Ontology-Based Link Prediction in the *LiveJournal* Social Network

Doina Caragea, Vikas Bahirwani, Waleed Aljandal and **William H. Hsu**
Department of Computing and Information Sciences, Kansas State University
234 Nichols Hall, Manhattan, KS 66506, USA

Abstract

LiveJournal is a social network journal service with focus on user interactions. As for many other online social networks, predicting potential friendships in the *LiveJournal* network is a problem of great practical interest. Previous work has shown that graph features extracted from the graph associated with the network are good predictors for friendship links. However, contrary to the intuition, user data (e.g., interests shared by two users) does not always improve the predictions obtained with graph features alone. This could be due to the fact that features constructed from a large number of user declared interests cannot capture the implicit semantic of the interests. To test this hypothesis, we use a clustering approach to build an interest ontology, and explore the ability of the ontology to improve the performance of learning algorithms at predicting friendship links, when interest-based features are used alone or in combination with graph-based features. The results show that ontology-based features can help improve the performance of several machine learning classifiers (in particular, random forest classifiers) at the task of predicting links in the *LiveJournal* social network.

Introduction

In the recent years we have witnessed the advent of many online social networks and social network activities. Many of these networks, including *LiveJournal* online service (Fitzpatrick 1999), are focused on user interactions. *LiveJournal* users can tag other users as their friends. In addition to tagging friends, users can also specify their demographics and declare their interests.

Given the emphasis on user interaction, the *LiveJournal* network can be represented as a graph, wherein the nodes of the graph correspond to users of the online journal service (together with the information associated with them, e.g. user interests) and edges correspond to friendships between users. Assuming that a user *Jim* has tagged user *Sue* as his friend, then there exists a directed edge from *Jim* to *Sue* in the underlying graph. However, if *Sue* has not tagged *Jim* as her friend yet, there is no link from *Sue* to *Jim* in this graph. In general, the graph corresponding to a social network is an undirected graph.

One desirable feature of an online social network is to be able to suggest potential friends to its users. This task can be cast as a link prediction problem (Taskar et al. 2003) and addressed using machine learning prediction algorithms. More precisely, the task is to predict the existence (true or false) of a friendship link from user A to user B in a social network. Intuitively, knowledge of the tagged friends of a user (encoded as graph-based features) together with knowledge of the interests that the user shares with other users (encoded as interest-based features) should be predictive of friendship links. Indeed, previous work (Taskar et al. 2003; Liben-Nowell and Kleinberg 2003; Hsu et al. 2006) has shown that graph-based features are predictive of new friendships.

However, interest-based features have not proven to be effective when predicting friendships. One possible explanation for this is that interest-based features cannot capture the implicit semantic of the large number of declared user interests. To illustrate this issue, suppose that user Joe is interested in $football$ and user $Mike$ is interested in $basketball$. Furthermore, they have a common friend, $Jerry$, that is interested in $tennis$. While $football, basketball$ and $tennis$ are different interests, they are all $sports$ and it might be possible for Joe and $Mike$ to become friends, as they are both interested in $sports$ and have a mutual friend, $Jerry$.

In this paper, we study the effect of an interest ontology on the performance of the friendship link prediction problem in the *LiveJournal* social network, when interest-based features are used alone or in combination with graph-based features. An ontology can be seen as an explicit description of the concepts and relationships among concepts in a domain of interest (Gruber 1993). In particular, in this work we will construct and use a simple hierarchical ontology (expressing is-a relationships among interests), such as the $Sports$ ontology shown in Figure 1. We expect that higher levels of abstraction in the hierarchy can capture the implicit common semantic of interests, such as $football$, $basketball$ and $tennis$, thus resulting in better performance when used with machine learning algorithms.

The rest of the paper is organized as follows: We first describe the procedure used to construct the interests ontology. Then, we formally define the link prediction problem and the features (graph-based and interest-based) used to predict links, using a variety of traditional machine learning algo-

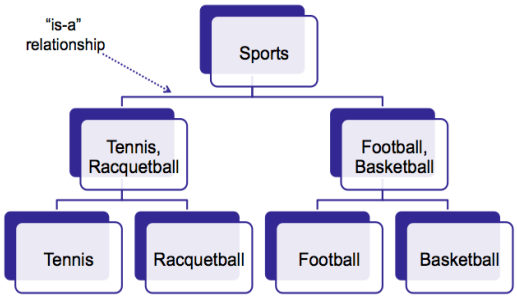

Figure 1: Sports Ontology

rithms. Next, we describe our experimental design and results. We end with a discussion of the results, conclusions and ideas for future work.

Interests Ontology

The data set used in this work consists of 1000 users of the *LiveJournal* online service. Surprisingly, there are approximately 22,000 interests that these users have collectively declared. Our goal is to organize these interests into an ontology. The procedure we have designed for this task consists of three main steps that are meant to make the ontology construction process as fast as possible and at the same time to produce a *sensible* and *useful* ontology. Briefly, the three steps are as follows:

- In the first step, the algorithm fetches descriptions of interests expressed by *LiveJournal* users, from three main sources: WordNet-Online, IMDB (movies) and Amazon (books). An interest may have multiple descriptions if the corresponding word has several meanings. Every description of an interest represents a "concept" that will be included in the resulting ontology.
- The second step divides the resulting concepts into different high-level clusters based on the sources from which the descriptions were fetched and based on the "genres" of books specified as interests (as there is a large number of book interests in our data).
- At the final step, our algorithm constructs a concept hierarchy in a bottom-up fashion for each high-level cluster. It produces a tree whose root collectively represents all concepts in that cluster and whose nodes represent concepts at various levels of abstraction in the cluster.

We describe the above steps in detail in what follows.

Obtaining Interest Descriptions

Each of the 21,811 unique interests is read from a text file and queried against three different sources for potential definitions or descriptions. We seek information from WordNet-Online for the meanings of valid words, Internet Movie Database (IMDB) for descriptions of movies, and Amazon.com for descriptions of books via the Amazon Associates Web Services (AWS). We have chosen to retrieve specific descriptions for movies and books, because many interests in our data are related to book and movie concepts. Each description we fetch corresponds to a concept in the data set to be fed to the clustering algorithm. In general, an interest can have more than one description (e.g., an interest word can have multiple meanings or an interest can be both a movie and a book), thus generating more than one concept.

Figure 2 illustrates the process of obtaining interest definitions from the three sources mentioned above. As can be seen, each interest is queried to *WordNet-Online*, *IMDB* and *AWS* in no particular order. The descriptions fetched from each source are separated into tokens (after removing stop words) and represented as a list of tokens. Each description corresponds to a concept that will be part of the hierarchy constructed by algorithm.

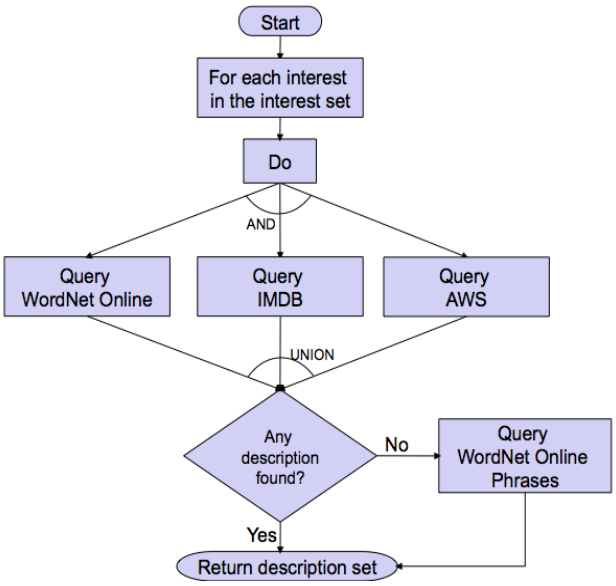

Figure 2: Process of obtaining interest definitions

To illustrate the outcome of this process, we fetch the descriptions of the interest "character" from the three data sources considered, *WordNet-Online*, *IMDB* and *AWS*. All sources have descriptions of this interest and the resulting description set looks like (in fact, this is only a subset of the actual description set):

- WordNet Online
  - character | grapheme | graphic | symbol | written | used | represent | speech | ...
  - character | genetics | functional | determined | gene | group | ...
- IMDB
  - character | reality | film | fantasy | history | character | movie | dream | rocky | ...
- AWS
  - character | novel | butcher | covey | davenport | detective | effective | favor | files |...

The *WordNet-Online* was not queries for phrases in this case, as the description set obtained from *WordNet-Online*, *IMDB* and *AWS* was non empty. Similarly, if we search for the interest "Harry Potter", we will find descriptions in *IMDB* and *AWS* (there are no descriptions in *WordNet-Online* and no need to search in *WordNet-Online-Phrases*).

- IMDB
  - harry potter | fantasy | adventure | chris | columbus | family | magic | hogwarts |...
- AWS
  - harry potter | books | deathly | half-blood | harry | phoenix | potter | rowling |...

However, if we query the interest *aim prank*, no definitions are found in *WordNet-Online*, *IMDB* and *AWS*. Therefore, for this interest we query *WordNet-Online-Phrases* for an alternate description. This is achieved by querying *WordNet-Online* for each of the constitutive words of the "phrase" *aim prank* and concatenating their descriptions. The resulting description set in this case will look like:

- WordNet Online - Phrases
  - aim | purpose | intention | design | pranks | buffoonery | prank | acting | clown | ...

The procedure for obtaining interest descriptions receives 21,811 user interests as input and fetches 42,096 descriptions (as an interest can have multiple descriptions). The clustering algorithm will consider the fetched descriptions as independent concepts. As a result, different concepts corresponding to the same interest word (e.g., "jaguar": car and "jaguar": animal; or "Harry Potter":book and "Harry Potter":movie) will possibly be placed in different clusters by the algorithm. This feature of the approach can be exploited to address the semantic heterogeneity problem (in particular, word sense disambiguation). Thus, if a user A is interested in "jaguar" and it is friends with other users that are interested in "luxury cars", then we can infer that the user A is interested "jaguar" as a "car". Similarly, if the user is interested in "Harry Potter", based on its friends interests we can infer if the user if interested in "fantasy books", "fantasy movies" or maybe both.

Divisive Clustering Step

After descriptions for interests are fetched, the next step is to divide the resulting concepts into four major clusters based on source, as shown in Figure 3. The first cluster consists of all the concepts that are described in terms of meaningful words from WordNet-Online. The second cluster consists of movie descriptions fetched from IMDB. The third cluster comprises of book descriptions from AWS and the fourth contains concepts with descriptions obtained by querying the WordNet-Online for phrases. The 21,811 unique interests queried for descriptions generate 17,753 valid word descriptions, 4,189 movie descriptions, 18,168 book descriptions and 1,986 alternate word descriptions resulting in precisely 42,096 concepts 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) as shown in Figure 3. Similar to books, movies could also be divided based on their genres. However, since the number of movie concepts is relatively small compared with the number of book concepts, the "movie" cluster is not further divided in this step. Thus, the final number of high-level clusters obtained as a result of the divisive clustering step is 19.

There are two main advantages we gain by dividing the data into several high-level clusters before applying the hierarchical agglomerative clustering algorithm. First, we can apply this algorithm in parallel to the high-level 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.

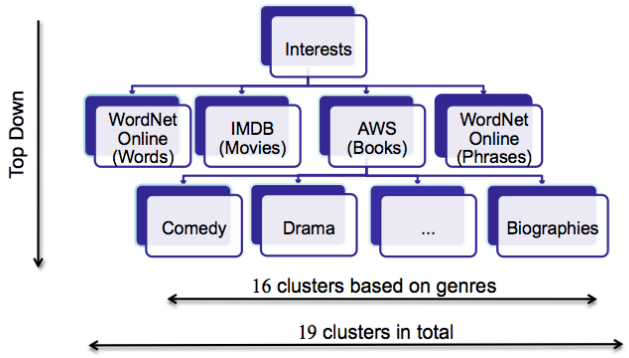

Figure 3: Divisive Clustering Step: Interest descriptions are divided into clusters based on source and book genres.

Agglomerative Clustering Step

The hierarchical agglomerative clustering algorithm is independently applied to each of the nineteen clusters obtained in the previous step. It works in a bottom-up fashion and groups instances together based on their similarity, as described below. The agglomerative clustering step takes as input the set of interest descriptions (or basic concepts) in a particular high-level divisive cluster. Initially, each concept is considered to be a singleton cluster. The set of all active clusters is called the "working set" and it initially consists of the singleton clusters. The working set is updated every time a new cluster is formed: the newly formed cluster is added to the working set, while its constitutive subclusters are removed from the working set. The algorithm ends when the working set contains only one element or elements that cannot be merged anymore (resulting ontology denoted by O).

To complete the description of this step, we need to explain how we calculate the similarity between two clusters. First, similarity between two singleton clusters is defined as the number of common tokens describing the corresponding

concepts. Similarity between two non-singleton clusters is considered to be the average similarity among pairs formed with elements from the two clusters (i.e., average linkage). A cluster A is grouped with another cluster B if the similarity between clusters A and B is maximum among all the possible pairs of clusters in the current cluster set.

When no more groupings are possible (zero similarity), the remaining clusters in the working set are combined to form a single cluster, which represents the root of the corresponding hierarchy. We have applied this algorithm to the nineteen high-level clusters (obtained in the divisive step), using a multi-threaded execution paradigm, and obtained nineteen sub-ontologies. Together, these sub-ontologies form a unified ontology of interests of the 1000 users of *LiveJournal* social network, considered in our study.

The resulting ontology contains interest concepts at various levels of abstraction. Each global cut through the ontology corresponds to a level of abstraction, the root corresponding to the most abstract level. In this work, we consider depth-guided global cuts through the ontology. This means that concepts at depth n from the root are on the n-th level, together with concepts at more abstract levels that don't have descendants at the n-th level. This enumeration procedure results in 43 levels.

Friendship Link Prediction

The main goal of this work is to study the effect of the interest ontology on the performance of learning algorithms at the tasks of predicting friendship links in the *LiveJournal* social network. The graph network used in the study consists of 1,000 nodes (users) and 7,500 links (declared friendships). Collectively, the users in the network specify 21,811 interests which correspond to 42,096 concepts. These concepts are organized in an ontology, O, as described in the previous section. Global cuts through these ontologies result in concepts specified at higher or lower levels of abstraction.

The prediction task addressed can be stated as follows: given a pair of users $< A, B >$, predict the existence (true or false) of a directed link from user A to user B. To do that, we need to represent pair of users $< A, B >$ as a feature vector. In preliminary work on this problem (Bahirwani et al. 2008), we considered three types of features: interest-based nominal features, interest-based numerical features and graph-based features. Given the poor results obtained using interest-based nominal features, we don't include them in the current study. Thus, the focus here will be on interest-based numerical features (constructed with or without the ontology) and graph-based features.

Interest-Based Numerical Features

The intuition behind the interest-based numerical features we construct is that if two users have many interests in common, then it is possible that they are friends, regardless of exactly what those interests are. Furthermore, if the users don't have many common interests, but share a very rare interest, they might also be friends. We derive eight numerical features that capture this intuition by "measuring the interestingness" of a the set of common interests that two users A and B share (Aljandal et al. 2008). This is achieved by regarding each interest as an itemset whose elements are the users having that interest. From each interest itemset i, association rules of the form $A(i) \rightarrow B(i)$ (meaning, if A has interest i, then B has interest i) are derived. The resulting association rules are used to define eight objective measures of rule interestingness, from which eight interest-based numerical features are constructed.

Suppose that $numInt_A$ denotes the number of interests of user A, $numInt_B$ denotes the number of interests of user B and $numInt_{AB}$ denotes the number of mutual interests of users A and B. For a user pair $< A, B >$, we define the following probabilities:

- Probability that A has an interest: $P(A) = \frac{numInt_A}{TotNumInt}$
- Probability that B has an interest: $P(B) = \frac{numInt_B}{TotNumInt}$
- Probability that A and B have a common interest: $P(AB) = \frac{numInt_{AB}}{TotNumInt}$

Using these probabilities and the *Bayes' Theorem*, the following eight measures of interestingness can be derived:

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. $Conviction(A \rightarrow B) = \frac{P(A) - P(\neg B)}{P(A\neg B)}$
6. $Match(A \rightarrow B) = \frac{P(AB) - P(A) * P(B)}{P(A) * (1 - P(A))}$
7. $Accuracy(A \rightarrow B) = P(AB) + P(\neg A \neg B)$
8. $Leverage(A \rightarrow B) = P(B|A) - P(A)P(B)$

As pointed out in (Aljandal et al. 2008), these measures do not take into account the relative size of the itemset to which each candidate pair $A \rightarrow B$ belongs. For example, some interests that have low membership (i.e., small itemset size), such as *DNA replication* (an example of rare interest), often imply a more signicant association between users listing them, than common interests, such as $music$ or $games$ that are shared by many users. To address this limitation, Aljandal et al. (2008) have derived a normalization factor that takes into account the popularity of particular interests that two users share, with the most popular interests (held by a signicant proportion of users) being slightly less revealing than rarer interests. Experimental results (Aljandal et al. 2008) show that link prediction algorithms presented with normalized interestingness measures (as numerical features) give better results than those presented with the features obtained from non-normalized interestingness measures. Hence, we use the normalized versions of the eight association rule measures mentioned above to derive eight interest-based numerical features in our experiments.

Graph-Based Features

Similar to (Hsu et al. 2006), for each directed edge $A \rightarrow B$ in the social network graph, we construct and use the following graph-based features in our study:

1. *In-degree* of A: The number of incoming edges to the node corresponding to user A (represents the popularity or importance of A).
2. *In-degree* of B: Similar to *in-degree* of A, this is the number of incoming edges to the node corresponding to B.
3. *Out-degree* of A: The number of friends of user A (except for user B). This number is computed by counting the number of outgoing edges (except for $A \rightarrow B$) from the node corresponding to user A in the social network graph.
4. *Out-degree* of B: Similar to *out-degree* of user A, this represents the number of friends of B except for A.
5. *Mutual friends of A and B* - 4 types considered:
   - Number of mutual friends C, s.t. $A \rightarrow C$ and $C \rightarrow B$.
   - Number of mutual friends C, s.t. $B \rightarrow C$ and $C \rightarrow A$.
   - Number of mutual friends C, s.t. $A \rightarrow C$ and $B \rightarrow C$.
   - Number of mutual friends C, s.t. $C \rightarrow A$ and $C \rightarrow B$.
6. *Backward distance* from B to A: The minimum distance from the node corresponding to user B to the node corresponding to user A in the graph.

Experimental Design and Results

We evaluate the ability of the interest ontology to improve the performance of traditional learning algorithms such as decision trees (J48), support vector machines (SVM), random forests (RF) and logistic regression (LR) classifiers (Mitchell 1997) at the task of predicting friendship links. To do this, interest-based numerical features (with and without the ontology) are used by themselves and in combination with graph-based features. The expectation is that when combining the two types of features, the classification results are better than the results obtained using only one type of features alone (either interest-based or graph-based).

Experimental Design

As mentioned before, our data set consists of $1,000$ users and approximately $7,500$ declared friendship links (out of 1000×1000 possible links in an undirected graph with $1,000$ nodes). We make the assumption (obviously, violated in practice) that the graph network is complete, i.e. all declared friendships are positive examples and all non-declared friendships are negative examples. That means, our data set is highly skewed towards the negative class, the ratio between the positive and negative classes being less than $1 : 99$. We randomly divide the original data set into three subsets (sample without replacement):

- A training set consisting of 50% positive links and 50% negative links (approximately 3,700 friendships and 3,700 non-friendship links). This data set is used to train the classifiers considered.
- A validation set consisting of data that has the original distribution (approximately, 1850 friendship links and 240,000 non-friendship links). This data set is used to find the best level of abstraction for an ontology.
- A test set also consisting of data having the original distribution (approximately, 1850 friendship links and 240,000 non-friendship links). We use this data set to evaluate the true performance of the classifiers, given the best level of abstraction for the ontology.

From each data set we remove the links that go across data sets to ensure that the three sets are independent. Interest-based features are constructed using a particular level in the ontology. For the training set, graph-based features are constructed using all the available friendship links. However, for the validation and test sets, we want to predict the friendship links. To be able to construct graph-based features for these sets, we follow the approach in (Taskar et al. 2003). More precisely, we assume that a certain percent of the links are known in the validation/test graphs (in particular, we explore scenarios where 10%, 25% or 50% links are known), construct graph-features based on the known links only and predict the "unknown" links. As our data set is highly imbalanced, we report the performance of a learning algorithm as the area under the ROC curve (called AUC). The ROC curve depicts the tradeoff between the true positive rate and the false positive rate.

The experiments we have designed are meant to address several questions, including:

- How does the performance of a classifier vary with the number of concepts used to construct the interest-based features (i.e., with the level of abstraction in ontology)?
- What is the performance of an algorithm when interest-based numerical features (constructed with or without ontology) are used by themselves?
- Does the ontology help improve the results obtained using only graph-based features?

We considered two baseline models in our study, a model using interest numeric features constructed from the original interest descriptions (leaf level in the ontology) and another one using graph-based features only. We compared these models against models that use interest numerical features constructed based on the ontology or models that use both interest-based numerical features and graph-based features. Each experiment that we performed was repeated 5 times and the results were averaged over the 5 runs. A *paired t-test* (Mitchell 1997) was performed to investigate if one model is significantly better than another one.

Results

Weka implementations (http://www.cs.waikato.ac.nz/ml/weka/) of the learning algorithms considered in our study were used (default parameters). Table 1 shows the results of the comparison between classifiers that use numerical features constructed with or without making use of the ontology. The results for O(best level) are averaged over the 5 runs performed, and the best level of abstraction for each run is shown under the corresponding AUC value. Values

highlighted in bold are statistically significant according to the t-test. In particular, the performance of the SVM and LR classifiers that use the best level of abstraction is significantly (based on t-test) greater than the corresponding classifiers that use numeric features based on original descriptions (leaf level in the ontology). However, this is not the case for the RF and J48 classifiers.

Tables 2 show the results of the comparison between classifiers built using graph-based features only and classifiers built using graph-based and interest-based numerical features (without ontology, and in the presence of the ontology - values for best levels for each run averaged and best levels shown under the corresponding average value). Each horizontal section in the table shows the results for cases when 10%, 25% and 50% links are known, respectively. As before, values highlighted in bold are statistically significant according to the t-test. As expected, the performance increases with the percent of links known. Furthermore, when graph based features are used in combination with numeric interest based features, the use of the ontology results in improvements of the performance consistently for all classifiers, although sometimes the best level is obtained for the level just above the leaf level (most often for the SVM and logistic regression classifiers).

As can be seen from the table, the best results are obtained using the random forest classifiers (next best being SVM), while the worst are obtained using decision trees. Another important fact to note is that the best level is usually very high (low number) for decision tree classifiers and sometimes for random forests as well. For example, one of the random forests runs shows that as few as 2207 (level 9) abstract interests (out of 42,096 possible interests) used with random forests can provide results better than those obtained with more specific interests. It is also worth observing, that in the presence of ontology, SVM outperforms random forests in several cases (e.g., 50% links know, graph features only or graph features and interest-based numerical features without ontology). However, the results are always better using the random forest algorithm when the ontology is used to construct interest-based numerical features.

Figure 4 shows the variation of the AUC performance with the level of abstraction (and implicitly, number of nodes used) in the case of the random forest classifier, which uses interest-based numerical features. All 5 runs are shown. Similar graphs (not shown here) are obtained from the other classifiers constructed.

Related Work

There is a significant body of work related to ontology construction and link prediction. We discuss several ontology construction approaches, but we are not aware of any work that specifically addresses the problem of building an ontology over interests specified by users of a social network. Among the many approaches to link prediction in social networks, we discuss two that are the most relevant to our work.

Ontology Construction

Recent work in machine learning has focused on learning implicit concepts and concept hierarchies from data using clustering approaches. Kang et al. (2004) introduce a hierarchical agglomerative clustering algorithm for constructing attribute value taxonomies from data. This algorithm constructs a taxonomy (binary tree) by recursively grouping attribute values based on the Jensen-Shannon divergence (Slonim and Tishby 1999) between distributions of classes associated with attribute values. Experimental results show that learning algorithms that make use of attribute values taxonomies to reduce the number of features (e.g. Naive Bayes) learn simpler still accurate models as compared with classifiers that do not use these taxonomies. Our ontology construction approach is similar in spirit to this approach, although the details differ significantly.

As opposed to the bottom-up approach in (Kang et al. 2004), Punera et al. (2006) present a top-down approach for constructing an n-ary hierarchy of classes, given a set of labeled data points. Their algorithm uses a divisive clustering paradigm to build an n-ary hierarchy by computing the similarity between sets of class labels also based on the Jensen-Shannon divergence. Experimental results show that n-ary taxonomy aware classifiers yield better results than classifiers that use binary taxonomies.

Kim and Chan (2003) address the problem of modeling web user interests by classifying the web pages that a particular user visits. They introduce a top-down divisive hierarchical clustering method to recursively divide parent clusters into child clusters until a termination criterion is met. Godoy and Amandi (2005) describe a practical way to implement the method introduced in (Kim and Chan 2003). One main limitation of this approach is that is does not allow an interest to belong to two concepts. The hierarchical agglomerative clustering algorithm that we used does not present this limitation, as we find all descriptions of an interest and regard them as distinct concepts.

Link Prediction

Two commonly addressed problems in social networks are: object classification (labeling the nodes of a graph) and link prediction (labeling the links in a graph). Object classification is usually performed by assuming a complete set of known links (Getoor 2003). As opposed to that, link prediction problems are usually addressed by assuming a fully observed set of node attributes (Getoor 2003). However, in many real world domains, node attributes and links are often missing or incorrect. Thus, the object classification algorithm is not provided with all the relevant links, while the link prediction algorithm is not provided with all the node attributes needed for accurate prediction. Bilgic et al. (2007) 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 this approach, the object classification algorithm is provided with information about known and predicted links, while the link prediction algorithm is provided with information about known and predicted node classifications, until both the object classifications and the link predictions converge. Experimental results show that the collective classification algorithm performs better than the "flat" prediction approach (where each problem is addressed independently).

Table 1: AUC for different classifiers presented with interest-based numerical features, constructed without and with ontology. Average values over 5 runs shown. For O(best level), the best levels for the 5 runs are also shown under the corresponding average value. Values highlighted in bold are statistically significant according to the t-test.

Features	SVM	LR	RF	J48
Without O	0.65±0.01	0.66±0.01	0.78±0.01	0.66±0.01
O (best level)	**0.67±0.00**	**0.68±0.00**	0.77±0.01	0.66±0.01
	(42,42,42,42,42)	(25,27,42,40,42)	(16,24,30,38,29)	(27,19,26,24,28)

Table 2: AUC for different classifiers presented with graph featuers and interest-based numerical features (constructed without and with ontology). 10%, 25% and 50% links known, respectively. Average values over 5 runs shown. For O(best level), the best levels for the 5 runs are also shown under the corresponding average value. Values highlighted in bold are statistically significant according to the t-test.

Features	SVM	LR	RF	J48
Graph only	0.69±0.01	0.67±0.01	0.70±0.04	0.61±0.08
Graph, without O	0.68±0.01	0.68±0.01	0.69±0.05	0.57±0.09
Graph, O (best level)	**0.70±0.00**	**0.69±0.01**	**0.74±0.04**	**0.64±0.06**
	(42,35,37,42,34)	(42,28,17,21,17)	(9,13,38,26,27)	(2,3,5,22,6)
Graph only	0.71±0.01	0.67±0.01	0.72±0.02	0.67±0.05
Graph, without O	0.74±0.01	0.72±0.01	0.71±0.03	0.65±0.04
Graph, O (best level)	**0.76±0.01**	**0.74±0.01**	**0.79±0.02**	**0.71±0.05**
	(42,36,42,41,23)	(42,40,42,29,32)	(42,36,19,31,27)	(6,22,2,5,6)
Graph Only	0.82±0.01	0.79±0.01	0.80±0.01	0.77±0.03
Graph, without O	0.85±0.01	0.83±0.01	0.82±0.02	0.76±0.02
Graph, O (best level)	**0.86±0.01**	**0.85±0.01**	**0.86±0.02**	**0.78±0.02**
	(42,42,42,27,23)	(42,23,21,29,42)	(42,36,26,18,27)	(6,28,2,26,27)

Taskar et al. (2003) have also used a collective classification approach, based on relational Markov networks, to the problem of link prediction in relational data. Relational Markov networks can be used to define a joint probability distribution over a graph (both node attributes and links). One of the data sets in their study is a student social network data set and the focus is on predicting friendship links form student information and a subset of known links (in particular, they assume that 10%, 25% and 50% are known, respectively). We used a similar experimental design.

Conclusions and Future Work

We have addressed the problem of building an interests ontology and predicting potential friendship links between users in the social network *LiveJournal*, using interest-based numerical features (constructed in the presence of ontology), in addition to graph-based features. With respect to this problem, our main goal was to organize user 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 friends.

One contribution of this work is an approach for constructing an interest concept hierarchy. Our approach relies on extracting interest descriptions from several sources and using an agglomerative hierarchical clustering algorithm to group descriptions into concepts. The algorithm implementation is general and can be used to construct similar ontologies in other domains.

Another contribution of our work is the construction of a set of ontology-aware interest-based numerical features that can be used to build classifiers for link prediction. We have performed experiments to explore the effectiveness of the interest-based numerical features at predicting friendships. Our results show that ontology-aware interest-based numerical features perform better at capturing interest information (and hence at predicting friendships) than their non-ontology-aware counterparts, especially when used in combination with graph-based features. Best results were obtained for the random forest algorithm.

Future work ideas include:

- Predicting friendship links in incomplete social networks (where the absence of a link between two users does not necessarily mean the two users cannot be friends).
- Including user-defined ontologies, when available, in the process of ontology engineering (building a global ontology on the top of small user-defined ontologies).
- Incorporating information from Wikipedia and Google in the ontology engineering process (by using them as sources of interest descriptions).
- Using other similarity measures for comparing descriptions, such as cosine similarity or similarity based on *Latent Semantic Analysis* (Deerwester et al. 1990).
- Refining interest-based concepts using a variable-depth cut through the ontologies.

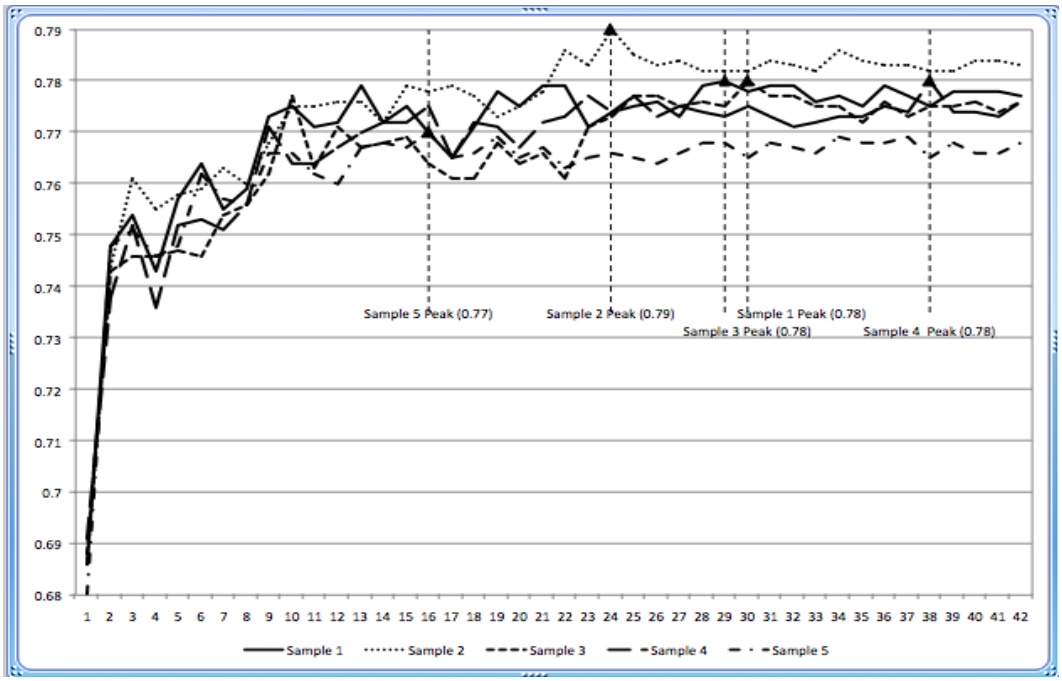

Figure 4: Dependency of the AUC performance on the level of abstraction. Results shown for the random forest classifiers that use interest-based numerical features. All 5 runs are shown. The best values for each run are also marked.

Acknowledgments

This work was funded by the National Science Foundation grant number NSF 0711396 and US Department of Defense.

References

Aljandal, W.; Hsu, W. H.; Bahirwani, V.; Caragea, D.; and Weninger, T. 2008. Validation-based normalization and selection of interestingness measures for association rules. In *Proc. of Artificial Neural Networks in Engineering*.

Bahirwani, V.; Caragea, D.; Aljandal, W.; and Hsu, W. H. 2008. Ontology engineering and feature construction for predicting friendship links in the live journal social network. In *Proc. of the 2nd ACM Workshop on Social Network Mining and Analysis (SNA-KDD)*.

Bilgic, M.; Namata, G. M.; and Getoor, L. 2007. Combining collective classification and link prediction. In *Workshop on Mining Graphs and Complex Structures at ICDM*.

Deerwester, S.; Dumais, S. T.; Furnas, G. W.; Landauer, T. K.; and Harshman, R. 1990. "indexing by latent semantic analysis. In *Journal of the American Society for Information Science*, volume 41.

Fitzpatrick, B. 1999. Live journal: online journal service. Created in 1999.

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

Godoy, M., and Amandi, A. 2005. Modeling user interests by conceptual clustering. http://www.sciencedirect.com.

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

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

Kang, D. K.; Silvescu, A.; Zhang, J.; and Honavar, V. 2004. Generation of attribute value taxonomies from data for data-driven construction of accurate and compact classifiers. In *Proc. of the 4th IEEE Int. Conf. on Data Mining*.

Kim, H. R., and Chan, P. K. 2003. Learning implicit user interest hierarchy for context in personalization. In *Proc. of the 8th Int. Conf. on Intelligent User Interfaces (IUI)*.

Liben-Nowell, D., and Kleinberg, J. 2003. The link prediction problem for social networks. In *Proc. 12th Int. Conf. on Information and Knowledge Management (CIKM)*.

Mitchell, T. M. 1997. *Machine learning*. McGraw-Hill Companies Inc., international edition.

Punera, K.; Rajan, S.; and Ghosh, J. 2006. Automatic construction of n-ary tree based taxonomies. In *Proc. of the Workshop on Ontology Mining and Knowledge Discovery from Semistructured Documents at ICDM*.

Slonim, N., and Tishby, N. 1999. Agglomerative information bottleneck. In *Proc. of the 13th Neural Information Processing Systems (NIPS)*.

Taskar, B.; fai Wong, M.; Abbeel, P.; and Koller, D. 2003. Link prediction in relational data. In *Proc. of the 17th Neural Information Processing Systems (NIPS)*.