Ontology Engineering and Feature Construction for Predicting Friendship Links in the Live Journal Social Network

Vikas Bahirwani
Kansas State University
vikas@ksu.edu

Doina Caragea
Kansas State University
dcaragea@ksu.edu

Waleed Aljandal
Kansas State University
waleed@ksu.edu

William H. Hsu
Kansas State University
bhsu@ksu.edu

ABSTRACT

An ontology can be seen as an explicit description of the concepts and relationships that exist in a domain. In this paper, we address the problem of building an interest ontology and predicting potential friendship relations between users in the social network *Live Journal*, using features constructed based on the interest ontology. Previous work has shown that the accuracy of predicting friendship links in this network is very low if simply interests common to two users are used as features and no network graph features are considered. Thus, our goal is to organize users' interests in an ontology (specifically, a concept hierarchy) and to use the semantics captured by this ontology to improve the performance of learning algorithms at predicting if two users can be friends. We have designed and implemented a *hybrid clustering* algorithm, which combines hierarchical agglomerative and divisive clustering paradigms, and automatically builds the interest ontology. Furthermore, we have explored the use of this ontology to construct interest-based features and shown that the resulting features improve the performance of various classifiers for predicting friendship links.

Keywords

Social Network Analysis, Interest Ontology, Clustering, Machine Learning, Friendship Link Prediction

1. INTRODUCTION

An ontology is an explicit, formal specification of a shared conceptualization of a domain of interest [16], where "formal" implies that the ontology should be machine readable and "shared" implies that it is accepted by a group or community [6]. In other words, an ontology is an explicit description (similar to the formal specification of a program) of the concepts and relationships that exist in a domain [15].

The 2nd SNA-KDD Workshop '08 (SNA-KDD'08), August 24, 2008 , Las Vegas, Nevada , USA .

Ontologies can be seen as high-level "umbrella" structures, which can be inherited and extended in order to organize data and knowledge [25]. They can also be seen as metadata that are used to provide a better understanding of the data or to facilitate data integration [23, 11, 8, 10, 27].

Ontology-based data management systems (which organize data based on the semantic knowledge of a domain) find their applications in Web-based learning [29], software development (e.g., consistency checking, support and validation testing) [25, 17], flexible querying of heterogeneous data sources [7], etc. In social networks, ontologies can provide a crisp semantic organization of the knowledge available in such networks. For example, users' interests can be grouped in a concept hierarchy that makes explicit the *implicit* relationships between various interest concepts, thus helping in the process of data understanding and analysis.

In previous work, Hsu et al. [19] have addressed the task of predicting if two users of the social network *Live Journal* can be friends or not. Their results suggest that features constructed only from the common interests of two users are ineffective at predicting friend relationships. While network graph features prove to give good results, several questions can be still raised: Can we improve the performance of the algorithms that use graph features by combining them with interest features? What if the graph features are not available or are incomplete? What if we are interested in other prediction problems, such as the prediction of the users' interests themselves? We hypothesize that organizing interests into an ontology and constructing interest features based on the ontology can help to successfully address some of these questions.

To illustrate the importance of incorporating semantic information in the data used by prediction algorithms, we consider a simple example. Suppose that a user A is interested in "laptops" and a user B is interested in "desktops". A naive learner will consider "laptops" and "desktops" to be two different entities due to their lexical differences - a semantically incorrect assumption. If an interest ontology was available, "laptops" and "desktops" would probably be grouped together into a more abstract concept called "computers", and thus, the learner would be able to semantically link the more specific concepts "laptops" and "desktops" and use this information in the prediction process.

It has been previously shown [33, 18] that the use of metadata (e.g., concept hierarchies) in addition to data, can

improve the quality (accuracy and interpretability) of the learned predictive models in several domains. In this paper, we investigate the use of ontologies to improve the performance of several learning algorithms at predicting friendship links in a social network. To do this, our first construct an ontology based on declared interests of users in the social network *Live Journal*.

The rest of the paper is organized as follows: Section 2 presents background information for the problem addressed. Section 3 introduces the details of the algorithm for ontology building. Sections 4 and 5 describe the procedure used to evaluate the ontology extracted from the data and actual experimental results obtained when predicting friend relationships using the ontology, respectively. A discussion of the related work can be found in Section 6. We conclude and present several directions for future work in Section 7.

2. BACKGROUND

Clustering is an unsupervised learning task that can be defined as the process of partitioning data into groups, corresponding to higher level concepts, of similar entities [3]. Clustering algorithms are highly dependent on the selection of a distance metric that assigns a score to every pair of entities that may be grouped together. The distance metric captures the extent of similarity (or dissimilarity) between candidate pairs. The distance between two clusters is usually computed as the average, maximum or minimum distance among all distances between the possible pairs of entities contained in the two clusters. A *hierarchical clustering* algorithm builds a tree of clusters by successively grouping the closest cluster pairs, until no further grouping is possible. In the resulting tree (often called *dendrogram*), each cluster node at an intermediate level is associated with a parent cluster, one or more child nodes and one or more sibling nodes. The hierarchical clustering approach allows the exploration of the data at different levels of granularity. Thus, parent nodes represent abstract notions of the detailed concepts that their children embody [3].

Hierarchical clustering approaches are categorized as *agglomerative* and *divisive* according to their algorithmic structure and operation [21]. The agglomerative approach starts by considering each instance as a distinct singleton cluster, and based on the similarity criterion, successively merges clusters together until the termination conditions are satisfied [21]. On the other hand, the divisive approach begins with a root cluster that represents all data instances, and successively splits the clusters based on the cohesiveness (or dissimilarity) of the instances constituting the cluster members, until the termination conditions are met [21].

We have designed and implemented a hybrid clustering algorithm, called *Hierarchical Agglomerative and Divisive* (HAD) clustering, to *automatically* organize the users' interests into a concept hierarchy. Several other approaches were considered for this task. We will briefly describe them and their limitations in the next few paragraphs and show how our HAD algorithm overcomes these limitations.

There has been a lot of undergoing research in modeling users' interests by classifying the web pages that a particular user visits. Kim [22] and Godoy [14] present unsupervised web document clustering algorithms that perform incremental concept learning. Kim [22] introduces a top-down divisive hierarchical clustering method to recursively divide parent clusters into child clusters until a terminating criterion is met. This division is based on how similar two interests in the parent cluster are to each other. The similarity of two interests is based on *content* (i.e., words and/or phrases describing interests). For each pair of interests the similarity function is used to produce a score. If the score exceeds a threshold, all similar interests are separated out from the parent in a child cluster. The termination condition for the recursive partitioning is satisfied when no parent cluster can be further divided because of lack of similarity among the interests it contains.

Godoy [14] describes a practical way to implement the method introduced by Kim [22]. The procedure begins with a root cluster, which represents "everything". Initially, interests are added one-by-one to this root cluster and with each addition, the cluster is evaluated for its *cohesiveness*. If all interests in a cluster are similar enough to bring the cohesiveness measure above a threshold, they are combined into a new child cluster. Every cluster, except for the root cluster, has a *concept* associated with it. This concept is a set of terms that describe the interests in the corresponding cluster. Godoy's approach requires each individual interest "entering" into the clusters below the root to be matched against the associated cluster concepts. If an individual does not match a particular concept, it is either matched against other peer concepts or added to the parent cluster, if there are no matching peers. Thus, this idea of recursive partitioning extends from root to leaves, and partitioning of a cluster takes place when a newly added individual increases the cohesiveness of the cluster above a threshold.

Our initial approach to building an interest ontology was based on the notion of similarity introduced in [22] and the algorithm proposed in [14], but it presented several shortcomings in the social network domain. First, these two papers consider a bag of words (BoW) approach to describe instances (web pages), and thus each instance is represented as a fixed length array. In the social network context, users' interests can be seen as instances, but cannot be naturally described by a fixed length bag of words, as every user has a relatively small number of interests compared to the total number of possible interests.

Second, the approach in [14] does not allow instances to belong to several concepts. An instance is restricted to belong to only one of the learned concepts. However, this restriction is not desirable for interest concepts, as interests like "notebooks" can belong to more than one category, for example "school supplies" and "computers".

Third, the incremental nature of the unsupervised learning approach used in [14] is not appropriate for our social network data. Specifically, the conceptual clustering approach tries to derive a concept out of a cluster every time a new instance is added to it. If a concept is generated, new instances will be added to a cluster below this concept only if they are similar to the terms describing the instances in the cluster. If instances in a cluster contain small definitions (description of an interest), the associated concept will contain less terms and hence will allow instances with small definitions to pass through it (the same can happen if instances with long definitions are considered). Furthermore, checking if each instance belongs to the newly formed concepts can make the algorithm slower.

Finally, if according to the algorithm, a cluster is cohesive enough to produce a concept out of it, but the concept cannot be assigned descriptive terms (because, maybe, all valid

terms have been used to describe its parents), then further expansion of the conceptual hierarchy from this concept will not be possible. This makes the concepts, which are away from the root, more stringent to "classifying" new instances.

Based on these considerations, we could not use the approaches in Kim [22] and Godoy [14] to construct an ontology over the interests of users in a social network. Instead, we propose a hierarchical agglomerative and divisive clustering approach, which overcomes these issues and produces a useful ontology of interests. We evaluate this ontology with respect to the improvement in the performance of algorithms for predicting friend relationships among network users, when the algorithms are presented with features computed based on the ontology.

3. HIERARCHICAL AGGLOMERATIVE AND DIVISIVE CLUSTERING

As the name of the algorithm suggests, Hierarchical Agglomerative and Divisive (HAD) clustering algorithm is a hybrid between the hierarchical agglomerative and divisive clustering paradigms. The algorithm is designed to make the ontology extraction process as fast as possible and at the same time to produce a *sensible* and *useful* ontology. Primarily, it consists of three steps:

- In the first step, HAD fetches definitions of interests expressed by *Live Journal* users, from various sources such as WordNet-Online and IMDB. Every definition of an interest forms an instance that will be included in the resulting ontology.
- The second step divides the instances into different clusters based on the sources from where the definitions are fetched and other factors such as "genres" of books or movies specified as interests.
- At the final step, HAD engineers the concept hierarchy in a bottom-up fashion to produce a tree whose root collectively represents all instances and whose nodes represent concepts at various levels of abstraction.

Subsections 3.1, 3.2 and 3.3 describe the above steps in detail.

3.1 Obtaining Interest Definitions

Social network data consisting of 1000 users, their interests and declared friends, is obtained from the *Live Journal*. Surprisingly, the 1000 users have nearly 22,000 unique interests, from which we derived approximately 45,000 unique individuals or instances, as explained below.

Each of the 22,000 unique interests is read from a text file and queried against different sources for potential definitions or descriptions. We seek information from WordNet-Online for the meanings of valid words, Internet Movie Database (IMDB) for description of movies, and Amazon.com for descriptions of books via Amazon Associates Web Services (AWS). We have chosen to retrieve specific definitions for movies and books, because many user interests in our data are related to such concepts. Usually, an interest word can have more than one meaning, generating more than one instance. Furthermore, instances of the same interest word may belong to different parts of speech. At last, an interest may be a movie and/or a book with a specific genre associated with it. The definitions retrieved from the different sources capture such information associated with interests.

As an example, we have three definitions for the interest "character". The first definition is obtained using WordNet-Online and results in the following instances for "character":

```
character, n, reference| character| formal|
describing| qualifications| dependability...
character, n, grapheme| graphic| symbol| written|
symbol| used| represent| speech...
character, n, genetics| functional| determined|
gene| group| genes...
character, v, engrave| inscribe| characters...
```

Two more definitions are obtained from IMDB and AWS, by querying these sources for "character" movies and books:

```
IMDB: character, reality tv, reality| film| fantasy|
history| character| movie| dream| rocky...
AWS: character, novel, butcher| covey| davenport|
detective| dresden| effective| favor| files...
```

These definitions are used to create individuals, which are instances to be provided as input to the clustering algorithm and have the following format:

```
<interest>, <part of speech/genre>, <gloss>
```

where `<gloss>` is the set of words describing a particular interest. The gloss is extracted by filtering the text describing the interest, so that stop-words such as articles and prepositions are removed.

Furthermore, for interests that are neither single words, nor movies/books, an "alternate definition" is formed. An alternate definition is the combination of definitions (fetched from WordNet-Online) of individual words that form the phrase for which the sources considered failed to provide descriptions. For instance, the alternate definition for "aim pranks" is as follows:

```
aim pranks, alternate, aim| purpose| intention|
design| pranks| buffoonery| clowning|
prank| acting| like| clown| buffoon
```

Apart from the interests that have valid definitions obtained from a variety of sources, users often specify interests that do not form valid interest words (e.g., "?????" or ":)"). There are approximately 500 such "interests" and we do not included them in the list of interests provided as input to the clustering algorithm. It is worth mentioning that for interests that have multiple definitions, HAD will consider these definitions as independent instances and will try to place them in relevant and possibly different clusters. This takes care of the major shortcoming described in Section 2, mainly that an interest can only belong to one cluster.

3.2 HAD: Divisive Clustering Step

After definitions for interests are fetched, the next step is to divide the resulting instances into four major clusters. The first cluster consists of all the instances that are described in terms of meaningful words from WordNet-Online. The second cluster consists of movie definitions fetched from IMDB. The third cluster comprises of book definitions and the fourth contains instances with alternate definitions. About 22,000 unique interests queried for definitions generate 17,753 valid word definitions, 4,189 movie definitions, 18,168 book definitions and 1,986 alternate word definitions resulting in a total of 42,096 individuals to be

clustered. Given the large number of book instances and the prior knowledge about genres, the "book" cluster is further divided into a set of sixteen sub-clusters based on genres (Action, Fantasy, Drama, Children, etc). Similar to books, movies can also be divided based on their genres. However, since there are only about 4,200 movie interests, compared to almost 18,000 book-interests, the "movie" cluster is not further divided in this phase of the algorithm; the grouping based on genre will be performed by the hierarchical agglomerative clustering.

There are two advantages we gain by dividing the data into several clusters before applying the hierarchical agglomerative clustering algorithm. First, we can apply the algorithm in parallel to these clusters, resulting in faster ontology construction. Second, the source from which the definitions of an interest are obtained can inform us about other concepts that this interest can be associated with. The prior cluster division makes it possible to exploit this information.

3.3 HAD: Agglomerative Clustering Step

The hierarchical agglomerative clustering algorithm is independently applied to each cluster obtained in the divisive phase of HAD. It works in a bottom-up fashion and takes as input the set of instances in a particular cluster. Initially, each instance is considered to be a singleton cluster and the set of all singleton clusters is called the "current" set. At each iteration, HAD considers every cluster present in the "current" set and aims to find another cluster matching it. If a match is found between two clusters, the two clusters are merged to form a new parent cluster, which will be added to the "current" set to be used in the next iteration of the algorithm. If no match is found for a cluster, that cluster will be added to the new "current" set as it is.

The similarity between two instances (or equivalently, two singleton clusters) is defined as the number of common terms describing the instances. Furthermore, the similarity between two non-singleton clusters is considered to be the average similarity between all elements of the two clusters (average linkage). A cluster is said to "match" another cluster if the similarity between them is maximum among all the possible pairs of clusters in the current set.

We have applied this algorithm to all nineteen clusters (obtained in the divisive phase of HAD), using a multi-threaded execution paradigm. The clusters obtained from each thread are combined to form a "unified" set. At this stage, the hierarchical agglomerative clustering is again applied to the unified set of clusters to complete the formation of the ontology. The algorithm is said to convergence (or complete its job) in any of the following two cases: either the clusters in the current set do not match to any of their peers or the new current set has only one element.

3.4 Visual Inspection of the Ontology

Visual inspection of an ontology can provide useful insights about how various instances have been organized into concepts that capture the semantic knowledge in a domain. We have used the open source tool "Cytoscape" [28] to inspect the ontology of interests constructed by our algorithm. Because HAD involves no human intervention during the ontology building process, semantically incorrect concepts may get introduced into the hierarchy. In principle, visual inspection of the ontology is useful in finding such incorrect concepts and tools like Cytoscape can help editing and correcting the ontology by adding/deleting nodes and edges. We have not edited the ontology returned by HAD, but performed visual analysis to get an idea about its quality.

One small subset of the ontology constructed from the *Live Journal* data using HAD is shown in Figure 1, which delineates the organization of several interests related to the concept of "networking". It is comprised of the following concepts:

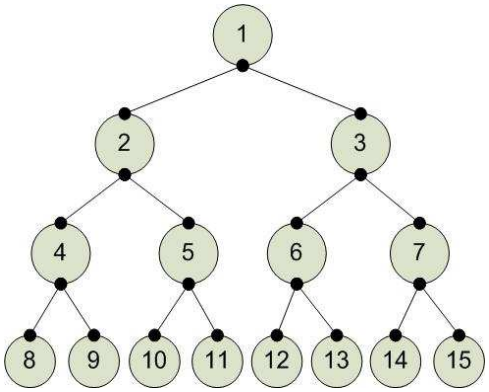

Figure 1: Ontology of terms related to "networking"

```
1: networks, topology, networks, networking, wireless,
radio, networks, wifi
2: networks, networks, networking, topology
3: radio, networks, wireless, wifi
4: networks, topology
5: networks, networking
6: radio, wireless
7: networks, wifi
8: networks (defined as)-
net| network| mesh| meshing| meshwork| wire
9: topology (defined as)-
network| components| connected| mesh| wire
10: networks (defined as)-
network| communicate| group
11: networking (defined as)-
network| communicate| group
12: wireless
wireless| communication| electromagnetic| waves
13: radio (defined as)-
wireless| communication| based| broadcasting|
electromagnetic|waves| transmit| radio| waves
14: networks (defined as)-
broadcasting| communication| stations|
transmit| programs
15: wifi (defined as)-
wireless| network| high| frequency| radio|
signals|transmit| receive| data| broadcasting
```

Figure 1 shows that there are several instances describing the "networking" interests and they combine together (based on similarity) in a meaningful way to form more abstract concepts.

4. EVALUATION OF THE ONTOLOGY THROUGH LEARNING

As the main goal of our work is to improve the performance of learning algorithms at the task of predicting friend

relationships in a social network, we will formally evaluate the ontology through the results of prediction algorithms that use interest features constructed using the interest ontology.

4.1 Types of Features

Several types of features, described below, can be constructed from social network data:

1. *Interest-based features*:

   (a) User interests themselves can be indicative of user friendships and could be used as *interest-based nominal features*. Intuitively, if two users share a rare interest, then the two users can potentially be friends. Given the large number of interests in our data, it is not feasible to use the original user interests as features. However, one can try to use user interests at higher levels in the ontology of interests.

   (b) On the other hand, one may think that if two users have many interests in common, then it is possible that they are friends, regardless of what exactly those interests are. Several *interest-based numerical features* capturing this intuition can be derived and used to predict friendships. Section 4.2 presents an overview of our related work [2], where a variety of interest-based numerical features are investigated.

2. *Graph-based features*:

   The social network *Live Journal* is essentially a graph with nodes representing the users and edges representing the friendships among users. Similar to Hsu et al. [19], we consider several graph-based features, such as *in-degree of users*, *out-degree of users*, *forward deleted distances*, etc. For precise definitions of these features, please see [19].

To evaluate the usefulness of the interest ontology in predicting friendships in the *Live Journal* we compare the performance of several learning algorithms at this task, when used with interest-based features only, graph-based features only and a combination of the two sets of features, both in the presence and absence of the ontology. When deriving interest-based numerical features using the ontology, the interests of users can viewed at different levels of abstraction and experiments are conducted to reveal the levels in the hierarchy that give the best performance (see Section 5).

4.2 Computing numerical interest based measures using Association Rules

Association rules are rules of the form A→B, where both A and B are subsets of an observed item set $L = \{I_1, I_2, \ldots, I_k\}$. There are many approaches for association rule learning from item set data. Also, many measures for estimating rule interestingness have been proposed. Geng and Hamilton [12] review the interestingness measures for rules and summaries, classify them from several perspectives and compare their properties. They present 38 probability-based objective interestingness measures for association rules. In another survey, Tan et al. [31] discuss the properties of 21 objective interestingness measures and conclude that there is no "single" measure that is consistently better than others in all application domains. Each measure captures different rule characteristics in a domain.

In the context of association rule based measures, user interests can be seen as item sets from which rules of the form A→B are formed. Eight objective measures of rule interestingness are defined in [2]. Using these measures, we construct eight interest-based numerical features (as described below) and use them alone or in combination with graph features to predict friends.

For a pair of users and a corresponding association rule A→B, the interestingness measures considered for this rule are normalized by a factor that takes into account the *popularity* of particular interests the users hold in common, with the most popular interests (held by a significant proportion of users) being slightly less revealing than rarer interests.

We use the following eight normalized association rule measures, along with the number of common interests, as features for the prediction problem at hand:

1. $Support(A \rightarrow B) = P(AB)$
2. $Confidence(A \rightarrow B) = P(B|A)$
3. $Confidence(B \rightarrow A) = P(A|B)$
4. $Lift(A \rightarrow B) = \frac{P(B|A)}{P(B)}$
5. $Conviction(A \rightarrow B) = \frac{P(A)-P(\rightharpoonup B)}{P(A \rightharpoonup B)}$
6. $Match(A \rightarrow B) = \frac{P(AB)-P(A)*P(B)}{P(A)*(1-P(A))}$
7. $Accuracy(A \rightarrow B) = P(AB) + P(\rightharpoonup A \rightharpoonup B)$
8. $Leverage(A \rightarrow B) = P(B|A) - P(A)P(B)$

5. EXPERIMENTAL SETUP AND RESULTS

This section describes the details of our experimental setup and the results obtained. We conducted a set of eight experiments designed to investigate the performance of several classification algorithms at predicting friend relationships, when presented with different sets of features. When used alone, interest-based numerical features may not be very effective at predicting friendships. However, one may expect the algorithms to perform better when presented with graph-based features as well. Our results support this intuition.

5.1 Experimental Setup

In each experiment, the training and test data sets are independent and each consists of 1000 user pairs. The training set contains approximately 50% friend pairs and 50% non-friend pairs, while the test set contains user pairs selected randomly from the original distribution. Positive examples are obtained from the links in the network graph. To generate negative examples, we make the assumption that two users are not friends if there is no direct link between them in the network graph. Any overlap in terms of users, between the training and test data sets, is removed.

Moreover, in each experiment, we consider the following classifiers: J48 decision trees, support vector machines (SVM) with *build logistic model* option enabled, random forests, logistic regression (Logistic) and one-attribute-rule (OneR) classifiers, whose implementations are provided by the WEKA data mining software [32]. The performance of

each algorithm is measured by the area under the *Receiver Operating Characteristic* (ROC) curve, i.e. the curve depicting the tradeoff between the *true positive rate* vs. *false positive rate*.

In our first experiment, we considered the original interests of users and aimed to investigate if they can be used to predict friendships. There are 22,000 interests, which give a set of about 44,000 possible interest-based nominal features. The input vector corresponding to each candidate pair comprises of 88,000 interest-based nominal features (44,000 for each user in the pair) and the class "friendship". Given the large number of features, Weka [32] crashed due to insufficient memory, so this experiment could not be run.

In our second experiment, the interests are refined according to the more abstract levels of the interests hierarchy. As in the first experiment, a list of interest-based nominal features for each candidate user pair is presented to the prediction algorithms. Since, there are fewer interests at higher levels of abstraction, refining the data makes it possible to run Weka directly on the interest features.

In our third experiment, we use the interest-based numerical features proposed in [2]. These features are computed on the original data set *without* considering the concept hierarchy of interests.

In the fourth experiment, we modify the original data by considering interests at different levels of abstraction in the concept hierarchy and compute the interest-based numerical features from the modified data. The resulting features are then used to predict friendships among users. Since the performance of classifiers is evaluated for each modified version of the data (each level of abstraction), this experiment also reveals the best level of abstraction in the interest ontology. Intuitively, these classifiers are expected to show an improvement in performance, compared to the classifiers that use numeric features computed from the original data.

The fifth experiment addresses the friend prediction problem by exploiting graph-based features, while the sixth experiment considers interest-based nominal features (as used in the second experiment) combined with graph-based features and aims to explore possible improvements in the performance of classifications algorithms when presented with both types of features as compared to only one type of features.

The seventh experiment considers graph-based and interest-based numerical features derived without making use of the interests ontology. The classification algorithms are expected to show an improvement over the previous results for they are exposed to more information than the classifiers in those experiments.

The final experiment uses graph-based and interest-based numerical features to predict friendships. Interest-based numerical features are derived using the ontology. By varying the interest level according to the ontology, the best level of abstraction for the given data can be obtained. When used with interest-based numerical features at the "best" level of abstraction in the concept hierarchy, classification algorithms are expected to show improvements in performance when compared to results from previous experiments.

5.2 Results

The main goal of the experiments described above is to investigate the contribution of the interest ontology to the performance of algorithms for predicting friendships links. Indeed, the results indicate the usefulness of the ontology constructed.

Table 1 shows the AUC (area under curve) values for various classifiers used in our experiments. There are no AUC values for the first experiment as this experiment could not be run (due to memory limitations). In the second experiment, however, the ontology proved to be useful in helping the classifiers run successfully. Interests specified at higher levels of abstraction (levels 1 to 4 out of 17 possible levels) fostered smaller data sets that classifiers could handle. The numbers of interests at levels 5 to 17 were large (level 5 contains around 1900 interests) and produced large training and test data sets. Table 1 enlists the AUC values for the classifiers considered, when they were presented with interest-based nominal features specified at level 4 in the ontology.

Moreover, the interest-based nominal features were combined with the graph-based features to investigate incremental improvements in the prediction performance of classifiers. Table 1 reveals that graph-based features alone are very effective in predicting friendships. However, interests alone are not effective at predicting friends. Furthermore, they don't show a significant improvement in the AUC values, when combined with graph features.

Numerical measures aim at capturing the interest-based information in the data, and at the same time ensure that the training and test data sets are not too large for the classifiers to handle. Table 1 indicates an improvement in the performance of classifiers, when interest-based numerical features are used instead of interests themselves. The AUC values show that numerical measures are better at summarizing the interest-based information, when compared to the interests themselves.

In addition, generally classifiers show improvement in the performance of the friend prediction problem, when provided with semantic knowledge captured in the ontology. Every classifier shows a 25% increase in the AUC value when interest-based numerical features are constructed using the ontology.

Also from Table 1, we can see that graph-based features alone are very effective at predicting friends in the given social network. However, when using the ontology, the SVM, logistic regression and random forest classifiers perform even better (AUC value for SVM = 0.993). This shows the effectiveness of using the ontology and the fact that it does improve the performance of the classification algorithms.

A point worth mentioning here is that our experiments show that out of 17 possible levels of abstraction (other than the root representing "everything" and the leaves representing the original interests), Level 15 seems to be the best level of abstraction when predicting friends using only interest-based numerical features, while Level 10 seems to be the best level when using both graph-based and interest-based numerical features. Thus, in the experimental results, when the ontology is used to modify user interests and compare the performance of different classifiers, the ontology is considered from the respective best levels of abstraction.

Figure 2 shows the ROC curves for all the classifiers used to predict friends with graph-based and interest-based numerical features. As mentioned above, Level 10 of the ontology is used to specify interests and to compute numerical features from them. Figure 2 and Table 1 show that the best classifier for the given prediction problem is SVM.

Table 1: Prediction performance of different classifiers when exposed to a variety of feature sets

Features Used	J48	SVM	Logistic	OneR	Random Forest
Nominal interest-based without Ontology	-	-	-	-	-
Nominal interests-based with Ontology	0.504	0.487	0.456	0.439	0.491
Numerical interest-based without Ontology	0.694	0.711	0.68	0.597	0.61
Numerical interest-based with Ontology	0.83	0.893	0.894	0.677	0.839
Graph-based	0.977	0.952	0.946	0.9	0.979
Graph-based and nominal interest-based with Ontology	0.977	0.95	0.944	0.9	0.98
Graph-based and numerical interest-based without Ontology	0.95	0.948	0.936	0.9	0.98
Graph-based and numerical interest-based with Ontology	0.95	0.993	0.98	0.864	0.982

Furthermore, the classifier OneR (one-attribute-rule) performs the worst among the set of classifiers considered. OneR algorithm tries to find the one attribute, which can be use to classify a new instance with the smallest number of errors. When comparing OneR with other classification algorithms, it becomes clear that the algorithm is not able to find a single attribute that can distinguish between friend and not-friend links effectively.

Figure 3 delineates the ROC curves for SVM when different sets of features are used to address the prediction problem at hand. Figure 3 shows that SVM performs the best when it uses graph-based and interest based numerical features, in the presence of the ontology.

6. RELATED WORK AND DISCUSSION

This section provides a short review of the areas of research related to the work presented in the paper. Subsection 6.1 presents the progress in the field of semantic information extraction in social networks using simple clustering-based approaches. Subsection 6.2 describes recent advancements in the field of social network analysis.

6.1 Information Extraction in Social Networks

"The success and popularity of social network systems have generated many interesting and challenging problems to the research community" [24]. Li et al. [24] have addressed the problem of discovering social interests shared by a group of users of the social network system "del.icio.us". Their approach is based on the key observation that "human users tend to use descriptive tags to bookmark or annotate the web-based content they are interested in". A system called "Internet Social Interest Discovery" (ISID) that clusters user-generated tags, bookmarked URLs and users who annotated the URLs with these tags is described in [24] and used to predict user interests. The results show that ISID successfully discovers interests of nearly 90% users. Similar to the ISID system presented by Li et al., our approach uses clustering techniques to group user interests together. In *Live Journal*, interests are analogous to bookmarked URLs in "del.icio.us" and are explicitly specified by the users. Furthermore, "user-tags" in "del.icio.us" are analogous to descriptions of interests extracted by HAD from various sources, in our approach. However, HAD does not cluster users and their interests together, and uses the semantic information captured in the ontology to address the problem of predicting friend relationships.

In other related work, Mori et al. [26] have investigated the use of semantic information extracted from web documents to discover relationships in social networks. Specifically, given entity pairs in a political social network (e.g., George W. Bush - United States, Junichiro Koizumi - Japan), the goal is to extract labels for describing the relations of the respective entity pairs (i.e., to discover relevant terms that relate a politician to a location in this example). Their approach first builds a context model for each entity pair, based on the web based contents that describe the participating entities. The algorithm, then, clusters entity pairs according to the similarities among the corresponding context models and finally completes by selecting representative labels to describe relations from each cluster. With reference to our approach, entity-pairs are analogous to interests whose descriptions (context models) can be extracted from the information available on the web. Moreover, interests are clustered based on their descriptions to capture the underlying semantics. Unlike Mori et al., however, we use the resulting clusters (ontology) to predict only one kind of relationship in Live Journal, i.e. "Friendships".

6.2 Social Network Analysis

"Social network discovery" refers to the general task that comprises of specific subtasks of analysis (prediction) and visualization (exploration) of social networks. This section provides information on the current research and accomplished work related to analyzing and visualizing social networks.

Broadly speaking, analyzing a social network is a challenging problem in itself. In addition, it largely depends on the quality of data given to address the challenge. A common approach to improving data quality is "Entity Resolution" [4]. Bilgic et al. [4] have developed an interactive tool "D-Dupe", which provides a stable visual layout for optimized entity resolution and allows users to combine entity resolution algorithms including data mining algorithms for entity resolution and task-specific network visualization. As the paper describes, the entity resolution process can be seen as an iterative process: as pairs of nodes are resolved, additional duplicates may be revealed; therefore, resolution decisions are often chained together. D-Dupe users resolve ambiguities either by merging nodes or by marking them distinct. In addition they can apply sequences of actions to produce a high quality entity resolution result [4].

In terms of social network analysis, two commonly addressed, but independently studied problems are: object classification (labeling the nodes of a graph) and link prediction (predicting the links in a graph). Object classification is performed assuming a complete set of known links. Coffman and Marcus [9] have addressed the task of object classification by characterizing actors in a simulated dataset as

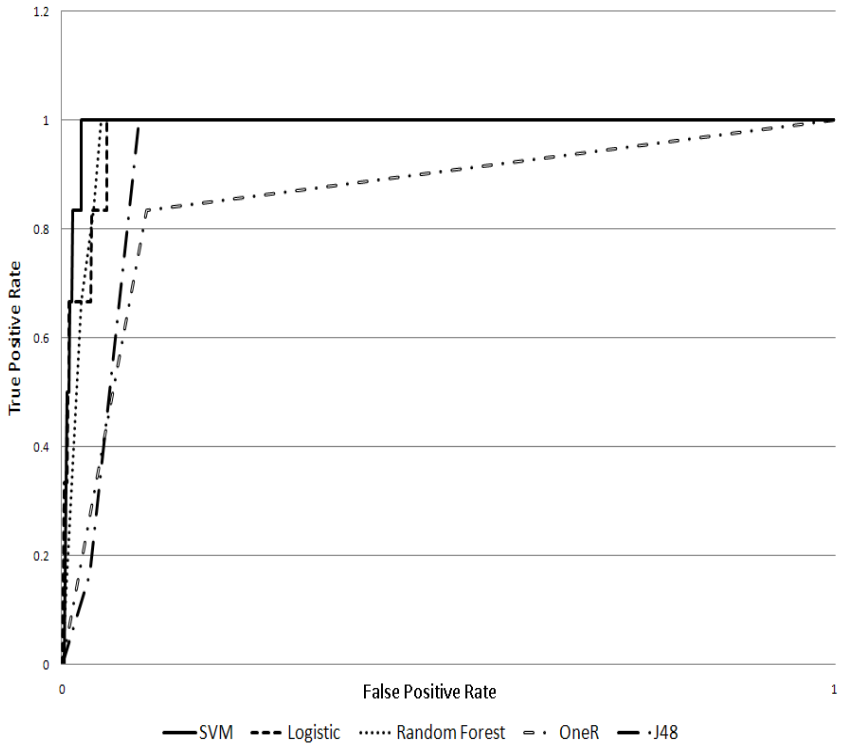

Figure 2: ROC curves for different classifiers when using graph-based and interest-based numerical features, derived using the ontology, to predict friends

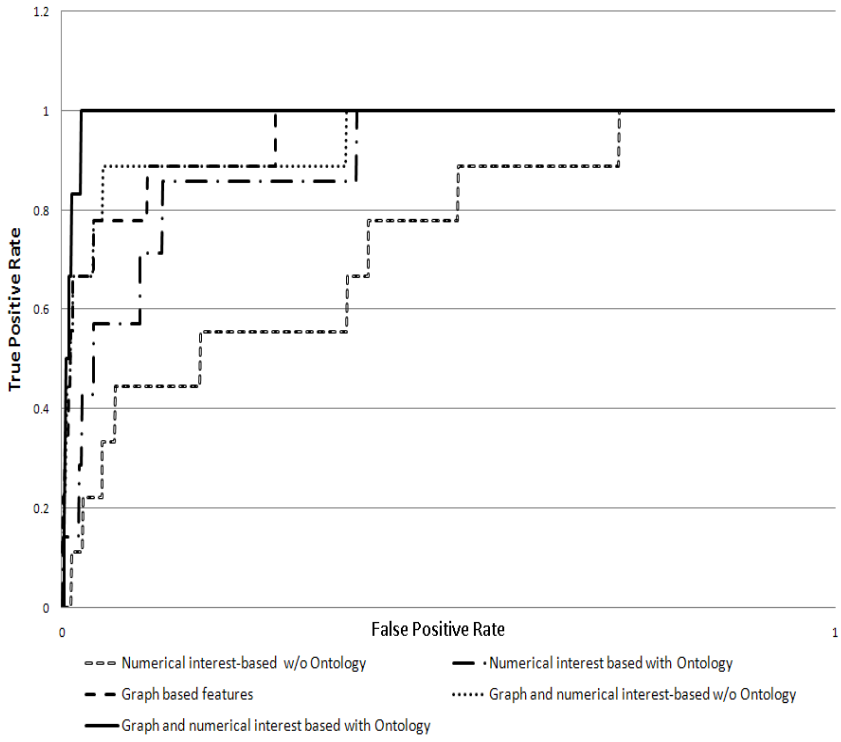

Figure 3: ROC curves for SVM when using different sets of attributes to predict friends

terrorists or non-terrorists by applying statistical classifiers to their social network analysis (SNA) metric values. As the case study describes, the simulated datasets modeled the social interactions that occur within Leninist cell organizations as well as in more typical social structures. Furthermore, the authors achieve an accuracy of 86% in a three-class classification problem (cell leader, cell member, or non-terrorist) and an accuracy of 96% in a two-class classification problem (terrorist or non-terrorist). On the other hand, problems such as link-based classification, identifying the link type, predicting the link strength and cardinality, that come under the umbrella of link prediction are addressed assuming a fully observed set of node attributes [13]. Hsu et al. [20] have addressed the problems of predicting, classifying, and annotating friendship relations in a social network, based on the network structure and user profile data. In their approach, all the node attributes are available from the blog service *Live Journal*. They address a set of link prediction tasks such as predicting existing links and estimating inter-pair distance and achieve an accuracy of about 98%.

In most real world domains, however, attributes and links are often missing or incorrect. Object classification is not provided with all the links relevant to correct classification and link prediction is not provided with all the labels needed for accurate link prediction [5]. Bilgic [5] have developed an approach that addresses these two problems by interleaving object classification and link prediction in a "simple yet general" collective classification algorithm. In their approach they provide the object prediction task with information about the available node and its links. Once a prediction about a node or its particular attribute is made, it is used to address the prediction problem for related links and in general for other links in the network as well. The results show that the algorithm performs well (nearly 90% accuracy) when compared to "flat" prediction approach (addressing each problem independently) [5].

The second subtask of social network discovery i.e. visual mining of the underlying network has also received much attention in the recent research on social networks. Singh et al. [30] have developed a tool "Invenio" which makes use of a wide range of interactive visualization features included in "Prefuse" [1] the open source tool kit for graph visualization, graph mining algorithm support from JUNG and construction of views from both database and graph-based operations. With "Invenio" they aim to explore the multimodal multi relational social networks. As they describe, "modal" refers to number of node-sets in N and "relational" refers to the number of relationship types in N, where N is an extended multi-modal multi-relational network. Besides the features provided by "prefuse" [1], "Invenio" provides newly developed features such as attribute guided subgraph generation, graph mining and visual application analysis.

7. CONCLUSION AND FUTURE WORK

The problem of link prediction in social networks can be addressed by using a variety of features, e.g. interest-based features and graph-based features. We have shown that incorporating semantic knowledge into interest-based features, helps improving the performance of classifiers trained to predict friends.

The HAD algorithm, proposed in this paper, constructs a concept hierarchy of interests of users, enabling classification algorithms to use semantic information in link prediction. The algorithm is implemented by combining hierarchical agglomerative and divisive clustering approaches to overcome the shortcomings of other related approaches. Furthermore, the algorithm implementation is general and can be used to construct similar ontologies in other domains. Our investigation shows that the ontology so produced is very useful in predicting friendships in the absence of graph features. In addition, the combination of graph-based and interest-based numerical features derived in the presence of the ontology is most effective in addressing this problem.

Furthermore, there are several related problems that we would like to address in future work. First, we would like to consider the problem of predicting interests of users by using the ontology we have constructed. The rationale behind this is simple. If two users are friends and they have a lot of other friends in common, and if one is interested in "laptops and gaming", then we may predict that the other user is also interested in "computers" - an abstract concept which will contain laptops and gaming in the concept hierarchy.

Second, similar to the ontology of interests, ontologies of schools, communities etc. can be derived and used to improve the performance at predicting friendships. Moreover, similar to "Live Journal" social networks, other social networks such as "Flickr" or "Facebook" can be considered. Specifically, these social networks allow users to tag their interests, resulting in small "user defined" ontologies. One idea for future work is to try to incorporate these user-defined ontologies in the extracted ontology with the aim to improve the latter.

Finally, a network (social or otherwise) does not always have well defined links among nodes. Predicting links using the concept hierarchies can help in completing the network, which can then be used for other problems.

8. ACKNOWLEDGEMENTS

This work is supported by the National Science Foundation under Grant No. 0711396. We thank Tim Weninger and the research group - Knowledge Discovery in Databases (KDD) for valuable discussions, comments and suggestions.

9. REFERENCES

[1] Prefuse: interactive information visualization toolkit. Released on October 21st 2007.

[2] W. Aljandal and W. H. Hsu. Validation-based normalization and selection of interestingness measures for association rules. Submitted to ANNIE 2008.

[3] P. Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002.

[4] M. Bilgic, L. Licamele, L. Getoor, and B. Shneiderman. D-dupe: an interactive tool for entity resolution in social networks. In *Visual Analytics Science and Technology, (VAST)*, Baltimore, MD, USA, October 2006.

[5] M. Bilgic, G. M. Namata, and L. Getoor. Combining collective classification and link prediction. In *Workshop on Mining Graphs and Complex Structures at the IEEE International Conference on Data Mining, (ICDM)*, 2007.

[6] P. Buitelaar, P. Cimiano, and B. Magnini. Ontology learning from text: an overview. IOS Press, 2003.

[7] D. Calvanese, G. D. Giacomo, M. Lenzerini, and M. Y. Vardi. View-based query processing: on the relationship between rewriting, answering and losslessness. In *Proceedings of the 10th International Conference on Database Theory (ICDT)*, volume 3363. Springer, 2005.

[8] D. Caragea, A. Silvescu, J. Pathak, J. Bao, C. Andorf, D. Dobbs, and V. Honavar. Information integration and knowledge acquisition from semantically heterogeneous biological data sources. In *Proceedings of the Second International Workshop on Data Integration in Life Sciences, (DILS)*, San Diego, CA, 2005. Berlin: Springer-Verlag. LNCS.

[9] T. R. Coffman and S. E. Marcus. Pattern classification in social network analysis: a case study. In *Aerospace Conference*, volume 5, pages 3162–3175, 2003.

[10] A. Doan and A. Halevy. Semantic integration research in the database community: a brief survey. *AI Magazine, Special Issue on Semantic Integration*, 26(1):83–94, 2005.

[11] B. Eckman. A practitioner's guide to data management and data integration in bioinformatics. *Bioinformatics*, pages 3–74, 2003.

[12] L. Geng and H. J. Hamilton. Interestingness measures for data mining : a survey. *ACM Computer Survey*, 2006.

[13] L. Getoor. Link mining: a new data mining challenge. *SIGKDD Explorations*, 5(1):85–89, 2003.

[14] M. Godoy and A. Amandi. Modeling user interests by conceptual clustering. http://www.sciencedirect.com. accepted February 2005.

[15] T. R. Gruber. A translation approach to portable ontology specifications. Technical Report 5(2):199-220, Knowledge Systems AI Laboratory, Stanford University, April 1993.

[16] T. R. Gruber. Towards principles for the design of ontologies used for knowledge sharing. In *International Journal of Human and Computer Studies*, volume 43, pages 907–928, 1994.

[17] H. Happel and S. Seedorf. Application of ontologies in software engineering. In *Semantic Web enabled software engineering*, 2006.

[18] A. Hotho, S. Steffen, and G. Stumme. Ontologies improve text document clustering. In *Proceedings of The Third IEEE International Conference on Data Mining*, 2003.

[19] W. H. Hsu, A. L. King, M. S. R. Paradesi, T. Pydimarri, and T. Weninger. Collaborative and structural recommendation of friends using weblog-based social network analysis. In *Proceedings of Computational Approaches to Analyzing Weblogs, (AAAI)*, 2006.

[20] W. H. Hsu, J. Lancaster, M. S. R. Paradesi, and T. Weninger. Structural link analysis from user profiles and friends networks: a feature construction approach. In *Proceedings of the International Conference on Weblogs and Social Media, (ICWSM)*, March 2007.

[21] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[22] Hyoung R. Kim and Philip K. Chan. Learning implicit user interest hierarchy for context in personalization. In *Proceedings of the 8th international conference on Intelligent user interfaces, (IUI)*, pages 101–108, New York, NY, USA, 2003. ACM.

[23] A. Levy. Logic-based techniques in data integration. In *Logic-based artificial intelligence*, pages 575–595. Kluwer Academic Publishers, 2000.

[24] X. Li, L. Guo, and Y. Zhao. Tag-based social interest discovery. In *Proceedings of the 17th International World Wide Web Conference, (WWW)*, Beijing, China, 2008.

[25] D. L. McGuinness. Ontologies come of age. In Dieter Fensel, Jim Hendler, Henry Lieberman, and Wolfgang Wahlster, editors, *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press, 2003.

[26] J. Mori, T. Tsujishita, Y. Matsuo, and M. Ishizuka. Extracting relations in social networks from the web using similarity between collective contexts. In *Proceedings of the 5th International Semantic Web Conference, (ISWC)*, Athens, Georgia, 2006.

[27] N. Noy and H. Stuckenschmidt. Ontology alignment: an annotated bibliography. In Y. Kalfoglou, M. Schorlemmer, A. Sheth, S. Staab, and M. Uschold, editors, *Semantic Interoperability and Integration*, number 04391 in Dagstuhl Seminar Proceedings, 2005.

[28] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. In *Genome Research*, pages 2498–2504, 2003.

[29] B. Shridharan, A. Tretiakov, and Kinshuk. Application of ontology to knowledge management in web based learning. In *IEEE International Conference on Advanced Learning Technologies*, 2004.

[30] L. Singh, M. Beard, L. Getoor, and M. B. Blake. Visual mining of multi-modal social networks at different abstraction levelsx. In *L. Singh, M. Beard, L. Getoor, M. Blake. Visual mining of multi-modal social networks at different abstraction levels. IEEE Conference on Information Visualization - Symposium of Visual Data Mining (IV-VDM)*, 2007.

[31] P. N. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In *Proceeding of the ACM SIGKDD International Conference on Knowledge Discovery in Databases, (KDD)*, pages 32–41. ACM SIGKDD, 2002.

[32] I. H. Witten, E. Frank, L. Trigg, M. Hall, G. Holmes, and S. J. Cunningham. Weka: practical machine learning tools and techniques with java implementations. In Nikola Kasabov and Kitty Ko, editors, *Proceedings of the ICONIP/ANZIIS/ANNES'99 Workshop on Emerging Knowledge Engineering and Connectionist-Based Information Systems*, pages 192–196, 1999. Dunedin, New Zealand.

[33] J. Zhang, D. Caragea, and V. Honavar. Learning ontology-aware classifiers. In *Proceedings of the Eight International Conference on Discovery Science, (DS)*, volume 3735. Berlin: Springer-Verlag, 2005.