Fridays, hence the name of the hashtag. To rank the human-generated recommendations, several features are extracted and used with the Rotation Forest algorithm to train a binary classifier. The features are grouped into three categories: user, relation, and format. While the first category includes user popularity and level of activity in Twitter, the second considers level of communication between users and similarity (content-based and geographical), and the third includes recommendation repetition (i.e. how many times a user has been recommended) and context (e.g. day of the week). The results were evaluated using mean average precision, and the relation features were found to be the most useful; on the contrary, the format features were the least useful.

Another recent approach concerns the *Twilite* system, which is introduced by Kim and Shim (2013); preceded by a similar model-based approach named *Twitobi* (Kim and Shim, 2011), the Twilite system sits on Latent Dirichlet Allocation and matrix factorization, which help to recover the process of tweeting and following users. The resulting probabilistic generative model (whose parameters are learned via expectation maximization) is used for followee and tweet recommendation.

Still within the context of recommendation in social media are the seminal works by Chen et al. (2009, 2010) and Liu and Lee (2010). The first of these works (Chen et al., 2009) proposes and evaluates four algorithms that are applied on Beehive, an enterprise social networking site of IBM. One of these algorithms is purely content-based, as it creates tf-idf vectors for user profiles and generates the recommendation list based on cosine similarity; the second algorithm (which is hybrid) extends the former by increasing similarity if the users are linked. The third algorithm, purely structural, is based on FoF's and mutual friendships, and the last algorithm (named

SONAR) is leveraged by organizational information, such as patents, the organizational chart, and project wikis. It is important to note that, for every recommendation, an explanation is provided (this feature being also used in works by other authors). To assess the four algorithms, the authors conducted a survey where users were asked a variety of questions regarding the received recommendations; another form of evaluation consisted of a controlled field study where a recommender widget was introduced in Beehive and monitored for a sample of users. While all algorithms showed strengths and weaknesses, in general SONAR was considered as a fair alternative, since it takes advantage of more information. Posterior works by these authors focus on Twitter, specifically on URL recommendation with topic relevance and social voting (Chen et al., 2010). With respect to the work by Liu and Lee (2010), this work — which is applied on the Cyworld SNS — shows that collaborative filtering can be enhanced by combining nearest neighbors with friends for item recommendation.

The approach by Li et al. (2011) suggests experts by using a fuzzy linguistic method and fuzzy text categorization to analyze task documents. Other items recommended using a fuzzy engine include electronic products (Cao and Li, 2007), music (Park et al., 2006), and academic resources inside universities (Porcel and Herrera-Viedma, 2010); fuzzy logic has been used, as well, for learning and constructing user profiles (Castellano et al., 2010; Xiang-Wei et al., 2009). More recently, a friend-to-friend recommendation system is proposed by Yigit et al. (2015), where a classification of data in four categories is made and a mathematical equation is used to generate a recommendation; this approach performs better than extended FoF, a graph-based system, and a conceptual fuzzy set based algorithm. Also, a fuzzy system is used to extract useful interaction behavior of Twitter users (Fu and Shen, 2014). Collective user behavior is analyzed to generate tweet *reply* or *not reply* categories by means of fuzzy association rules that consider activity time, number of friends/followers, and number of tweets as influential factors. Other fuzzy systems have been used for sentiment analysis on social networks, for example, the analysis of opinions generated by Twitter and Facebook with the use of fuzzy propagation modeling (Trung and Jung, 2014). In the approach by Bollen et al. (2011), a self-organized fuzzy neural network is used to relate opinions generated in Twitter by mood tracking tools with changes in Dow Jones industrial average over time.

3.1 Discussion

Reviewing the aforementioned related approaches, we can see that none of them does followee recommendation in Twitter using a fuzzy-based method (hence the uniqueness of the solution we propose). The state of the art, to the best of our knowledge, instead poses two kinds of approaches: followee recommendation in Twitter using various methods and fuzzy approaches for various tasks and contexts (other than followee prediction and recommendation in Twitter). Let us discuss related work of each kind.

The followee recommendation task is not new and has been explored previously; however, a number of the proposed methods have been developed over less complex (smaller, homogeneous, less noisy) contexts. Considering that Twitter is by no means a simple context, the few works that actually concentrate on this social network still have room for improvement — either talking about precision, efficiency, or robustness. For example, the works by Gavilanes et al. (2013) and Hannon et al. (2010) use a considerable number of variables (the greater the number of variables, the more time it takes to extract this information), while our approach proposes only three. On the aspect of precision, the work by Armentano et al. (2011b) reports results that could be further improved. On the aspect of robustness, Tsourougianni and Ampazis (2013) state that bots (noise) affect the performance of the approach, and for this reason our approach is based on fuzzy logic. In summary, our approach attempts to improve — in any aspect — the current state of the art, or at least bring in new knowledge about followee recommendation in Twitter.

With respect to fuzzy expert systems for various tasks and contexts, as we have seen, these approaches so far have not been devoted to the task of followee prediction and recommendation in Twitter; this is, therefore, an opportunity area for the field of expert systems. Tackling this opportunity area could bring novel perspectives on challenging problems and show that the solution proposed by expert systems keeps fitting into new contexts (no need of "reinventing the wheel").

As a result, it is possible to state that our contribution lies within two areas: fuzzy expert systems and followee recommendation in Twitter and similar contexts. For the fuzzy expert systems area, our contribution consists of a fuzzy system capable of predicting the degree of friendship between a pair of users of a *sui generis* (massive, dynamic, heterogeneous) social network; for the followee recommendation area, our contribution consists of a robust, efficient approach that combines several information types (text, structure) in a manner that has not been tried before.

4 Approach

To recommend followees, we designed a fuzzy system that takes as input three similarity values between a target user u_i and a candidate user u_j: comment (tweet) similarity, followee similarity, and followee tweet similarity. The rules of the system produce the predicted degree of friendship (follower-followee relation) between the users, and this value is used to determine if u_j is worth recommending. To calculate the similarities, we first extract user profiles.

4.1 Generation of user profiles

As previously mentioned, a profile contains information that allows to characterize the user and measure similarity with respect to other users, thus enabling recommendations. We consider that a user profile can be constructed with (a) the *comments of the user*, (b) the *comments of the user's followees*, and (c) the *id's of the followees*. This approach for generating user profiles includes both content and structure: while the former is given by comments that comprise topics and opinions, the latter is given by links that reveal how the user is connected. It also — from the recommender systems perspective — combines both content-based and collaborative filtering strategies by employing information from the user and information from contacts, respectively. In that sense, the approach is *hybrid*.

Upon generating a user profile, it is important to take into consideration the amount of data per user; when this amount is not easily manageable, the profile might as well be generated with a random sample of comments and contacts. To the best of our knowledge, there is no standardized number of followees to include or the criteria to select these.

4.1.1 Formal conceptual framework for profile generation

Let us formally depict the profile of a user u_i with a triplet

$$p_i = (\mathbf{c_i}, \mathbf{e_i}, \mathbf{f_i})$$

of document vectors, where the vector $\mathbf{c_i}$ represents the comments of u_i, the vector $\mathbf{e_i}$ represents the comments of u_i's followees, and the vector $\mathbf{f_i}$ represents the id's of u_i's followees; for later explanation, let us assume that a function $f_p(u_i)$ produces p_i. All vectors consist of tf-idf weights that are computed by conceiving Twitter as a heterogeneous information network and constructing documents from this network.

Let $G = (V, E)$ represent a heterogeneous information network of *users* and *comments*. G is unweighted, vertex-labeled, and consists of two subnetworks: a follower-followee directed network $G_f = (V_f, E_f)$ and a user-comment undirected bipartite network $G_{cf} = (V_{cf}, E_{cf})$. An example of this kind of network is illustrated by Figure 4.

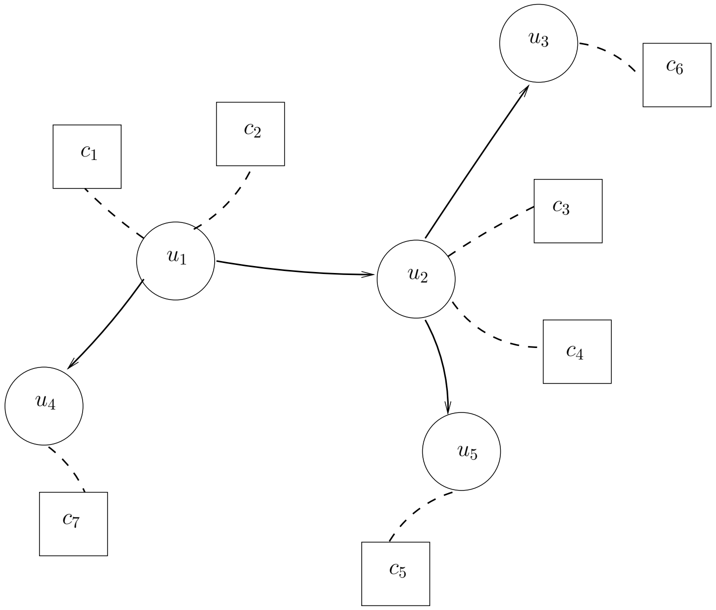


Figure 4: Example of a heterogeneous information network.

For G_f, each vertex $v \in V_f$ stands for a user and each arc $(a, b) \in E_f$ represents a "who-follows-who" relationship where a follows b; these arcs only join pairs of vertices in V_f, such that $E_f \subset V_f \times V_f$. With respect to G_{cf}, it represents a set of comments V_c generated by the set of users V_f (note that $V_c \cup V_f = V_{cf} = V$); an edge $\{a, c\} \in E_{cf}$ where $a \in V_f$ and $c \in V_c$ indicates that user a made comment c. From our example, we can see that $V_f = \{u_1 \dots u_5\}$, $V_c = \{c_1 \dots c_7\}$, $(u_1, u_2) \in E_f$, and $\{u_1, c_1\} \in E_{cf}$.

To retrieve followees and comments from users, let us introduce three kinds of neighborhoods on the vertices of G: the f-neighborhood, the comment neighborhood, and the extended comment neighborhood.

The *f-neighborhood* of a vertex i, which we will denote by $\Gamma_f^+(i)$, is the out-neighborhood of i in G_f and represents the set of followees for a given user (see Eq. 1 in Section 2.2). In our example, $\Gamma_f^+(u_2) = \{u_3, u_5\}$.

The *comment neighborhood* of a vertex $i \in V_f$, which we will denote by

$\Gamma_c(i)$, is the neighborhood of i in the bipartite subgraph G_{cf} and represents the set of comments (tweets) for a given user:

$$\Gamma_c(i) = \{v : \{v, i\} \in E_{cf}, i \in V_f\}\,.$$

In our example, $\Gamma_c(u_2) = \{c_3, c_4\}$.

The *extended comment neighborhood* of a vertex $i \in V_f$, which we will denote by $\Gamma_e(i)$, corresponds to the comment neighborhoods of the f-neighbors for i and represents the comments of the followees for a given user:

$$\Gamma_e(i) = \{v : (i, j) \in E_f, v \in \Gamma_c(j)\}\,.$$

Regarding our example, $\Gamma_e(u_2) = \Gamma_c(u_3) \cup \Gamma_c(u_5) = \{c_5, c_6\}$.

It is also important to introduce a function $L(i)$ that maps a vertex i to a label l_i, where the latter consists of a user id for $v \in V_f$ or a comment (text) for $v \in V_c$.

Taking the previous definitions into consideration, three types of documents can be generated per user: the *comment document*, the *extended* comment document, and the *followee document*. These document types all emerge from the same procedure: the concatenation of labels that belong to a given neighborhood. Let us represent this procedure with a function

$$\Phi(N) = \bigparallel_{j \in N} L(j).$$

In this case $c_i = \Phi(\Gamma_c(i))$, $e_i = \Phi(\Gamma_e(i))$, and $f_i = \Phi(\Gamma_f^+(i))$. For our example, let us assume that $L(u_2) = \text{“@newton”}, L(c_3) = \text{“shoulders of giants”}$, $L(c_4) = \text{“action and reaction”}, L(u_3) = \text{“@archimedes”}, L(c_6) = \text{“eureka”}$, $L(u_5) = \text{“@galileo”}$, and $L(c_5) = \text{“it moves”}$. Here, $c_{u_2} = \text{“shoulders of giants action and reaction”}, f_{u_2} = \text{“@galileo @archimedes”}$, and $e_{u_2} = \text{“eureka it moves”}$.

By extending this concept, three corpora are constructed from G: a comment corpus C, an extended comment corpus E, and a followee corpus F. Each corpus consists of the union of its respective documents. Formally,

$$X = \bigcup_{i \in V_f} x_i$$

where $X \in \{C, E, F\}$ and $x_i \in \{c_i, e_i, f_i\}$.

Taking these considerations into account, the calculation of the tf-idf scores for the document vectors $\mathbf{c_i}$, $\mathbf{e_i}$, and $\mathbf{f_i}$ is straightforward from Equation 8.

Given two profiles p_i and p_j, we create a *similarity vector* where each element corresponds to the cosine similarity (Eq. 9) between document vectors of the same type. Let us denote this vector with $\mathbf{s}_{(i,j)} = \left[c_{(i,j)}, e_{(i,j)}, f_{(i,j)}\right]$, where $c_{(i,j)} = \text{cosim}(\mathbf{c_i}, \mathbf{c_j})$, $e_{(i,j)} = \text{cosim}(\mathbf{e_i}, \mathbf{e_j})$, and $f_{(i,j)} = \text{cosim}(\mathbf{f_i}, \mathbf{f_j})$. For further explanation, let us assume that a function $f_{\text{sim}}(u_i, u_j) = \mathbf{s}_{(i,j)}$.

For supervised learning approaches, the similarity vectors can actually be turned into a matrix $\mathbf{S}$, where each row is an input vector. An additional vector $\mathbf{y_D}$ of desired outputs is needed and serves for the training and test sets; each output $o_{(i,j)} \in \{0, 1\}$ of this vector indicates whether the users are contacts or not:

$$\mathbf{S} = \begin{bmatrix} c_{(1,2)} & e_{(1,2)} & f_{(1,2)} \\ \vdots & \vdots & \vdots \\ c_{(n-1,n)} & e_{(n-1,n)} & f_{(n-1,n)} \end{bmatrix} \text{ and } \mathbf{y_D} = \left[o_{(1,2)}\, o_{(1,3)} \dots o_{(n-1,n)}\right]^{\mathrm{T}}. \quad (16)$$

4.2 Fuzzy system design

As previously stated, our fuzzy system has three inputs and one output. The linguistic variables for the inputs are *tweet similarity*, *followee tweet similarity*, and *followee similarity* between a pair of users, while the output is the predicted degree of friendship between these users; this result could also be interpreted as the probability of a link (follower-followee relationship) creating between the users. A diagram of the fuzzy system is shown in Figure 5.

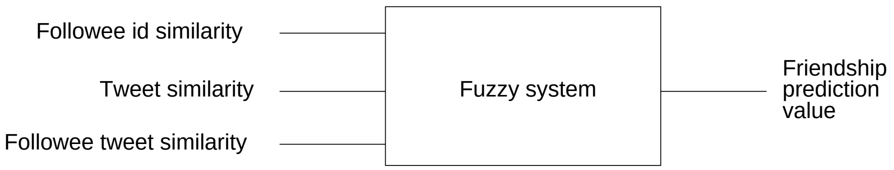


Figure 5: Architecture of the fuzzy system for recommendation prediction.

Each input variable has nine partitions, i.e. nine linguistic values (fuzzy sets): *Extremely low*, *Very low*, *Considerably low*, *Low*, *Medium*, *High*, *Considerably high*, *Very high*, and *Extremely high*. As we can see from Figure 6, the fuzzy sets at both extremes are trapezoidal functions (half-trapezoids) and the rest are triangular functions; these functions are not only common but also computationally cheap. These functions are described in Eq. 17:

$$\text{type1}(x,n) = \begin{cases} \text{trapezoidal}(x,0,0,0.1,0.2) & \text{if } n=1 \\ \text{triangular}(x,0.1,0.2,0.3) & \text{if } n=2 \\ \text{triangular}(x,0.2,0.3,0.4) & \text{if } n=3 \\ \text{triangular}(x,0.3,0.4,0.5) & \text{if } n=4 \\ \text{triangular}(x,0.4,0.5,0.6) & \text{if } n=5 \\ \text{triangular}(x,0.5,0.6,0.7) & \text{if } n=6 \\ \text{triangular}(x,0.6,0.7,0.8) & \text{if } n=7 \\ \text{triangular}(x,0.7,0.8,0.9) & \text{if } n=8 \\ \text{trapezoidal}(x,0.8,0.9,1,1) & \text{if } n=9 \end{cases} \tag{17}$$

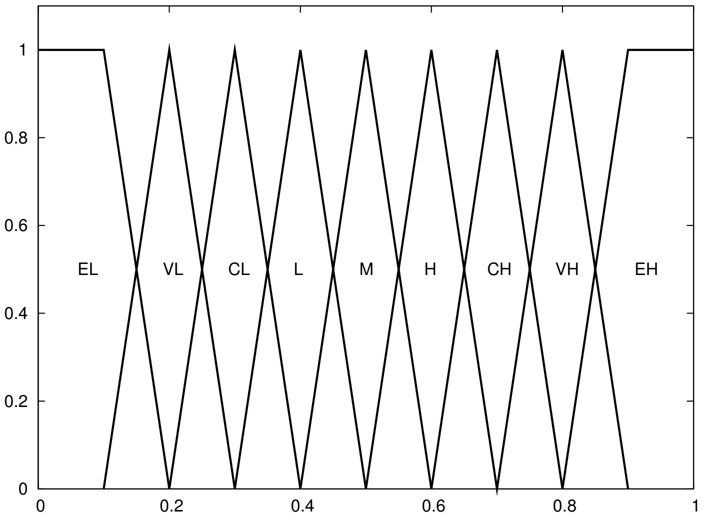


Figure 6: Linguistic values used (represented as memberships functions). EL="Extremely low", VL="Very low", CL="Considerably low", L="Low", M="Medium", H="High", CH="Considerably high", VH="Very high", EH="Extremely high".

To fuzzify the inputs, the singleton method (Eq. 12) is used, as this is the one usually employed and the input does not require further processing. With respect to the fuzzy rule base, there are $9^3 = 729$ rules possible; however, a smaller set was derived by manually analyzing 200 cases in which a pair of users was connected (had a follower-followee relationship). Regarding the inference engine, we use Mamdani's max-min engine (Eq. 14). To defuzzify the output, the centroid (center of mass) method (Eq. 15) is used.

4.2.1 Recommendation algorithm

To make recommendations for a target user, we select a set of candidate followees (a suitable choice is to use a friends-of-friends or *FoF* approach) and include in the recommendation list those candidates that exceed an acceptance threshold τ for the prediction value obtained by the fuzzy system; in our experiments, τ was set to 0.5. Additionally, only the top k elements of the list may be chosen (in this paper, we do not apply this filter). This procedure is shown in Algorithm 1.

Algorithm 1 Recommendation algorithm

```
Description: Receives a target user $u_i$, a set $C$ of candidate followees, an acceptance threshold $\tau$, and a list size $k$. Returns a recommendation list $L_k$.
function RECOMMEND($u_i$, $C$, $\tau$, $k$)
    $p_i \leftarrow f_p(u_i)$
    $L \leftarrow ()$
    for all $u_c \in C$ do
        $p_c \leftarrow f_p(u_c)$
        $\mathbf{s}_{(\mathbf{i},\mathbf{c})} \leftarrow f_{\text{sim}}(p_i, p_c)$
        $y_{(i,c)} \leftarrow \text{FS}(\mathbf{s}_{(\mathbf{i},\mathbf{c})})$        ▷ Get prediction value from fuzzy system (FS).
        if $y_{(i,c)} \geq \tau$ then
            append($u_c, L$)
        end if
    end for
    sort($L$)                                  ▷ Sort $L$ according to prediction values.
    $L_k \leftarrow L[1:k]$                        ▷ Get the top $k$ elements of $L$.
    return $L_k$
end function
```

5 Experiments and Results

To test our approach, we created 30 datasets with information crawled from Twitter and measured recommendation accuracy for each dataset, as it is conventionally done for evaluating recommender systems; we compared our results against a supervised variant of our approach, two state-of-the-art methods (previously described in Section 3) and a baseline. With the intent of evaluating our approach in other contexts, we also created a pair of datasets for the Arxiv High-Energy Physics Theory citation network (Cit-HepTh) and obtained accuracy for these datasets as well. Finally, to assess the performance of input variables (followee similarity, tweet similarity, and followee tweet similarity), we calculated the correlation of each input variable with respect to the obtained output.

5.1 Experimental Setup

Sample networks. Because the Twitter network is very large to handle, experiments were performed using smaller sample networks, where each sample network consisted of a heterogeneous network such as the one described in Section 4; each sample network had one *target user*, the *followees* of this target user, and a number of *non-contacts* (i.e., users who were neither followees nor followers of the target user); the main idea was for the fuzzy system to be able to recover the target user's followees as recommendations. As such, we evaluated the fuzzy system by calculating the accuracy (percentage of matches) between the resulting recommendation list and the actual followee list of the network's target user. Since sample networks were grouped into *datasets*, we obtained the average accuracy per dataset.

Twitter data. To generate the sample networks, a repository of users and comments was first crawled from Twitter[4]. We collected a *seed set* of 100 users, which resulted from searching "Monterrey" (a northeastern city of Mexico) in Twitter and selecting those users that complied with four criteria: having Monterrey as their location, writing in Spanish, having at least 30 followees, and having no more than 3,000 followers. The last two criteria intended to discard companies and celebrities such as singers, actors, and famous politicians; processing this kind of users is left as future work. To have a larger repository, the seed set was expanded by including those followees and followers that complied with the criteria previously mentioned; the expansion was breadth-first, such that the initial level consisted of seed users, the second level consisted of seed user contacts, the third level of contact contacts, and so on. The current repository contains 18,499 users[5]. From this repository, we randomly selected 6,000 users and approximately 2 million comments. These comments were filtered (stopwords were removed) and spelling correction was performed.

Twitter datasets. With the data from Twitter, we constructed 3,000 heterogeneous sample networks, which were equally distributed to create 30

[4] All information was obtained using the Twitter API, available at `https://dev.twitter.com/`.

[5] This repository has also been used for other studies involving Twitter, such as opinion mining in Spanish (Rodríguez, 2013).

datasets with 100 networks each. While the number of users (size) of each network was variable, on average, there were 24.44 users per network.

Cit-HepTh datasets. To assess our approach in a different context, we also created datasets for the Arxiv High-Energy Physics Theory (Cit-HepTh) citation network[6], which is analogous to the Twitter heterogeneous network. In this case, we are dealing with *reference recommendation*, i.e. recommending which paper to cite. In that sense, papers can be seen as equivalent to users, a paper b that is cited by a paper a can be seen as a's followee, and the abstract of a is equivalent to the set of a's tweets. For the case of Cit-HepTh, we generated two datasets, where each dataset consisted of 30 heterogeneous sample networks. Each network contained 100 papers on average.

Comparative experiments. With the intent of having a wider perspective on the effectiveness of the fuzzy approach, we performed a comparison against two state-of-the-art methods: Twittommender (Hannon et al., 2010) and the co-followers structural strategy followed by Armentano et al. (2011b); for Twittommender, we used the ensemble strategy. We additionally introduced a baseline, which consisted of a simple friends-of-friends (FoF) approach — i.e., recommending the followees of the target user's followees (note that this would yield a high accuracy if the followee and FoF lists had a considerable overlap).

Our supervised variant. Another point of comparison was given by the *supervised variant* of our approach, which was implemented to further assess the combination of our three input variables. This supervised variant consisted of a *multilayer perceptron* neural network (we ran experiments using WEKA[7] and its default parameters). We shall refer to this approach as "NN variant" or simply "NN".

5.2 Results and Discussion

A summary of our results is shown in Table 1 (note that this is the average of the average results per dataset), and a box plot for the Twitter datasets is depicted in Figure 7. As we can see from these results, our approaches (the fuzzy system and the neural network) are highly competitive with regard

[6] This network is available as part of Stanford's Large Network Dataset Collection. URL: `http://snap.stanford.edu/data/cit-HepTh.html`.

[7] Available at: `http://www.cs.waikato.ac.nz/ml/weka/`.

to the rest of the methods; the neural network, in particular, clearly outperforms all competitors (the difference with respect to Twittommender in the Twitter datasets is, in fact, statistically significant with $p = 1.16 \times 10^{-6}$ using a two-sided paired t-test). This suggests that our three input variables act as reasonable link predictors and are thus feasible for recommendation. Moreover, let us note that almost all methods performed better on the Cit-HepTh datasets, except for the co-followers strategy, which seems to be better tailored for Twitter. This apparently suggests that Twitter's nature is more "messy" (e.g. user's comments) and probably more chaotic; it could be therefore convenient to employ the *sui generis* features of Twitter to try to improve accuracy. However, it would be desirable to have more results (more datasets) on the Cit-HepTh network to have a more confident point of view on this perspective. This is left as future work.

Table 1: Average recommendation accuracy (30 Twitter datasets, two Cit-HepTh datasets). FS=Fuzzy system, NN=supervised variant, TM=Twittommender, CF=Co-followers, FoF=friends-of-friends (baseline).

	FS	**NN**	**TM**	**CF**	**FoF**
Twitter	61.57%	**71.75%**	65.13%	29.99%	3.15%
Cit-HepTh	83.25%	86.77%	**86.86%**	10.84%	8.74%
Global	72.41%	**79.26%**	75.99%	20.41%	5.94%

As we can also see from the results, the fuzzy system is a strong competitor both inside and outside Twitter. Although it ranks third, there is an evident advantage over the baseline and the co-follower strategy (needless to say, there is a statistically significant difference with these approaches, since $p < 2.2 \times 10^{-16}$); in that sense, our approach remains competitive. With respect to Twittommender, we can see that the fuzzy system's accuracy lies behind for approximately 3%, but let us also note that the fuzzy system is only using three variables whereas Twittommender is using five (user tweets, followee id's, followee tweets, follower tweets, and follower id's), i.e. it needs to extract more information. Another disadvantage of Twittommender is that it is more of a "hit or miss" approach — at least with our Twitter results — and this can be appreciated in the box plot (Figure 7). On the contrary, the fuzzy system tended to give more stable results.

With respect to the NN variant, the fuzzy system also stays behind.

In general, neural networks are more precise than fuzzy systems, but they are also black boxes and no explanation can be extracted from them; on the other hand, fuzzy systems explain the relationship between inputs and outputs by means of linguistic variables and rules. For example, consider the rule:

IF v_1 is *Low* and v_2 is *Medium High* and v_3 is *Extremely High* THEN y is *Very High*,

where v_1 is user tweet similarity, v_2 is followee tweet similarity, v_3 is followee id similarity, and y is the possibility of being contacts. In a linguistic manner, this rule explains that *if there is a low user tweet similarity and there is a medium high followee tweet similarity and there is an extremely high followee id similarity, then there is an extremely high possibility to be contacts.*

We can choose any fuzzy rule and make the same interpretation. Fuzzy systems have the disadvantage of not being very accurate against other paradigms such as neural networks; however, because NN's are black boxes, it is very difficult to establish an interpretation of the relationship between inputs and outputs. Moreover, a fuzzy system has a high interpretability because it generates rules that explain in an explicit and linguistic way the relationship between variables.

5.2.1 Individual variables

As we stated previously, with a fuzzy approach it is possible to know how each input variable influences the output. To measure the individual influence of input variables, we calculated the correlation coefficient between each input variable and the output. Our results for both Twitter and Cit-HepTh are displayed in Table 2. As we can see from this table, for both cases, the variable with the highest influence on the fuzzy system corresponds to *followee tweets.* This result allows us to gain insight on the factors that affect prediction, and thus make adjustments for future work.

5.2.2 Summary

In summary, our results show that the fuzzy approach is suitable for link prediction in recommender systems, both in Twitter and in other contexts. The approach is highly competitive with other methods. In some cases, it clearly surpasses the accuracy obtained by its competitors; in other cases, it stays behind by a small percentage but excels in other aspects, such as

Table 2: Correlation between input variables and prediction output.

	Twitter	Cit-HepTh
User tweets	0.12	0.43
Followee id’s	0.42	0.32
Followee tweets	0.85	0.65

amount of information used and capability of explanation. This last goal is important for recommender systems in general (Bonhard and Sasse, 2006).

6 Conclusions and Future Work

A new fuzzy system for followee recommendation in Twitter has been proposed in this paper. This system handles recommendation mainly as a link prediction problem and uses three linguistic variables that express similarity between users: tweet similarity, followee tweet similarity, and followee similarity. Every similarity contributes in a different way for the prediction. To calculate these similarities, a profile is built for each user. The profile consists of vectors that contain information from comments and followee relationships, and this information is extracted by considering the Twitter social networking site as a heterogeneous information network.

From an objective point of view, the approach has both strengths and weaknesses. On one hand, as with any expert system, creating the rules requires knowledge and is an additional overhead of the approach; also, as we could see from the results, accuracy is still below other methods and could still be further improved. On the other hand, the system only requires three types of information (on the contrary of other methods which require more), is self-explanatory, and is capable of producing a *degree* instead of only a *yes/no* binary answer. The self-explanatory capability, in particular, allows the system to explain how variables interact to produce an output, as opposed to “black-box” methods, such as neural networks or support vector machines. With respect to the friendship degree, besides helping to construct the recommendation list for a particular user, it can also be used to carry out other tasks, such as discovering latent user communities or clusters.

With respect to the impact of this work, our fuzzy approach brings in a new perspective on the use and construction of expert systems for link prediction and contact recommendation in microblog-oriented social networks. Our method, in particular, brings new knowledge on how to address recommendation in a *sui generis*, heterogeneous, massive, dynamic environment, which is subject to inaccuracies (think, for example, about language inconsistencies) and incomplete information. One of our main findings, in particular, reveals that the combination of user tweet similarity, followee id similarity, and followee tweet similarity yield accurate followee recommendation results (these results are even better when the combination is used in a supervised approach); in that sense, the use of only three information types is promising — specially in the neural network case. Even when the fuzzy approach can be further improved, we have taken the first step towards a robust, efficient, accurate solution to the problem. Another theoretical contribution that it is important to highlight and that might be of interest is the conceptual model of Twitter as a heterogeneous network, as this model not only facilitates the understanding of Twitter's underlying dynamics, but also serves as a basis for other data mining tasks, such as cyberbullying detection and interest mining.

As future research directions, we propose several courses of action. One of these consists of modifying the existing rule base to incorporate more input variables (e.g. follower tweets, follower id's, retweets, multimedia content, etc.) with the intent of improving accuracy and exploiting Twitter's unique features; of course, this implies an additional overhead and the tradeoff between quality and efficiency must be assessed. Now, beyond the sole introduction of more variables, lies the issue of finding an *optimal* design for the fuzzy system (fuzzification and defuzzification methods, membership functions, partitions, etc.); in that sense, adaptive methods — such as genetic algorithms, PSO, and similar — could be employed to find this optimal design.

Another important future research direction concerns testing and adapting the fuzzy expert system for other contexts, both inside and outside Twitter. In that sense, it seems valuable to discover how general the system could be, and if it is possible to have a tailored version enhanced for Twitter (or a specific language, such as Chinese or Arab, whose alphabet is different) and a general version for heterogeneous networks analogous to Twitter (such as citation networks). Consequently, there is still plenty of room to create new "flavors" of the original fuzzy expert system. Related to this is the *hybridization* of the fuzzy system with a supervised technique to create a more powerful approach.

A final future research direction is related to community discovery and graph clustering. Because the fuzzy system produces a friendship degree for a pair of users, it is possible to view this degree as a network weighted edge and use this information to construct a friendship network (or at least parts of) and develop a local probabilistic graph clustering algorithm to find groups of similar users in Twitter (or other complex networks). On the contrary of the *follow graph*, which would be normally used for the clustering task in Twitter, this friendship network constructed from the fuzzy system could give new insights to community discovery in Twitter and social networks in general.

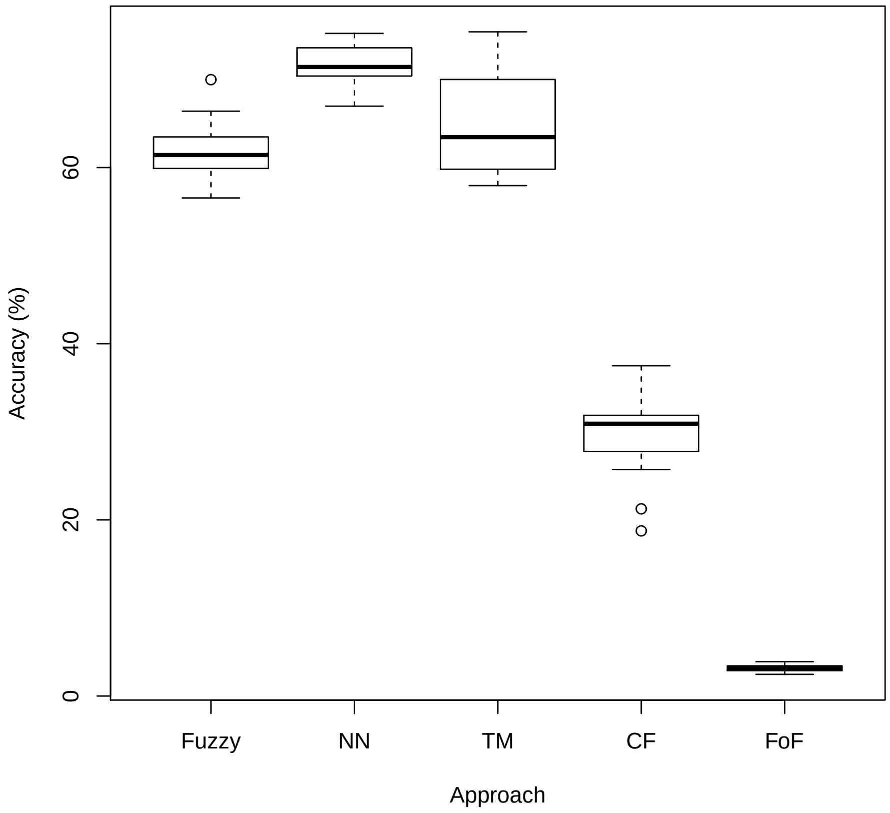


Figure 7: Comparison using the Twitter datasets. Each box shows the minimum, first quartile, median, third quartile, and maximum for the accuracy obtained from the 30 datasets (one box per approach). NN= neural network (multilayer perceptron), TM= Twittommender, CF=co-followers strategy, FoF=friends-of-friends strategy.

References

ARMENTANO, M. G., GODOY, D., and AMANDI, A. (2011a) Towards a followee recommender system for information seeking users in Twitter, in *Proceedings of the Workshop on Semantic Adaptive Social Web (SASWeb 2011), CEUR Workshop Proceedings*, volume 730, 27–38.

ARMENTANO, M. G., GODOY, D. L., and AMANDI, A. A. (2011b) A topology-based approach for followees recommendation in Twitter, in *Proceedings of the 9th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems*, Barcelona, Spain.

BAEZA-YATES, R. and RIBEIRO-NETO, B. (1999) *Modern information retrieval.* ACM Press, New York, NY, USA.

BENITO-RUIZ, E. (2009) *Handbook of research on Web*, chapter Infoxication 2.0, 60–79. IGI Global. doi: 10.4018/978-1-60566-190-2.ch004.

BOLLEN, J., MAO, H., and ZENG, X. (2011) Twitter mood predicts the stock market, *Journal of Computational Science*, 2(1), 1–8.

BONHARD, P. and SASSE, M. (2006) Knowing me, knowing you: Using profiles and social networking to improve recommender systems, *BT Technology Journal*, 24(3), 84–98.

CAO, Y. and LI, Y. (2007) An intelligent fuzzy-based recommendation system for consumer electronic products, *Expert Systems with Applications*, 33(1), 230–240.

CASTELLANO, G., CASTIELLO, C., DELL'AGNELLO, D., FANELLI, A., MENCAR, C., and TORSELLO, M. (2010) Learning fuzzy user profiles for resource recommendation, *International Journal Of Uncertainty, Fuzziness, and Knowledge-based Systems*, 18(4), 389–410.

CHEN, J., GEYER, W., DUGAN, C., MULLER, M., and GUY, I. (2009) Make new friends, but keep the old: Recommending people on social networking sites, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems:* ACM, 201–210.

CHEN, J., NAIRN, R., NELSON, L., BERNSTEIN, M., and CHI, E. (2010) Short and tweet: experiments on recommending content from information streams, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*: ACM, 1185–1194.

CULOTTA, A. (2010) Towards detecting influenza epidemics by analyzing Twitter messages, in *Proceedings of the 1st Workshop on Social Media Analytics*: ACM, 115–122.

FU, X. and SHEN, Y. (2014) Study of collective user behaviour in Twitter: a fuzzy approach, *Neural Computing and Applications*, 25(7-8), 1603–1614.

GANTZ, J. and REINSEL, D. (2010) The digital universe decade: Are you ready? IDC Review.

GAVILANES, R. G., O'HARE, N., AIELLO, L. M., and JAIMES, A. (2013) Follow my friends this Friday! An analysis of human-generated friendship recommendations, in *Social Informatics*: Springer, 46–59.

GETOOR, L. and DIEHL, C. (2005) Link mining: a survey, *ACM SIGKDD Explorations Newsletter*, 7(2), 3–12.

HANNON, J., BENNETT, M., and SMYTH, B. (2010) Recommending Twitter users to follow using content and collaborative filtering approaches, in *Proceedings of the 4th ACM conference on Recommender Systems*: ACM, 199–206.

HANNON, J., MCCARTHY, K., and SMYTH, B. (2011) Finding useful users on Twitter: Twittomender the followee recommender, in *Advances in Information Retrieval*: Springer, 784–787.

HSU, W. H., KING, A. L., PARADESI, M. S., PYDIMARRI, T., and WENINGER, T. (2006) Collaborative and structural recommendation of friends using weblog-based social network analysis, in *Proceedings of the 2006 AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*: AAAI, 55–60.

HUBERMAN, B., ROMERO, D. M., and WU, F. (2008) Social networks that matter: Twitter under the microscope, *First Monday*, 14(1). doi: http://dx.doi.org/10.5210/fm.v14i1.2317.

JAVA, A., SONG, X., FININ, T., and TSENG, B. (2007) Why we twitter: understanding microblogging usage and communities, in *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*: ACM, 56–65.

JI, M., HAN, J., and DANILEVSKY, M. (2011) Ranking-based classification of heterogeneous information networks, in *Proceedings of the 17th*

ACM SIGKDD International Conference on Knowledge Discovery and Data mining: ACM, 1298–1306.

KIM, Y. and SHIM, K. (2011) Twitobi: A recommendation system for Twitter using probabilistic modeling, in *Proceedings of the IEEE 11th International Conference on Data Mining (ICDM)*: IEEE, 340–349.

KIM, Y. and SHIM, K. (2013) TWILITE: A recommendation system for Twitter using a probabilistic model based on latent Dirichlet allocation, *Information Systems*, 42, 59–77. doi: http://dx.doi.org/10.1016/j.is.2013.11.003.

KWAK, H., LEE, C., PARK, H., and MOON, S. (2010) What is Twitter, a social network or a news media?, in *Proceedings of the 19th International Conference on World Wide Web*: ACM, 591–600.

KYWE, S. M., LIM, E.-P., and ZHU, F. (2012) A survey of recommender systems in Twitter, in *Social Informatics*: Springer, 420–433.

LI, M., LIU, L., and LI, C.-B. (2011) An approach to expert recommendation based on fuzzy linguistic method and fuzzy text classification in knowledge management systems, *Expert Systems with Applications*, 38(7), 8586–8596.

LIU, B. (2007) *Web data mining: Exploring hyperlinks, content, and usage data*. Springer, Chicago, IL, USA.

LIU, D. and CHEN, X. (2011) Rumor propagation in online social networks like Twitter–a simulation study, in *Proceedings of the 3rd International Conference on Multimedia Information Networking and Security (MINES)*: IEEE, 278–282.

LIU, F. and LEE, H. J. (2010) Use of social network information to enhance collaborative filtering performance, *Expert systems with applications*, 37 (7), 4772–4778.

NEWMAN, M. (2010) *Networks: An introduction*. Oxford University Press, New York, USA.

PARK, D. H., KIM, H. K., CHOI, I. Y., and KIM, J. K. (2012) A literature review and classification of recommender systems research, *Expert Systems with Applications*, 39(11), 10059–10072.

PARK, H.-S., YOO, J.-O., and CHO, S.-B. (2006) A context-aware music recommendation system using fuzzy bayesian networks with utility theory, in *Fuzzy systems and knowledge discovery*: Springer, 970–979.

PORCEL, C. and HERRERA-VIEDMA, E. (2010) Dealing with incomplete information in a fuzzy linguistic recommender system to disseminate information in university digital libraries, *Knowledge-Based Systems*, 23(1), 32–39.

RICCI, F., ROKACH, L., and SHAPIRA, B. (2011) Introduction to recommender systems handbook. In Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors, *Recommender Systems Handbook*, pages 1–35. Springer US. ISBN 978-0-387-85819-7. doi: 10.1007/978-0-387-85820-3_1. URL `http://dx.doi.org/10.1007/978-0-387-85820-3_1`.

RODRÍGUEZ, F. M. (2013) Cuantificación del interés de un usuario en un tema mediante minería de texto y análisis de sentimiento. Master's thesis, Universidad Autónoma de Nuevo León.

SAKAKI, T., OKAZAKI, M., and MATSUO, Y. (2010) Earthquake shakes Twitter users: real-time event detection by social sensors, in *Proceedings of the 19th International Conference on World wide web*: ACM, 851–860.

SILVA, N. B., TSANG, R., CAVALCANTI, G. D., and TSANG, J. (2010) A graph-based friend recommendation system using genetic algorithm, in *Proceedings of the 2010 IEEE Congress on Evolutionary Computation (CEC)*: IEEE, 1–7.

SUN, Y., YU, Y., and HAN, J. (2009) Ranking-based clustering of heterogeneous information networks with star network schema, in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*: ACM, 797–806.

SUN, Y., HAN, J., AGGARWAL, C. C., and CHAWLA, N. V. (2012) When will it happen?– Relationship prediction in heterogeneous information networks, in *Proceedings of the 5th ACM International Conference on Web Search and Data mining*: ACM, 663–672.

TRUNG, D. N. and JUNG, J. J. (2014) Sentiment analysis based on fuzzy propagation in online social networks: a case study on tweetscope, *Computer Science and Information Systems*, 11(1), 215–228.

TSOUROUGIANNI, E. and AMPAZIS, N. (2013) Recommending who to follow on Twitter based on tweet contents and social connections, *Social Networking*, 2, 165–173. doi: http://dx.doi.org/10.4236/sn.2013.24016.

TUMASJAN, A., SPRENGER, T., SANDNER, P., and WELPE, I. (2010) Predicting elections with Twitter: What 140 characters reveal about political sentiment, in *Proceedings of the 4th International AAAI Conference on Weblogs and Social Media*, 178–185.

WANG, J., VARSHNEY, K. R., and MOJSILOVIC, A. (2012) Legislative prediction via random walks over a heterogeneous graph, in *Proceedings of the 2012 SIAM International Conference on Data Mining*: SIAM, 1095–1106.

XIANG-WEI, M., YAN, C., LI-LI, Q., and TAO-YING, L. (2009) A new user profile model based on intuitionistic fuzzy set for personalized information analysis and sharing, in *Proceedings of the International Conference on Management Science and Engineering (ICMSE)*: IEEE, 64–69.

YIGIT, M., BILGIN, B. E., and KARAHOCA, A. (2015) Extended topology based recommendation system for unidirectional social networks, *Expert Systems with Applications*, 42(7), 3653–3661.

ZADEH, L. (1965) Fuzzy sets. *Information and control*, 8(3), 338–353.