LDA BASED APPROACH FOR PREDICTING FRIENDSHIP LINKS IN LIVE JOURNAL SOCIAL NETWORK

by

ROHIT PARIMI

B.E., Andhra University, India, 2008

A THESIS

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas
2010

Approved by:

Major Professor
Doina Caragea

Copyright

Abstract

The idea of socializing with other people of different backgrounds and cultures excites the web surfers. Today, there are hundreds of Social Networking sites on the web with millions of users connected with relationships such as “friend”, “follow”, “fan”, forming a huge graph structure. The amount of data associated with the users in these Social Networking sites has resulted in opportunities for interesting data mining problems including friendship link and interest predictions, tag recommendations among others. In this work, we consider the friendship link prediction problem and study a topic modeling approach to this problem. Topic models are among the most effective approaches to latent topic analysis and mining of text data. In particular, Probabilistic Topic models are based upon the idea that documents can be seen as mixtures of topics and topics can be seen as mixtures of words. *Latent Dirichlet Allocation* (LDA) is one such probabilistic model which is generative in nature and is used for collections of discrete data such as text corpora. For our link prediction problem, users in the dataset are treated as “documents” and their interests as the document contents. The topic probabilities obtained by modeling users and interests using LDA provide an explicit representation for each user. User pairs are treated as examples and are represented using a feature vector constructed from the topic probabilities obtained with LDA. This vector will only capture information contained in the interests expressed by the users. Another important source of information that is relevant to the link prediction task is given by the graph structure of the social network. Our assumption is that a user “A” might be a friend of user “B” if a) users “A” and “B” have common or similar interests b) users “A” and “B” have some common friends. While capturing similarity between interests is taken care by the topic modeling technique, we use the graph structure to find common friends. In the past, the graph structure underlying the network has proven to be a

trustworthy source of information for predicting friendship links. We present a comparison of predictions from feature sets constructed using topic probabilities and the link graph separately, with a feature set constructed using both topic probabilities and link graph.

Table of Contents

List of Figures

List of Tables

Acknowledgments

The following thesis, while an individual work, benefited from the insights and direction of several people. I would like to thank them all for their help and support.

First and foremost, I offer my sincerest gratitude to my adviser, Dr. Doina Caragea, who has supported me throughout my thesis with her patience and knowledge. Her perpetual energy and enthusiasm in research motivated all her advisees, including me. She was always accessible and willing to help her students with their research. In addition, Dr. Caragea provided timely and instructive comments and evaluation at every stage of the thesis process, allowing me to complete this project on schedule. I attribute the level of my Masters degree to her encouragement and effort and without her this thesis, too, would not have been completed or written. One simply could not wish for a better supervisor.

I was delighted to interact with Dr. Gurdip Singh by attending his classes and to have him as my thesis committee member. His insights into operating system concepts is second to none. His classes are a great source of information and motivate his students to think for naive solutions to some of the challenging problems in operating systems.

I am also thankful to Dr. Torben Amtoft, my thesis committee member for educating me with some of the important concepts of algorithms and ways to tackle some of the hardest problems. His classes helped me to analyze some of the algorithms on graphs, which form crucial part of my thesis.

My deepest gratitude goes to my parents, Mr. Saradhi Parimi and Mrs. Nirmala Parimi and my brother Ravi Parimi for their love and support at every stage of my life. It is because of their motivation and encouragement that I am able to complete my Masters and plan for a PhD.

I also thank my friends and colleagues especially Ashok, Lakshman, Sandeep, Samuel and Roshan for helping me in the early days of my masters and valuable discussions.

Finally, I thank Dr. Hsu and the KDD (Knowledge Discovery in Databases) group for sharing their data and learnings on previous work.

The research described in this thesis was supported by a grant from the National Science Foundation NSF 0711396.

Chapter 1

Introduction

Since the introduction of Social Network Sites (SNSs) such as *MySpace, Facebook, Orkut, LiveJournal* and *Bebo*, they have attracted millions of users [Boyd and Ellison, 2007]. Social network sites selected from among those with at least 10 million visitors worldwide have grown at a rate of more than 50 percent during the past year [1]. The share of adult internet users who have a profile on a social networking site has more than quadrupled in the past four years, from 8% in 2005 to 35% in 2009 [2]. Recent statistics from United Kingdom suggest that social networks have overtaken search engines in terms of usage [3]. This shows how users of SNSs have integrated these sites into their daily practices.

Many of these SNSs, including *LiveJournal* online services [Fitzpatrick, 1999] are focused on user interactions. Users in *LiveJournal* can tag other users as their friends. In addition to tagging friends, users can also specify their demographics and interests in this social network. Thus, *LiveJournal* in itself forms a graph structure with users (along with their specific information, e.g. user interests) corresponding to nodes in the graph and edges corresponding to friendship links between the users. In general, the graph corresponding to a social network is undirected. However, in *LiveJournal* social network, the edges are directed i.e., if a user A specifies another user B as its friend, then it is not necessary for

[1] As of 2007. $http : //www.comscore.com/Press_Events/Press_Releases/2007/07/Social_Networking_Goes_Global$

[2] $http : //pewresearch.org/pubs/1079/social - networks - grow$

[3] As of June 2010. $http : //eu.techcrunch.com/2010/06/08/report - social - networks - overtake - search - engines - in - uk - should - google - be - worried/$

user A to be the friend of user B.

Social network sites (SNSs) are increasingly attracting the attention of academic and industry researchers who are intrigued by their affordances and reach. Data mining techniques have been very effective in using the content and graph structure available, to solve various problems such as ranking web documents (Vector Space Model [Manning et al., 2008] and Page Rank [Page and Brin, 1998]), Citation Analysis [Geetor and Lu, 2003], Genre Classification [Na and Thet, 2009], Spam Detection for web-pages [Castillo et al., 2007] etc. One desirable feature of an online social network is to be able to suggest potential friends to its users [Taskar et al., 2003]. This task is also described as the link prediction problem where, we predict the existence of a friendship link from user 'A' to user 'B'. In this thesis, we aim at using the ability of machine learning algorithms to take advantage of the content and graph structure of a social network site, *LiveJournal* to predict friendship relations. The work described here, addresses this social network problem using topic modeling (a machine learning technique used to uncover latent structure in text) on the interests specified by users of the *LiveJournal* social network. We also study the scalability of our approach with the number of users in *LiveJournal* social network.

This work is an extension of prior work on link prediction in the *Machine Learning and Bio-informatics* (MLB) lab at Kansas State University and is meant to address the limitations of the prior work. Previously, Bahirwani [2008] of MLB, focused on constructing an interest ontology by obtaining the interest definitions from well known resources such as WordNet-Online, Internet Movie Database (IMDB) and Amazon Associates Web Services (AWS). An interest can have multiple definitions and each definition can be seen as an instance. By querying the online resources for definitions of the interests, they capture the semantics of interests and use them to predict/recommend friends for each user. Similarity between instances is computed as the dot product of vectors representing the instances. These instances are grouped into hierarchies using clustering algorithms. Although the approach proved to be effective in improving the predictions, it has some limitations. Firstly, for some interests, e.g. interests related to sports persona such as *Rafeal Nadal*, *Michael Schumacher*, no definitions are found in the three sources used. Secondly, the nomenclature used to represent the parent of two child nodes in the ontology constructed, does not necessarily capture the semantics of the interest. For example, if we have two interests say *Lenovo* and *Dell*, the approach used by Bahirwani [2008] represents their parent in the ontology as *Lenovo&Dell*, as opposed to *Computers* which is semantically more meaningful. Also, the ontology constructed in this work is a binary tree like ontology in which each node has at most two children. However, in principle, a node in an ontology can have more than two children.

Haridas [2009] of MLB, in his work, used different ontology engineering approaches that rely on more comprehensive knowledge sources such as Wikipedia and Directory Mozilla. In the first approach, he used Wikipedia to get the definitions of user interests and used Latent Semantic Analysis (LSA) to calculate the similarity between interest definitions (seen as documents) to construct a hierarchy. While the Wikipedia/LSA approach used by Haridas [2009] produced a more sensible ontology than the one produced by the approach in Bahirwani [2008], the ontology is still a binary tree and consists of internal clusters labeled based on child information. To overcome these limitations Haridas [2009] in his next approach, took advantage of existing hierarchies such as Wikipedia Category Graph (WCG) and Directory Mozilla (DMoz) respectively, to build ontologies over interests. Even though the ontology constructed using WCG was n-ary instead of binary and was able to produce parent classes which are semantically more meaningful, it is highly incomplete as WCG did not have good coverage for the interest set used in their work. This problem is resolved through the use of the ontology constructed using DMoz.

Although the ontology created by Haridas [2009] using Directory Mozilla was able to resolve the limitations reported by Bahirwani [2008], it was very dense, resulting in gain or loss of information at a rapid pace when constructing features at various levels in the ontology. Also, considerable time is taken to fetch the definitions of interests from the repositories and to organize them in a hierarchy. Furthermore, the presence of a large number of instances in the ontologies constructed by Bahirwani [2008] and Haridas [2009] made it difficult to use them directly to construct features. Instead, Associative Rule Mining (ARM) techniques were used to construct the feature set to represent user pairs for the task of predicting friendship links.

To avoid the problems that arise when building an ontology, in this work we consider a Topic Modeling approach which provide an easy way to analyze large volume of data. We explore the ability of topic modeling techniques such as Latent Dirichlet Allocation (LDA) to uncover latent structure in user interests. LDA is a probabilistic model which is generative in nature and is used to identify latent topics in each item of a collection of documents (in our case users). Number of topics to be modeled on the collection of documents results in abstract or specific topics. While we do not construct an ontology explicitly, we can simulate an ontology by varying the number of topics. Thus we bypass the problems of having a binary tree in the ontology or naming of the parent nodes in the ontology. Our goal is to investigate the ability of topic modeling approach to predicting friendship links. Recent work by Chen et al. [2009] aimed at recommending communities in a social network to its users reported that LDA performs consistently better than Associative Rule Mining.

Our approach overcomes the limitations and drawbacks reported by Bahirwani [2008] and Haridas [2009] for constructing and using ontologies for the task of friendship prediction as we do not construct any ontology. However, we implicitly simulate an ontology by varying the number of latent topics.

1.1 High Level Overview

Our goal is to apply LDA on the user interests to identify the abstract topics. To accomplish this, each user in the dataset is seen as a “document” with his interests corresponding to the content of that “document”. *MALLET: A Machine Learning for Language Toolkit* [McCallam, 2002] has an implementation of LDA and is executed over the user collection. Topic distribution thus obtained is used to construct the features that represent each user pair. We call the features generated from the above technique as *Interest Based Features* as they are generated by applying LDA over the user interest set. Some of the well known classifiers like Logistic Regression, Support Vector Machines and Random Forest are used to learn predictive models from the training data. Next, we use the existing graph structure between the users in the *LiveJournal* social network to calculate various *Graph Based Features* such as *In-degree, Out-degree, Mutual Friends* and *Backward Distance* between each user pair in the dataset. Again, Logistic Regression, Support Vector Machines and Random Forest classifiers are used to learn predictive models over *Graph Based Features*. Previous work by Hsu et al. [2007], Bahirwani [2008] and Haridas [2009] reported that the feature set constructed from both *Graph Based Features* and *Interest Based Features* was more predictive compared to the feature set constructed from either *Graph Based Features* or *Interest Based Features*. In our work, we would like to see if the *Interest Based Features* that we construct using a generative probabilistic model, LDA would improve the predictions for link prediction problem. We would also like to test our approach for improvement in predictions by different classifiers with the feature set constructed from both *Graph Based Features* and *Interest Based Features*. We first perform the above mentioned experiments on a dataset containing 1,000 *LiveJournal* users. We then repeat all the experiments with a 5,000 user dataset and 10,000 user dataset of *LiveJournal* users to study the scalability of our approach.

This thesis is organized as follows: Chapter 2 discusses previous work on social network analysis and network graph done outside MLB lab at Kansas State University. Chapter 3 describes the concept of Topic Modeling and *Latent Dirichlet Allocation* (LDA) and its implementation. In chapter 4, we formulate the problem of Link Prediction in *LiveJournal* social network. In Chapters 5 and 6, we present the experimental setup and discuss the results from the experiments, respectively. We conclude this work and present directions for future work in Chapter 7.

Chapter 2

Related Work

This chapter provides a review of previous work on techniques employed for social network analysis and Topic Modeling applications. Over the past decade, growth of social network sites have drawn the attention of many researches. Some of the challenges involved in mining richly structured datasets for tasks such as hypertext classification, segmentation, information extraction, searching and information retrieval, discovery of authorities and link discovery are discussed in Geetor [2003]. Data mining techniques are a natural choice for many researches for social network analysis. Hsu et al. [2007] have addressed the problems of predicting, classifying, and annotating friendship relations in a social network, based on the network structure and user profile data. They have used features generated from the network structure existing between the users of *LiveJournal* social network along with some numerical features calculated from the user profiles for the task of predicting link existance for a user pair. They also analyzed the ability of some of the features constructed like "Backward Distances", "Mutual Friend Count" and "Degree measures" for predicting friendship links. Their experimental results suggest that features constructed from the graph structure existing between users of a social network along with user profiles were effective in link analysis between users in the network.

A framework for modeling link distributions, taking into account object features and link features for link-based classification is proposed in [Geetor and Lu, 2003]. In this work, the authors modeled link distributions, which describe the neighborhood of links around an object, and can capture the correlation among links. They proposed an Iterative Classification Algorithm for link based classification and used Logistic Regression model for both links and content to capture joint distributions of the links. They applied this approach on datasets including both web and citation collections and reported that using link distribution improves accuracy in all cases.

Taskar et al. [2003] studied the application of a relational Markov network (RMN) framework for the task of Link Prediction. The RMN framework is used to define a joint probabilistic model over the entire link graph which includes the attributes of the entities in the network as well as the links. To facilitate the application of RMN to this task, they defined probabilistic patterns over subgraph structures. This method is applied to two relational datasets, one involving university web pages and the other a social network. They reported that the classification approach of RMSs and the introduction of subgraph patterns over link labels significantly improved the accuracy of classification task.

Castillo et al. [2007] described the importance of features computed using the content of world-wide web documents as well as the web graph formed as a result of hyper links for the task of web spam detection task. They proposed a spam detection system that combines link-based and content-based features and used the topology of the web graph by exploiting the link dependencies among the web pages. In their approach they used several link-based features such as Degree related measures, e.g *in-degree,out-degree, edge-reciprocity* etc, *PageRank*, *TrustRank*, *Truncated PageRank* calculated from the pages in their collection. Content-based features such as *Corpus Precision* and *Recall*, *Query Precision* and *Recall*, *Compression rate* were also used for spam detection. They tested their approach on large public dataset of web pages and reported that the system was accurate in detecting spam pages.

Several approaches to construct ontologies and techniques to capture the semantic information represented by an ontology for friendship link prediction task in *LiveJournal* social network are discussed in [Caragea et al., 2009], [Bahirwani et al., 2008] and [Haridas, 2009].

With the growth of data on the web with addition of new articles, web documents, social networking sites and users daily, there is an increased need to accurately process this data for extracting hidden patterns. Topic modeling techniques were successful in identifying the inherent topics in the underlying data and performed well in predicting word association and the effects of semantic association and ambiguity on a variety of language-processing tasks [Steyvers and Griffiths, 2007; Steyvers et al., 2007]. Latent Dirichlet Allocation (LDA) [Blei et al., 2003b] is one such generative probabilistic model used over discrete data such as text corpora. LDA has been applied to many tasks such as Word Sense Disambiguation [Blei et al., 2007], Named Entity Recognition [Guo et al., 2009], Tag Recommendation [Krestel et al., 2009], Community recommendations in Orkut social network [Chen et al., 2009] etc. Recent work by Liu and Niculescu-Mizil [2009] have proposed a unified framework that identifies latent topics from the content of blog data as well as take into account the author social network to find links with other authors using a hierarchical Bayesian model, LDA. However, to the best of our knowledge, an approach that uses LDA over interests specified by users of *LiveJournal* for Friendship link prediction has not yet been explored.

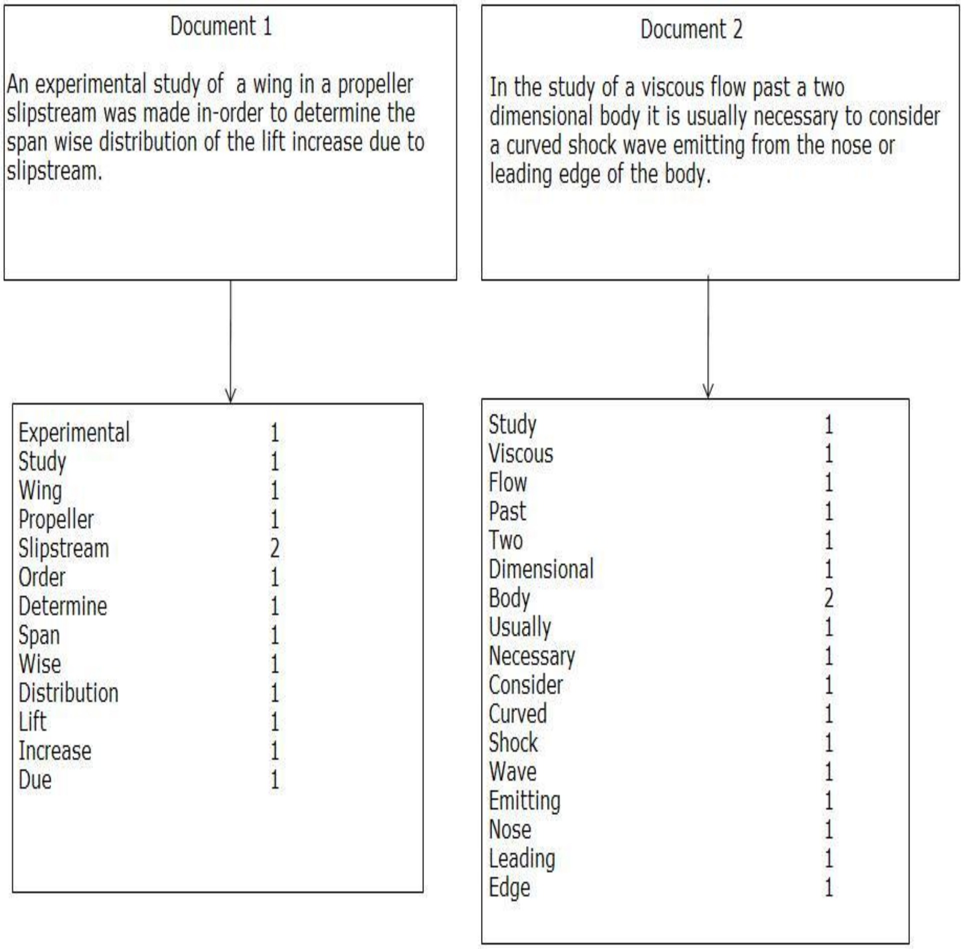
Chapter 3

Topic Models and Latent Dirichlet Allocation

3.1 Need for Topic Modeling

With the growth of data on the web in the form of web sites, articles, social networking sites, news etc., there is an increased need to process this data to extract hidden patterns and information from them. Data Mining techniques like vector space model were used in the past to extract patterns from text. Vector space model (or term vector model) is an algebraic model for representing text documents as vectors of identifiers. It uses the bag of words representation (documents are seen as vectors in the word space) to represent each document in the document corpus [Manning et al., 2008]. Fig. 3.1 depicts the bag of words representation.

However, the bag of words representation is a sparse representation with the size of the vocabulary much greater than the number of documents in the corpus. The vector space model also suffers from its inability to cope with synonymy and polysemy. Thus, there is a greater need to reduce the dimensionality and define conceptual closeness. Latent Semantic Indexing (LSI) also known as Latent Semantic Analysis (LSA) was proposed to address these two problems with the vector space model for Information Retrieval [Deerwester et al., 1990]. LSI can retrieve relevant documents even when they don't share any words with a given query. LSI uses Singular Value Decomposition (SVD) technique to reduce the



Document 1

An experimental study of a wing in a propeller slipstream was made in-order to determine the span wise distribution of the lift increase due to slipstream.

Experimental	1
Study	1
Wing	1
Propeller	1
Slipstream	2
Order	1
Determine	1
Span	1
Wise	1
Distribution	1
Lift	1
Increase	1
Due	1

Document 2

In the study of a viscous flow past a two dimensional body it is usually necessary to consider a curved shock wave emitting from the nose or leading edge of the body.

Study	1
Viscous	1
Flow	1
Past	1
Two	1
Dimensional	1
Body	2
Usually	1
Necessary	1
Consider	1
Curved	1
Shock	1
Wave	1
Emitting	1
Nose	1
Leading	1
Edge	1

Figure 3.1: *Bag of Words Representation for two documents*

dimensionality of the vocabulary obtained from the corpus. Then, it computes similarity between entities in the lower dimension space called the semantic space. The LSA approach makes three claims: that semantic information can be derived from a word-document co-occurrence matrix; that dimensionality reduction is an essential part of this derivation; and that words and documents can be represented as points in an Euclidean space [Steyvers and Griffiths, 2007]. Topic Modeling approaches on the other hand are consistent with the first two claims but differ with the third. These approaches express semantic properties of words and documents in terms of probabilistic topics. Topics are useful latent structures to explain the semantic association between documents in a collection. Topic models treat each document in the corpus as a distribution over topics and each topic as a distribution over words. They provide a better dimensionality reduction on the document corpus compared to Latent Semantic Analysis. Fig. 3.2 shows the matrix factorization of LSA and Topic Models.

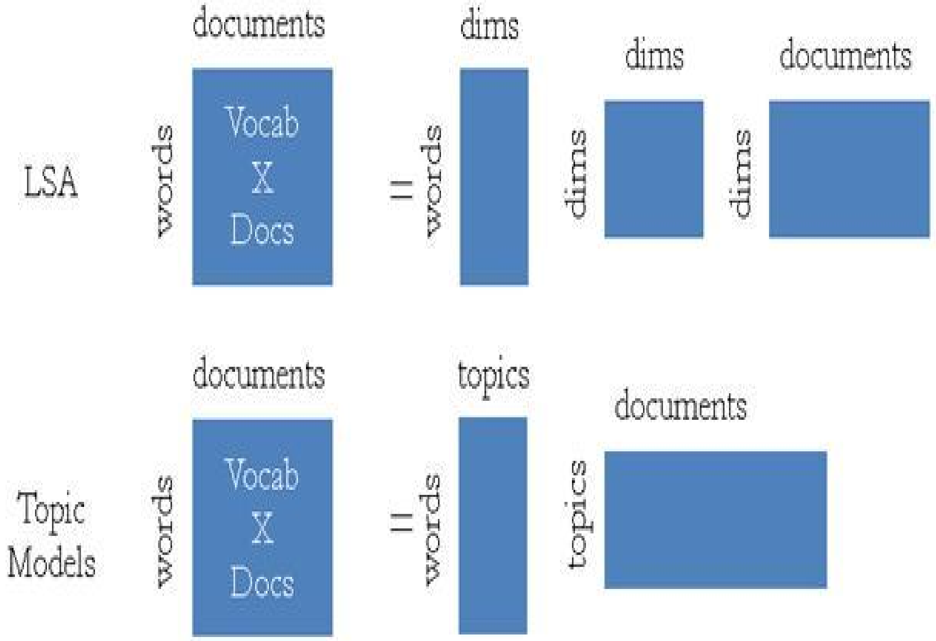


Figure 3.2: *The matrix factorization of the LSA model compared to the matrix factorization of the topic model (image adapted from [Steyvers and Griffiths, 2007])*

A topic model, in general, is a generative model for documents i.e it specifies a probabilistic way in which documents can be generated. Fig. 3.3 illustrates the topic modeling approach as a generative model. Probabilistic Latent Semantic Analysis (pLSA) is one such generative model used to model documents. However, it is reported that pLSA has severe over-fitting problems. The number of parameters grows linearly with the number of documents. In addition, although pLSA is a generative model of the documents in the collection used to estimate the model, it is not a generative model of new documents. Latent Dirichlet Allocation proposed by Blei et al. [2003b] solves the problems with over-fitting and increased number of parameters. The LDA is also a true generative model i.e. it has the ability to generate documents that do not belong to the corpus on which the model is built.

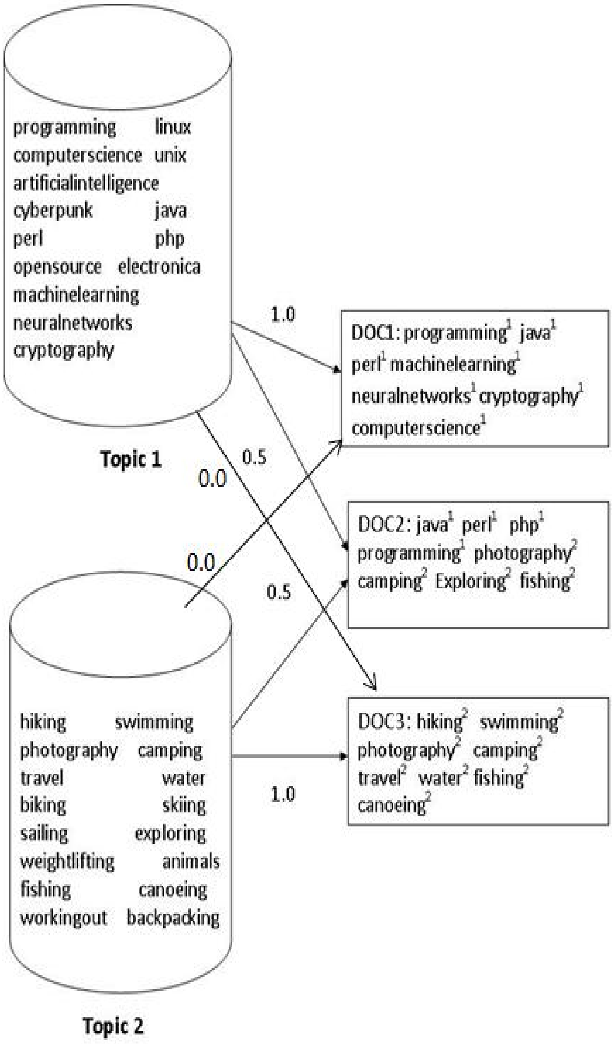

programming linux
computerscience unix
artificialintelligence
cyberpunk java
perl php
opensource electronica
machinelearning
neuralnetworks
cryptography
Topic 1
1.0
0.5
0.0
0.0
0.5
1.0
DOC1: programming^1 java^1 perl^1 machinelearning^1 neuralnetworks^1 cryptography^1 computerscience^1
DOC2: java^1 perl^1 php^1 programming^1 photography^2 camping^2 Exploring^2 fishing^2
DOC3: hiking^2 swimming^2 photography^2 camping^2 travel^2 water^2 fishing^2 canoeing^2
hiking swimming
photography camping
travel water
biking skiing
sailing exploring
weightlifting animals
fishing canoeing
workingout backpacking
Topic 2

Figure 3.3: *Illustration of Probabilistic Generative Process (adapted from [Steyvers and Griffiths, 2007])*

3.2 Latent Dirichlet Allocation: Model

Latent Dirichlet Allocation (LDA) is a probabilistic generative model for collection of discrete data such as text corpora. It was first introduced by Blei et al. [2003b]. LDA was aimed at solving the disadvantages exhibited by the pLSA model. LDA is almost similar to pLSA, except that in LDA the topic distribution is assumed to have a Dirichlet prior. Fig 3.4 illustrates the graphical model representation of LDA.

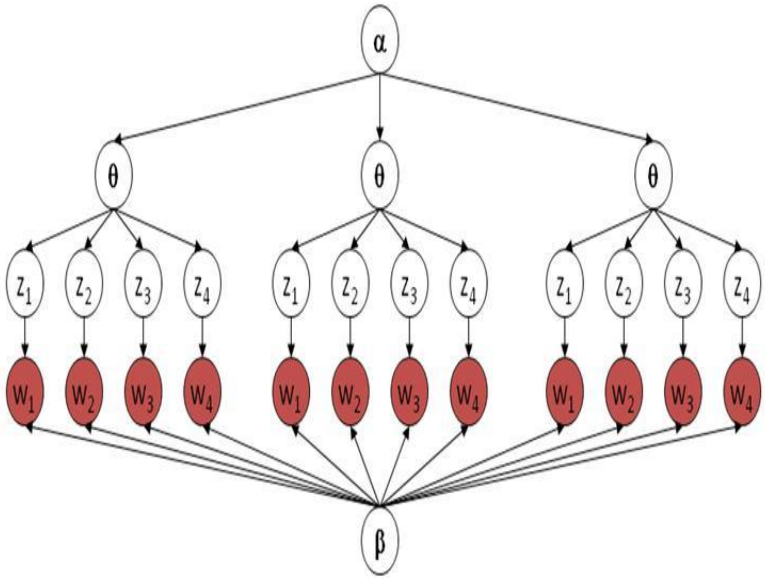


Figure 3.4: *Graphical notation representing LDA Model*

3.2.1 Terminology and Notation

The terminology and notations used to describe topic modeling using LDA is described below.

- A *word* is the fundamental unit to describe discrete data. It is an item from a vocabulary indexed by $1, \cdots, V$. Words are represented using a vector and v^{th} word in the vocabulary is represented by a V-vector w such that $w^v = 1$ and $w^u = 0$ for $u \neq v$.

- A *document* is defined as a sequence of N words denoted by $\mathbf{w} = (w_1, w_2 \cdots w_N)$, where w_n is the n^{th} word in the sequence.

- A *corpus* is a collection of M documents denoted by $D = \{\mathbf{w_1}, \mathbf{w_2} \cdots \mathbf{w}_M\}$.

The generative process that LDA assumes for each document $\mathbf{w}$ in a corpus D is presented below:

1. Choose $\theta \sim Dirichlet(\alpha)$.
2. For each of the N words w_n:
    - Choose a topic $z_n \sim Multinomial(\theta)$.
    - Choose a word w_n from $p(w_n|z_n, \beta)$, a multinomial probability conditioned on the topic z_n

Please refer to [Blei et al., 2003b] for more details on the generative process of LDA.

Some of the assumptions that we make to obtain a generative probabilistic LDA model of a corpus are as follows. Firstly, we assume that the dimensionality k of the Dirichlet distribution hence the dimensionality of topic variable z is known and fixed. Second, we assume that the word probabilities are parameterized by a k X V matrix β where $\beta_{ij} = p(w^j = 1|z^i = 1)$, which is treated as a fixed quantity that is to be estimated.

A k-dimensional Dirichlet random variable θ which lies in the (k-1)-simplex has the following probability density on this simplex:

$p\left(\theta \mid \alpha\right) = \frac{\Gamma\left(\varepsilon_{i=\mathit{1}}^{k} \alpha_i\right)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} ... \theta_k^{\alpha_k - 1}$ (1)

where α is a k-vector with components $\alpha_i > 0$, and where $\Gamma\left(x\right)$ is the Gamma function.

Given the parameters α and β, the joint probability distribution of the topic mixture θ, a set of N topics z, and a set of N words $\mathbf{w}$ is given by:

$p\left(\theta, z, \mathbf{w} \mid \alpha, \beta\right) = p\left(\theta, \alpha\right) \prod_{n=1}^{N} p\left(z_n \mid \theta\right) p\left(w_n \mid z_n, \beta\right),$ (2)

where $p\left(z_n \mid \theta\right)$ is simply θ_i for the unique i such that $z_n^i = 1$. Integrating this over θ and summing over z, we obtain the marginal distribution of a document:

$p\left(\mathbf{w} \mid \alpha, \beta\right) = \int p\left(\theta, \alpha\right) \left(\prod_{n=1}^{N} \sum_{z_n} p\left(z_n \mid \theta\right) p\left(w_n \mid z_n, \beta\right)\right) d\theta.$ (3)

Ultimately, we take the product of the marginal probabilities of single documents to obtain the probability of the corpus:

$p\left(D \mid \alpha, \beta\right) = \prod_{d=\mathit{1}}^{M} \int p\left(\theta_d \mid \alpha\right) \left(\prod_{n=\mathit{1}}^{N_d} \sum_{z_{dn}} p\left(z_{dn} \mid \theta_d\right) p\left(w_{dn} \mid z_{dn}, \beta\right)\right) d\theta_d.$ (4)

From Fig 3.4 we can see that the LDA model is a three level representation. The parameters α and β constitute to the outermost level of the model, parameter θ_d forms the middle level and parameters z_{dn} and w_{dn} are at the innermost level of the model. The parameters α and β are corpus level parameters, in the sense that they are assumed to be sampled once in the process of generating a corpus. The variables θ_d are document-level variables sampled once per document and the variables z_{dn} and w_{dn} are at the word level. These variables will be sampled once for each word in each document.

For estimating the parameters of the LDA model, α, β in particular, many techniques have been proposed. Blei et al. [2003b] used empirical Bayes approach for parameter estimation, Steyvers and Griffiths [2004] used Gibbs sampling technique. *MALLET: A Machine Learning for Language Toolkit* [McCallam, 2002] which has an implementation of LDA is used in this work for modeling interests and uses Gibbs sampling technique for parameter estimation.

Chapter 4

Problem Formulation

This chapter describes in detail, the task of predicting friendship links in the *LiveJournal* social network and the approach used in this work. It also puts forward some of the questions that we would like to answer in this work.

As mentioned in Chapter 1, we exploit the ability of machine learning algorithms to take advantage of the content and graph structure of the social network site, *LiveJournal* to predict friendship links between its users. Users of *LiveJournal* social network can specify their demographics and interests along with tagging other users of the social network as friends. It was shown that the graph formed as a result of a user tagging other users with a relation, "Friend" in *LiveJournal* was very helpful in predicting and recommending potential friends to the users of this social network [Hsu et al., 2007]. We believe that the interests provided by users of *LiveJournal* are also useful in recommending potential friends to those users. Thus, in this work we study the usefulness of these interests specified by the users of *LiveJournal* for friendship recommendation to its users using topic modeling technique. We would also like to see if we can improve the accuracy with which friendship links are predicted using user graph structure of *LiveJournal* if we use interests along with them. Figure 4.1 depicts the approach that we use in this work for the task of predicting friendship links between users of *LiveJournal* social network. This chapter is organized as follows: In Section 4.1, we describe the approach of using user interests for predicting friendship links. In Section 4.2, we describe the approach of using user graph for predicting friendship links. In Section 4.3, we describe the approach of using user interests along with user graph structure for predicting friendship links. In Section 4.4, we list the research questions that we address in this work.

4.1 Friendship Link Prediction Using User Interests

In this section, we describe the approach of predicting friendship links using the interests specified by users of *LiveJournal*. These interests belong to a wide variety of domains, including *Movies, Books, Sports, Social* and *Current Issues*, among others. Past work by Bahirwani [2008] and Haridas [2009] constructed an ontology for these interests. They believed that in social networks, ontologies provide a crisp semantic organization of the knowledge available in such networks. In our approach, we use a Topic Modeling technique to capture semantic information from user's interests. In our work, we use LDA, a generative probabilistic topic model, to find inherent topics from the user interests. To model user interests in *LiveJournal* social network using LDA, we treat *LiveJournal* as a document corpus with each user in the social network representing a "document". Interests specified by each user form the content of the user document. We then run MALLET: A Machine Learning for Language Toolkit [McCallam, 2002] which has a java-based implementation for LDA. We vary the number of topics to be modeled for this user collection, thus implicitly simulating an ontology. The topic probabilities obtained as a result of modeling user interests are then used to construct the features for the friendship prediction task. These features are henceforth referred to as Interest Based Features. Construction of interest based features for a user pair is described in Section 5.1.

4.2 Friendship Link Prediction Using User Graph

Previous work by Hsu et al. [2006], Bahirwani [2008], Haridas [2009], Caragea et al. [2009] have shown that the graph structure existing between the users of *LiveJournal* social network acts as a good source of information for predicting friendship links. In this work, we follow the method described in [Hsu et al., 2006] for predicting friendship links using the graph structure of *LiveJournal* dataset. Construction of features from graph structure is described in Section 5.2 and these features are henceforth referred to as Graph Based Features.

4.3 Friendship Link Prediction Using User Interests and User Graph

In this method, we combine the features that were constructed by applying topic modeling on user interests i.e interest based features with graph based features constructed from the user graph network and use them for predicting friendship links in the *LiveJournal* social network. Our hypothesis is that the features constructed from topic probabilities obtained as a result of applying LDA on user interests will improve the prediction accuracy when combined with graph based features as compared to using just graph based features.

4.4 Research Questions

The questions that we address in this work are the following.

- Are the interest based features generated by applying LDA on user interests better in terms of prediction accuracy compared to graph based features? If not, how useful are these features for link prediction problem?

  According to Bahirwani [2008] and Haridas [2009], graph based features constructed using user graph of *LiveJournal* social network outperformed interest based features constructed using interest ontology which is engineered from user interests. However, they reported that interest based features generated using interest ontology when combined with graph based features were much effective in predicting friendship links as compared to using just graph based features. In this work, we would like to see if interest based features generated using LDA on user interests will be more effective for the link prediction task compared to the graph based features. We also use these interest based features in combination with graph based features to see if we can improve the accuracy of predicting friendship link as compared to using either LDA interest based features or graph based features, separately.

- What is the effect of varying the number of topics while modeling user interests using LDA?

  The number of latent topics to be found from the user interest documents by applying LDA is varied from 20 to 200. The intuition is that the lower the number of topics to be discovered, the more abstract the topics are. As the number of topics to be modeled increases, the inherent topics discovered from the interests becomes more specific. However, if the number of topics to be modeled is very large, it is equivalent to treating each interest as a separate topic. We believe that the more abstract the topics, the less informative they are. Similarly, the more specific the topics, the less useful they would be in providing information for prediction. Thus, our intuition is that the ROC curve would increase gradually, reach a peak and then decrease as we increase the number of topics to be modeled and we want to test this intuition by varying the number of topics from 20 to 200.

- How does the approach perform with an increase in the number of users?

  We would like to see if our approach of constructing interest based features from topic probabilities to predict friendship links would be consistent/improve in terms of the AUC values, as the number of users in the dataset used to construct these features is increased. We believe that our approach will improve the prediction accuracy as more number of users get added to the dataset.This might be because of an improved estimation of topic probabilities by LDA as a result of an increase in the number of interests specified by the users. We would also like to see if the computation of features using our approach will have any affect as we add more users to the dataset.

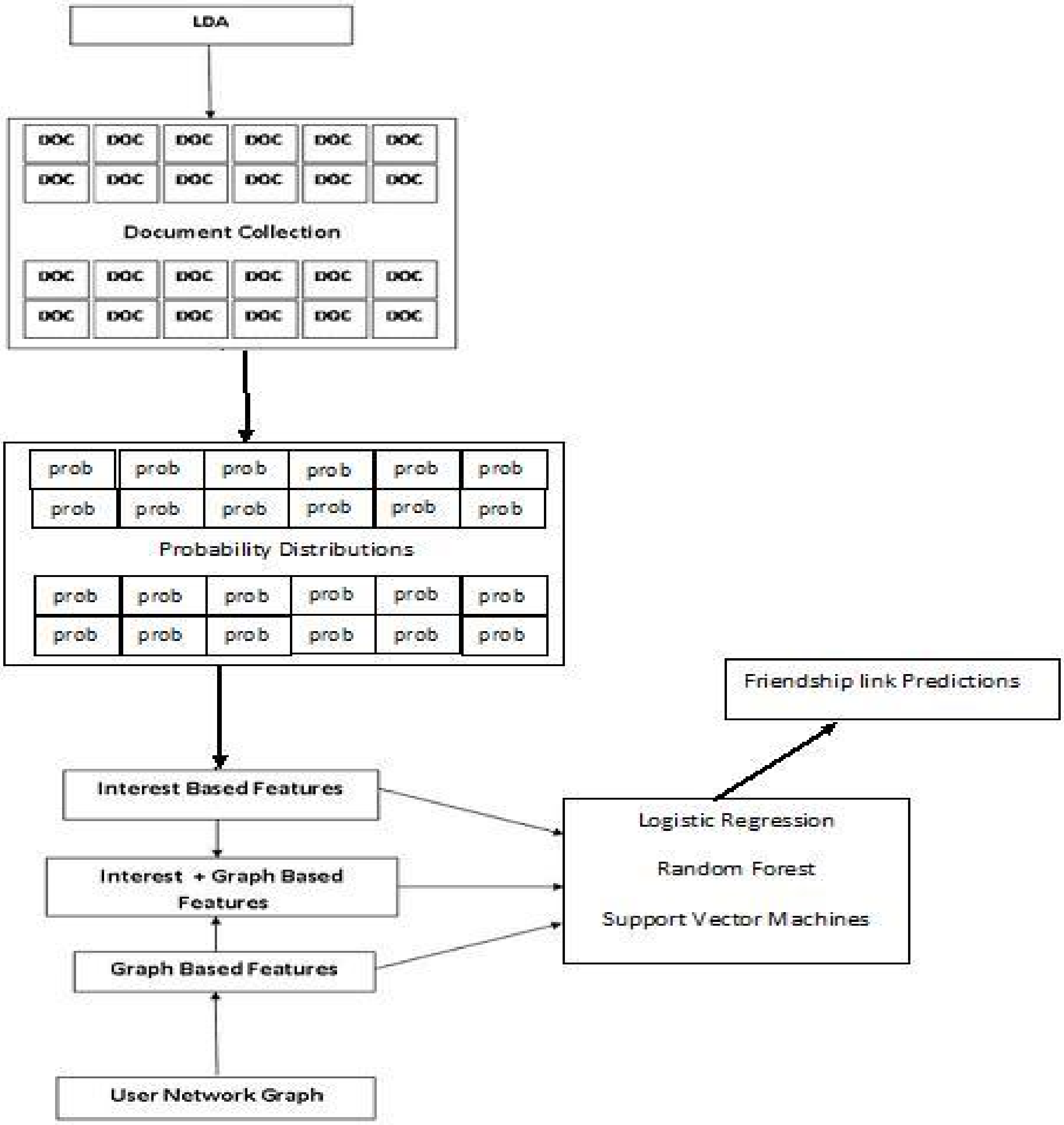


Figure 4.1: *Link Prediction approach based on LDA and the network graph*

Chapter 5

Experimental Setup

In this chapter, we describe the dataset used in this work and the experiments designed to evaluate our approach of using LDA for the link prediction task. We have conducted various experiments with several classifiers to investigate their performance in predicting friendship links between users of the *LiveJournal* social network. This chapter is organized as follows: In Section 5.1, we describe the construction of Interest Based Features. In Section 5.2, we describe the construction of Graph Based Features. In Section 5.3, we describe various machine learning algorithms that are used for the friendship prediction task. The dataset used in this work is described in Section 5.4. The dataset that we used in this work is highly imbalanced. Thus, there is a need to balance the data before we build predictive models. We describe the data balancing techniques in Section 5.5. Finally, we list the experiments designed in this work in Section 5.6.

5.1 Interest Based Features

The intuition behind interest based features is that, two users ‘A’ and ‘B’ might be friends if ‘A’ and ‘B’ have some interests in common irrespective of what those interests are. We try to capture this information from the feature set that we construct using the user interests. We hypothesize that the feature set constructed will improve the accuracy of the friendship prediction task. The topic probabilities obtained during modeling these user interests using LDA provide an explicit representation of each user. For predicting friendship link, we treat each user pair as an instance and construct the feature set for this instance using their respective topic probabilities.

Suppose $A\,[1 \cdots n]$ represents the topic distribution for user 'A' and $B\,[1 \cdots n]$ represents the topic distribution for user 'B' at a particular number of topics, n. The feature set, $F' < A, B >$ for the user pair $< A, B >$ is constructed as follows

$F' < A, B >= F' < |A\,[1] - B\,[1]\,|, |A\,[2] - B\,[2]\,| \cdots |A\,[n] - B\,[n]\,| >$

5.2 Graph Based Features

We follow the approach in [Hsu et al., 2006] for constructing Graph Based features. For each user pair $< A, B >$ in the network graph, we consider the following graph based features derived from graph and use them for prediction task:

- *In-degree* of “A” (InDeg(A)): The number of edges that originate at a node corresponding to some user in the network and terminate at the node corresponding to user “A” are denoted as the *in-degree* of that user.

- *In-degree* of “B” (InDeg(B)): Similar to *in-degree* of user “A”, this is the number of edges terminating at the node corresponding to user ”B” in the social network graph.

- *Out-degree* of “A” (OutDeg(A)): The number of friends of user “A” except user “B” are denoted as the *out-degree* of user “A”. These are computed by counting the number edges originating at the node corresponding to user “A” (except the edge $A \rightarrow B$) in the social network graph.

- *Out-degree* of “B” (OutDeg(B)): Similar to *out-degree* of user “A”, this represents the number of edges originating at the node corresponding to user “B” (except the edge $B \rightarrow A$)

- *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 $C \rightarrow B$

- *Backward deleted distance* from “B” to “A” (BDD(A,B)): The minimum distance from the node corresponding to user “B” to the node corresponding to user “A” in the network graph ignoring the edge $B \rightarrow A$ if any, is denoted as the *backward deleted distance* between the two users.

5.3 Machine Learning Algorithms Used for Friendship Prediction Task

- Support Vector Machines (SVM) with *build logistic model* option enabled: Support Vector Machines are a set of supervised learning algorithms used for classification and regression. Given a set of training examples, each example belonging to a positive or a negative class, it builds a model that predicts whether a new example belongs to the positive or the negative class. More formally, SVM views each input example as an n-dimensional vector. SVM then constructs a hyperplane or a set of hyperplanes in this high dimension space so that the positive examples and the negative examples are divided by a clear gap that is as wide as possible. New examples are predicted to one or the other class based on which side of the gap they fall in [Mitchell, 1997].

- Random Forests: Random Forest classifier is an ensemble of many decision tree classifiers. The output for the class label for an example from the random forest classifier, is the class label, that is predicted by majority of the decision trees in the random forest collection [Mitchell, 1997].

- Logistic regression (Logistic): Logistic regression or logistic models predicts the class label of an example by calculating the probability of it belonging to a positive class or a negative class by fitting the data from that example to a logistic curve. This logistic curve is derived from the data used in the training phase of the classifier [Mitchell, 1997].

5.4 Dataset Description and Pre-Processing

The dataset that we used for this work has approximately 37,000 users from *LiveJournal* online service. We used 3 subsets of the *LiveJournal* dataset with 1,000 users, 5,000 users and 10,000 users, respectively. The interests expressed by the users belong to a wide variety of domains, including *Movies, Books, Sports, Social* and *Current Issues*, among others. As part of the preprocessing step, we cleaned the interest set to remove symbols, numbers, foreign language. Interests, whose frequency is less than 5 are also removed. We then concatenated strings of words in an interest into a single word so that mallet treats each interest as a single word. For example, we have two interests 'Artificial Intelligence' and 'Artificial neural networks' expressed by the user, after pre-processing step, these interests will be converted into 'AtrificialIntelligence' and 'AtrificialNeuralNetworks'. We also removed those users, whose in-degree and out-degree is zero as well as users who do not have any interests declared, from all the three datasets we used in this work. Thus, we are left with 801, 4,026 and 8,107 users in the three datasets, respectively, and approximately 14,000, 32,000 and 39,700 interests after preprocessing. There are around 4,400 declared friendship links (out of 801 by 801 possible links in an undirected graph with 801 nodes) in the 1,000 user dataset, around 40,000 declared friendship links (out of 4,026 by 4,026 possible links in an undirected graph with 4,026 nodes) in the 5,000 user dataset and approximately 49,700 number of declared friendship links (out of 8,107 by 8,107 possible links in an undirected graph with 8,107 nodes) in the 10,000 user dataset. We make an assumption that the graph is complete i.e. all declared friendship links form the positive instances and all non declared friendships are negative examples [Caragea et al., 2009]. However, this assumption does not hold in real world.

In-order to generate training and test instances, we partition the graph network formed as a result of user declared friendships, into 2 parts with 2:1 ratio. We use the partition with $2/3^{rd}$ of the users to generate training instances and the partition with $1/3^{rd}$ of the users for test instances. The assumption of having a complete user graph network made the data highly skewed. For example, the ratio between positive instances to negative instances is 1:144 in the 1,000 user dataset that we used. Initial experiments with the skewed training set resulted in poor predictions. Hence, we made sure that the training data is balanced before we built predictive models. This is achieved with the use of SpreadSubSample filter described in Section 5.5.

5.5 Undersampling Techniques

The most important factor influencing classification accuracy is the training data. However, the data in real-world applications often is imbalanced, i.e. most of the examples belong to one class often called as 'majority class' and few examples belong to other class called as 'minority class'. In this case, as the training data is skewed towards the majority class, the classifier tends to predict the class label as majority class for many examples (which actually belong to minority class) in the test set. One way to avoid this is to balance the training data (while the distribution of the test data remains unchanged). Undersampling is a technique in which we sample a subset of the majority class examples which are selected randomly. In this work, we use SpreadSubsample filter whose implementation is provided by the WEKA data mining software [Witten et al., 1999]. SpreadSubsample filter produces a random subsample (without replacement) of the input dataset. This filter also allows us to specify the maximum "spread" between the rarest and most common class. For example, we can specify that we want a 2:1 ratio in class frequencies. As part of the experiments, we used 2:1 spread and 1:1 spread between the rarest class and most common class.

5.6 Experiments

The following experiments have been performed in this work.

1. In the first experiment, we test the performance of the predictive models trained on interest based features constructed with topic distributions as explained in 5.1 for the 1,000 user dataset. This experiment is henceforth referred to as Experiment 1.

2. In the second experiment, we test the models that are trained on graph features described in 5.2 for the 1,000 user dataset. To be able to construct the graph features for test data, we assume that a certain percentage of links are known. To be precise, we explore scenarios where 10%, 25% and 50% links are known in the test set, respectively. We then construct features only for the unknown links using known links. This experiment is henceforth referred to as Experiment 2.

3. In the third experiment, graph based features are used in combination with interest-based features to see if they can improve the accuracy with which the trained models predict for the 1,000 user dataset. As in experiment 2, graph features constructed with 10%, 25% and 50% known links in the test set are combined with interest features. This experiment is henceforth referred to as Experiment 3.

In each experiment, we build predictive models using training data that is balanced using SpreadSubsample filter with 1:1 spread and 2:1 spread between the rarest and most common class.Thus, Experiment 1 with 1:1 spread is referred to as Experiment 1(a) and with 2:1 spread is referred to as 1(b). Similarly, Experiments 2 and 3 with 1:1 spread will be referred to as 2(a) and 3(a) and Experiments 2 and 3 with 2:1 spread will be referred to as 2(b) and 3(b). Next, we repeat the above mentioned experiments for the 5,000 user dataset and for the 10,000 user dataset. The experiments on the 5,000 user dataset with 1:1 spread will be referred to as Experiment 4(a), Experiment 5(a) and Experiment 6(a), respectively, and with 2:1 spread will be referred to as Experiment 4(b), Experiment 5(b) and Experiment

6(b), respectively. Similarly, the experiments on the 10,000 user dataset with 1:1 spread will be referred to as Experiment 7(a), Experiment 8(a) and Experiment 9(a), respectively, and with 2:1 spread will be referred to as Experiment 7(b), Experiment 8(b) and Experiment 9(b), respectively, henceforth. We consider the classifiers addressed in section 5.3, whose implementations are provided by the WEKA data mining software [Witten et al., 1999] for all the experiments.

Chapter 6

Results

In this chapter, we discuss the results from the predictive models built for the experiments listed in Section 5.6 of Chapter 5. These experiments were designed to investigate the contribution of interest features generated by using LDA over user interests for the link prediction task in the *LiveJournal* social network when used by themselves, as well as in combination with graph features. As hypothesized, interest based features created using LDA were able to improve the accuracy of the prediction task when used in combination with graph based features compared to graph based features alone. This chapter is divided as follows: we discuss the results corresponding to link prediction problem in Section 6.1. In Section 6.2, we discuss the scalability of our approach.

6.1 Predicting Friendship Links

In this section, we explain results of various classifiers used to predict friendship links between *LiveJournal* users. We also discuss the effectiveness and usefulness of the interest based features constructed for predicting friendship links, as well as the impact on the accuracy when the number of topics to be modeled on the user interests is varied. The results reported in these experiments are the AUC values from the classifiers used to predict friendship links between *LiveJournal* users. As described in Section 5.4, we partition the user graph into 2 parts. $2/3^{rd}$ of the users are used to generate the training set and $1/3^{rd}$ of the users used to generate the test set. All AUC values reported are averaged over five different Train and Test sets created as a result of 5 different partitions of the user graph. Section 6.1.1 reports results for experiments 1, 2 and 3. In Section 6.1.2, results for experiments 4, 5 and 6 are described. Finally, in Section 6.1.3, we report results for experiments 7, 8 and 9. Please refer to Section 5.6 for description of each experiment.

6.1.1 Results for 1,000 user dataset

Table 6.1 tabulates the AUC values obtained for experiments 1(a), 2(a) and 3(a) using classifiers Logistic Regression, Random Forest and Support Vector Machines. Table 6.2 tabulates the AUC obtained for experiments 1(b), 2(b) and 3(b) using classifiers Logistic Regression, Random Forest and Support Vector Machines. The AUC values obtained using interest based features is compared with the AUC values obtained using graph features and interest+graph features at a certain percentage of known links. Highest AUC value obtained at each percentage of known links, for each of the three classifiers is highlighted in **BOLD-FACED**. Number of topics at which highest AUC value is observed for interest features and interest+graph features is listed in the braces '()'.

Table 6.1: *AUC values for Logistic Regression (LR), Random Forests (RF) and Support Vector Machines (SVM) classifiers with interest, graph and interest+graph based features using 1:1 spread for the 1,000 user dataset. We assume that k% links are known in the test set, where k is 50, 25 and 10, respectively. The known links are used to construct graph features and interest+graph features.*

Exp#	Features	Logistic Regression	Random Forest	SVM
1(a)	Interest	0.6258(160)	0.5808(90)	0.6158(90)
2(a)	Graph 10%	**0.7258**	0.5698	**0.7418**
3(a)	Interest+Graph 10%	0.625(140)	**0.6292**(90)	0.648(80)
2(a)	Graph 25%	**0.7624**	0.7036	**0.7924**
3(a)	Interest+Graph 25%	0.723(20)	**0.8024**(60)	0.7684(80)
2(a)	Graph 50%	0.8538	0.7798	0.8624
3(a)	Interest+Graph 50%	**0.8594**(20)	**0.872**(30)	**0.8808**(60)

We can see from tables 6.1 and 6.2 that graph features with 10% and 25% known links outperformed interest as well as interest+graph features in terms of AUC values for both Logistic Regression and SVM. However, in the case of the Random Forest classifier, the

Table 6.2: *AUC values for Logistic Regression (LR), Random Forests (RF) and Support Vector Machines (SVM) classifiers with interest, graph and interest+graph based features using 2:1 spread for the 1,000 user dataset. We assume that k% links are known in the test set, where k is 50, 25 and 10, respectively. The known links are used to construct graph features and interest+graph features.*

Exp#	Features	Logistic Regression	Random Forest	SVM
1(b)	Interest	0.625(160)	0.5782(90)	0.6198(160)
2(b)	Graph 10%	**0.74**	0.578	**0.7738**
3(b)	Interest+Graph 10%	0.6226(140)	**0.6664**(20)	0.6606(20)
2(b)	Graph 25%	**0.7684**	0.7106	**0.8104**
3(b)	Interest+Graph 25%	0.7406(20)	**0.8188**(40)	0.7983(20)
2(b)	Graph 50%	0.8526	0.8008	0.8692
3(b)	Interest+Graph 50%	**0.8648**(20)	**0.877**(60)	**0.8918**(20)

AUC values are the highest when interest features are used along with graph features and outperformed both interest features and graph features for all percentages of known links. Figures 6.1 through 6.6 are the AUC plots for the 1,000 user *LiveJournal* dataset for all three classifiers with both 1:1 and 2:1 spread. AUC values obtained using interest features, graph features and interest+graph features are plotted across different number of topics modeled on the user interests. In the case of the Logistic Regression classifier (fig. 6.1 and 6.2), we can observe that graph features alone outperformed interest features alone across all topic levels. Also, graph features were better compared to interest+graph features when 10%, 25% are known. Even though highest AUC value when 50% of links are known is obtained using interest+graph features, it is not consistent across all topic levels. However, this is not the same in the case of the Random Forest classifier (fig. 6.3 and 6.4). Interest+graph features with 10%, 25% and 50% known links are better across all topic levels compared to graph features with 10%, 25% and 50% known links respectively. For SVM classifier (fig. 6.5 and 6.6), like for the Logistic Regression classifier, graph features with 10% and 25% known links are better compared to interest+graph features with 10% and 25% known links, respectively, across all numbers of topics. When 50% of links are known, AUC values with interest+graph features are better compared to AUC values from corresponding graph features until number of topics is 110 but decreased as number of topics reached 200.

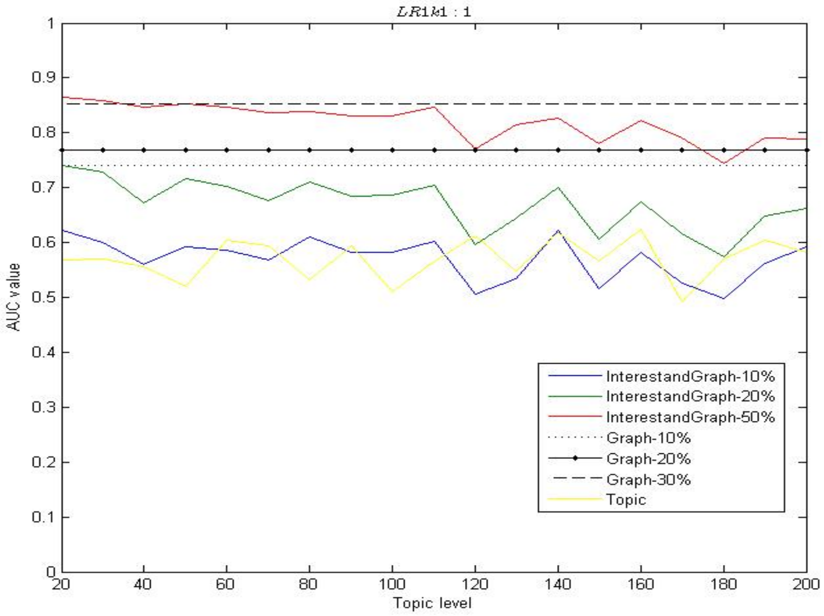


Figure 6.1: *Graph of reported AUC values v/s no. of topics used for modeling, for Logistic Regression classifier with 1:1 spread for the 1,000 user dataset*

6.1.2 Results for 5,000 user dataset

The AUC values obtained for experiments 4(a), 5(a) and 6(a) using classifiers Logistic Regression, Random Forest and Support Vector Machines are tabulated in 6.3. Table 6.4 shows the AUC values for experiments 4(b), 5(b), 6(b) with all 3 classifiers listed above. Similar to Section 6.1.1, we compare the AUC values obtained using interest based features with the AUC values obtained using graph features and interest+graph features at a certain percentage of known links. Highest AUC value obtained at each percentage of known links, for each of the three classifiers is highlighted in **BOLD-FACED**. Number of topics at which highest AUC value is observed for interest features and interest+graph features is listed in the braces '()'.

The results for 5,000 user dataset were different compared to the results from the 1,000 user dataset. In the case of the Logistic Regression classifier for the 5,000 user dataset, we

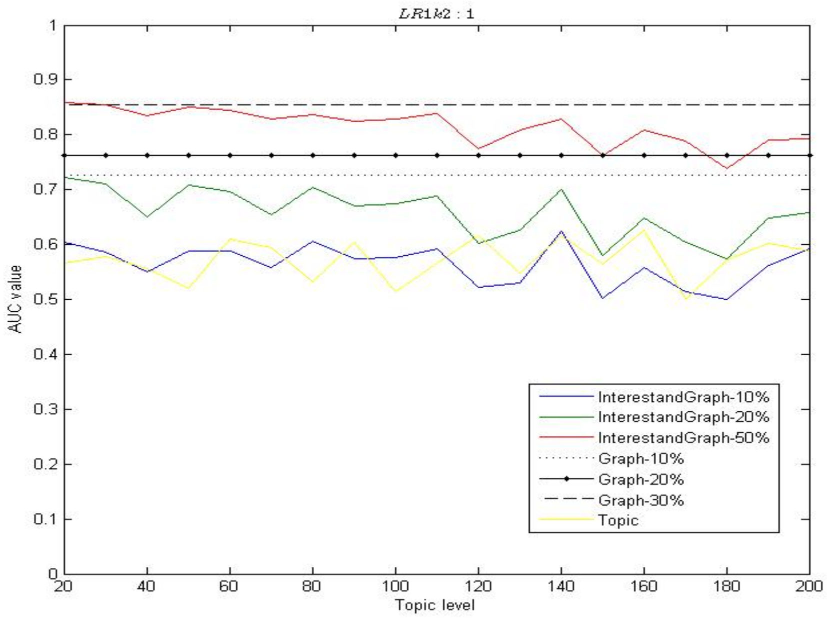


Figure 6.2: *Graph of reported AUC values v/s no. of topics used for modeling, for Logistic Regression classifier with 2:1 spread for the 1,000 user dataset*

can observe that the AUC value obtained using interest features was better compared to that of graph features or interest+graph features when 10% of links are known for 1:1 spread, as well as for 2:1 spread. In the case of 25% and 50% known links, interest+graph features were found to be effective. However, for Random Forest and SVM classifiers, interest+graph features proved to be very helpful at the task of predicting friendship links as we can see that the AUC values obtained using these features are the highest for all three percentages of known links in both 1:1 and 2:1 spread. Figures 6.7 through 6.12 are the AUC plots for the 5,000 user *LiveJournal* dataset for all three classifiers with both 1:1 and 2:1 spread. The AUC values obtained using interest features, graph features and interest+graph features are plotted as a function of number of topics. In the case of the Logistic Regression classifier (fig. 6.7 and 6.8), the AUC values with interest features are better across all number of topics compared to graph features and interest+graph features with 10% known links. However, interest+graph features with 25% and 50% known links outperformed graph features alone

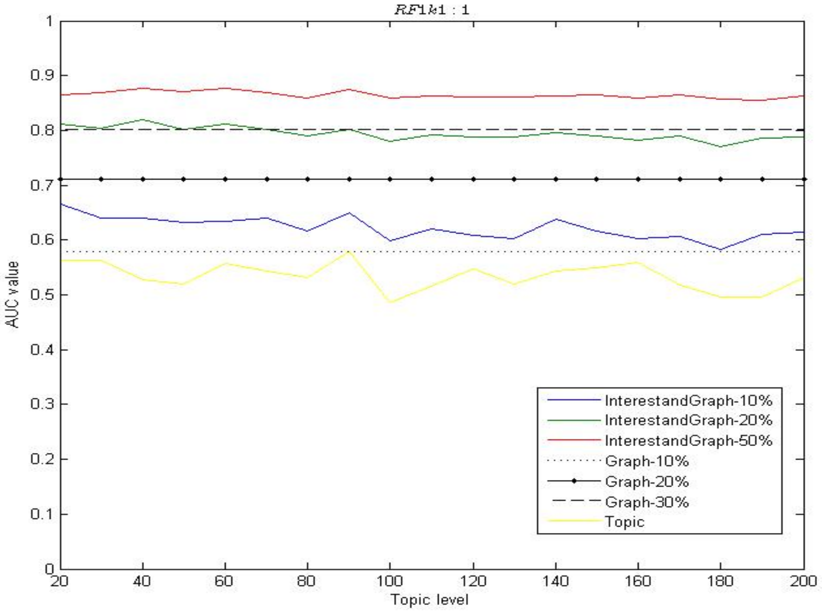


Figure 6.3: *Graph of reported AUC values v/s no. of topics used for modeling, for Random Forest classifier with 1:1 spread for the 1,000 user dataset*

with 25% and 50% known links and interest features alone across all number of topics proving the usefulness of interest features calculated using LDA for link prediction problem. For the Random Forest classifier (fig. 6.9 and 6.10), interest+graph features across for all percentages of known links are better compared to the corresponding graph features. In the case of the SVM classifier (fig. 6.11 and 6.12), interest features were slightly better than interest+graph with 10% of links known for 2:1 spread and interest+graph with 10% of links known is slightly better than interest features for 1:1 spread. Not much difference can be found between graph features with 25% known links and interest+graph features with 25% known links across all number of topics. However, when 50% of the links are known, interest+graph features outperformed corresponding graph features for all number of topics.

6.1.3 Results for 10,000 user dataset

Due to time, space and memory constraints, we were not able to execute experiments 8 and 9. However, we were able to run experiment 7 in which we use just interest based features

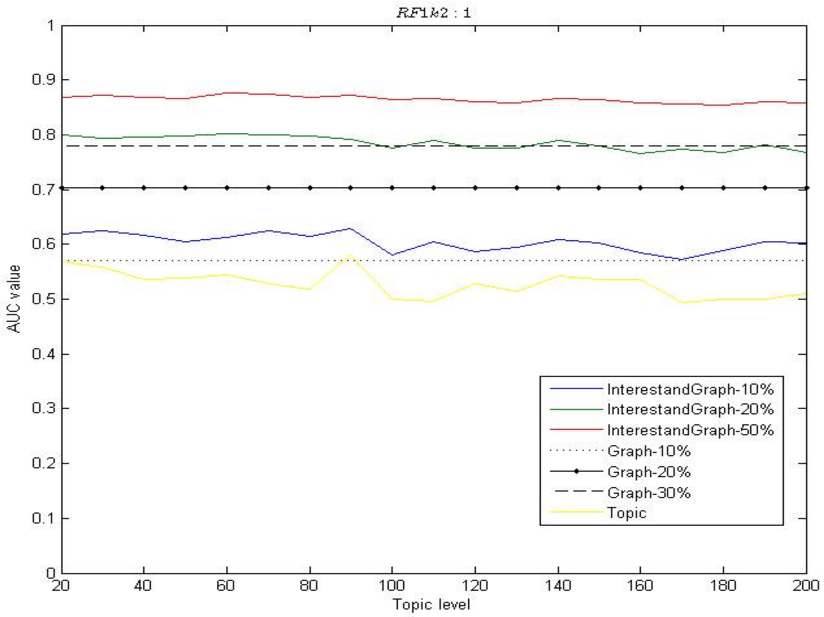


Figure 6.4: *Graph of reported AUC values v/s no. of topics used for modeling, for Random Forest classifier with 2:1 spread for the 1,000 user dataset*

and build predictive models. Similar to sections 6.1.1, 6.1.2, we use Logistic Regression, Random Forest and SVM for experiment 7 with 1:1 and 2:1 spread. The AUC values with interest features for 10,000 dataset were better across all number of topics compared to those of 5,000 dataset as well as 1,000 dataset for all three classifiers. This can be observed in plots 6.13, 6.14, 6.15, 6.16, 6.17 and 6.18.

6.2 Scalability

In this section, we explain the scalability of our approach in terms of results obtained, with increase in the number of users in the dataset that we use and support this explanation with evidence. As mentioned in Section 1.1, we incrementally added users to the dataset that we used initially which has 1,000 users from *LiveJournal* social network. As a first step to test the scalability of our approach, we added 4,000 users to the initial dataset that was used making it a 5,000 user dataset. Data cleaning and pre-processing is similar to that of

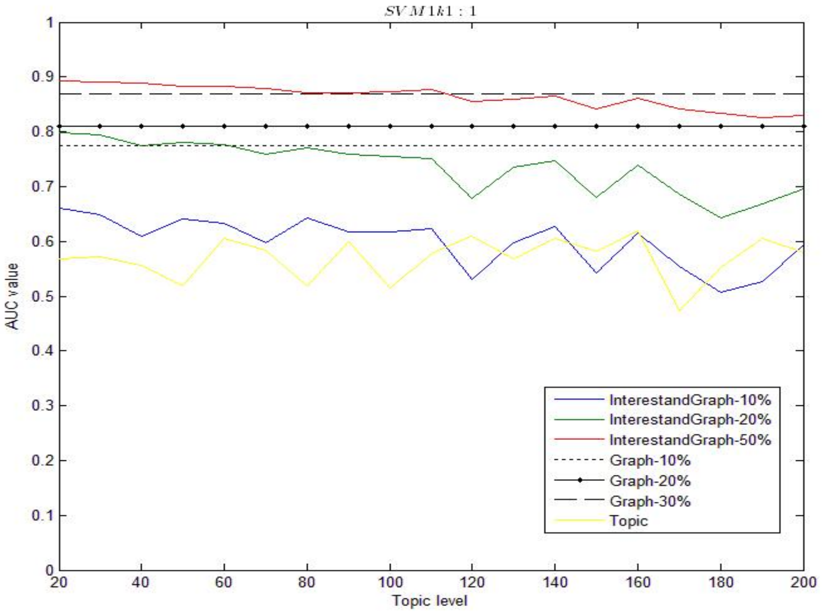


Figure 6.5: *Graph of reported AUC values v/s no. of topics used for modeling, for SVM classifier with 1:1 spread for the 1,000 user dataset*

1,000 user dataset. We then constructed features from the 5,000 user dataset and executed experiments 4,5 and 6 (refer to Section 5.6 for the description of the experiments). Similarly, we used 10,000 uses from *LiveJournal* social network and constructed features from them to execute experiments 7,8 and 9. We plotted the AUC values obtained for each classifier with interest features across all numbers of topics for 1,000 user, 5,000 user and 10,000 user datasets with 1:1 and 2:1 spreads to see if there is an improvement in the results obtained with an increase in the number of users. As hypothesized, for all classifiers used, AUC values with interest based features, across all numbers of topics for 5,000 user dataset are better than those of 1,000 dataset. Similarly, AUC values with interest based features for 10,000 user dataset are better than those of 5,000 dataset. This can be observed from plots 6.13, 6.14, 6.15, 6.16, 6.17 and 6.18.

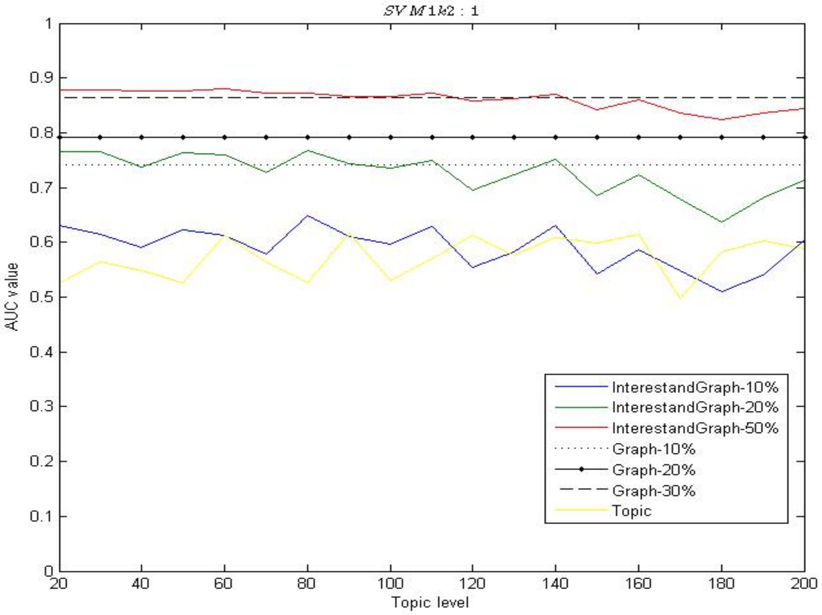


Figure 6.6: *Graph of reported AUC values v/s no. of topics used for modeling, for SVM classifier with 2:1 spread for the 1,000 user dataset*

Table 6.3: *AUC values for Logistic Regression (LR), Random Forests (RF) and Support Vector Machines (SVM) classifiers with interest, graph and interest+graph based features using 1:1 spread for the 5,000 user dataset. We assume that k% links are known in the test set, where k is 50, 25 and 10, respectively. The known links are used to construct graph features and interest+graph features.*

Exp#	Features	Logistic Regression	Random Forest	SVM
4(a)	Interest	**0.6954**(170)	0.6276(20)	0.6938(140)
5(a)	Graph 10%	0.649	0.5936	0.692
6(a)	Interest+Graph 10%	0.6718(150)	**0.6566**(20)	**0.6998**(80)
5(a)	Graph 25%	7022	0.6716	0.7896
6(a)	Interest+Graph 25%	**7384**(70)	**0.7846**(50)	**7986**(20)
5(a)	Graph 50%	0.8456	0.7086	0.883
6(a)	Interest+Graph 50%	**0.8696**(50)	**0.8908**(190)	**0.9046**(20)

Table 6.4: *AUC values for Logistic Regression (LR), Random Forests (RF) and Support Vector Machines (SVM) classifiers with interest, graph and interest+graph based features using 2:1 spread for the 5,000 user dataset. We assume that k% links are known in the test set, where k is 50, 25 and 10, respectively. The known links are used to construct graph features and interest+graph features.*

Exp#	Features	Logistic Regression	Random Forest	SVM
4(b)	Interest	**0.695**(170)	0.6268(20)	**0.6978**(170)
5(b)	Graph 10%	0.6462	0.5764	0.6644
6(b)	Interest+Graph 10%	0.6706(150)	**0.6276**(20)	0.6934(180)
5(b)	Graph 25%	0.696	0.6512	0.7744
6(b)	Interest+Graph 25%	**0.7318**(150)	**0.7778**(90)	**0.7906**(110)
5(b)	Graph 50%	0.8432	0.6996	0.8778
6(b)	Interest+Graph 50%	**0.8646**(50)	**0.8874**(130)	**0.9006**(110)

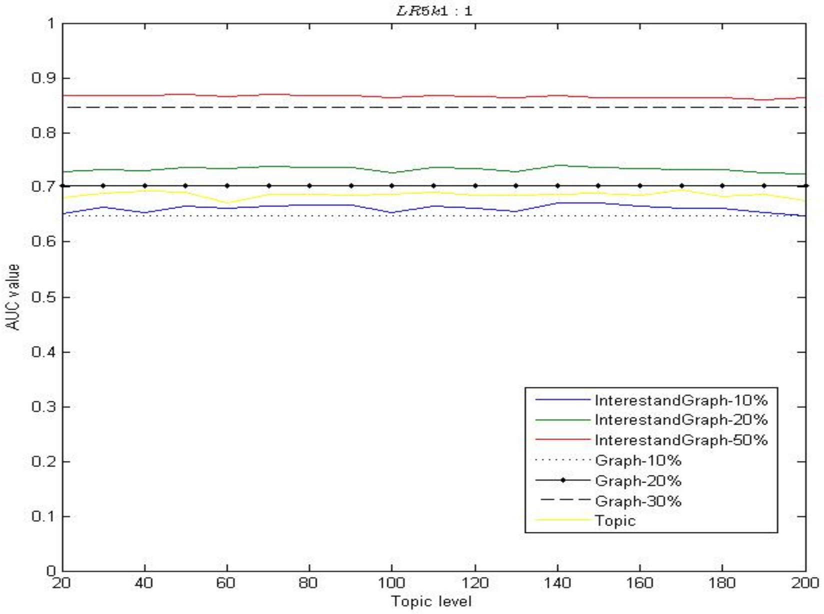


Figure 6.7: *Graph of reported AUC values v/s no. of topics used for modeling, for Logistic Regression classifier with 1:1 spread for the 5,000 user dataset*

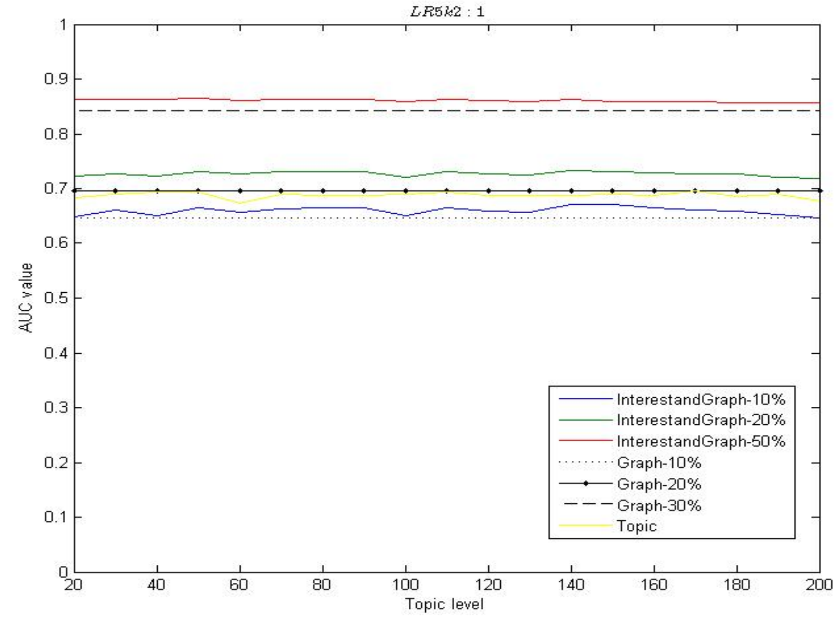


Figure 6.8: *Graph of reported AUC values v/s no. of topics used for modeling, for Logistic Regression classifier with 2:1 spread for the 5,000 user dataset*

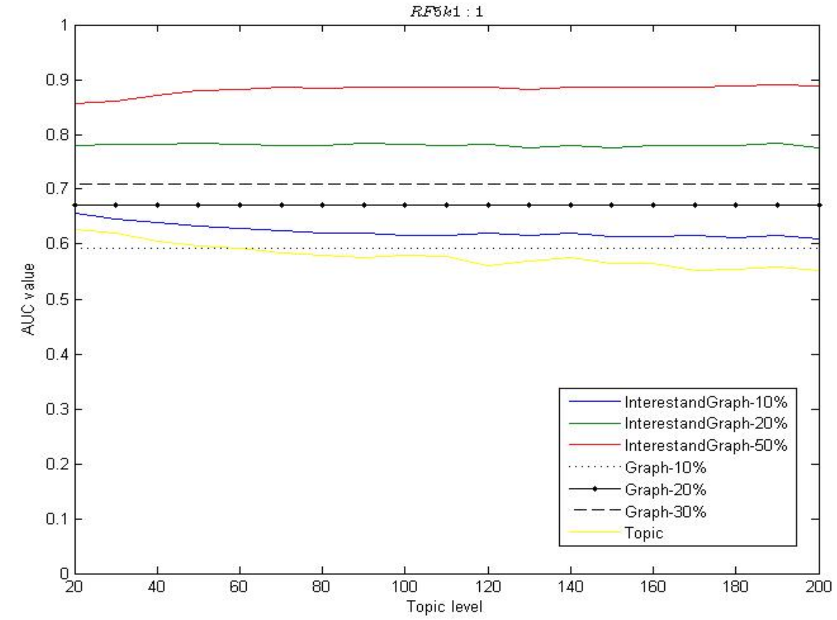


Figure 6.9: *Graph of reported AUC values v/s no. of topics used for modeling, for Random Forest classifier with 1:1 spread for the 5,000 user dataset*

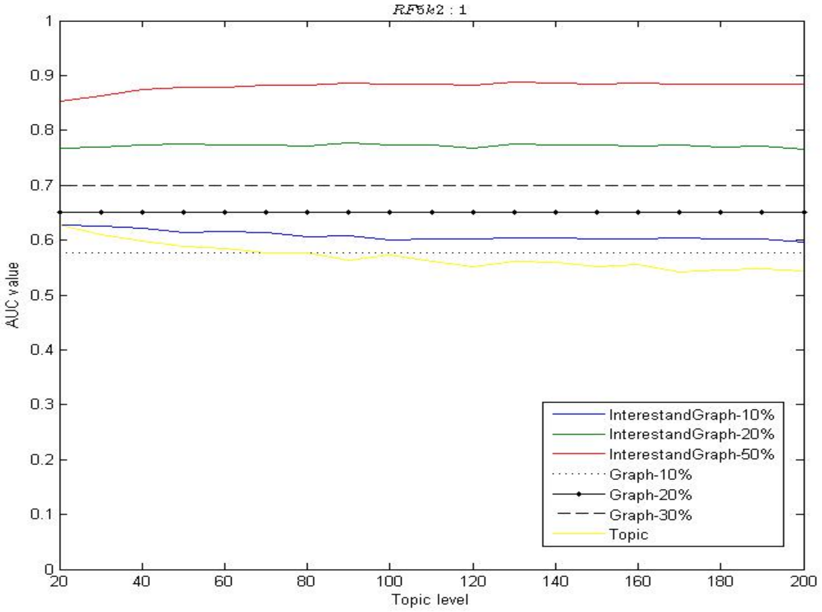


Figure 6.10: *Graph of reported AUC values v/s no. of topics used for modeling, for Random Forest classifier with 2:1 spread for the 5,000 user dataset*

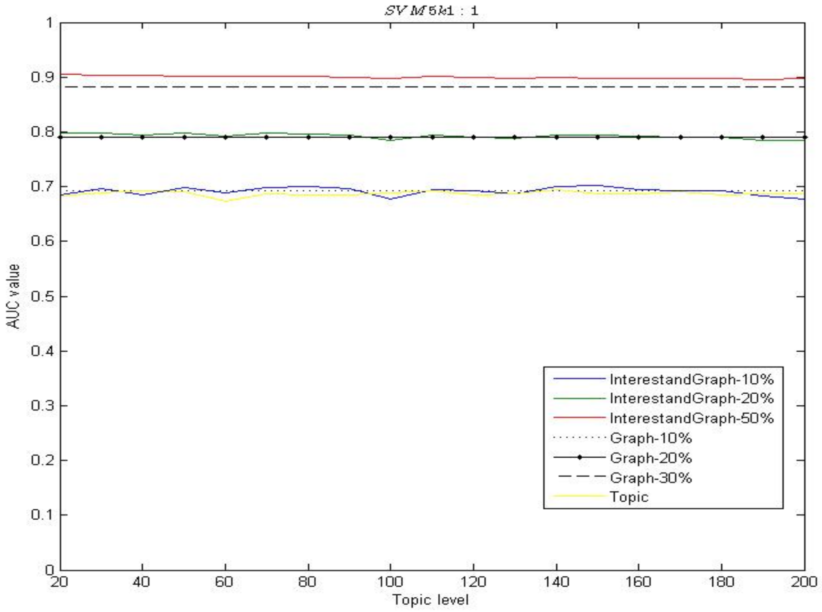


Figure 6.11: *Graph of reported AUC values v/s no. of topicsused for modeling, for SVM classifier with 1:1 spread for the 5,000 user dataset*

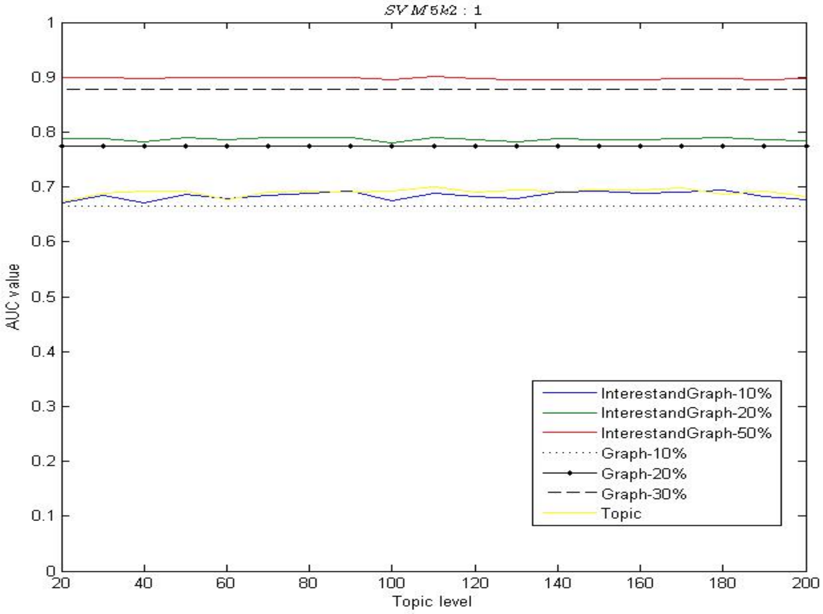


Figure 6.12: *Graph of reported AUC values v/s no. of topics used for modeling, for SVM classifier with 2:1 spread for the 5,000 user dataset*

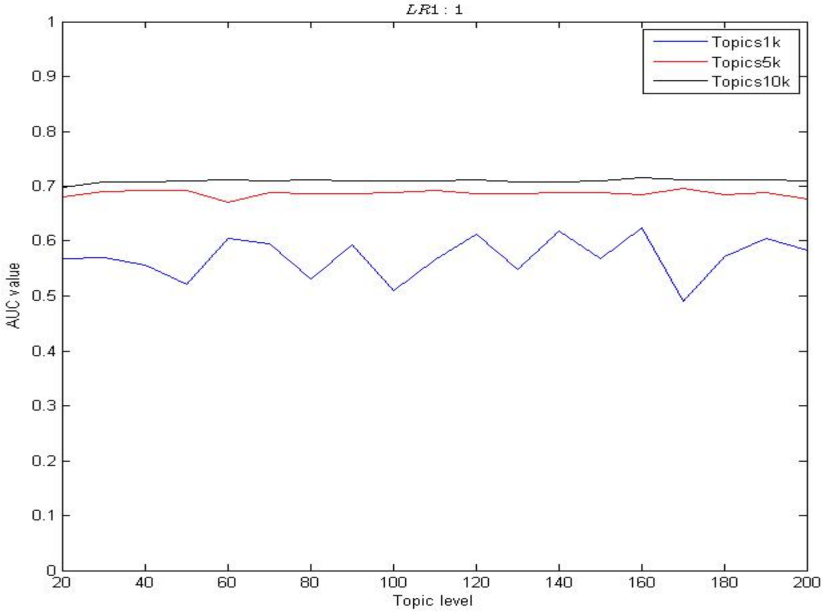


Figure 6.13: *Graph of reported AUC values for interest features v/s no. of topics used for modeling, for Logistic Regression classifier with 1:1 spread for 1,000, 5,000 and 10,000 user datasets*

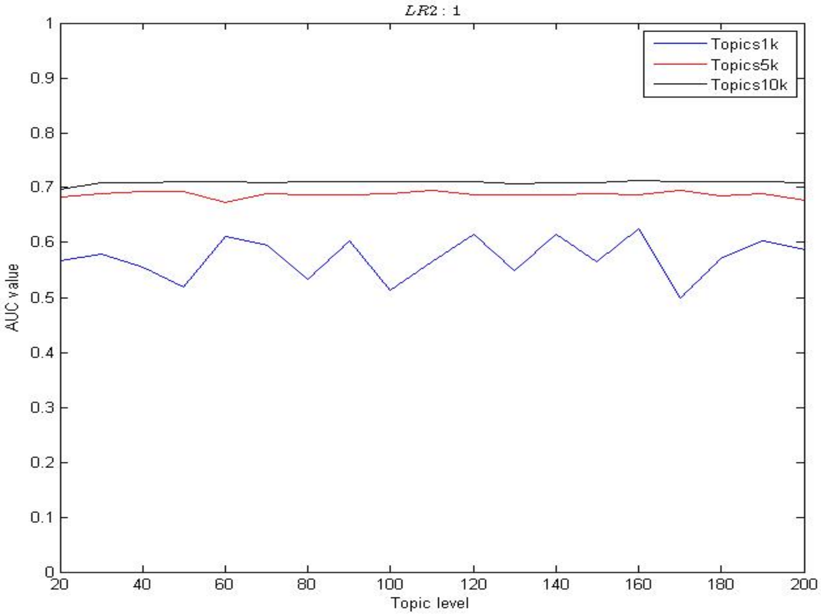


Figure 6.14: *Graph of reported AUC values for interest features v/s no. of topics used for modeling, for Logistic Regression classifier with 2:1 spread for 1,000, 5,000 and 10,000 user datasets*

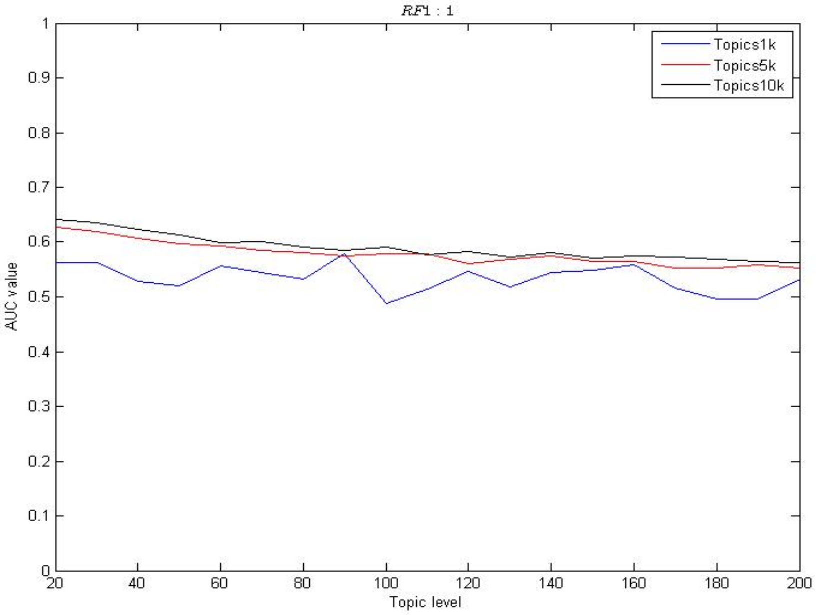


Figure 6.15: *Graph of reported AUC values for interest features v/s no. of topics used for modeling, for Random Forest classifier with 1:1 spread for 1,000, 5,000 and 10,000 user datasets*

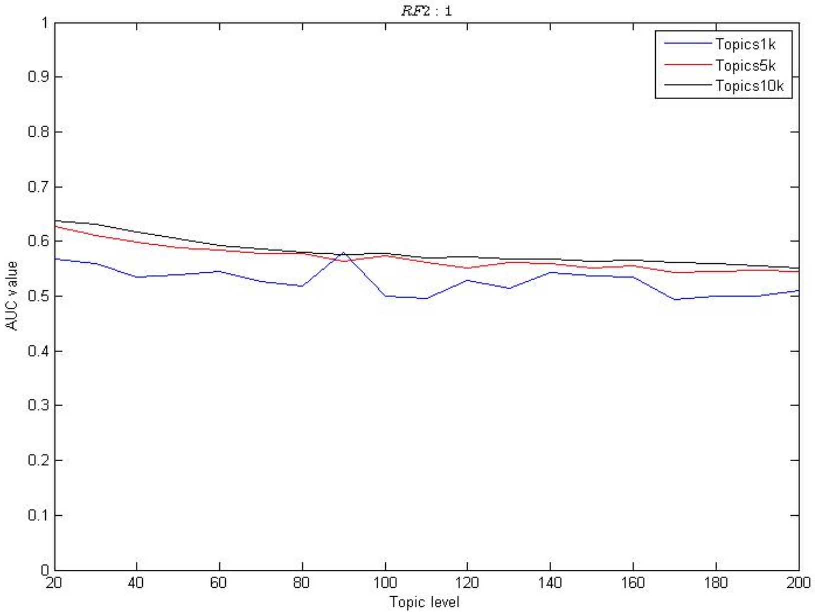


Figure 6.16: *Graph of reported AUC values for interest features v/s no. of topics used for modeling, for Random Forest classifier with 2:1 spread for 1,000, 5,000 and 10,000 user datasets*

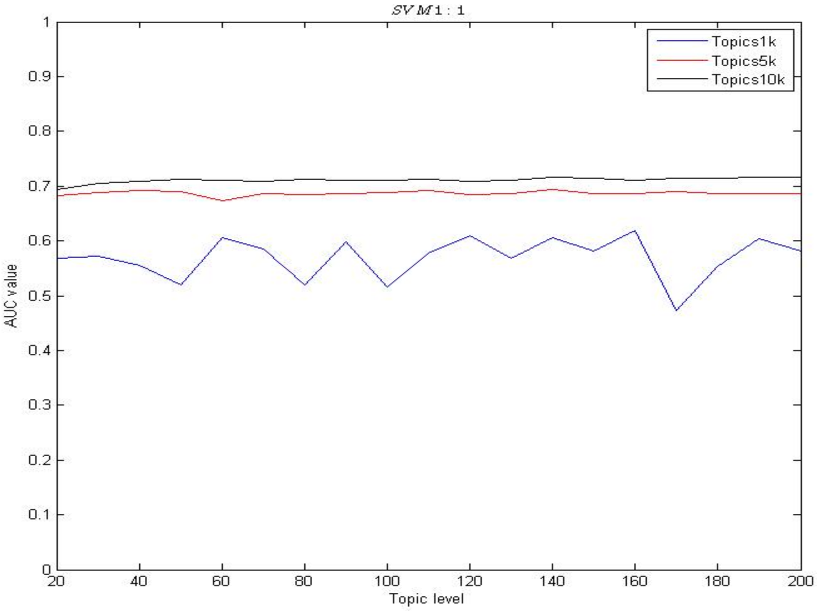


Figure 6.17: *Graph of reported AUC values for interest features v/s no. of topics used for modeling, for SVM classifier with 1:1 spread for 1,000, 5,000 and 10,000 user datasets*

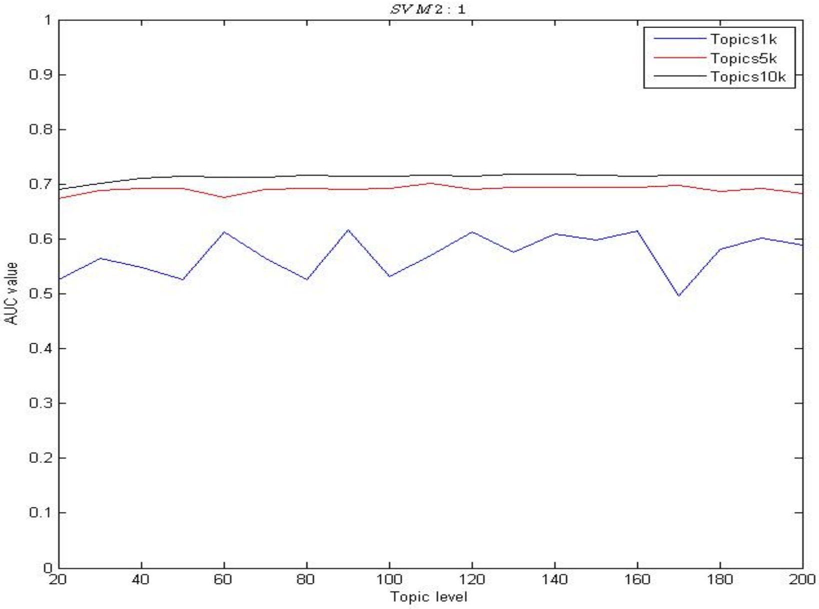


Figure 6.18: *Graph of reported AUC values for interest features v/s no. of topics used for modeling, for SVM classifier with 2:1 spread for 1,000, 5,000 and 10,000 user datasets*

Chapter 7

Conclusion and Future Work

In this chapter, we will discuss the questions raised in Section 4.4 using the results reported in Chapter 6. We also report some limitations of our approach of using LDA for Friendship Link prediction in Section 7.1. Some improvements and new directions for this work are proposed in Section 7.2.

7.1 Conclusion

We use results from tables 6.1, 6.2, 6.3, 6.4 to discuss questions raised earlier.

- Are the interest based features generated by applying LDA on user interests good in comparison with graph based features? If not, how useful are these features for link prediction problem?

  Even though interest based features alone were not very helpful compared to graph based features alone for the 1,000 user dataset, they were better compared to graph features and interest+graph features when 10% of known links in case of Logistic Regression Classifier for the 5,000 user dataset. Also, interest features when used in combination with graph features yielded better results for Logistic Regression and SVM classifiers, in case of the 1,000 user dataset when 50% of links are known and for all percentages of known links when Random Forest classifier is used. For the 5,000 user datatset, either interest features alone or interest features used along with graph features were better compared to graph features alone at the task of predicting friendship links. Thus, interest featured generated using LDA on user interests are useful when used along with graph features for predicting friendship links even though interest features alone are not as informative as graph features alone in case of the 1,000 user dataset.

- What is the effect of varying the number of topics while modeling User Interests using LDA?

  From figures in Chapter 6, we can observe that there is not much difference between the AUC values when the number of topics to be modeled are varied from 20 to 200. This does not match with our hypothesis of having a curve that would gradually increase, reach a peak and then decrease as we increase the topics. One reason for this might be that the variation that we used to model topics in our approach is not significant, given the large number of interests. We also, might not have satisfied the assumption of having abstract topics when the number of topics are close to 20 and specific topics when the number of topics are close to 200. We might have to try this approach with higher number of topics and by varying the topics by say 50 or 100 rather than just 10.

- How does the approach perform with an increase in the number of users?

  From figures 6.13, 6.14, 6.15, 6.16, 6.17 and 6.18 we can observe that our approach of using LDA over user interests of *LiveJournal* social network yielded better results as more number of users are added to the dataset that we used. We tested our approach with 1,000 users, 5,000 users and 10,000 users in the dataset. We believe that as we increase the number of users further, our approach might still improve the accuracy of link prediction problem between *LiveJournal* users. The reason is that, as we increase number of users in the dataset, more interests get added to the interest set improving the topic probabilities of each user in the collection obtained by the modeling technique. However, with an increase in the number of topics to be modeled on the user interests, size of the test files increased significantly leading to memory issues. Memory used to compute interest features for 10,000 dataset was 40GB. Thus, we can say that the approach used in this work to compute features from topic probabilities does not scale easily in terms of more users being added to the dataset.

Even though our approach of using topic modeling on user interests for link prediction proved to be effective, it has some limitations as well. One of the biggest problems that we faced was memory and storage issues. As we increased the number of topics to be modeled, the size of the test files grew larger and larger because of our assumption of having a completely connected graph. Thus, there is a need for a system with huge memory space and processing unit if we want to add more number of users to the dataset. Secondly, the amount of time it took to compute the feature 'Backward Deleted Distance' for graph features increased significantly with an increase in the number of users in the graph. As a result, we were unable to compute graph features and interest+graph features for the 10,000 user dataset. There is a need to reduce the amount of time it takes to compute this backward deleted distance feature by using efficient algorithms or a suitable replacement to this feature has to be found out.

Also, a limitation of the LDA technique is that it fails to capture the correlation between the underlying topics which in principle is most common in text documents e.g in case of scientific articles, a document about genetics may be more likely to contain information about health and diseases rather than astronomy or games. The basis for this limitation in LDA is because of the independence assumptions implied in the Dirichlet distributions on topic proportions. Also, for a given number of topics to be modeled, for instance there might be two inherent topics *Data Mining* and *Machine Learning* in the collection that was modeled. In principle, *Machine Learning* is a sub-category of *Data Mining* and the current implementation of LDA does not capture this information as well.

Another important limitation of this approach is that it takes into account, the static image of *LiveJournal* social network. In principle, this assumption of having a static network graph does not hold in real world scenario. Based on user interactions in the social network, the graph might change rapidly due to the addition of more users as well as friendship links. Also, users may change their demographics and interests regularly. Our approach does not take into account these changes in the network graph of the user profiles.

7.2 Future Work

As part of the future work, we would like to test our approach on complete *LiveJournal* dataset which has around 37,000 users. We would like to improve the efficiency of the algorithm that we used to calculate graph based features, as well as maintain a ratio between the number of positive instances and negative instances in the test set rather than adhering to the complete graph assumption. Secondly, we would like to increase the number of topics to model on user interests further to see if we can get a convex shaped curve like AUC plot for link prediction problem rather than having a near straight line as is the case in this work. Also, instead of using topic probabilities for constructing features as done in this work, we would like to use the most probable words for a topic (mallet outputs this information) and construct features using them.

Furthermore, modeling techniques have become more powerful in the past few years with introduction of techniques like Hierarchical Topic Models [Blei et al., 2003a] which learn topic hierarchies from data and Correlated Topic Models [Blei and Lafferty, 2007] in which topic proportions exhibit correlations. We might benefit even more if we can design an approach to capture these topic correlations and hierarchies for predicting friendship links and this is left as a future work.

Bibliography

V. Bahirwani. Ontology engineering and feature construction for predicting friendship links and users' interests in the live journall social network. Master's thesis, Kansas State University, 2008.

V. Bahirwani, D. Caragea, W. Aljandal, and H. W. Hsu. Ontology engineering and feature construction for predicting friendship links in the live journal social network. In *Proceedings of The 2nd SNA-KDD Workshop 08 (SNA-KDD 08), Las Vegas, Nevada, USA*, 2008.

D. Blei and D. J. Lafferty. A correlated topic model of science. *The Annals of Applied Statistics*, 1:17–35, 2007.

D. Blei, L. T. Griffiths, Jordan I. M., and Tenenbaum B. J. Hierarchical topic models and the nested chinese restaurant process, 2003a.

D. Blei, Y. A. Ng, and I. M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003b.

D. Blei, J. Boyd-Graber, and X. Zhu. A topic model for word sense disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1024–1033, 2007.

M. D. Boyd and B. N. Ellison. Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13, 2007. URL http://jcmc.indiana.edu/vol13/issue1/boyd.ellison.html.

D. Caragea, V. Bahirwani, W. Aljandal, and H. W. Hsu. Ontology-based link prediction in the *livejournal* social network. In *Proceedings of Association of the Advancement of Atrificial Intellignece*, pages 192–196, 2009.

C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri. Know your neighbors: Web spam detection using the web topology. In *Proceedings of SIGIR'07, Amsterdam, Netherlands*, 2007.

W. Chen, J. Chu, J. Luan, H. Bai, Y. Wang, and Y. E. Chang. Collaborative filtering for orkut communities: Discovery of user latent behavior. In *Proceedings of International World Wide Web Conference*, 2009.

S. Deerwester, T. S. DEmais, W. G. Furnas, Landauer K. T, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

B. Fitzpatrick. Livejournal: Online service, 1999.

L. Geetor. Link mining: A new data mining challenge. In *SIDKDD Explorations*, volume 5(1), pages 85–89, 2003.

L. Geetor and Q. Lu. Link-based classification. In *Proceedings of the Twelth International Conference on Machine Learning (ICML-2003), Washington DC*, 2003.

J. Guo, G. Xu, X. Cheng, and H. Li. Named entity recognition in query. In *Proceedings of SIGIR'09, Boston, USA*, 2009.

M. Haridas. Exploring knowledge bases for engineering a user interests hierarchy for social network applications. Master's thesis, Kansas State University, 2009.

H. W. Hsu, T. Weninger, and R. S. M. Paradesi. Predicting links and link change in friends networks: supervised time series learning with imbalanced data. In *Proceedings of Artificial Neural Networks in Engineering, (ANNIE)*, 2006.

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

R. Krestel, P. Fankhauser, and W. Nejdl. Latent dirichlet allocation for tag recommendation. In *Proceedings of RecSys'09, New York, USA*, 2009.

Y. Liu and A. Niculescu-Mizil. Topic-link lda: Joint models of topic and author community. In *Proceedings of the 26 th International Conference on Machine Learning, Montreal, Canada*, 2009.

D. C. Manning, P. Ragahavan, and H. Schtze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

K. A. McCallam. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

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

J. C. Na and T. T. Thet. Effectiveness of web search results for genre and sentiment classification. *Journal of Information Science*, 35, Issue 6:709–726, 2009.

L. Page and S. Brin. The anatomy of a large-scale hypertextual web search engine. In *Computer Networks and ISDN Systems*, pages 107 – 117, 1998.

M. Steyvers and T. Griffiths. Probabilistic topic models, 2007.

M. Steyvers and T. Griffiths. Finding scientific topics. In *Proceedings of National Academy of Sciences, U.S.A*, pages 5228–5335, 2004.

M. Steyvers, T. Griffiths, and J. B. Tenenbaum. Topics in semantic representation. *American Psychological Association*, 114(2):211–244, 2007.

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

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