Link Discovery Through Iterative Link Classification — Towards a Real-Time Analysis of Graph Evolution

A Dissertation

Presented to the

Graduate Faculty of the

University of Louisiana at Lafayette

In Partial Fulfillment of the

Requirements for the Degree

Doctor of Philosophy

Murali Krishna Pusala

Summer 2018

Link Discovery Through Iterative Link Classification — Towards a Real-Time Analysis of Graph Evolution

Murali Krishna Pusala

APPROVED:

Vijay V. Raghavan, Chair
Professor of Computer Science

Chee-Hung Henry Chu
Professor of Computer Science

Mohsen Amini Salehi
Assistant Professor of Computer Science

Raju N. Gottumukkala
Assistant Professor of Mechanical Engineering

Ryan G. Benton
Assistant Professor of Computer Science
University of South Alabama

Mary Farmer-Kaiser
Dean of the Graduate School

To
my parents, Vengaiah Pusala and Pushpa Latha,
my wife, Harika,
my sisters, Lakshmi and Leela,
and
my brother-in-laws, Narayana and Venkat,
who encouraged me to pursue my dreams and finish my dissertation

Acknowledgments

I would like to express my sincere gratitude to my advisor, Dr. Raghavan for his excellent guidance and support throughout this dissertation. Thanks to my dissertation committee, Dr. Ryan Benton, Dr. Raju Gottumukkala, Dr. Amini Mohsen, and Dr. Henry Chu, whose invaluable feedback and constructive comments improved and enhanced my research. I would also like to thank my friends Satya Katragadda, Siva Venna, Prabhakar Vemavarapu, Naveen Tammareddy, and Kiran Sanagapaty for their support and encouragement.

Table of Contents

List of Figures

List of Tables

1 . Introduction

In recent years, the advent of the internet and developments in information technology have led to tremendous growth in data generation from sources such as sensors, social media, the World Wide Web, and the Internet of Things (IoT). Managing, mining, and extracting useful insights from this data has become increasingly popular among researchers in various fields. Most real-world data are modeled as graphs because of the inherent strength of the graph model to represent complex relationships. Since many real-world applications such as social networks, biomedical networks, the World Wide Web, citation graphs, and patent networks are modeled as graphs, it is necessary to investigate techniques for mining and analyzing graphs.

Graph mining entails analyzing and extracting knowledge from graphs using statistical and machine learning techniques. One of the most important problems in graph mining is link prediction, which refers to predicting the likelihood of new relationships forming by analyzing the current snapshot of the graph. Link prediction has various practical applications in social networks, biomedical research, recommendation systems, and information systems.

1.1 Background to Link Prediction

In the field of complex network analysis, link prediction plays an important role in evaluating the likelihood of future node associations, typically by using topological and statistical information. Link prediction has real-world applications in many domains such as biomedical research, social networks, recommendation systems, and communication networks. For example, users and communities in social networks are modeled as nodes, and the interactions, collaborations, and preferences of users are represented as edges [1]. Link prediction is employed in social networks to recommend friends and communities based on predictions of future relationships between entities. In biomedical research, protein-protein interactions are represented as a protein networks, where each protein is a node and interactions between proteins are edges [2][3][4]. As the graphs grow in size and complexity, mining them to predict future links becomes extremely challenging. Therefore, link prediction has become an important activity in graph mining research.

Probabilistic models [5][6] and similarity-based models [7][8][9] have been successfully applied for link prediction on static graphs. However, many real-world networks are both large and highly dynamic, due to high volumes and high velocity of relational data. Given that the topology of these dynamic graphs changes with the creation and removal of links, new methods must be investigated. Recent work on scalable, distributed approaches to graph mining offers a promising direction for link prediction on large and dynamic graphs [7][10][11]. More details about link prediction and related work are presented in Chapter 2.

1.2 Motivation

The concept of link prediction was first proposed by Liben-Nowell et al. to predict collaborations between authors in co-author networks using network topology features [1]. Although several advances have been made in link prediction, there are still many problems with the use of existing link prediction frameworks for real-time applications. Therefore, the primary motivation of this dissertation is to develop link prediction solutions for real-time applications.

Aggarwal and Subbian categorize a network based on the rate at which edges are added or updated [12]. These categories include *slowly evolving networks* and *fast-evolving* *networks* or *steaming networks*. In both of these categories, a link prediction method is required to update the model frequently as new data arrive. Most early link prediction methods, including the supervised method adapted for this research, use single snapshot data for training and testing generated models. Evaluation of these prediction models is performed using model validation techniques such as cross-validation. However, cross-validation has proved adequate for evaluating proposed methodologies but not for understanding a model's performance in real-world applications.

Developing computationally efficient link prediction methods for evolutionary graphs requires knowledge on *how the performance of a link prediction model deteriorates over time? how frequently should the model be updated?* and *what strategies are suitable for achieving an optimal performance?* Given the computational complexity involved in continuous feature extraction, and retraining the model to detect changing topological properties, it is essential to understand the optimal learning rate, and strategies to reduce graph size.

Ideally, the prediction model should be updated whenever new data arrives. We can approach this in two ways using traditional machine learning techniques. We can either train the model using recent data whenever new data arrive, or we can train the model using historical data along with recent data. By using only recent data, we lose historical information, but training the model using all of the data is computationally very expensive. Bearing in mind these shortcomings, we propose an incremental learning algorithm to update the model using recent information, while also preserving historical information.

The link prediction model predicts a large number of possible links in a network at any given time, and not all the predicted links are equally interesting. When this is the case, it is vital to quantify the degree of interestingness of the predicted. Since in real-world applications only interesting predictions should be conveyed to decision-makers, predictions need to be ordered and ranked based on their interestingness.

The problem of link prediction has been studied extensively in several domains where data is modeled as a network and a link prediction method is used to predict future links. In other words, link prediction is used for forecasting the future state of a network based on the present state. We investigate whether exploiting the predicted network enables us to iteratively predict more links in a future network than would be possible using the original network. This question forms the foundation of our iterative link classification approach to predicting future links. The intuition behind iterative link classification is that the formation of new relationships in a network increases the likelihood of more new relationships forming. This approach shows a significant improvement in overall accuracy compared to traditional link prediction approaches.

1.3 Contributions

Our contributions are as follows:

- Development of iterative link classification, a novel approach that uses predicted knowledge of a network to predict more relevant links, thus improving the overall accuracy of predicting new links by 6% and increasing the number of relevant links discovered by 10%
- Proposal of guidelines for maintaining the link prediction model in real time, based on our observations and our comprehensive study of supervised link prediction method

- Proposal of an *interestingness-based link ranking* algorithm, which would be the first domain-independent method to order and rank predicted links by interestingness, utilizing the concept of association interestingness from data mining
- Demonstration of the feasibility of using incremental machine learning in link prediction for dynamically evolving data and using both historical and current data

1.4 Dissertation Organization

This dissertation is organized as a compilation of chapters, each of which discusses proposed solutions to a specific challenge in order to develop link prediction for real-time applications.

Chapter 2: Analysis of Supervised Link Prediction for Temporal Graphs: Since most real-world applications change over time, it is crucial to evaluate predictive models on constantly changing data. In this chapter, we define the temporal aspects of time-varying networks and study their effect on the performance of link prediction models. To handle large volumes of network data in real-time applications, we propose two different semantic-type-based pruning of network data. We study the effect of pruning on the prediction of new links. We also presented an evidence-based evaluation of link prediction model as automatic hypothesis generation by rediscovering the know relationships.

Chapter 3: Supervised Interestingness-based Link Ranking: In this chapter, we propose a domain-independent, supervised method that predicts the rank of future links based on objective interestingness measures. An analysis of thirteen common interestingness measures indicates that predicting future interestingness values is difficult. However, we can predict the relative ordering of links with low error.

Chapter 4: Link Prediction using an Incremental Learning Approach: This chapter presents an incremental learning process to address the link prediction problem. A supervised learning strategy based on an incremental support vector machine (SVM) is adapted to generate an incremental model for the link prediction problem. The incremental SVM updates the model using recently gathered data.

Chapter 5: Link Discovery Using Iterative Link Classification: In this chapter, we propose a novel link prediction approach, which uses iterative classification to predict the formation of new links in a network. In this approach, the links predicted in an iteration are fed back into the network data to predict the links in the subsequent iteration. We observed considerable improvement in the overall performance of the link prediction model.

Chapter 6: Conclusions and Future Work: We discuss the findings of our research and make suggestions for future work.

2 . Analysis of Supervised Link Prediction for Temporal Graphs

2.1 Introduction

The link prediction problem can be defined as the challenge of predicting the likelihood of a future association between two nodes that are not connected in the current state of the network [13]. Several real-world applications generate large volumes of highly dynamic graph data, which add new nodes and links that can change the original topological properties of graphs. Efficient algorithms and data management techniques are needed at every stage from pre-processing to feature extraction through to training and prediction. These algorithms and techniques must be able to accommodate the memory and computational requirements of evolutionary graphs generated from real-time data sources. Link prediction on static graphs has proved reasonably accurate [1][7][14]. However, link prediction on dynamic graphs remains in its infancy. The development of new methods for use on dynamic graphs requires deeper investigation into the dynamic aspects including temporal aspects, graph pruning, training the model, and prediction measures.

In this chapter, we study some important aspects of how a dynamic link prediction model behaves when applied to temporal and massive graphs. To apply link prediction to dynamic graphs, two major aspects need to be considered. First, unlike those used for static network, prediction models for dynamic networks need to be updated as new data arrives. It is important to understand how the trained prediction model performs on new data. This will provide insights into how frequently the model needs to be updated. In addition, it is important to know the optimum size of the network snapshot in order for the model to achieve its best computational performance. Therefore, it is important to ask whether the network size affects the performance of the model. Removing nodes and edges that do not contribute to the model's performance also helps to improve the performance of the model. Earlier approaches [7][15] reduce the size of the network using various pruning techniques based on the co-occurrence of a link in the graph. While these techniques are less cumbersome and are easy to implement, they fail to consider the prominence of a link or node when pruning the graph. Nor do they account for the problem of determining the optimal threshold for pruning the graph in a dynamic setting. In many applications, domain experts are used to identify non-prominent node types. We investigate the effects of using semantic information from the nodes to prune the graph. This semantic-based approach considers the prominence of a node before pruning the graph to reduce its size for efficient processing. Identifying the semantic type requires the involvement of a domain expert. This can be done less frequently for a dynamic graph as the semantic information does not change.

We use the link prediction-based hypothesis discovery application for evaluating the behavior of the dynamic link prediction model behavior when applied to temporal and massive graphs. Hypothesis discovery, also known as literature-based discovery (LBD), extracts information from published literature to generate new hypotheses. LDB is modeled as a link prediction approach through the generation of a concept network using concepts extracted from published literature as nodes; co-occurrence of concepts suggests relationships between them. The prediction of a link between two concepts in a concept network is translated into a hypothesis of a link between the concepts. Several studies use the link prediction approach to discover hypotheses [2][7][16][17]. For this work, we adopt a supervised hypothesis generation approach proposed by Katukuri et al. [7].

In this work, our contributions can be summarized as follows:

- A thorough investigation of a supervised link prediction method for different temporal aspects and their effects on the overall performance and stability of the prediction algorithms
- A study of the impact of semantic-type-based pruning on the performance of the link prediction model
- A study of the effectiveness of our supervised link prediction algorithm by testing it on well-known and established scientific facts

2.2 Related Work

Literature-based discovery. In 1986, Don R. Swanson developed a methodology for discovering concept-concept relationships by mining published literature [15]. The assumption behind his methodology was that when two concepts that are not directly related have significant indirect relations through common concepts, there is a possibility that the two concepts might be related. Using this methodology, Swanson hypothesized a relation between fish oil and Raynaud's syndrome and later published a hypothesis regarding a possible relationship between migraines and magnesium [20]. Initially, most of LBD work was carried out by manually searching literature databases. Gordon and Lindsay [21] replicated and automated Swanson's LBD process using computer-based algorithms and information retrieval (IR) techniques. They used the dataset from the MEDLINE [13] database and applied IR techniques to identify the relationship between Raynaud’s syndrome and fish oil. In 1997, Swanson and Smalheiser [22] developed a similar system named Arrowsmith. Most early work attempted to replicate Swanson's methodology using different IR algorithms. Along with co-occurrence information, some of the earlier work included semantic information and lexical statistics of concepts to generate hypotheses [23][24][25]. Goodwin et al. proposed a method that uses the process of spreading activation as well as graph properties for LDB.

Literature-based discovery using link prediction. In recent studies[18][2][7][3], link prediction is used to generate possible hypotheses by modeling publications as concept graphs. A biomedical concept network is generated using concepts extracted from publications, with biomedical concepts as nodes and their co-occurrences as relationships between nodes (i.e., edges). A biomedical concept network, generally referred to as a concept network, evolves over time with the formation of new relationships between nodes. Link prediction analysis is applied to the concept network to predict future relationships between node pairs that are not directly connected.

Őzgür et al. [18] proposes a hypotheses generation method that applies a link prediction technique to predict relationships between genes and diseases. Initially, a seed list is generated from a list of genes that are related to a specific disease. Later, the seed list is used to construct a disease-specific gene interaction network by automatically mining literature using dependency parsing and support vector machines. Genes are ranked based on centrality measures calculated for each node in the network, and it is hypothesized that central gene(s) have a possible association with the specific disease. Jeong et al. [2] also proposed a similar method using degree centrality to predict lethal mutation in a yeast protein interaction network.

Katukuri et al. [7] propose a supervised link prediction method by modeling a biomedical literature repository as a comprehensive network of biomedical concepts. They use topological and semantic features to hypothesize possible relationships between biomedical concepts. They also propose an automatic labeling approach to generate labels for node pairs using two consecutive snapshots of the network. In the biomedical domain, Andrej et al. [26] propose an unsupervised learning method for predicting relationships in biomedical literature networks. The authors optimized the network by determining the non-useful edges using Pearson’s Chi-Square test and removing them from the network.

Most of these methods are evaluated using data extracted from a given static network. However, in practice, most real-world networks are temporal. To the best of our knowledge, few studies consider temporal evaluations of link prediction models. A study by Yang [27] discusses the effects of the temporal distance and temporal duration of a testing graph on link prediction performance. They divide the testing dataset into subsets of equal temporal duration. To evaluate the temporal distance, they consider a subset at a time as a testing dataset. To evaluate temporal duration, they agglomerate to create a test dataset of different temporal durations. This provides an excellent start for evaluating the temporal aspects of the link prediction model using social network data. We adopted a similar evaluation method to that used on the MEDLINE data to evaluate the performance of the prediction model in discovering unknown relationships. We also studied the temporal aspects of both the training dataset and the testing data. As discussed above, the link prediction dataset is generated using two snapshots of the graph. We investigated the effects of the temporal duration of these snapshots on the overall performance of the prediction model. As well as evaluating different temporal aspects of the link prediction model, we performed an evidence-based evaluation of link prediction, testing it on relationships that have been established in empirical studies.

2.3 Methodology and Experimental Setup

Terminology.

- G (V, E) denotes the network where V is a set of nodes and E is set of edges in the network
- G_f is a snapshot of a network used to generate the feature set
- G_s is a labeling snapshot used to label the node pairs
- $\tau(s)$ represents the neighbors of node s
- $Score(s, t)$ denotes the similarity score between node s and node t
- The training and testing datasets are generated by combining the features and labels

Supervised link prediction methodology. For this work, we adopt a supervised link prediction method proposed by Katukuri et al. [7]. They also propose a feasible approach to generating the labeled dataset without involving a domain expert. They use Map-Reduce algorithms to generate and analyze the large-scale graph in a distributed environment. The promising results of their experiments motivated us to consider this link prediction method for our study. Below, we discuss the link prediction methodology in detail.

This methodology uses a concept graph generated by parsing the publication metadata in the MEDLINE dataset [13]. We extracted publication information on authors, dates, document ID, keywords from fields such as MeSH Heading List, Chemical Compounds List and Gene Symbol List in the MEDLINE dataset. These keywords are part of MeSH (Medical Subject Headings), a medical vocabulary thesaurus curated by the National Library of Medicine [28]. We consider these keywords as concepts in concept network. For a given concept network, a node represents a medical concept that has appeared in one or more publications and an edge between two concepts indicates that the concepts co-occur in one or more publications. The node count represents the number of documents in which the concept appears, and the edge count (edge weight) indicates how many documents the two concepts appeared together in.

We generated snapshots of the concept network at two consecutive, non-overlapping points in time. We refer to the first snapshot as the *feature snapshot* and to the second snapshot as the *labeling snapshot*. The topological features used in our supervised method are extracted from the feature snapshot. The labeling snapshot is compared to the feature snapshot to generate the class labels.

The supervised method uses a heterogeneous feature set extracted from the training snapshot to discriminate between positive and negative links. The feature set is composed of neighborhood and proximity features:

- **Common Neighbors**[29]: This is a fundamental and direct approach for measuring the similarity between two nodes. Let two nodes be s and t. The number of neighbors common to both of them represents the similarity. The similarity score is defined as follows:

$$Score_{CN}(s,t) = |\tau(s) \cap \tau(t)|$$

- **Adamic/Adar** [30]: The Adamic and Adar measure is like the common neighbors measure, but it also weighs the common neighbors:

$$Score_{AA}(s,t) = \sum\nolimits_{z \epsilon \tau(s) \cap \tau(t)} \frac{1}{\log|\tau(z)|}$$

- **Jaccard Coefficient** [31]: The Jaccard coefficient is commonly used in informational retrieval to measure the similarity between two sets. It measures the probability that a selected node from the union of two neighborhoods is in both the neighborhoods.

$$Score_{JC}(s,t) = \frac{|\tau(s) \cap \tau(s)|}{|\tau(s) \cup \tau(s)|}$$

- **Preferential Attachment** [29]: Preferential Attachment models the growth of networks on the presumption that the node with a higher number of neighbors has a higher chances of being part of a newly formed edge. For our purposes, the score between two nodes is calculated as:

$$Score_{PA}(s,t) = |\tau(s)|.|\tau(t)|$$

- **Cycle-Free Effective Conductance (CFEC)** [32]: The cycle-free effective conductance (CFEC) proposed in measures the proximity between two nodes in a network [32] . The CFEC between the nodes s and t is formulated as

$$Score_{CFEC}(s,t) = deg_s \,.P_{cf.esc}(s \to t)$$

where deg_s denotes the degree of node s. The cycle-free escape probability $P_{cf.esc}(s \to t)$ is the probability that a random walk that beings at s will reach t without visiting any node more than once. It is represented as

$$P_{cf.esc}(s \to t) = \sum\nolimits_{r \epsilon R} Prob(r)$$

where R is the set of simple paths between the nodes s and t. In a random walk, the probability of a path P ($Prob(P)$), where the transition probability from node *i* to node *j*, is $p_{ij} = \frac{w_{ij}}{deg_i}$, is defined as:

$$Prob(R) = \prod\nolimits_{i=1}^{N} \frac{w_{v_i v_{i+1}}}{deg_{v_i}}$$

To make this method portable across different datasets and domains, we excluded the domain-specific semantic features proposed in proposed by Katukuri et al. [7].

The training dataset consists of node pairs that are not associated in the training snapshot (G_f) and are associated in the labeling snapshot (G_s). For each pair in the training dataset, features are extracted from G_f and corresponding class labels are generated using an automatic labeling approach. An automatic labeling approach compares the training snapshot with the labeling snapshot to generate the labels for the training set. A node pair is labeled *positive* if the pair is not connected in G_f but is strongly connected in G_s. The pair is labeled *negative* if it is weakly connected in G_s. An edge is determined as a strong or weak connection based on the user-defined parameters *minimum support* and *margin* [7]:

- Connection is strong if $S \geq minimum\ support$
- Connection is emerging if $margin \times minimum\ support \leq S < minimum\ support$
- Connection is weak if $S < margin \times minimum\ support$
- No connection exists if $S = 0$

For our experiments, we considered strong and weak connections.

Dataset. We used biomedical literature data from MEDLINE[13] for the years 1969 to 2005 to build a concept network. Table 2-1 shows some of the statistics relating to this concept network.

In the concept network, the total number of edges not formed (classed as negative) is significantly higher than the edges formed (classed as positive). Therefore, the dataset that is generated is highly unbalanced, which can result in a biased classifier. To remedy this, we randomly selected an equal number of instances from both classes to generate a balanced dataset.

Evaluation of link prediction model. Link prediction can be evaluated as a classification problem since the supervised method is modeled as a binary classification. For this study, we use a C4.5 decision tree to train the link prediction classifier. A classification model can be evaluated using several different measures, of which we consider accuracy, precision, recall, and the F1 score.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\ score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

True positive (TP) and true negative (TN) are the respective numbers of positive and negative labels correctly classified, whereas false positive (FP) and false negative (FN) are the numbers of those incorrectly classified. We employed a 10-fold cross-validation evaluation approach to estimate the prediction performance of the link prediction model.

Table 2-1 Statistics of the concept network 1969-2005

Total number of publications	9584017
Total number of concepts (Nodes)	195358
Total number of concept pairs (Edges)	43908827

2.4 Experiments and Results

In this section, we describe the experiments performed to evaluate the supervised link prediction approach for generating hypotheses from the literature data. The following experiments were designed to assess different aspects of the link prediction method.

- As a baseline experiment, we replicated the implementation of the prediction model with the user settings explained by Katukuri et al. in order to assess the model's performance [7].
- We also perform the experiments with pruned datasets to evaluate the model's performance at different pruned levels.
- An experiment was performed to investigate the influence of the temporal duration of a snapshot on the overall performance of the predictive model.
- Another experiment assessed the impact of the temporal distance between the training and testing snapshots on the performance of the predictive model.
- To investigate the effectiveness of the link prediction model as a hypothesis discovery application, we performed an evidence-based evaluation by testing the model on proven relationships.

Baseline experiment. A baseline experiment is designed with default user-defined values and settings as proposed in methodology by Katukuri et al. [7]. The results of this analysis help to test the methodology and to interpret the results of rest of the experiments.

From the dataset, two graph snapshots, G_f and G_s, are extracted from two time periods. G_f is a feature snapshot that is used to extract the features mentioned in the methodology section. G_s is the labeled snapshot that is used to label every node pair as positive or negative based on the strength of the link. In the current experimental setup, the thresholds are set as 5.0 for minimum support and 0.4 for margin.

Table 2-2 Datasets and time periods used in the baseline experiments

	Training Dataset		**Testing Dataset**	
	Feature Snapshot	*Labeling Snapshot*	*Feature Snapshot*	*Labeling Snapshot*
1	1975-1979	1980-1984	1980-1984	1985-1989
2	1980-1984	1985-1989	1985-1989	1990-1994
3	1985-1989	1990-1994	1990-1994	1995-1999

A C4.5 decision tree is applied to the features extracted from the dataset to generate a classifier model. The classification accuracy, precision, recall and F1 score are evaluated using a10-fold cross-validation. The experiment is repeated with training data generated from different temporal snapshots of the network (shown in Table 2-2). The characteristics of the data used for the classification are presented in Figure 2-1.

Figure 2-2 shows the accuracy, positive precision, positive recall, positive F-score, negative precision, negative recall, and negative F-score. The accuracy of the prediction model was found to vary from 73.12% to 76.80% for different training data generated at different time periods. The model predicted the link formation with a recall of 73% and a precision of 75%. It is important for a link prediction model to be able to classify negative (non-forming) links as well as positive links. Results show that our model can predict negative links with a precision of 75% and a recall of 76%.

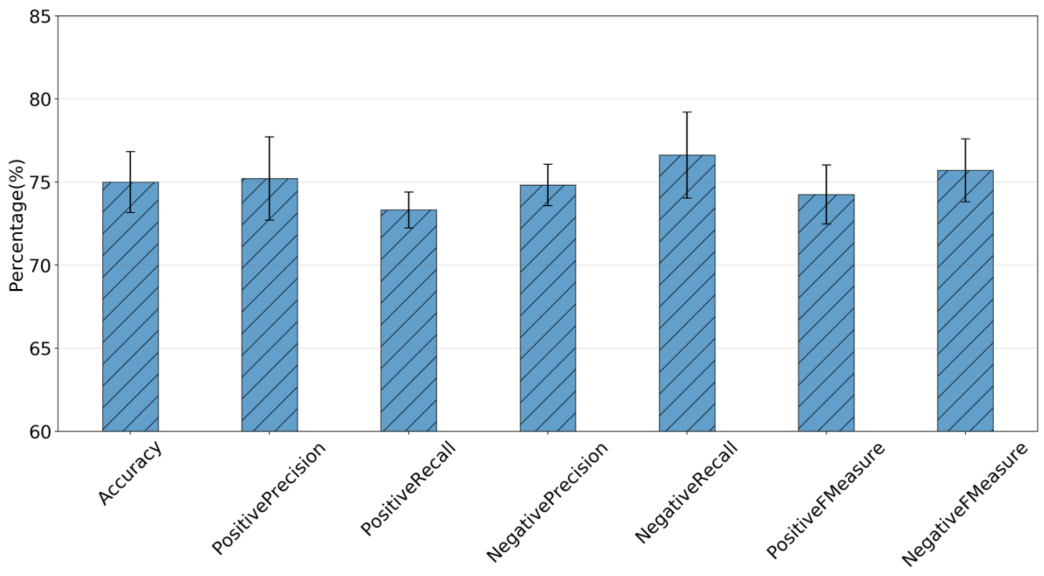

Figure 2-1 Baseline experiment: classification accuracy, precision, recall and F1 score for three training datasets (1975-79, 1980-84, 1985-1989) with temporal durations of five years

Semantic-type-based pruning. Typically for a graph analysis approach such as link prediction, the network is pruned in order to be able to process the massive graphs. Pruning (or filtering) reduces the size of the graph, which reduces the cost of extracting information from a large and dense graph. In this section, we investigate the use of semantic information for pruning the graph and evaluate the impact on the link prediction model of pruning based on semantic types.

Co-occurrence-based pruning eliminates edges that are lower than a threshold, i.e., if two words appear less frequently than a user-defined threshold these edges are eliminated. In semantic-type-based pruning, the semantic type of the node is considered when pruning the network. Unlike co-occurrence-based pruning, whereby edges are pruned, semantic-type-based pruning entails pruning the nodes in the network. Appropriate semantic types of nodes are selected for pruning, which ensures that important and relevant nodes and relationships are preserved in the network.

Three levels of pruning are considered:

- ***Comprehensive***: All the concepts extracted from the literature or publication data are considered irrespective of their semantic types.
- ***Selective***: Certain concepts whose semantic types are not informative are excluded at this level.
- ***Precise***: At this level, only specific concepts belonging to semantic types of interest are considered.

In the case of MEDLINE data, each node in the network is a biomedical concept with a MeSH-defined semantic type. At the comprehensive level, the network consists of all the nodes irrespective of their semantic type.

At the selective level of pruning, the network is pruned based on the semantic types that are not informative or relevant to the target prediction. In this experimental setup, link prediction is used to predict relationships between drugs and diseases. Therefore, concepts that belong to semantic types such as, "administrative" and "health care," are not used in the proposed link prediction scenario. Pruning unwanted semantic types reduces the noise in the network. At this level of pruning, the network includes concepts belonging to other relevant semantic types, such as “phenomena and processes,” and “psychiatry and psychology,” in addition to the target semantic types. At the precise level, the network only includes concepts belonging to the targeted semantic types.

To study the effect of semantic-type-based pruning of the network on the link prediction algorithm, we replicated the baseline experiment with the three proposed levels of pruning techniques. The experiments use the datasets shown in Table 2-2.

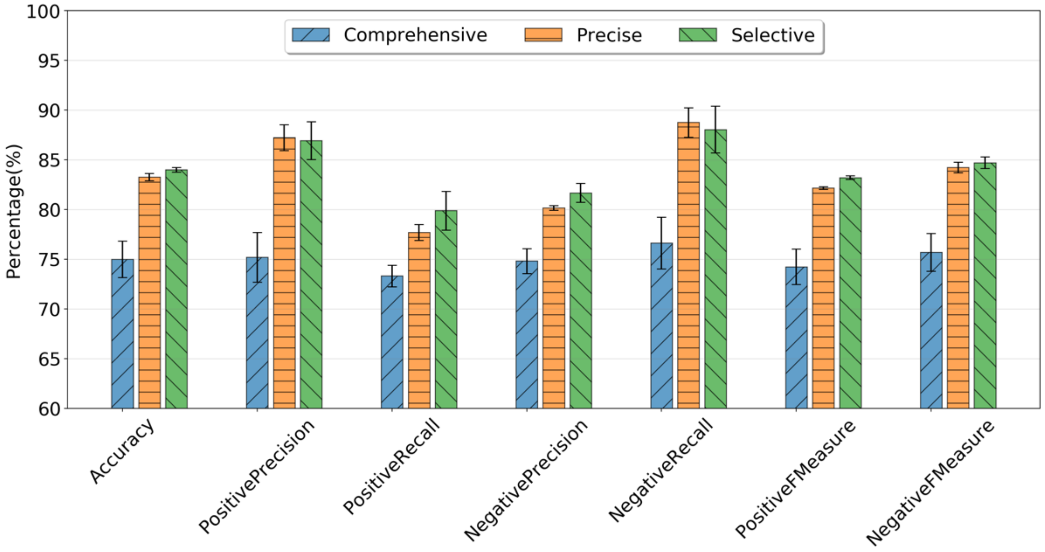

Figure 2-2 Classification accuracy, precision, recall and F1 score for the three levels of pruning based on semantic types: comprehensive, selective and precise.

The accuracy, positive precision, positive recall, positive F1 score, negative precision, negative recall, and negative F1 score for each level of pruning are presented in Figure 2-2. The Link prediction algorithm uses topological similarities between nodes to predict links. At the comprehensive level of pruning, concepts not relevant to the target prediction introduce noise into the network. The results show that:

- Pruning the network improves prediction performance by filtering irrelevant nodes that do not contribute to link prediction.
- Selective pruning produces more accurate predictions than precise pruning.

The effect of temporal duration. Link prediction is performed on a current network snapshot to predict relationships in a future snapshot of the network. In a dynamic setting, the topology of graphs changes over time. Thus, it is important to select the optimal size of the snapshot for achieving the best predictive performance. The time interval over which a snapshot is generated is referred to as the *temporal duration* of a snapshot. In this section, we study the effects of the temporal duration of a snapshot on the predictive performance of the link prediction model. For this experiment, we prune the network at the selective level.

As previously discussed, training and testing datasets are generated using two consecutive time snapshots; the feature snapshot (G_f) and the labeling snapshot (G_s). The impact of temporal duration on the model's predictive performance is evaluated by varying the temporal duration of G_f from 1 year to 11 years while maintaining the temporal duration of G_s at 5 years.

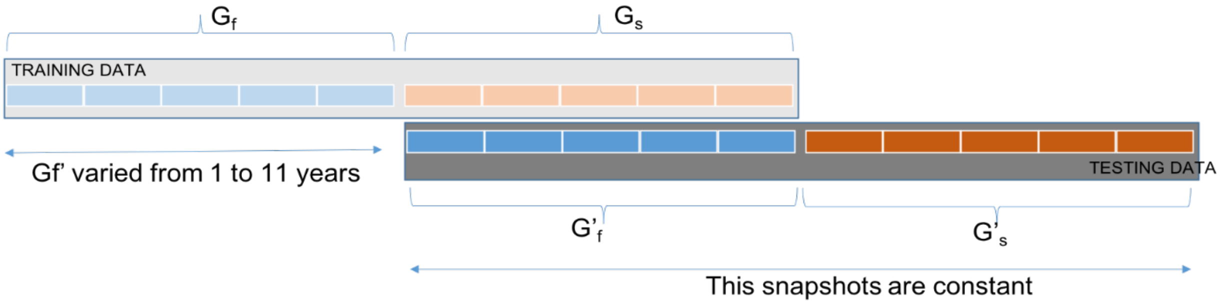

Figure 2-3 Varying the temporal duration on feature snapshot (G_f) of training dataset.

Varying the temporal duration of G_f of training data. In this experiment, we vary the temporal duration of snapshot G_f of training data from 1 to 11 years and maintain the temporal durations of all the other snapshots at 5 years (illustrated in Figure 3). The results of this experiment are plotted in Figure 2-4. It was found that as temporal duration increases, the accuracy of the model increases up to 5 years and then begins to decline. We also observed that positive recall is high at shorter temporal durations and decreases as temporal duration increases.

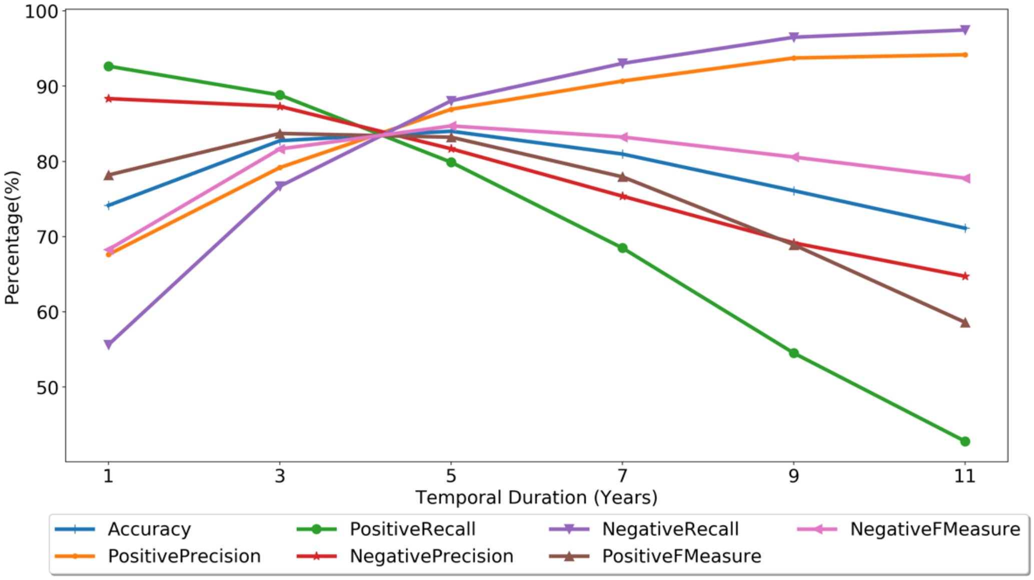

Figure 2-4 Accuracy, precision, recall and F1 score with a varying temporal duration of G_f of training dataset from 1 years to 11 years. Temporal duration of the G_s is constant at 5 years.

Graph snapshots are generated by aggregating the concepts (nodes) and their co-occurrences (edges) in the published literature for a given temporal duration. The shorter the duration, the less relationships are formed in the graph, resulting in a sparse graph. This is in contrast to the dense graph generated by aggregating over long durations (e.g., 10 years). The topological features of sparse graphs are different to those of dense graph. For example, the number of common neighbors for a pair of node in a sparse graph is less than that of the same nodes in a dense graph. We can observe similar variations in other topological features used in this study.

The model built with features extracted from a snapshot of short temporal duration to predict the links on test data generated from a snapshot of 5 years duration tended to classify negative pairs as positive which resulted in more false positives. This led to low positive precision and low negative recall. When the temporal duration of the training data was close to the temporal duration of the test data, this resulted in optimum precision and recall and, in turn, higher accuracy. When the model was trained using the topological features generated from snapshots of longer durations, it tended to classify positive pairs as negative. This resulted in lower positive recall and higher negative recall.

Varying the temporal duration of G_f of testing data. In this experiment, the temporal duration of G_f of testing data is varied, while the temporal duration of the training snapshot is kept constant at 5 years (illustrated in Figure 2-5). The results of this experiment are plotted in Figure 2-6. Unlike the experiment with the training dataset, it was found that at shorter temporal durations, the precision of the model's predictions was high and that it decreased as the temporal duration increased. However, recall was found to be lower at shorter durations and to increase as temporal duration increased.

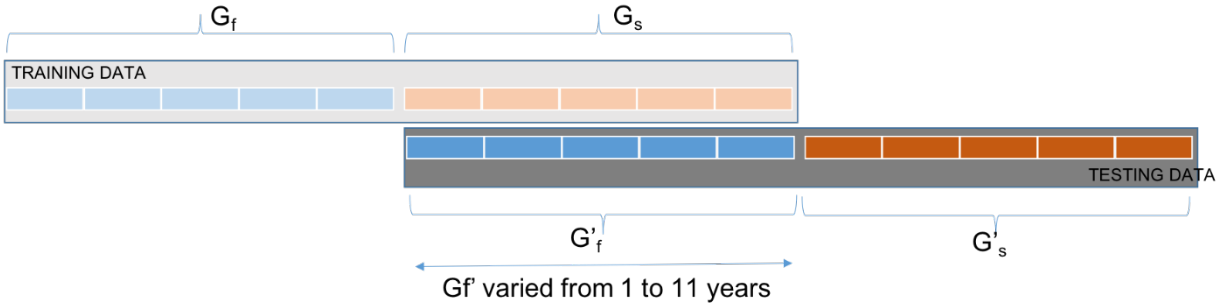

Figure 2-5 Varying the temporal duration on feature snapshot (G_f) of training dataset.

For this experiment, the model was trained using the graph snapshot of 5 years temporal duration. The test dataset was generated using the snapshot with temporal durations varying from 1 to 11 years. As found in the previous experiment, the features extracted from the shorter duration are weaker than those extracted from the snapshot of longer durations. At shorter temporal durations, the testing data has indistinct temporal features compared to the stronger topological characteristics of the training data. Due to this difference in features, the model classified more positive pairs as negative, which led to low positive recall and high negative recall. When the temporal duration of the test data was 5 years, which was the temporal duration of the training data, precision and recall reached optimum values, resulting in higher accuracy. When the temporal duration of the test data increased further and the training data was maintained at 5 years duration, the model tended to classify the positive pairs as negative, which resulted in higher positive recall and lower negative recall.

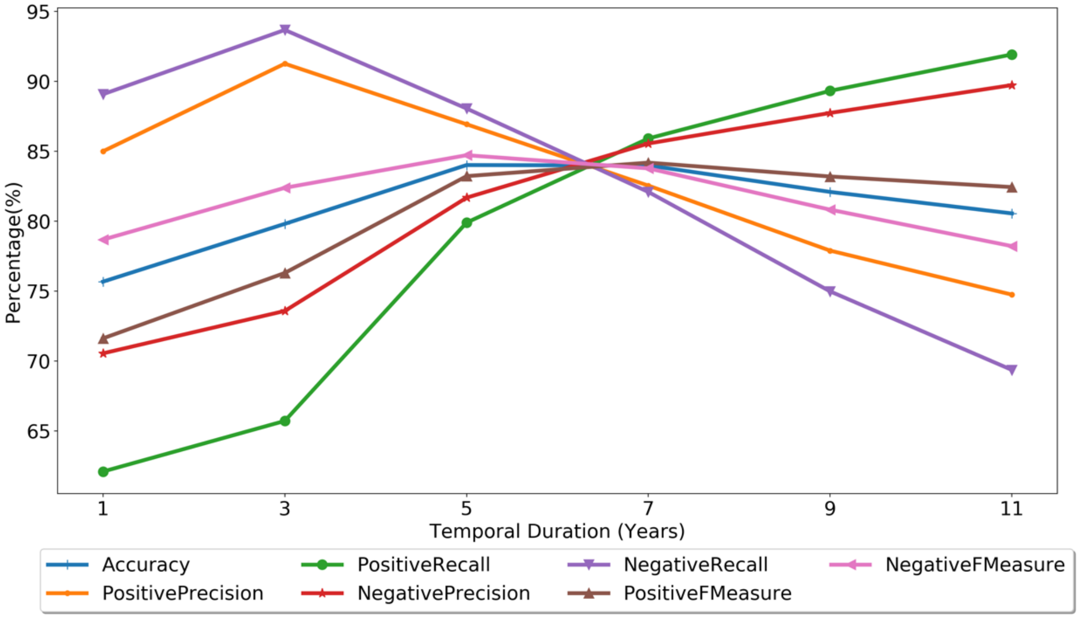

Figure 2-6 Accuracy, precision, recall and F1 score with varying temporal duration of testing dataset from 1 years to 11 years.

The effects of temporal distance. In a dynamic setting, the model must be updated as new data becomes available. To know how frequently the model requires updating, we first need to know how well the model performs predicting relationships in future data. The temporal distance between the data used to generate the model and the data used to test the model is referred to as *temporal distance* (illustrated in Figure 2-7).

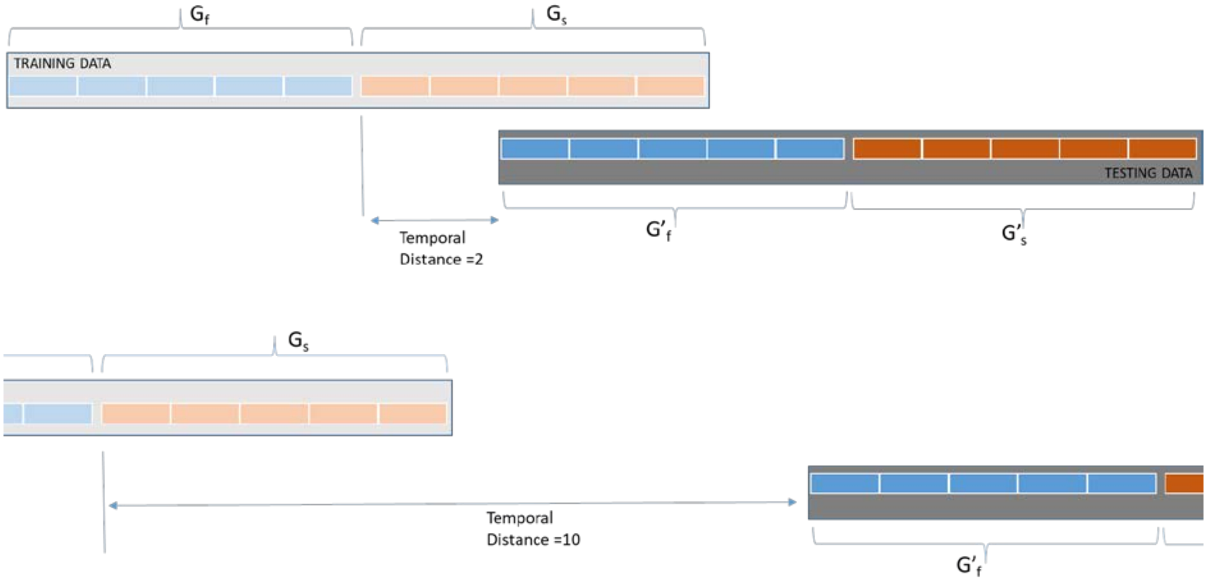

Figure 2-7 Illustration of the concept temporal distance.

In this section, we evaluate the impact of temporal distance on the predictive performance of the link prediction model. We generate several datasets varying the temporal distance between the training and testing data from 0 to 10 years in two-year steps. The temporal duration of each snapshot in this experiment is set at 5 years.

Figure 2-8 shows the results of the experiment with a temporal distance varying from 0 to 10 years between the training and testing datasets. From Figure 2-8, we can observe that accuracy decreases as temporal distance increases. Also, there is a notable decrease in positive recall and an increase in positive precision with an increase in temporal distance. This suggests that our link prediction model was able to predict most of the old patterns in the new data but failed to classify new patterns in the data. Even though accuracy decreased, the link prediction model performed considerably better predicting new links in future data at longer temporal distances, which eliminated the need to frequently retrain the model.

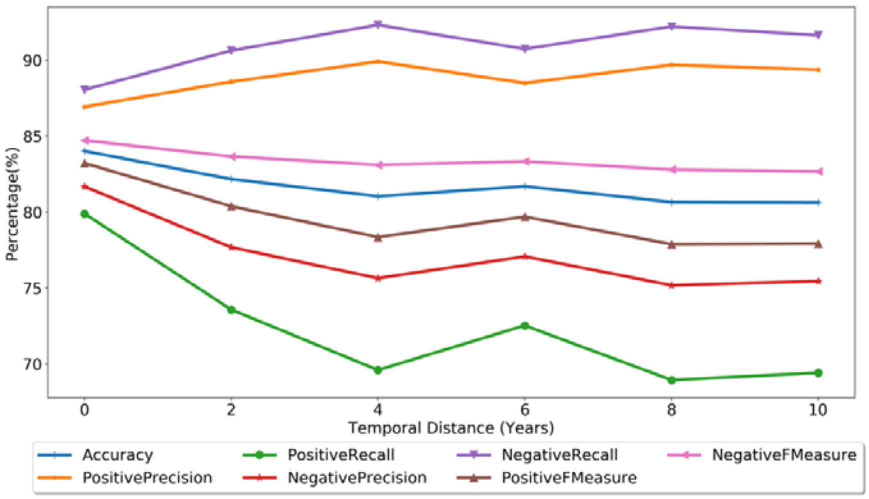

Figure 2-8 Accuracy, precision, recall and F1 score with temporal distance between training and testing dataset varying from 0 years to 10 years.

Evidence-based evaluation. We evaluated the link prediction model based on evidence from real-world instances of hypothesis testing. The evidence-based evaluation was intended to validate the ability of the prediction algorithm to predict known relationships in the literature.

Table 2-3 List of the known relationships that are hypothesized and proven in different studies

MeSH Concept	MeSH ID	MeSH Concept	MeSH ID	Year
Fish Oils	M0008518	Raynaud’s Disease	M0018534	1985
Migraine Disorder	M0013864	Magnesium	M0012884	1988
Indomethacin	M0011240	Alzheimer’s Disease	M0012884	1989
Somatomedin	M0007795	C Arginines	M0001683	1994
Calcium-Independent Phospholipase A2	M0275442	Schizophrenia	M0019489	1997
Estrogen	M0007795	Alzheimer’s Disease	M0000842	1995

Some of the de facto relationships that were discovered in the literature are listed in Table 2-3. The dataset was designed to include these relationships in the labeling graph of the test data. The training dataset was created by keeping test duration constant. Both the training and testing datasets were generated over 5-year durations. The threshold was set at 5 and margin was set to 0.4.

Table 2-4 shows the ability of the link prediction algorithm to discover relationships that are already known. In the results, Y denotes the ability of the model to predict the link, whereas N denotes that the model was not able to predict the link. Using the data pruned at a comprehensive level, the model discovered one relationship, compared to four out of six relationships discovered at selective and precise levels of pruning.

Table 2-4 Results of identifying known relationships using the link prediction approach for different semantic-type-based prune settings. In the table, Y represents the relationships discovered and N represents those not discovered.

Concept1	**Concept2**	**Comprehensive**	**Selective**	**Precise**
Fish Oils	Raynaud's Disease	N	Y	Y
Migraine Disorder	Magnesium	N	Y	Y
Indomethacin	Alzheimer's Disease	N	Y	Y
Somatomedin	C– Arginine	N	N	N
Calcium-Independent Phospholipase A2	Schizophrenia	Y	Y	Y
Estrogen	Alzheimer's Disease	N	N	N

2.5 Conclusion

This study considers the temporal aspects of the link prediction model: temporal duration and temporal distance. It addresses the necessity to consider these aspects when evaluating the model on a time-varying network. Our results indicate that the supervised link prediction model is reasonably reliable in predicting links in future datasets. This eliminates the need to update the model every time new data becomes available. To achieve maximum accuracy, the duration of the feature snapshot of the testing data should be similar to the duration of the feature snapshot of the training dataset. The testing dataset is less effected than the training dataset by the temporal duration of the feature snapshot.

The model, which was tested using semantic-type-based pruning, performs better at the selective pruning level than at the precise and comprehensive pruning levels. We also evaluate the link prediction approach by testing it on known links. The model was able to identify four out of six known links at the precise and selective pruning levels compared to one link at the comprehensive level.

In the future, we want to explore an incremental approach to updating the model instead of retraining the entire model. We are also interested in predicting the strength of the link as well as the link itself. We hope to take an ensemble approach using multiple methodologies to create a model that makes better predictions instead of relying on a single method.

3 . Supervised Interestingness-based Link Ranking

3.1 Introduction

In recent years, the number of scientific publications has increased at an unprecedented rate. For instance, the MEDLINE database [13], a U.S. National Library of Medicine bibliographic database for biomedical publications, has more than 24 million entries [14]. In 2016, more than 869,000 new citations were added, averaging approximately 2,380 citations a day [14]. One effect of this high rate of publication is that it poses difficulties for researchers to stay up to date with current research. This problem highlights the need for an automated approach to extract and interpret hidden links within published literature.

Literature-based discovery (LBD) is a field of research that seeks to derive hidden knowledge from the published literature. It is a branch of data mining that uses text mining and information retrieval (IR) techniques in order to discover and extract significant relationships in published literature. The concept was first introduced in 1986 by Swanson, [15] who hypothesized a relationship, unexplored at the time, between fish oil and Raynaud's disease. Swanson manually identified relationships between two sets of literature. The first set contained publications that mentioned an increase in blood viscosity due to Raynaud's disease, and the second set contained publications about the effect of fish oil on reducing the level of blood viscosity. The study linked Raynaud’s disease and fish oil via the common concept of blood viscosity and hypothesized that fish oil may be used to reduce the effects of Raynaud's disease. This was later proven accurate in clinical trials [16]. Several other works sought to replicate Swanson’s approach and to automate the process using computer-based algorithms and techniques.

One approach is to model knowledge from literature as a network (graph), where the nodes represent knowledge entities and edges represent relationships between entities. Networks are an intuitive way of representing the relationships between different entities and are an important data-modeling tool for many areas of research such as social networking, e-commerce, biomedicine and recommendation systems. For instance, in a biomedical study of protein-protein interactions, interactions are represented as network, where proteins are the nodes, and the interactions between the proteins are edges [2].

Mining these networks to discover interesting patterns is a growing area of research in network analysis. One important problem in network analysis is that of predicting the missing links within the networks, commonly known as the *link prediction problem*, where links represent relationships or associations between nodes. The goal of the link prediction problem is to predict future associations between nodes based on currently observed links and nodes and their features. Hence, one can treat the LDB problem as a link prediction problem.

A study published by Őzgür et al. is an early example of LBD, in which the researchers generated hypotheses regarding potential relationships between vaccines and genes. To accomplish this, they used gene interaction networks generated from the literature [18]. Jeong et al. studied the protein-protein interaction network of yeast to predict lethal mutations [2]. Most of the earlier works concentrated on specific semantic types and relationships in biomedical literature. More recent studies have developed hypothesis discovery methods using a comprehensive network of concepts, irrespective of semantic type [7][19].

One major issue with such hypothesis generation methods is the large number of links discovered using link prediction. Only a small number of the discovered relationships may be of interest to the user, while the rest of them may be irrelevant or unimportant. Given that most existing link prediction algorithms make a binary decision (“Yes” or “No”)[7][17][3] when predicting the formation of a future link, the user has the unappealing task of sorting through the predicted links to determine those that are of interest.

To address this problem, we propose a supervised method for ranking predicted links. This proposed method uses the structural features of the graph to generate a regression model, which forecasts the “interestingness” of the links. This interestingness measure is defined using one of the thirteen probability-based objective measures typically used in data mining. The links are ordered based upon the predicted interestingness. We validated this approach using the MEDLINE database. Results showed that predicting an exact level of interestingness is difficult. However, we can accurately predict the future rank using several different interestingness measures.

3.2 Related Work

Literature-based discovery. The approach by Swanson is based upon the assumption that any two concepts that are not directly related but have significant indirect relationships through common concepts might be related [15]. In addition to his initial work, where he identified the relationship between Reynold’s disease and fish oil, Swanson also published a hypothesis regarding a possible relationship between migraines and magnesium[20] using the same principle.

Initially, most of the work in LBD was done manually by searching literature databases. Gordon and Lindsay [21] replicated and automated Swanson's LBD process using computer-based algorithms and IR techniques. They used the dataset from the MEDLINE[13] database and applied traditional IR techniques to identify the relationship between Raynaud’s disease and fish oil. A similar system named Arrowsmith was developed by Swanson and Smalheiser [22] in 1997. Most of the earlier work attempted to replicate Swanson's methodology using various IR algorithms. Some of this earlier work on LBD also used semantic information and lexical statistics of concepts along with co-occurrence information to generate hypothesis [23][24][25].

In recent studies [2][3][7][18], link prediction analysis has been applied to biomedical concept networks to generate possible hypotheses. Taking this approach, a biomedical concept network is generated using concepts extracted from publications, where biomedical concepts are represented as nodes and the co-occurrence of two concepts is represented as an edge. This biomedical concept network, typically referred to as a concept network, evolves over time with the formation of new relationships between nodes and the creation and deletion of nodes. Link prediction analysis is applied to the concept network to predict future relationships between nodes. In other words, link prediction attempts to forecast a future connection between two nodes that are currently not connected.

Őzgür et al. [18] proposes a hypotheses generation method, which applies a link prediction technique to LBD for predicting relationships between genes and diseases. Initially, a seed list is generated from a list of genes that are related to a specific disease. Later the seed list is used to construct a disease-specific gene interaction network using automatic literature mining based on dependency parsing and support vector machines.

Genes in the network are ranked based on different centrality measures, which are calculated for each node in the network. The study hypothesizes that the central gene (or genes) has a possible association with the specific disease. Jeong et al. [2] proposes a similar method using degree of centrality as a measure to predict lethal mutation in a yeast protein interaction network.

Link prediction. Liben-Nowell et al. [1] introduced the link prediction problem and proposed a methodology to predict a missing link in a social network using the network properties of two nodes. In a similarity-based algorithm, every node pair is assigned a score, which represents the similarity or proximity of node pairs. These node pairs are ordered based on their scores, and the pairs with the higher similarity are regarded as likely to be associated in the future. While ordering can be used to confer a ranking, the ranking only indicates the likelihood of the link occurring. The proposed method by Hasan et al. uses several local and global topological features and a number of different classifiers for predicting the presence or absence of new links in a social network [17]. In biomedical networks, Tao Zhou et al. [33] use local information to predict missing links in a number of datasets, including a protein-protein interaction set. They examine how accurately the probable occurrence of a missing link can be predicted using a single measure. Lei et al. [34] propose the reconstruction of a protein-protein interaction network using global topological similarity. They use random walk-based similarity algorithms to determine the possibility of different protein interactions.

Predicting links using individual predictors has proven more effective than random predictors [1]. The performance of predictors varies depending on the type of the network. Most of the above approaches are designed for graphs in a specific domain and are not applicable to graphs in other domains. Instead of using a single-feature, classifier-based algorithms, multiple features can be used to determine the possibility of future links. Hasan et al. [17] propose a supervised learning method to predict future links in a social network using topological features extracted from the network. These topological features are combined with aggregated and semantic features to build a binary classifier model for predict future co-authorships. O'Madadhain et al. [35] propose a joint probabilistic model for link prediction using topological features and node attributes. They also develop an algorithm that can rank nodes and compare changes to node ranks over time. This approach does not predict future ranks; rather, it is aimed at facilitating comparisons of current nodes to past nodes based on importance.

Link importance. The above approaches, along with many other link prediction solutions, focus on the accuracy of link prediction. However, as noted previously, identifying the importance of predicted links is also crucial, particularly when there are hundreds or thousands of possible links. To achieve this goal, two critical issues need to be resolved: (a) how to define importance and (b) how to accurately predict importance.

Kahanda and Neville [36] propose a method for predicting the strength of a link using transactional information between two users in an online social network. Their study set out to predict if the link between two users indicated a strong friendship (positive) or not (negative), but this approach could potentially be used elsewhere. However, it requires that the links are labeled, which is can be time-consuming. Xiang et al. [37] propose an unsupervised method for representing a range of relationship strengths from “strong” to “weak” in a social network. The strength of a link is measured using profile similarity and interaction activity and represented as a continuous-valued relationship. The main premise of this study is that users are likely to have stronger links (friendships/relationships) with people who are similar. Their approach relies on two assumptions: (a) that every node is described by the same set of features, and (b) that similar nodes tend to form strong relationships. The framework proposed by Kamath et al. uses Twitter features such as historical interactions, type of interactions, similarity with other users and social graph features to predict the strength of the link between two users [38]. This work quantifies the strength of the link as the probability of future interaction between two users. The framework was designed specifically using Twitter data. Zhang and Dantu [39] propose a method for predicting tie strengths between phone users based on the call details of their mobile phones. Several works leverage user interaction data to predict or estimate the strength of links between different entities in social networks [38][40]. However, most of the prior work in this area utilizes, and assumes the availability of, rich information from social networks to model and quantify the relationships among entities in a network. Not all networks have the kind of rich information that social networks can provide. Hence, it is useful to measure the importance of links using more widely applicable methods.

Interestingness. Interestingness measures are used extensively in the field of data mining to select or filter discovered patterns. Liu et al. [41] proposes a method to rank interesting patterns using the users’ existing knowledge and general impression. Ohsaki et al. [42] evaluated the ranking of rules using various interestingness measures against the rankings ascribed by experts. A study by Tan et al. [43] proposes a different key property to select the right interestingness measure for a given dataset or domain. They consider twenty-one different interestingness measures in their analysis of support-based pruning and contingency table standardization and conclude that no single measure can achieve consistency across all domains and datasets. Lenca et al. [44], consider the properties of measures to identify the measures that fit users' needs.

In the field of link mining, Aljandal [45] proposes a supervised link prediction method representing interestingness measures as numerical features to improve the prediction of a link. To the best of our knowledge, this study is the first attempt to use interestingness measures to study the importance of a predicted link.

3.3 Methodology

The primary goal of this work is to score or rank predicted links. Therefore, it is useful if the number of features required to enable ranking is minimized; ideally, no features apart from the chosen interestingness measure need to be considered. To accomplish this goal, we use the same feature set used for an existing, accurate method for predict future links proposed by Katukuri et al. [7].

In Katukuri et al.'s work, biomedical literature is modeled as a series of concept networks, where each network represents a set of literature from a given time period; subsequent networks represent literature published in later time periods. Within a network, a node represents a biomedical concept and links represents the relationships between two concepts within that time period. Each node, in addition to representing a concept, has a count, which indicates the number of publications in which the concept appears. Each link also has a count, which represents the number of publications in which the two linked concepts appear together.

After the networks are generated, features are extracted and used to make predictions, and node pairs are labeled (link formed or did not form). This requires two temporally consecutive and non-overlapping networks. The features are calculated using the earlier snapshot while the later snapshot is used to generate the class labels.

The features, which are calculated for each pair of nodes that are not connected in the earlier network, are both neighborhood-based and random walk-based. Formally, $\tau(s)$ represents the neighbor set of node *s*; that is, the nodes that are linked to node *s*. The following comprise the set of features.

- **Common Neighbors** [29]:

This is simple approach for measuring the similarity between two nodes. For a given two nodes *s* and *t*, the higher the number of common neighbors, the higher the similarity between them. The similarity score is defined as:

$$\mathrm{Score}_{\mathrm{CN}}(\mathrm{s},\mathrm{t}) = |\tau(\mathrm{s}) \cap \tau(\mathrm{t})|$$

- **Adamic/Adar** [30]:

The Adamic and Adar measure is similar to, but more robust than, the common neighbors measure. This measure weighs the common neighbors of the two nodes *s* and *t*:

$$\mathrm{Score}_{\mathrm{AA}}(\mathrm{s,t})=\sum_{\mathrm{z}\epsilon\tau(\mathrm{s})\cap\tau(\mathrm{t})}\frac{1}{\log|\tau(\mathrm{z})|}$$

- **Jaccard Coefficient** [31]:

The Jaccard coefficient is popular in IR for measuring the similarity between two sets. It is formulated as:

$$\mathrm{Score}_{\mathrm{JC}}(\mathrm{s,t})=\frac{|\tau(\mathrm{s})\cap\tau(\mathrm{s})|}{|\tau(\mathrm{s})\cup\tau(\mathrm{s})|}$$

- **Preferential Attachment** [29]:

Preferential attachment measures the growth of networks on the presumption that the node with the highest number of neighbors has the highest chance of being part of a newly formed edge. For our purposes, the score between the two nodes is calculated as:

$$\mathrm{Score}_{\mathrm{PA}}(\mathrm{s,t}) = |\tau(\mathrm{s})|.|\tau(\mathrm{t})|$$

- **Cycle-Free Effective Conductance (CFEC):**

The cycle-free effective conductance (CFEC) proposed measures the proximity between two nodes in a network [32]. The CFEC between the nodes s and t is formulated as

$$\mathrm{Score}_{\mathrm{CFEC}}(\mathrm{s,t}) = \mathrm{deg}_{\mathrm{s}} \, . \, \mathrm{P}_{\mathrm{cf.esc}}(\mathrm{s} \rightarrow \mathrm{t})$$

where deg_s denotes the degree of node s. The cycle-free escape probability $P_{cf.esc}(s \rightarrow t)$ is the probability that a random walk that begins at s will reach t without visiting any node more than once. It is represented as

$$\mathrm{P}_{\mathrm{cf.esc}}(\mathrm{s} \rightarrow \mathrm{t}) = \sum\nolimits_{\mathrm{r} \epsilon \mathrm{R}} \mathrm{Prob}(\mathrm{r})$$

where R is the set of simple paths between the nodes s and t. For a random walk, the probability of a path P, Prob(P), with a transition probability, p_{ij}, from node i to node j is defined using following equation:

$$\mathrm{Prob}(\mathrm{P}) = \prod\nolimits_{\mathrm{i}=1}^{\mathrm{N}} \frac{\mathrm{W}_{\mathrm{v_i v_{i+1}}}}{\mathrm{deg}_{\mathrm{v_i}}}$$

We make one modification to the feature set described by Katukuri et al. [7] to make this methodology applicable for different datasets and domains, which is to ignore the domain-specific semantic features. The experimental results presented by Katukuri et al. [7] show that the impact of removing them is minimal.

After the features are generated, the next step is to assign class labels to node pairs. In the link prediction model, a node pair is labeled as “positive” or “negative” depending on the strength of the connection of the nodes in the second (later) snapshot. These connections are categorized using the parameters *minimum support* and *margin.* The connection is labeled as:

- strong if $S \geq$ minimum_support
- emerging if margin $*$ minimum_support $\leq S <$ mimium_support
- weak if $S <$ margin $*$ minimum_support
- no connection if $S=0$

where S represents is the strength of the link (S is 0 if there is no link).

A training instance is labeled “positive” if the connection is strong and “negative” if the connection is either weak or there is no connection. Once the labels are generated, a supervised classification algorithm is used to train the model to predict future links.

In order to score and rank future links, an interestingness score is calculated. The interestingness score is based on the weights of the two nodes and that of the edge between them. A supervised regression-based algorithm is used to train the model to predict scores. All the predicted links are ranked based on their scores.

The above steps enable us to automatically generate features, labels and build models from temporal network data. In order to rank predicted links, we introduce an interestingness score which is discussed in next section.

3.4 Interestingness Measures

There are two possible ways of generating an interestingness score. The first is to engage multiple experts to evaluate each link and score its importance. This can be problematic when dealing with large domains (such as MEDLINE data), since it is time-consuming and cost prohibitive. Another problem is that different experts often have different opinions, and resolving disagreements regarding scoring increases costs and leads to additional complexities. It limits the potential generalizability of the approach and means that the process for assigning a score cannot be automated.

The second approach is to use an objective measure. The challenges associated with this approach are (a) how to define the objective measure, and (b) how to determine if the objective measure is appropriate. To address the first question, we selected interestingness measures used in association mining to rank and filter frequent itemsets. In our case, the itemsets were the linked nodes. To determine which measures to use, we reviewed several sources [43][44][45][46][47][48] and created a list of all the measures mentioned in them. From those measures, we selected thirteen that appeared most often and were straightforward enough to implement. These are listed in Table 3-1.

With respect to the appropriateness of the measures, this can be interpreted in one of two ways. The first is to ask whether we can predict the future interestingness or the rank of the link when the interestingness is defined by measure X. The experiments presented later in this study address this issue. The second interpretation of appropriateness is to ask if the measure ranks a relationship in the same way that a domain expert would? This is a more difficult question, as different experts may have different internal criteria. We do not address this issue in the current paper, as we are trying to establish if it is even possible to predict the score or rank of future links accurately.

In the following section, we introduce the modified Kendall's Tau method employed to compare the predicted ranking list with the actual ranking list.

3.5 Kendall's Tau Method

To compare the predicted ranking list with the actual ranking list, we adopt a modified form of Kendall's tau method presented by Fagin et al. [49]. One of the assumptions made in that study was that no two objects have the same rank. This is not true in our case, where it is possible that different instances might have the same interestingness score, which is later translated as the same rank.

Table 3-1 List of interestingness measures

Interestingness Measure	**Formula**
Coherence (CO)	$\frac{P(AB)}{P(A)+P(B)-P(AB)}$
Conviction (CV)	$\frac{P(A)P(\neg B)}{P(A\neg B)}$
Cosine (COS)	$\frac{P(AB)}{\sqrt{P(A)P(B)}}$
Information Gain (IG)	$\log(\frac{P(AB)}{P(A)P(B)})$
Kulczynski (KUL)	$\frac{P(AB)}{2}\left(\frac{1}{P(A)}+\frac{1}{P(B)}\right)$
Klosgen (KLO)	$P(AB)*\max(P(BA)-P(B), P(AB)-P(A))$
Least Contradiction (LC)	$\frac{P(AB)-P(A\neg B)}{P(B)}$
Linear-Correlation (LCOR)	$\frac{P(AB)-P(A)(B)}{\sqrt{P(A)P(B)P(\neg A)P(\neg B)}}$
Loevinger (LOE)	$1-\frac{P(A)P(\neg B)}{P(A\neg B)}$
Odd Multiplier (OM)	$\frac{P(AB)P(\neg B)}{P(B)P(A\neg B)}$
Piatetsky-Shapiro (PS)	$P(AB)-P(A)P(B)$
Sebag-Schoenauer (SS)	$\frac{P(AB)}{P(A\neg B)}$
Zhang (ZH)	$\frac{P(AB)-P(A)P(B)}{\max(P(AB)P(\neg B),P(B)P(A\neg B))}$

The method developed by Fagin et al. [49] measures the distance between the top k objects in two lists by adding appropriate penalties based on the order of objects appears in both lists [49]. Let us consider two top k lists, τ_1 and τ_2. In our case, these lists are the actual ranking list and the predicted rank list. A set $P(\tau_1, \tau_2)$ is defined as the set of unordered possible pairs of all distinct elements from both of the lists. Kendall's tau distance $K(\tau_1, \tau_2)$ is calculated as the pair-wise disagreement penalties for all the pairs of element in $P(\tau_1,\tau_2)$. The penalty for a pair of objects $(i, j) \in P(\tau_1, \tau_2)$ is determined as follows:

1. If *i* and *j* appear in both lists and in the same order, there is zero penalty and if in reverse order there is a penalty of 1.
2. If *i* and *j* both appear in one list (say τ_1) and exactly one object either *i* or *j* appears in another list(say τ_2); if the element that appears in τ_2 is ahead in τ_1 penalty is 0. Otherwise it is 1.
3. If one object (say *i*) appears in one list (say τ_1) and another object (say *j*) appears in another list (say τ_2), penalty is 1.
4. If both *i* and *j* appear in one list but neither appear in another list, we penalize with 0.5.

In our data, there is a possibility that *i* and *j* have the same rank when they both appear in a list (in case 1 and case 2 above). In case 1, if the objects *i* and *j* are the same in both lists, we do not add a penalty. If they have the same values in only one list, we add a compensation of 0.5. In case 2, if *i* and *j* appear the same in a list, we add a compensation of 0.5. We normalize the Kendall's tau score using the number of elements in P (τ_1, τ_2).

$$\hat{k} = \frac{k}{\sum p_i}$$

3.6 Experimental Setup

Datasets. We generate the concept graph using the MEDLINE dataset[13] obtained in 2014. The MEDLINE data is stored in XML files that contains metadata about the publications. We parse these files to extract information about publications including authors, dates, document ID, and keywords from fields such as MeSH Heading List, Chemical Compounds List and Gene Symbol List in the MEDLINE dataset. These keywords are part of MeSH (Medical Subject Headings) [28], a medical vocabulary thesaurus curated by the National Library of Medicine. We generate concept graphs using these keywords (medical concepts) as nodes and co-occurrences of the concepts as relationships (edges) between the concepts. The total number of publications pertaining to a concept is represented as the node weight. The edge weight indicates the number of documents in which the two concepts appear together.

Table 3-2 Basic statistics of concept network 1991-1995

Number of Nodes (Concepts)	71681
Number of Edges	12550625
Number of Documents	1447582

For our experiments, we used the literature published between the years 1991 and 2000. We generated two consecutive and non-overlapping networks for the years 1991-1995 and 1996-2000, respectively. Some of the network statistics are shown in Table 3-2. In addition to the statistics on the network, we studied the frequency distribution of the interestingness values, which are shown in the Figure 3-1. We observed that the distribution of scores were skewed for most of the interestingness measures. To handle the skewed data, we divided the data into bin, some of which had a large number of instances. To balance the score distribution, we randomly sampled the instances in the bins which had higher frequencies than the predefined threshold. In other bins, all the instances were considered.

It should be noted that, while in practice we would first predict the links and then predict the ranking of the links, such action is not necessary for this study. Since we are interested in determining if we can correctly predict the interestingness of future links, we assume that we have a method that always correctly predicts whether a link will form in the future.

Evaluation. The performance of the regression-based classifier is evaluated by the measures “relative absolute error” (RAE) and “root relative squared error” (RRSE).

$$RAE = \frac{\sum_{i=1}^{N}\left|\widehat{\theta_\iota} - \theta_i\right|}{\sum_{i=1}^{N}\left|\bar{\theta} - \theta_i\right|}$$

$$RRSE = \sqrt{\frac{\sum_{i=1}^{N}\left(\widehat{\theta_\iota} - \theta_i\right)^2}{\sum_{i=1}^{N}(\bar{\theta} - \theta_i)^2}}$$

The predicted value is represented as $\hat{\theta}$ and the actual value as θ. $\bar{\theta}$ is the mean of the actual values. We considered relative error metrics instead of absolute error metrics since different interestingness measures fall within different value ranges.

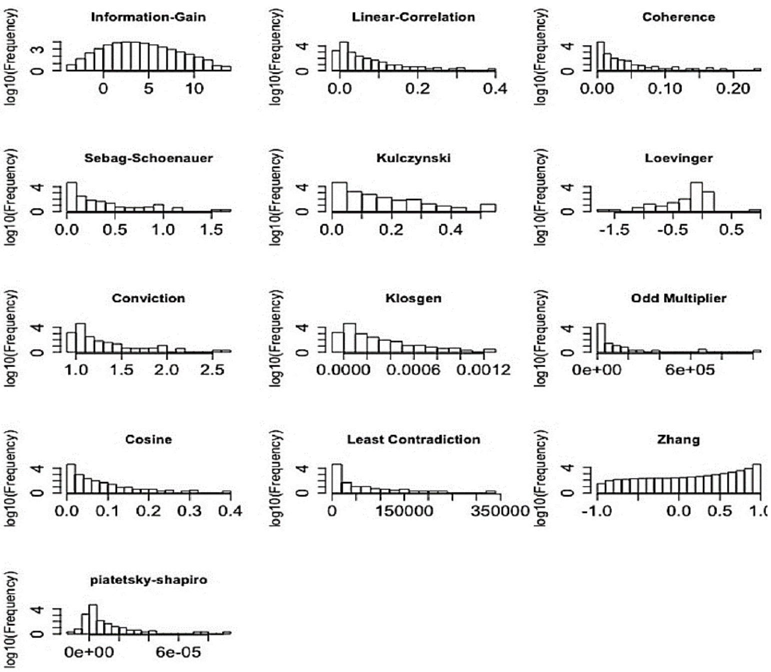

Figure 3-1 Distribution of interestingness values for different interestingness measures applied to biomedical datasets 1991-2000

3.7 Experiments

In this section, we evaluate the supervised ranking approach using different interestingness measures to select the interesting predictions from among all the predicted links. As a preliminary step, we studied the similarity between all the interestingness measures used in this paper. Later, we conducted the experiment using the supervised method to filter the interesting links. We evaluated two approaches:

- Predicting the interestingness scores of the predicted links.
- Predicting the ranking of interesting links from all predicted links.

Interestingness methods and their correlation. For any given dataset, we calculated interestingness using all the measures listed in Table 3-1and transformed these scores into ranks of predicted links. Some of the predicted links may have the same scores and hence the same ranking. To measure the similarity between two ranked lists generated using different measures, we adopted the Kendall's tau correlation, as discussed above.

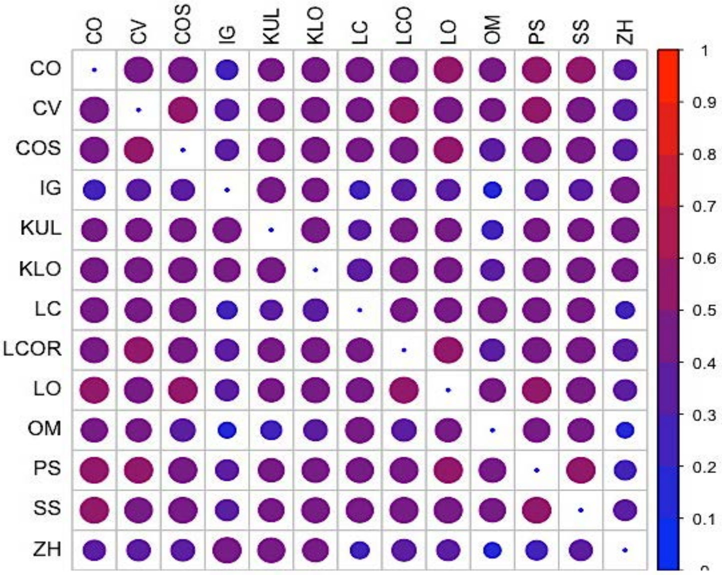

Figure 3-2 Kendall's tau correlation score for different interestingness measures

This correlation study helps us to understand the agreement and disagreement between the different measures. The Kendall's tau correlation score ranges from 0 (which implies that the measures agree in every case) to 1 (where the measures disagree completely). These correlation scores are illustrated in Figure 3-2, where the circles represent the correlation between different measures and the colors of the circles represent the discrete values of the scores. For instance, we can see from Figure 3-2 that "Information Gain" has the most similar rankings in contrast to "Least Contradiction", "Coherence" and "Odd Multiplier" and that the "Loevinger" measure mostly disagrees with other measures.

Predicting interestingness scores. To predict the interestingness scores for the predicted links, our proposed method uses topological features. For this experiment, we generated a labeled dataset for each measure as described above.

We applied the M5P regression algorithm [50] to train a model for each dataset. We evaluated the model using 10-fold cross-validation and computed the relative absolute error and root relative error. Figure 3-3 shows the results of the score predictions using the regression model. The results show that the relative absolute error varies from 30% to 85% and the root relative squared error varies from 40% to 140%. Even though the "Information Gain" measure shows relatively less error in predicting the interesting scores, it is still not encouraging. Hence, we conclude that trying to predict the scores directly is not feasible.

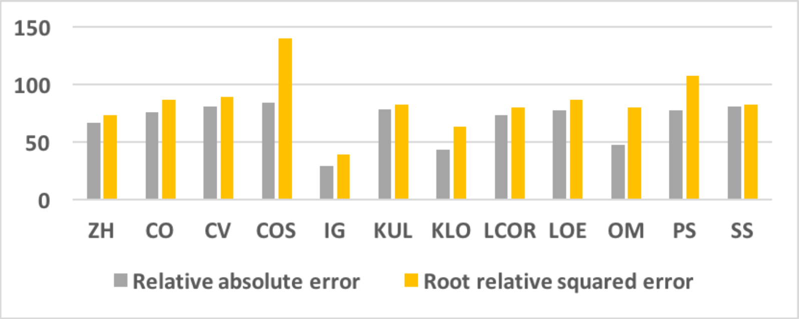

Figure 3-3 Ranking using predicted interestingness

Predicting interestingness ranks. For this experiment, we predicted the rank of the link instead of the actual score. We translated the predicted scores into rankings by ordering the links based on score. We evaluated the similarity of the top 100 instances of actual and predicted rankings using the Kendall's tau measure presented in the previous section.

Figure 3-4 illustrates the Kendall's tau score of the different interestingness measures, which vary from 0 to 0.45. The results show that the "Zhang" measure was able to predict the interestingness rankings of the links. Along with the "Zhang" measure, the "Information Gain", "Piatetsky-Shapiro" and "Klongen" measures showed promising results.

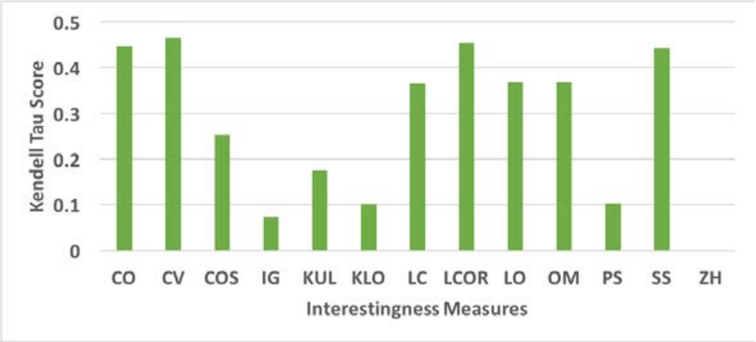

Figure 3-4 Ranking using predicted interestingness

We also evaluated the different values of the top k instances. Figure 3-5 illustrates the Kendall's tau scores when the number of top instances is increased from 100 to 1000 instances. We observed that the score is constant for different numbers of instances.

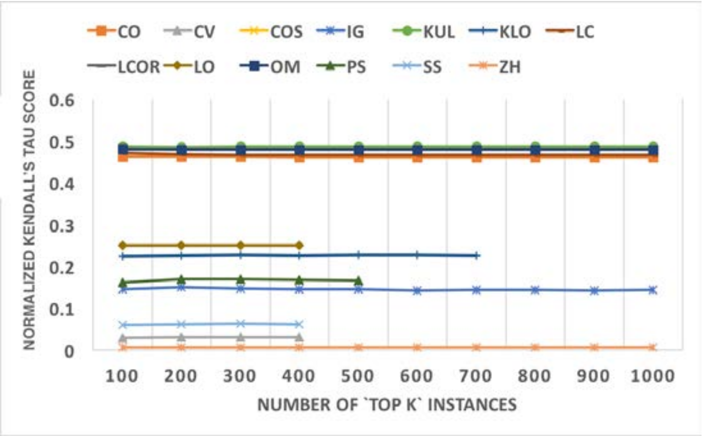

Figure 3-5 Normalized Kendall's tau score for the top k instances varying from top 100 to top 1000.

3.8 Conclusions

This paper proposes a supervised method for ranking predicted links using an interestingness measure. We evaluated the proposed method on MEDLINE data, by ranking future links between two medical concepts. We also studied the correlations between several interestingness measures and found some indication that some of the measures rank links similarly. We show that predicting the ranking of future links using interestingness measures is more precise than predicting the actual score.

A number of avenues are identified for future work. First, this work must be applied to additional domains, such as social networks and citation graphs, in order to test the domain independence of the approach. Second, only thirteen interestingness measures were used. Ideally, we should evaluate the possibility of ranking predicted links using all the interestingness measures found in the literature. Third, we should investigate how well the rankings generated from the interestingness measures correlate with the expectations of domain experts.

4 . Link Prediction using an Incremental Learning Approach

4.1 Introduction

In recent years, the link prediction problem, which concerns predicting the missing relationships in a network, has captured the attention of many researchers. The link prediction problem involves predicting the future association between nodes based on the observed features of links and nodes. Link prediction is studied extensively in relation to social networks, where it is used to predict missing information and to recommend communities, friends or topics of interest. It has also examined in other domains such as e-commerce and biomedicine. In e-commerce [51], link prediction is used to recommend content to users by analyzing the user and item network. In the biomedical domain [2], link prediction helps to predict interactions between proteins and pathways in metabolic networks.

Several link prediction methodologies have been proposed in the literature. Most of these use supervised learning approaches to predict future links [7][17][52]. Katukuri et al. [7] present a comprehensive study of a supervised link prediction model for large-scale concept networks extracted from biomedical literature. Several neighborhood and proximity features are extracted from the large-scale temporal concept network.

An expensive step in link prediction methodologies is the generation and updating of the prediction model using supervised learning. Typically, the prediction model is trained every time new data arrives. These approaches have the following drawbacks:

- The trained predicted model only uses recent data

- Retraining the model using the entire historical information is computationally expensive
- Removing the historical data and training the model only on the new data results in an incoherent model, due to a lack of historical information

Ade et al. describe two traditional incremental learning approaches: *data accumulation* and *ensemble learning* [53]. The data accumulation approach gathers both the new data and previous information to retrain a new model. The ensemble approach generates a new model using the new data and keeps the previous models active for final decision-making based on a voting mechanism. In the accumulation approach, the dataset for training the model grows gradually. The ensemble method entails developing a set of sub-models for each dataset, while a final weighted voting policy predicts the links in the future concept network. The hybrid method uses accumulated data, selected based on the ensemble method. The main goal of our work is to predict links in a future concept network using an incremental support vector machine (SVM) trained by the feature sets extracted from the temporal concept networks. Unlike previous link prediction methods, incremental learning can update the link prediction model online according to a large stream of data. The link prediction model can be adjusted to the new temporal data remarkably fast. Additionally, the incremental learning approach performs as accurately as the computationally expensive, conventional learning approaches. Therefore, fact that the dataset is both vast and temporal is not problematic when the incremental learning approach is employed. This investigation explores three incremental learning strategies that address the link prediction problem: data accumulation, ensemble learning and the hybrid method.

4.2 Related Work

Link prediction has been studied widely in relation to social networks. The problem of link prediction was first introduced by Liben-Nowell et al. [1], who proposed a methodology to predict the missing link in a social network using the network properties of a node pair. Similarly, Tao Zhou et al. [33] and Lei et al. [34] used the network structural features of protein-protein interaction networks to predict the missing links between the proteins. The above methods use individual network features as a predictor to determine the formation of new links. By contrast, supervised methods use different features to predict the future formation of a link. Hasan et al. [17] propose a supervised link prediction method using topological features extracted from the network to build a binary classifier for predicting the possibility of a new link. O'Madadhain et al. [35] propose a joint probabilistic model for link prediction using topological features and node attributes. Link prediction is also applied in LBD, where knowledge is modeled as a network and a hypothesis predicting a link between two knowledge entries is generated. Katukuri at el [7] propose a supervised link discovery method to generate hypotheses from a biomedical concept network. A biomedical concept network was built from MEDLINE dataset using the biomedical concepts that appeared in publications as nodes and the co-occurrence of concepts in the same publication as edges.

Several investigations that address incremental learning from the perspectives of unsupervised approach [54], supervised neural networks [55], genetic algorithm [56], and SVM [57][58][59] report its success on large online data streams. Yang et al. [60] propose an incremental SVM methodology combining classifiers using the voting principle.

4.3 Background

Definition: An SVM is a statistical pattern recognition approach which classifies instances into different classes using decision boundaries with maximum margin (minimum risk, Eq. 1) [61]. The problem of minimizing the risk is equivalent to the Lagrangian optimization (Eq. 2). Additionally, to increase the power of classification for non-linearly separable data, a convenient kernel function can be used to map the data to a larger feature space.

$$F = \frac{1}{2} \min \| w \|^2 + c \sum_i \xi_i \;\; st, \; y_i (w.x_i + b) \geq 1 - \xi_i \tag{1}$$

$$\begin{gathered} Max\, L(\lambda) = \sum_i \lambda_i - \frac{1}{2} \sum_i \sum_j \lambda_i \lambda_j z_i z_j x_i^t x_j^t \, , \\ 0 \leq \lambda_i \leq c \, , \sum_i \lambda_i z_i = 0 \end{gathered} \tag{2}$$

Thus, the hyperplane (which separates two classes) is obtained by using the following equations

$$w = \sum_i \lambda_i z_i x_i \tag{3}$$

$$w_0 = \frac{1}{z_i} - w^t x_i \tag{4}$$

where the non-zero λ_i specifies the support vectors (SV) forming the category boundary. This means that the hyperplane's location is specified via samples that are located close to the decision boundary (SV).

Training time complexity: According to the literature on the popular SVM library LibSVM, the training time complexity of the SVM model is directly related to the number of samples used in the training process [62]. The LibSVM time complexity has been reported as

O (n^3), which means that the redundant data elimination results in a drastic reduction in the training time complexity.

4.4 Incremental SVM

Support Vector Machines are computationally expensive learning algorithm for modeling large-scale data. Therefore, an incremental SVM is required to handle datasets that are a large and dynamic. The rest of this section explains the three incremental SVM approaches used in link prediction.

The accumulation method. Support Vector Machines have proved to perform well with high generalization ability on a vast variety of pattern recognition problems[63]. Due to the success of SVMs in categorizing and summarizing data in the form of support vectors, it is easy to sample large data streams [57][64]. An initial SVM model is trained using the initial dataset. As mentioned before, the location of the optimal hyperplane is only related to the linear combination of support vectors [65]. Thus, the support vectors are effective samples for future learning processes. Various incremental SVM algorithms are based on updating the model using concise and useful data as support vectors [57][66][59]. Syed et al. [57] develops an incremental SVM method whereby only the support vectors at each incremental step are preserved. Therefore, the support vectors of the initial model in combination with the next dataset form a new training dataset. The accumulation method is similar to the standard SVM batch learning process [57]. At each step, the obtained model represents the current data properties and the historical data characteristics through support vectors. Figure 4-1 shows the incremental SVM learning process using the accumulation methodology.

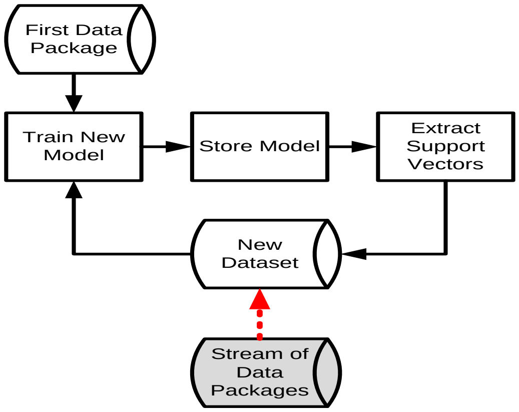

Figure 4-1 Incremental SVM based on accumulation method

The ensemble method. Unlike the accumulation method, the ensemble method does not use support vectors to sample historical data. For each new dataset (p), data are used to train a new model without modifying or retraining the previously trained models. Thus, M sets of SVM models are obtained corresponding to the new dataset batches in a given time duration T. The set of models learning in a given time duration is referred to as a *package* (P). Figure 4-2 shows the procedure for incremental SVM learning using he ensemble method.

Predicting a link between a pair of concept nodes in a concept graph is achieved using a voting policy that considers all the SVM models (P×M models). As the datasets are available at different time intervals, the *P* model sets may have different impacts on link prediction. To address this problem, a weighted voting method is defined in Equation 5 to predict the link with respect to the feature vector *x*

$$L(x) = \sum_{i=1}^{P} \sum_{j=1}^{M} w_i . \left(S_i(x) - C_i(x) \right) \qquad (5)$$

where w_i is the weight given to the model's prediction in package i. S_i and C_i are the numbers of SVM models that have respectively predicted and not predicted the link in the package i. The positive prediction value, $L>0$, indicates that the link exists.

The model's weight (w_i) influences prediction of the future link. To grant more importance to the most recent data, recent models are granted more weight than historical models. In this investigation, we examine three weight functions: uniform, linear, and exponential decay. The uniform function ascribes the same weight to the models. The linear and exponential decay functions ascribe less weight to older datasets. The ensemble voting weights for the data packages obtained from T_1 through T_p are calculated as follows

$$w_i = \begin{cases} 1 & \text{, Uniform} \\ 1-bt & \text{, Linear; } 0 \le bt \le 1 \\ e^{-at} & \text{, Exponential} \\ 0 & \text{, o.w} \end{cases} \qquad (6)$$

where a and b are constant parameters and t denotes the time distance (a decimal number in the range 0 to P) between the current state (P) and the data package arrival time (i) which is obtained by $t=P-i$.

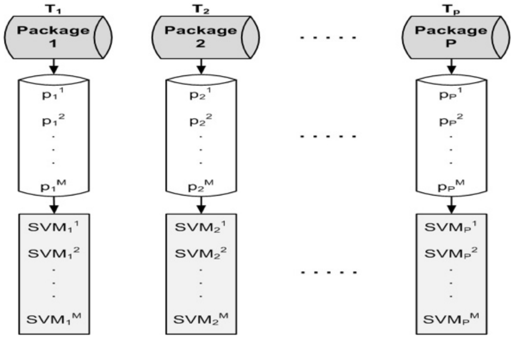

Figure 4-2 Incremental SVM learning process using the ensemble method. The data sets of interval a for time period Tb is represented

The hybrid method. The hybrid method combines the benefits of both the accumulation and the ensemble methods to update the SVM model incrementally. Historical information is sampled using support vectors obtained from preceding individual SVM models. In the hybrid method, each data package contributes to an SVM model containing support vectors. Updating the model at T_P (using data package P) requires an accumulated dataset consisting of the new data package and the support vectors sampled from the previous models.

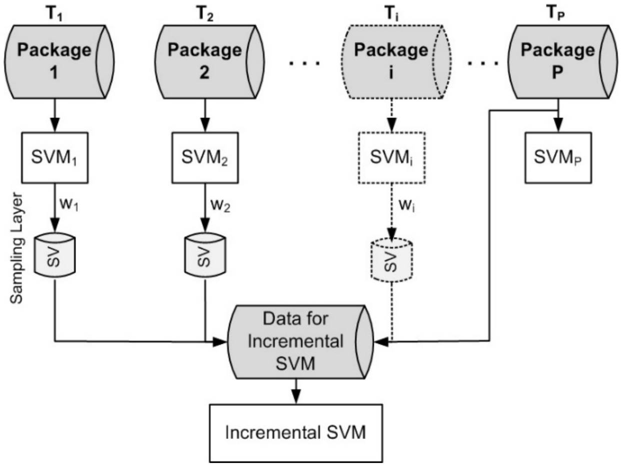

Figure 4-3 Incremental SVM learning using hybrid method. The data packages are available in order of T_1 through T_p.

There are two approaches that can be adopted for sampling support vectors: 1) all the support vectors can be taken into account, or 2) the support vectors of each SVM model can be sampled. The first approach is a specific version of the second one when the sampling rate is set to 1 (uniform function). The sampling rate is analogous to the weight function defined for the ensemble method. As a specific example, at T_3, the accumulated training data includes data-package-3, w_2 portion of the support vectors of the SVM_2, and w_1 portion of the support vectors of the SVM_1. The sampling rates (w_1 and w_2) in the linear and the exponential decay functions are in the range (0, 1) such that $w_2 > w_1$. Figure 4-3 depicts the procedure for incremental learning using the proposed hybrid method.

4.5 Experiments and Results

The proposed incremental SVMs are evaluated using the temporal datasets (say T_1 through T_5) and the measures "accuracy of prediction" and "learning time complexity." In this paper, the learning time complexity ($O(n^3)$) unit is 10^{12} (Tera) instructions (TI).

Data preparation. For these experiments, the MEDLINE dataset is used to create the training datasets. The training datasets are prepared using the procedure discussed in Chapter 2. Table 4-1 shows the datasets and the corresponding time durations.

Table 4-1 Datasets used in the experiments.

Time Interval	**Training Snapshot**	**Labeling Snapshot**
T1	1991-1993	1994-1996
T2	1994-1996	1997-1999
T3	1997-1999	2000-2002
T4	2000-2002	2003-2005
T5	2003-2005	2006-2008

Baseline experiment using batch mode learning. To assess the performance of the incremental SVM, we conduct a baseline experiment using the conventional SVM model (batch mode learning) for the link prediction problem. In batch mode learning, at any given time period, the data instances are accumulated from all previous time periods. We use 10-fold cross-validation to evaluate the performance of the prediction model. The model's performances metrics are shown in Table 4-2 along with the time complexity of learning the models. This table reports the performance of the conventional SVM learning model, which re-trains all the data instances to learn the link prediction model. The learning time complexity increases up to 66.4 TI and no significant improvement in accuracy is observed.

Experiments for the accumulation method. Initially, an SVM model is trained by the first dataset at T_1 and is then updated by the new data package at T_2 using the accumulation method. This is repeated for the datasets at T_3 and T_4. Although the model's size gradually increases, it is smaller than the model trained by the batch mode learning. Table 4-3 shows the model's accuracy and time complexity of learning in increments. As observed, the accuracy is close to the batch learning model. However, the learning time complexity of the accumulation method at T_4 (when all the data packages are available) is 2.7 times less than the batch mode learning.

Table 4-2 The link prediction performance reported by the conventional SVM model

Time interval	Positive Recall	Positive Precision	Accuracy	Time complexity (TI)
T1	73.3	72.2	72.1	1.0
T1-T2	77.6	71.5	72.6	8.3
T1-T3	79.3	70.3	72.3	28.0
T1-T4	71.8	74.6	73.7	66.4

The initial results show the success of the incremental learning approach, which preserves only a subset of the historical information for the model adaptation. However, one drawback of this model is the large number of support vectors extracted from the trained model (about half the data package). Therefore, updating the model is still time-consuming. To make the learning process faster, two approaches based on the ensemble method and the sampling method are introduced in the following sections.

Table 4-3 The link prediction performance reported by the incremental SVM model using the accumulation method.

Time interval	Recall	Precision	Fall out	Accuracy	Time complexity
T1	73.3	72.2	29.4	72.1	1.0
T1-T2	77.4	71.6	32.5	72.5	4.6
T1-T3	80.1	70.1	35.9	72.1	11.2
T1-T4	72.1	75.2	23.7	74.2	24.1

Experiments for the ensemble method. In this method, we demonstrate:

- The potential of the ensemble approach for decision-making
- The impact of historical data on the link prediction problem.

The data in each time period was divided into 5 sub-datasets for training the corresponding SVM models referred to as a package. Each sub-dataset is used to generate the model; the learning time complexity is much less than for previous methods. However, the prediction phase entails more computations to find the most likely predicted link. To vote among the models, we use three weight functions: uniform, linear decay and exponential decay.

Uniform weight function. This method ascribes the same weight to all the historical information, which is represented by individual SVM model sets. Table 4-4 shows the ensemble method's performance using the uniform weight function.

Linear decay weight function. Using the linear decay weight function (Equation 6), the most recent data package is assigned the largest weight (1) and the oldest package is assigned the smallest weight (0.25).

Table 4-4 The link prediction performance reported by the incremental SVM model using the ensemble method and the uniform weight function.

Data	Model set weights	Accuracy	Time complexity
T1	1	71.6	0.01
T1-T2	1, 1	71.5	0.01
T1-T3	1, 1, 1	71.6	0.01
T1-T4	1, 1, 1, 1	72.1	0.01

Table 4-5 shows the ensemble method's performance using the linear decay weight function. The results reveal that current data has more impact on future links than historical information does. Therefore, very old information can be eliminated to make the training phase faster.

Table 4-5 The link prediction performance reported by the incremental SVM model using the ensemble method and the linear decay weight function

Data	Model set weights	Accuracy	Time complexity (TI)
T1	1	71.6	0.01
T1-T2	0.75, 1	71.7	0.01
T1-T3	0.5, 0.75, 1	71.7	0.01
T1-T4	0.25, 0.5, 0.75, 1	72.5	0.01

Exponential decay weight function. As mentioned above, the most recent data has the greatest effect on link prediction. The exponential decay weight function computes a sharp weight reduction trajectory. To assess the impact of historical information, eight exponential decay functions obtained by Equation 10 with parameters *a= {0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2}* were examined. Figure 4-4 shows the weight functions calculated by

Equation 10. Table 4-6 shows the performance of the ensemble method using eight exponential decay functions.

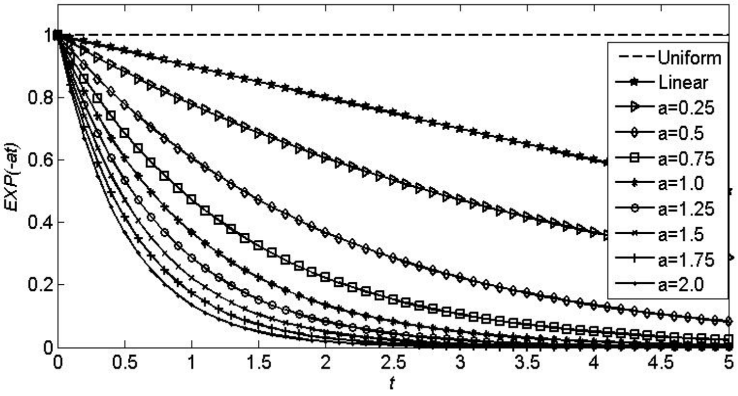

Figure 4-4 Uniform, linear, and exponential weight functions used in the voting policy for the ensemble method.

Table 4-6 The link prediction performance reported by the incremental SVM model using the ensemble method and the exponential decay weight function

Data	**Accuracy (%)**							
a	0.25	0.5	0.75	1	1.25	1.5	1.75	2
T1	71.6	71.6	71.6	71.6	71.6	71.6	71.6	71.6
T1-T2	71.7	71.7	71.6	71.6	71.6	71.6	71.5	71.5
T1-T3	71.7	72.1	71.8	71.8	71.8	71.8	71.7	71.6
T1-T4	72.5	72.5	72.7	72.7	72.7	72.7	72.5	72.5
Average	71.9	72.1	72.0	72.0	72.0	72.0	71.9	71.8

Table 4-6 shows that the exponential decay functions where the parameter *a* is in the range (0.5, 1.5) perform better. This range mostly concerns the current data package. However, it neither ignores historical information (a≥1.75) nor relies heavily on historical information (a≤0.25).

To compare the weight functions used for the voting, Table 4-4 and Table 4-5 show detailed and average accuracy rates of the ensemble method using the uniform, linear decay, and exponential decay weight functions. The exponential decay function where the parameter *a* was close to 1 outperformed the other weight functions.

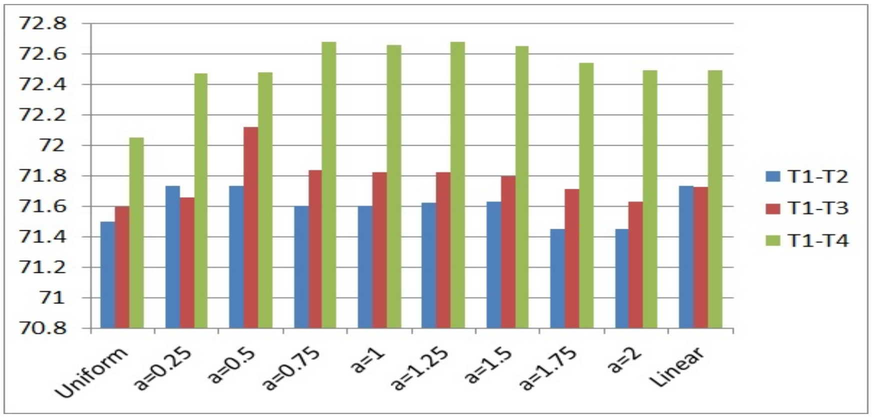

Figure 4-5 Accuracy of the ensemble method using the uniform, linear decay and exponential decay weight functions for voting. The results were exhibited for two, three, and four temporal data packages.

Experiments for the hybrid method. The ensemble method was fast learning and reported classification accuracy comparable with both the accumulation method and the baseline experiment. However, in the prediction phase voting among the different models was computationally more expensive than using a single model. Additionally, we learned from the ensemble method that link prediction is heavily influenced by the most recent data. The hybrid method uses both the accumulation approach and the ensemble approach. Similar to the accumulation approach, the hybrid method uses the support vectors obtained from the previous models and samples them in a similar way to the ensemble approach.

Table 4-7 shows the hybrid method’s performance with linear decay sampling. Table 4-8 shows the performance of the hybrid method using exponential decay sampling. The results show that the hybrid method is remarkably efficient and its accuracy is close to that of batch mode learning. The learning process of the hybrid method at the final step (T_1-T_4) is 25.5 times faster than that of batch mode learning. Furthermore, the hybrid method performance is slightly better than that of the other approaches.

Table 4-7 The link prediction performance reported by the incremental SVM model using the hybrid method and linear decay sampling.

Data	Recall	Precision	Fall out	Accuracy	Time complexity
T1	73.3	72.2	29.4	72.1	1.0
T1-T2	76.7	71.5	32.3	72.3	3.4
T1-T3	79.8	70.0	35.9	72.0	6.0
T1-T4	71.4	74.7	24.2	73.6	7.8

Figure 4-6 illustrates the accuracy rates and learning time complexities (components of the performance) reported for batch mode learning, the accumulation method, and the hybrid method. This figure shows the power of support vectors for sampling historical information.

Table 4-8 The link prediction performance reported by the incremental SVM model using the hybrid method and exponential decay sampling.

Data	Recall	Precision	Fall out	Accuracy	Time complexity
T1	73.3	72.2	29.4	72.1	1.0
T1-T2	77.0	71.2	33.0	72.1	2.0
T1-T3	79.3	70.2	35.5	72.0	2.4
T1-T4	72.3	74.4	24.9	73.7	2.6

Figure 4-6 shows that the hybrid model predicts the links between concept nodes as accurately as the conventional SVM (which uses all the information). However, the learning time complexity of the hybrid model is dramatically lower. Therefore, we achieved a desirable accuracy rate using the hybrid model, for which the training data only contained a portion of the previous data packages. Thus, the incremental SVM was deemed an efficient model.

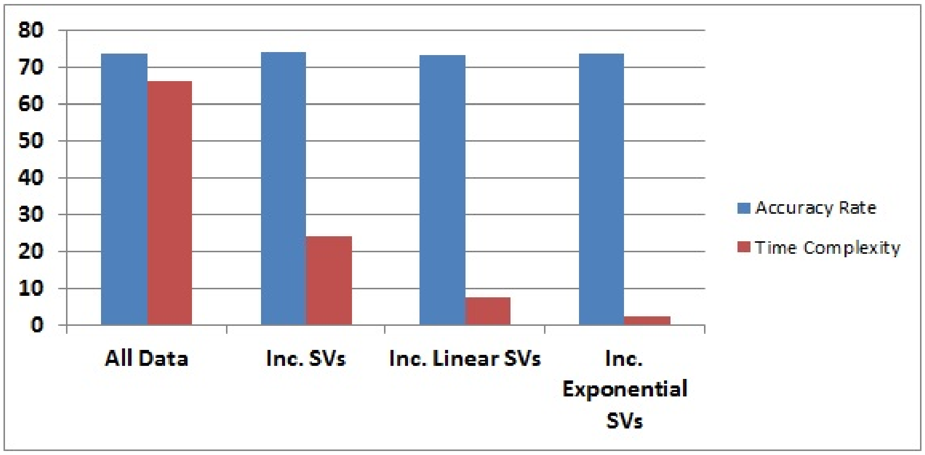

Figure 4-6 Performance of batch mode learning (all data), accumulation method (including support vectors), and two hybrid methods using linear decay (including linear support vectors) and exponential decay (including exponential support vectors) sampling in terms of accuracy rate and learning time complexity

4.6 Conclusion

This chapter proposes applying an incremental learning approach for the link prediction problem to the network concepts extracted from the biomedical literature. Specifically, three approaches to incremental SVM learning are proposed: 1) the accumulation method, 2) the ensemble method, and 3) the hybrid method. The accumulation method uses preserved information (as support vectors) and new data to update the model. The ensemble method creates a series of sub-models and predicts the link based on a voting system across sub-models. The hybrid method combines the benefits of the two previous methods, in which the support vectors are sampled and accumulated for updating the SVM model. The support vectors in the hybrid method are sampled using a weight function (linear decay or exponential decay) to represent historical information such that the recent data has a greater effect on link prediction.

The experiment results show that historical information sampling using exponential decay weights outperforms the other methods. Furthermore, the accuracy of the incremental SVM was comparable to the computationally expensive model trained by the conventional SVM. Our future work includes using the incremental learning approach for large streaming datasets.

5 . Link Discovery Using Iterative Link Classification

5.1 Introduction

In recent years, mining complex networks, known as *graph mining*, has drawn significant attention in the area of data mining. Due to its capacity for representing relationships in data and the theoretical and algorithmic advances in graph theory, graph mining has gained relevance in several domains including biomedicine, social media, and e-commerce. It offers a way of understanding relationships among entities in data. Link prediction, the predicting of a possible future association between any two entities in a network that are not currently associated, is one of the most common graph mining techniques used to predict friendship or follower relationships in social network data. Link prediction has been adapted for various other applications, such as predicting protein interactions in protein-protein interaction networks [67], identifying spurious links caused by inaccurate or incomplete information in networks [68][69], and as a collaborative filtering mechanism in e-commerce [70].

Link prediction generally uses various types of network structure information to predict the relationships among the entities in the network. Some studies have included semantic and domain-specific information as well as topological measures to predict links. In both cases, link prediction methodologies have proved reasonably accurate in forecasting unknown relationships in networks based on the past or present topological structure of networks.

Link prediction can also be considered a way of predicting the future state of an evolving network by predicting new links. This provides an opportunity to work on the evolved network before it actually evolves. In this work, we introduce an iterative classification-based link prediction methodology that propagates knowledge of predicted links by adding these links to the network.

The hypothesis underlying our approach is that if adding new information, in this case predicted links, to a network changes the neighborhood of a node pair, then this can increase the proximity (closeness) between nodes that were weak in the first place. Let us take social networks as an example. Social networks evolve rapidly through the addition of new relationships, which changes the topology of the subgraph that includes the users connected through these new relationships. Inferring a new relationship between user A and user B can increase the probability of new links forming between the friends of users A and B. This premise can be applied to several other networks, including user-item networks in recommendation systems and security analysis in covert networks. In recommendation systems, if we predict that a user will buy an item, we can recommend more items that may not have been possible to relate to the user without that initial prediction. In a covert network, if we can predict the interaction between persons using an iterative approach we can infer relationships between their associates.

An initial premise behind our iterative classification-based approach is that only predictions made with initially with high confidence should be considered candidates for addition into the network as actual links. This helps to predict more promising links in the network in subsequent iterations. Such an iterative method improves the possibility of predicting the relationships that are not possible to identify from the structure of the network in an initial iteration. In an iterative classifier scheme, a classifier is used iteratively to assign labels to unlabeled links and is trained using features and labels of known nodes. Our experimental results prove that iterative link prediction improves the accuracy of link prediction.

In this paper, we propose a link-propagation-based, supervised approach to boost the performance of link prediction. In this approach, a model is trained using the structural features of the network and then, for test purposes, applied to node pairs that are not associated in the updated network. The relations predicted with high probability are accepted as valid and added to the data as *known* links. The updated network is used to compute the features for next iteration. We expected that iteratively applying the link prediction algorithm would increase the possibility of predicting relationships that were missed in the previous prediction step. The evolution of the network involves both the addition of new links and the removal of existing links. Therefore, we wanted to investigate the possible effects of updating the graph by removing links that we predicted would disappear as well as adding predicted links. We also proposed a supervised approach for predicting links that would disappear using the topological measures extracted from the network. We evaluated the impact of links disappearing on our iterative model by using both the model for predicting disappearing links and the link prediction model to update the graph in each iteration.

To the best of our knowledge, ours is the first work to propose a link-propagation-based, iterative classification approach for predicting new links. The rest of the paper is organized as follows. Section 5.2 addresses recent studies on link prediction and iterative classification. Our approach is explained in detail in Section 5.3. Section 5.4 describes the experimental setup. The results of the hypothesis testing using MEDLINE data are discussed in Section 5.5. In Section 5.6, we discuss how the results of our iterative classification approach might be adapted for different domain networks, including co-authorship networks and protein-protein interaction networks. Finally, we present our conclusions in Section 5.7.

5.2 Related Work

Iterative classification. Iterative classifier algorithms have become an important research topic in the area of node classification. The basic idea behind iterative classification is that the node labels are predicted iteratively: the classifier makes an inference and uses that inference to make further inferences. Chakrabarti et al. [71] propose a method for classifying hypertext into a topic hierarchy. This model uses the local text in the document and the text from the related neighbor objects to improve the accuracy of classification. Neville et al. proposes an iterative classification method, updating entities' attributes iteratively and making inferences about entities in relational data [72]. The experiment results showed a significant increase in accuracy when the inferences made in the initial iteration are fed back. Heß and Kushmerick [73] propose an iterative classification for relational data that is similar to the method proposed by Neville et al. [72]. Unlike the approach proposed by Neville et al. [72], their method built separate classifiers for intrinsic and extrinsic features and used an ensemble approach to make a decision. In the process of identifying covert sub-networks, Galstyam and Cohen [74] propose an iterative node classification algorithm to classify groups of individuals within a large population.

Link Prediction. Liben-Nowell and Kleinberg[1] first introduced the concept of predicting new links in a network using the topological features of the network as prediction measures. Experiments demonstrated that several individual topological features outperform random predictors in predicting future links. Hasan et al. [17] propose a supervised link prediction method; instead of using an individual predictor, they propose a binary classifier which uses all the topological features as feature set. This follows several feature-based classification methods that use topological features[75][76], social features [77], and node attributes[78]. Several probabilistic methods use a probabilistic graph model, where each node pair is assign a probability based on their similarities and a transition probability in random walk[5][79]. Kashima et al. [80] propose a semi-supervised link prediction algorithm based on node information, which uses label propagation for predicting links in a network. To apply label propagation, they model a node pair into triplets (node, node, type). They predict the missing information in the triplets by applying the label propagation method on them. One of the major differences between this and our approach is that we propagate predicted link information in every iteration rather than propagating node attributes or labels to predict the similarity. The purpose of our work is not just to predict the links in a static network but also to use these predictions to predict relationships in future evolved networks. In the following section, we discuss the methodology of link-propagation-based prediction.

5.3 Iterative Link Classification

In this section, we introduce an iterative-based approach to predicting future associations among nodes in a network. Given that a new link changes the topological features that were originally used for training the model, it is possible that using topological features along with the new predicted links can help improve the predictive performance of a link prediction model.

Understanding iterative link classification. A supervised link prediction method uses the topological features from a current snapshot of a graph. With dynamic or evolutionary graphs, a new link changes the topological features of the graph. In some cases, these new links can significantly change topological features such as distance, centrality, neighborhood, etc. An iterative link classification approach uses these new topological features that are generated by adding the predicted links in the graph. Therefore, one of the necessary conditions is that the addition of new links changes the topology of the network.

With iterative link prediction, we use “predicted links” (i.e., links classified as true in the prior iteration) as actual links in the new iteration. This introduces an element of uncertainty because these links are only predicted and may or may not appear in future. Therefore, it is important that only links that are classified as positive with high confidence are added. The predictive performance of the base classifier is also important. It is vital to select a sound base classifier and to only included predicted links with high confidence to reduce error accumulation in an iterative classification model.

Methodology. Algorithm 1 presents the iterative link classification approach for predicting the formation of links in a network. The network is denoted by $G = (N, E)$, where N is a node set and E is an edge set. The training dataset (T) and the testing dataset (τ) consist of node pairs that are not connected in the current state of the network. Each instance in the datasets consists of the topological similarities between the node pair along with the known label.

Training model. An *initial model* is trained to predict the possibility of link formation using the training dataset (step 1). We train the model once and use it iteratively to predict the links.

```
Input: Graph $G$, training dataset ($T$), testing dataset ($\tau$), Number of iteration ($N$)
    1. Train classifier using $T$.
    2. At each iteration $i$:
        a. Compute the features for $\tau$ from $G$
        b. Apply classifier on $\tau$ to predict links
        c. Update $G$ with high confident links ($L$)
    3. Repeat Until $i <= N$ or $|L| > 0$
```

Algorithm 1: Iterative link classification

Feature extraction. A set of features (Φ) is extracted from the network (Step 2a). The feature set is composed of both neighborhood and proximity features:

- **Common Neighbors**[29]: This is a simple and direct approach to measuring the similarity between two nodes. Let the two nodes be s and t. The number of neighbors that are common to them represents the similarity. The similarity score is defined as

$$Score_{CN}(s,t) = |\tau(s) \cap \tau(t)|$$

- **Adamic/Adar** [30]: The Adamic and Adar measure is like the common neighbors measure but weighs the common neighbors:

$$Score_{AA}(s,t) = \sum\nolimits_{z\epsilon\tau(s)\cap\tau(t)} \frac{1}{\log|\tau(z)|}$$

- **Jaccard Coefficient** [31]: The Jaccard coefficient is commonly used in information retrieval to measure the similarity between two sets. It measures the probability that a selected node from the union of two neighborhoods is in both neighborhoods.

$$Score_{JC}(s,t) = \frac{|\tau(s) \cap \tau(s)|}{|\tau(s) \cup \tau(s)|}$$

- **Preferential Attachment**[29]: Preferential Attachment models the growth of networks on the presumption that the node with the highest number of neighbors has the highest chance of being part of the newly formed edge. In our case, the score between two nodes is calculated as:

$$Score_{PA}(s,t) = \ |\tau(s)|.\,|\tau(t)|$$

- **Cycle-Free Effective Conductance (CFEC)** [32]: The cycle-free effective conductance (CFEC) measures the proximity between two nodes in a network. The CFEC between the node s and t is formulated as

$$Score_{CFEC}(s,t) = \ deg_s\,.P_{cf.esc}(s \to t)$$

where deg_s denotes the degree of node s. The cycle-free escape probability $P_{cf.esc}(s \to t)$ is the probability that a random walk beginning at s will reach t without visiting any node more than once.

$$P_{cf.esc}(s \to t) = \sum\nolimits_{r \epsilon R} Prob(r)$$

where R is the set of simple paths between the nodes s and t. In a random walk, the probability of a path P ($Prob(P)$), where the transition probability from node *i* to node *j* is $p_{ij} = \frac{w_{ij}}{deg_i}$, is defined with the following equation:

$$Prob(R) = \prod_{i=1}^{N} \frac{w_{v_i v_{i+1}}}{deg_{v_i}}$$

Automatic labeling. uses both the current snapshot of the network at time t (G_f) and the network snapshot at time t+1 (G_s) to generate the labels for the training set. A node pair, which is not connected in Gf, is labeled as positive or negative depending on whether the link in G_s is strong or weak, respectively. The strength of the connection is determined as strong or weak based on the user-defined parameters *minimum support(δ)* and *margin(η)* [7]. The edge weight in G_s is represented by S:

- Connection is strong if $S \geq \delta$
- Connection is emerging if $\eta \times \delta \leq S < \delta$
- Connection is weak if $S < \eta \times \delta$ (or) $No\ Connection$

The training model is built only once during the initial phase, and this model is used during the subsequent link prediction phases for the updated graph. The process for iteratively updating the graph is presented in the next sub-section.

Classifying the test data. The initial trained model is used to classify the unknown relationships among node pairs in the testing dataset (step 2b). To classify the test instances, we use the features extracted in step 2a.

Identifying confidence links and updating the graph. Using the model, we classify several links in the test dataset as predicted links. Among these predicted links, a set of links with high confidence (L) are selected to add to the original graph as edges (Step 2c). High confidence predicted links are added to the graph. Updating the graph with the new predicted links changes the topology of the graph, which, in turn, modifies the features for every pair of nodes.

Termination condition. Step 2 is repeated until the termination condition is satisfied. The iterative prediction process is continued until there are no new links predicted to update the graph. It is not always guaranteed that this condition will be met. In such cases, we may choose to iterate for a fixed number of iterations.

We propose two setting variants depending upon how the test dataset is handled in each iteration: *selective* and *complete* settings. Selective settings use instances that are not classified as predicted in the previous iteration. By contrast, complete settings consider all instances in every iteration, irrespective of whether they are classified as predicted links.

Supervised method for predicting disappearance of links. Most networks evolve through the addition and deletion of links. To better forecast the evolution of a network, we need an approach that predicts the disappearance of existing links as well as the formation of new links. In this section, we propose an approach that uses the topology of the network to predict the disappearance of links.

Similar to the link prediction method, we predict which links will disappear using features extracted from the network and model it as a classification problem. We extract the homogenous feature set for the node pairs in the training dataset as presented in the link prediction methodology. In order to generate the labels for the training data, we compare the current snapshot with the consecutive snapshot. For example, G_f and G_s are two snapshots generated from two consecutive time periods. G_f is used to extract features and G_s is used to generate the labels for the training data.

In the training data, we consider the node pairs that are connected in the current snapshot (G_f) but may or may not continue to connect in the next snapshot (G_s). Labels are generated depending on the status of the connections in G_s. If the link is present in G_f and continues to be present in G_s, we label it as *"Stay"* and if it is not present in G_s, we label it as *"Disappear."*

5.4 Experiment Setup

Datasets considered for experiments. We evaluated the iterative links prediction using 3 datasets. These are presented below.

MEDLINE Dataset. We generated the concept graph using the freely available published literature dataset from MEDLINE [13]. The MEDLINE dataset is curated by National Institute of Health and is available in XML file format. The dataset contains metadata about the publications, which includes information on authors, publishing dates, document ID (PMID), and keywords. The keywords are from MeSH (Medical Subject Headings) [28], a medical thesaurus curated by the National Library of Medicine. In a concept graph, we consider the keywords (medical concepts) as nodes and the co-occurrence relationships between the concepts as edges. The total number of publications a concept appears in is represented as the node weight and the number of documents in which any two concepts (nodes) are mentioned together is represented as the edge weight between concepts.

For our experiments, we generated several snapshots of a graph using the literature published between the years 1991 to 2000, with each snapshot containing 3 years of data. Using these snapshots, we created training datasets as described in the supervised link prediction section. We divided each training dataset into five balanced datasets referred as d_0 to d_4.

Co-authorship dataset. The co-author network represents the collaboration of authors who publishing articles. In a co-author network, a node represents an author that has published at least one article and a collaboration between authors is considered an edge. The number of articles published by an author is consider a node weight, and the number of articles two authors publish together is considered an edge weight.

In this work, we generated the co-author networks from the Digital Bibliography & Library Project (DBLP) dataset for the period 1981 to 2010. We created multiple networks, each with duration of 10 years.

Protein-Protein Interaction Dataset. Protein-Protein Interaction networks are key for understanding the interactions between proteins. We generated our protein interaction network from Search Tool for the Retrieval of Interacting Genes/Proteins (STRING) database, which is one of the most extensive collections of known and predicted protein interaction data. The known sources are based on high-throughput lab experiments and knowledge from various databases such as Kyoto Encyclopedia of Genes and Genomes (KEGG). The interactions between the proteins are based on different prediction methods, including genomic context predictions and automated text mining. The dataset contains the pair of proteins and their scores based on evidence of their interaction derived from the corresponding data source. The scores range from 0-1000 based on the confidence of interaction, 0-150 means no or low confidence of interaction, and 900-1000 denotes the highest confidence of interaction.

Experiments. We designed several experiments to assess the performance of our iterative link prediction methodology:

- The performance of the model was evaluated based on accuracy, precision, and recall. We compared the performance of the typical supervised link prediction with our iterative algorithm.

- To evaluate the effect of the quality of the link added to network on the overall performance of the iterative algorithm, we filtered the links propagated to the next iteration using their classification probabilities.
- We evaluated the proposed link prediction algorithm using two different learning approaches to build the training model: a C4.5 decision tree and a support vector machine (SVM). The rationale behind this evaluation was to study the effectiveness of our link prediction algorithm for different machine learning approaches.
- We performed several experiments using knowledge of both link formation and the future disappearance of links to assess the impact of disappearing links on the performance of the iterative link classification model.

Experimental setup. Our supervised link prediction uses the graph's topological features to learn and predict the formation of new links. Initially, we build a model (m_0) using the dataset (d_0) to predict the formation of new links. Using this model, we iteratively perform the link classification on the datasets d_1 to d_4. The final evaluation metrics are the average of the individual performance metrics of all the data sets.

The experiments are designed to evaluate the performance of the iterative link classification model compared to traditional link prediction. The results for the initial iteration are considered the results of a traditional link prediction model.

We applied a C4.5 decision tree on the training dataset to generate a classifier model. We used evaluation metrics classification accuracy, precision, recall, and F1 score for evaluating the performance of the prediction models.

5.5 Experiment Results and Discussion

Performance boost with iterative link prediction model. We evaluated our algorithm by comparing it with typical supervised link prediction. We consider five subsets of labeled data. In each experiment, we considered a subset for building a classifier and applied the iterative method on the remaining subsets. The evaluation metrics were calculated by averaging the results across experiments performed on multiple subsets of data.

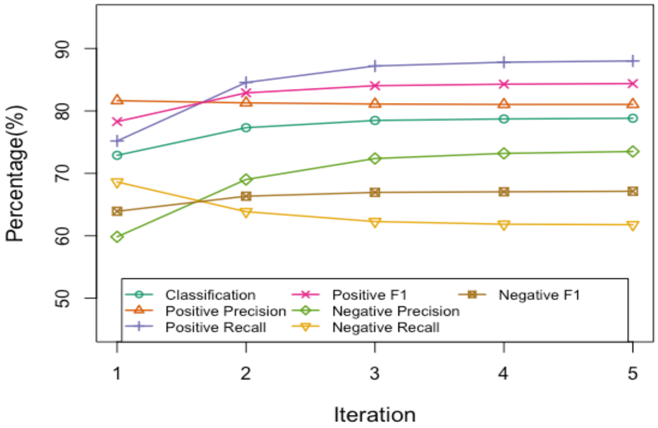

Figure 5-1 Classification accuracy, precision, and recall for the iterative algorithm in selective setting

The classification accuracy, precision, and recall of the iterative classification approach is presented in Figure 5-1. As can be observed, the results show an increase in the accuracy of the prediction model from 72.89% for the initial iteration to 78.82% for the fifth iteration. This is due to the increase in the number of relevant links predicted (positive recall) from 75.18% to 88.18%. The increase in the recall after the first iteration suggests that our approach is able to predict more links based on the links during the first iteration.

We repeat the same experiment for the complete setting. The results, shown in Figure 5-2, demonstrate no change in performance between the selective and complete settings.

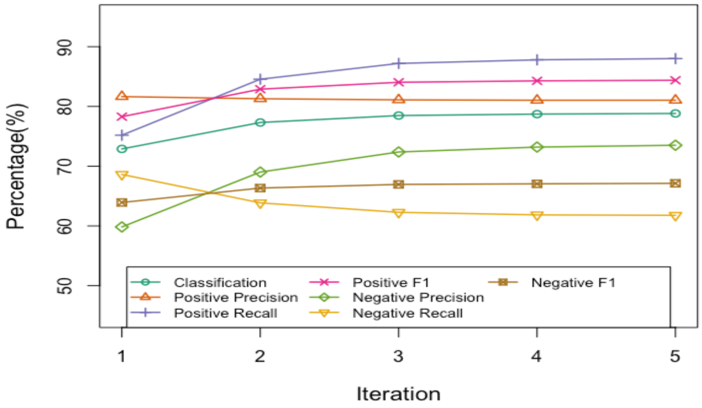

Figure 5-2 Classification accuracy, precision, and recall for the iterative algorithm in the complete setting.

The confidence of added links on performance. This experiment is designed to evaluate the impact of confidence of links added to the graph on the overall performance of our iterative prediction model. We used class membership probabilities (CMP) to identify high confidence links in every iteration, which were then used to update the graph. In general, the classifier assigns a class label to an instance if the CMP of that class label is greater than 50%. We use the CMP given by the classifier as the confidence.

In this set of experiments, we repeated the iterative link classification by varying the CMP threshold, which represents a confidence level from 30% to 90%. The results of the iterative link classification in both the selective and complete setting are plotted in Figure 5-3 and Figure 5-4, respectively. In the selective setting, the addition of low confidence links to the graph resulted in a decrease in positive precision and an increase in positive recall compared to the 50% CMP baseline. Adding lower confidence links or noisy links into the network had a negative effect on the topology of the network, which strengthened the features of most of the positive and negative instances. This resulted in a model biased towards positive labels which increased false positives and reduced negative recall.

By contrast, a higher CMP threshold led to a reduction in the number of links added to the graph. Adding fewer links to the network did not affect the topology sufficiently to predict new links in subsequent iterations. Results show that a confidence threshold of 50% to 70% led the best link prediction performance.

In the complete setting, the recall and precision were not affected by low confidence links, which suggests that the selective method is affected more by noisy links than the complete method.

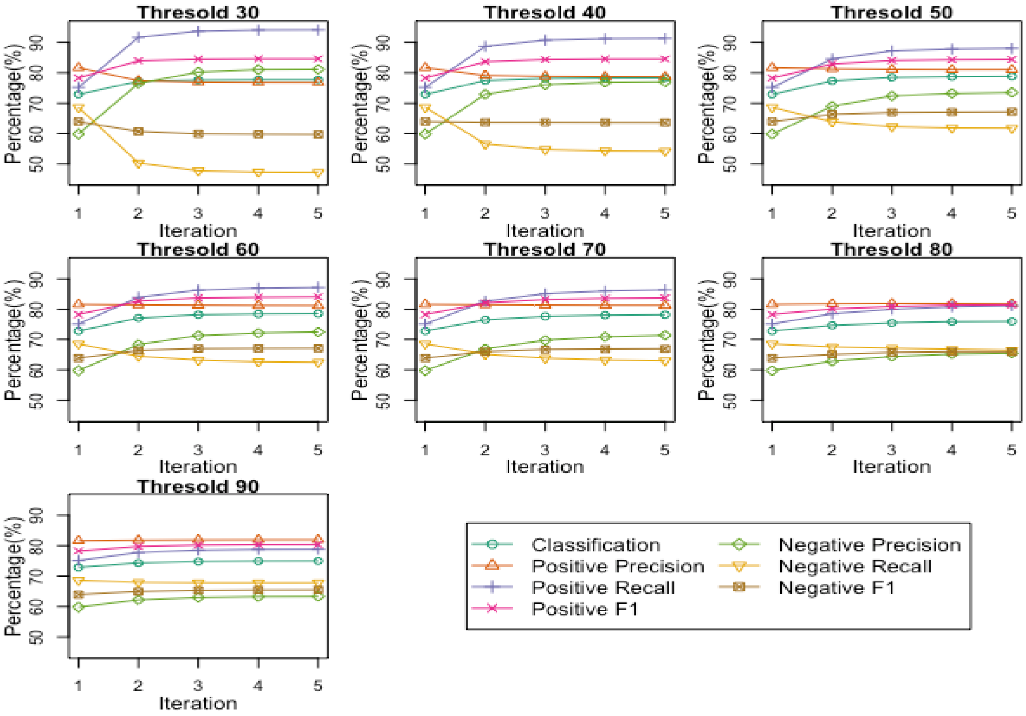

Figure 5-3 Performance of our prediction model, varying the confidence threshold from 30% to 90% in the selective setting

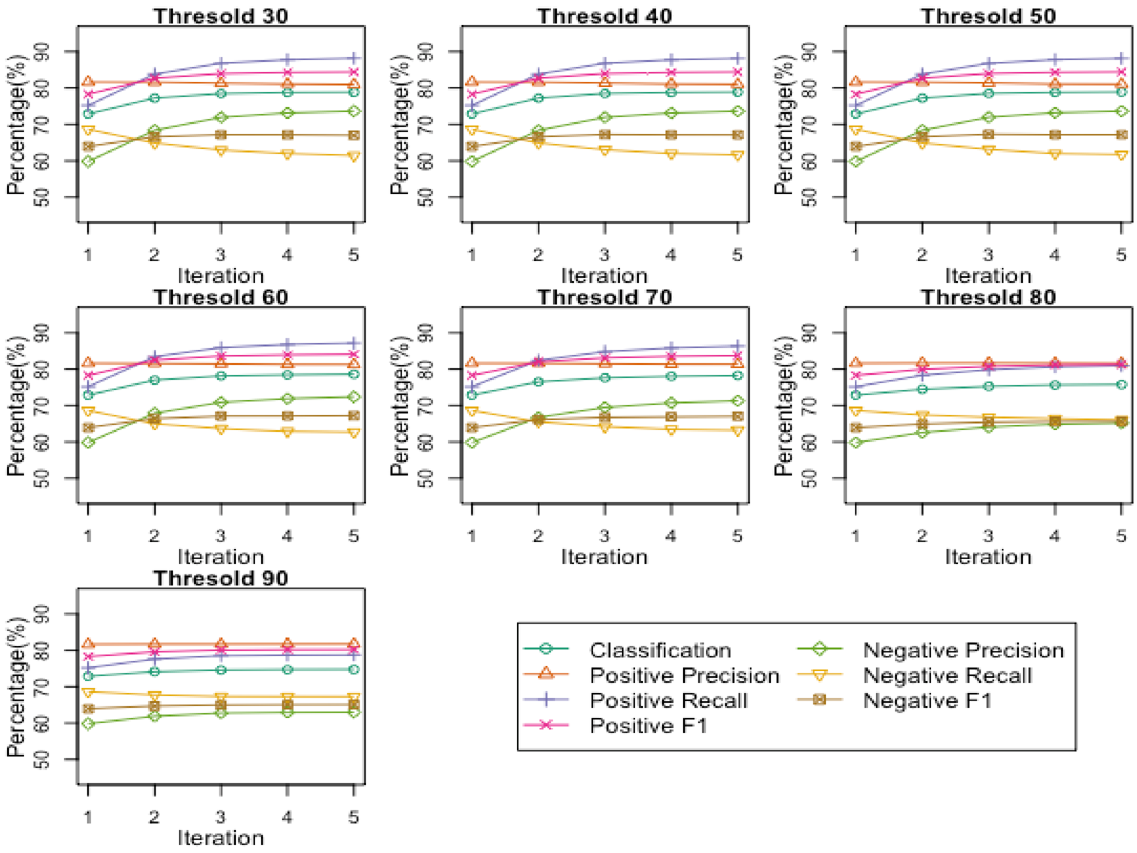

Figure 5-4 Performance of our prediction model, varying the confidence threshold from 30% to 90% in the complete setting

Disappearing links. For each experiment, we used two models: one to predict the links that will form in the future and one to predict the links that will disappear in future. At each iteration, we update the graph by adding the links predicted and removing the links predicted to disappear from the network.

Before incorporating the disappearing link prediction to our link prediction approach, we test the performance of the proposed supervised disappearing links model alone. The experiment tests for performance metrics, including accuracy, precision, recall, and F1 score. The results are plotted in Figure 5-5. Using our supervised model, we achieved an accuracy of 72.89% when predicting the disappearing links in a network.

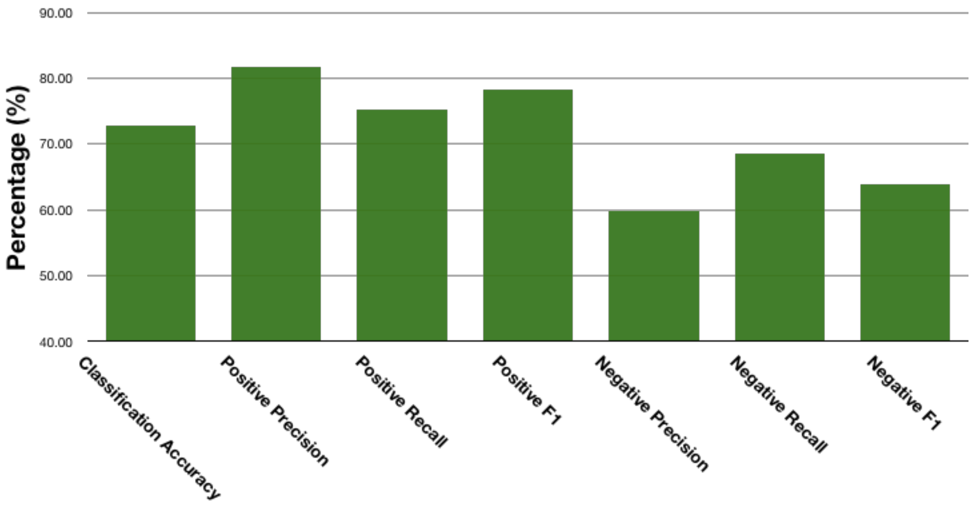

Figure 5-5 The performance metrics for the supervised method for predicting disappearing links.

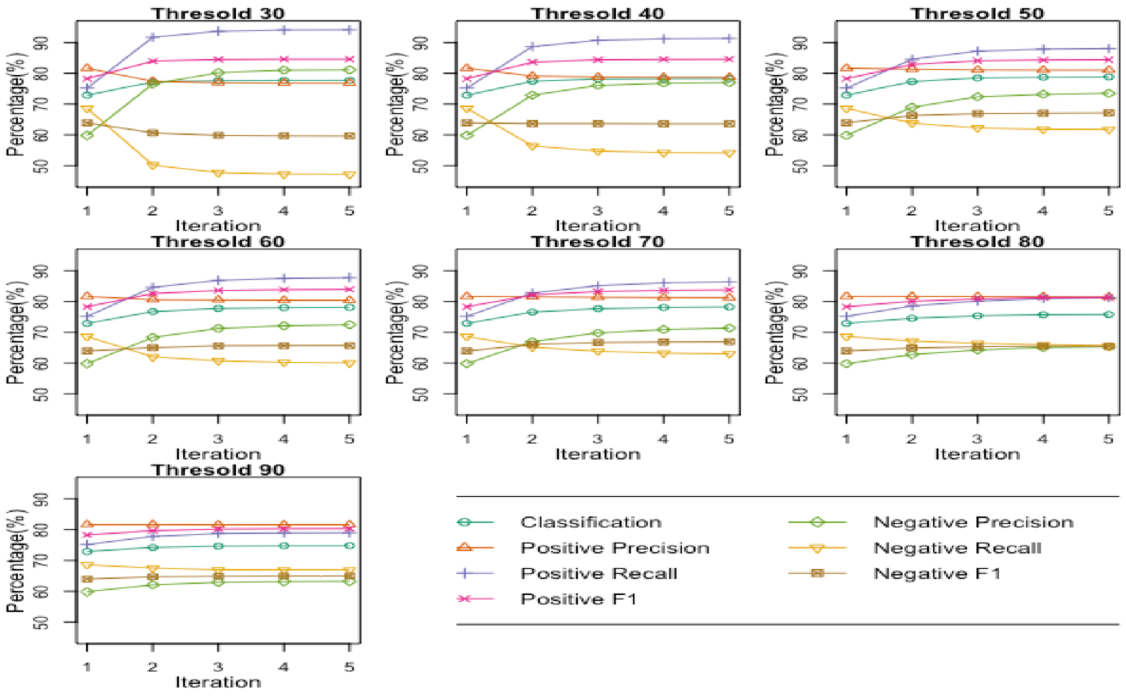

Figure 5-6 Performance of our prediction model, varying the confidence threshold from 30% to 90% in the selective setting. We consider the links predicted and those predicted to disappear to update the graph in every iteration.

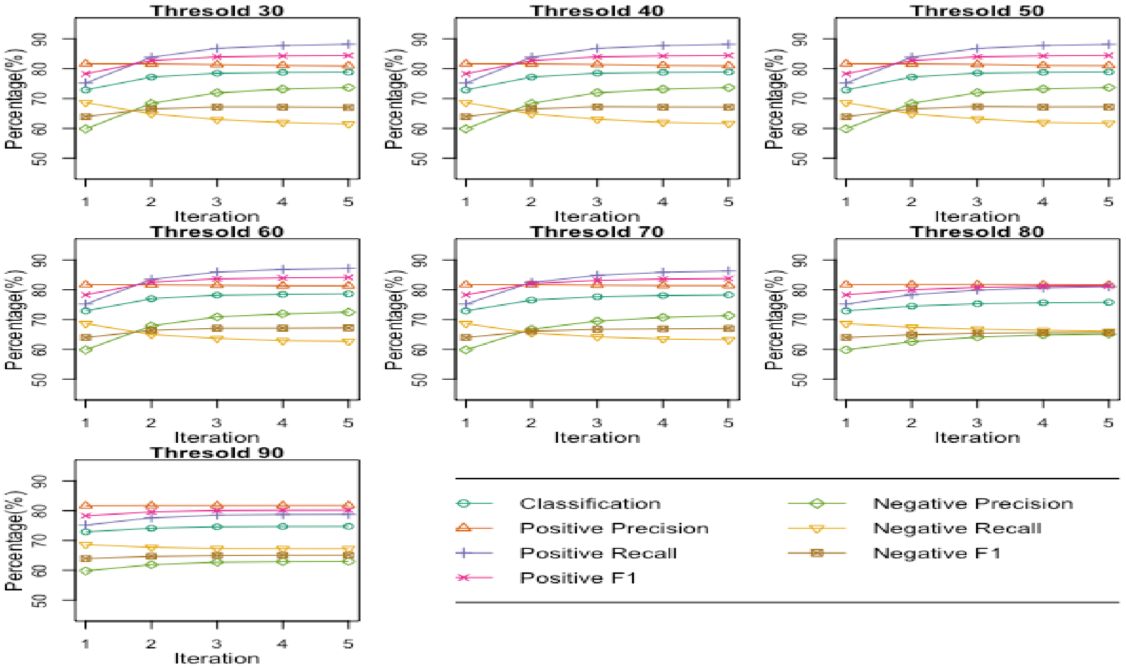

Figure 5-7 Performance of our prediction model, varying the confidence threshold from 30% to 90% in the complete setting. We consider the links predicted and those predicted to disappear to update the graph in every iteration.

We then evaluated the disappearing link phenomenon in the iterative link prediction model. During every iteration, as well as adding high confidence links, we also remove high confidence disappearing links from the graph. The results of experiment carried out in both the selective and complete settings are plotted in Figure 5-6 and Figure 5-7. In both the settings, results show that removing disappearing links does not influence the overall performance of the prediction model when compared with adding predicted links alone.

Different machine learning algorithms. To further assess the performance of our propagation model, we tested our approach using different machine learning algorithms. We considered an SVM and a C4.5 decision tree. Figure 5-8 and Figure 5-9 show the results for the machine learning algorithms, SVM and C4.5 decision tree, respectively. We observed an increase in accuracy of about 2-3% using the SVM compared to the C4.5 decision tree.

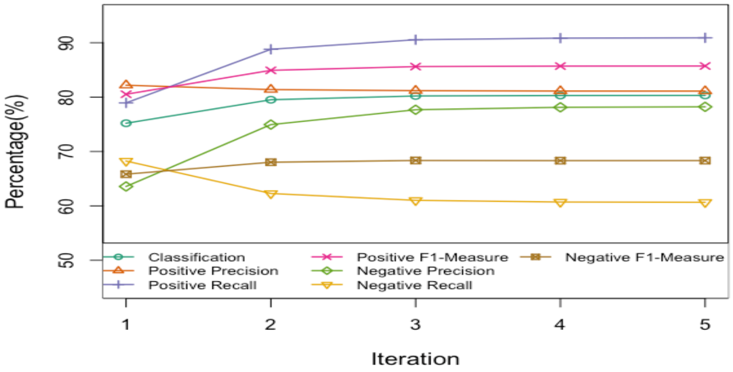

Figure 5-8 Classification metrics for propagation-based link prediction, using an SVM to train the model

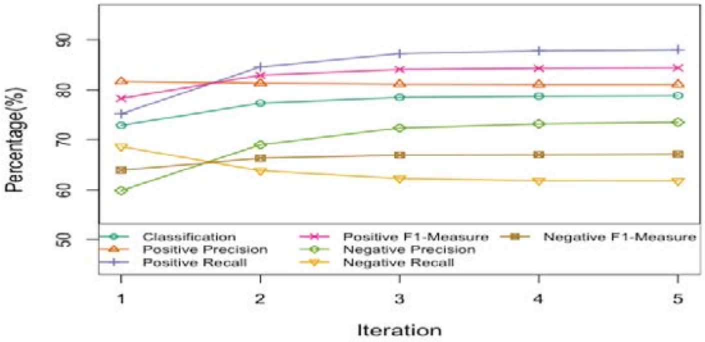

Figure 5-9 Classification evaluation metrics for iterative classification-based link prediction, using a C4.5 decision tree algorithm

5.6 Iterative Classification on other Domains

As discussed previously, link prediction applies to areas ranging from social network analysis to biomedical research. In this section, we validated our iterative classification approach on co-author data and on a protein dataset.

Co-authorship network. We performed the iterative link classification on the co-author network in both the selective and complete settings. The results, presented in

Figure 5-10 and Figure 5-11, show an increase in accuracy from 75.48% for the traditional link prediction method to 78.53% for the iterative link classification method.

Protein-Protein Interaction Network. For this experiment, we validated our iterative link prediction algorithm, by applying it to the protein interaction network to predict the interactions among the proteins. We extracted the topological features from the network and created labels from the dataset. We labeled the data by comparing the scores of known relationships from the database to the experimental scores. A pair of proteins with low confidence in the known database source is labeled *positive* if it has high evidence of interaction in the experimental data source. Otherwise, we label it *negative*.

The results of this experiment are presented for both selective and complete settings in Figure 5-12 and Figure 5-13, respectively. Using our methodology, we observed an increase in accuracy of approximately 1%.

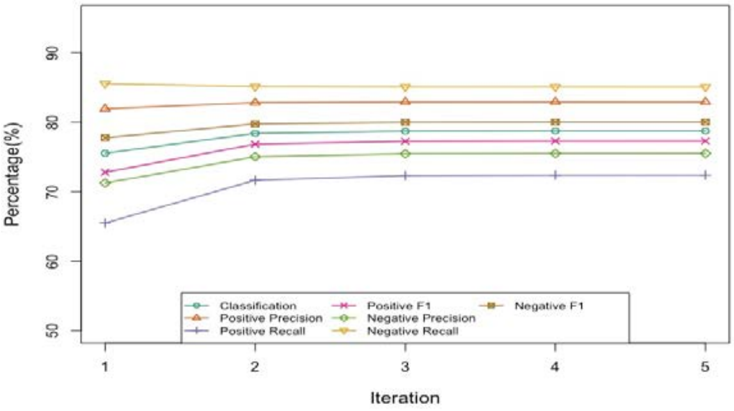

Figure 5-10 Classification accuracy, precision, and recall for the iterative algorithm on DBLP dataset with complete settings

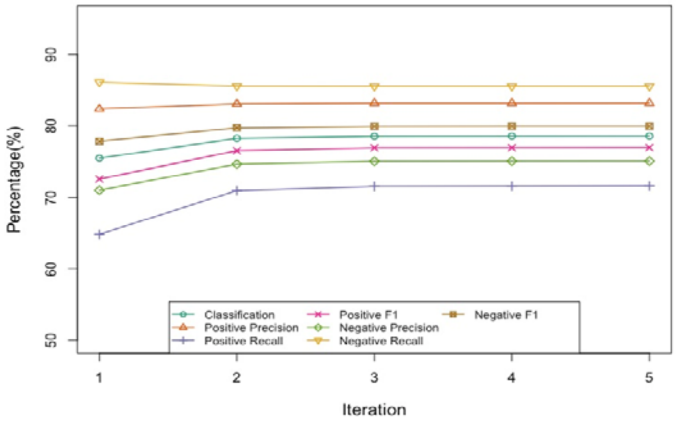

Figure 5-11 Classification accuracy, precision, and recall for the iterative algorithm on DBLP dataset with selective settings

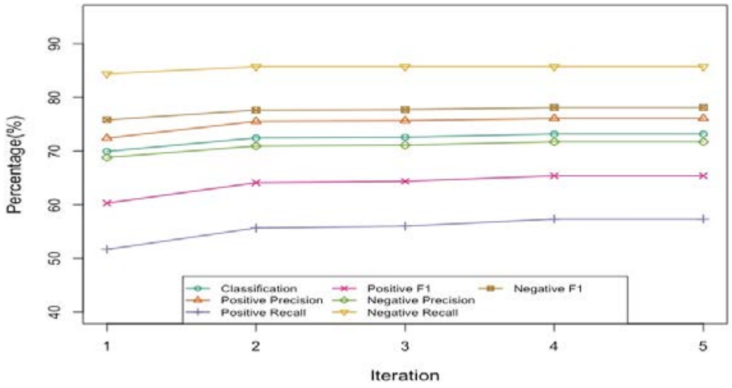

Figure 5-12 Classification accuracy, precision, and recall for the iterative algorithm on protein-protein interaction network with complete settings

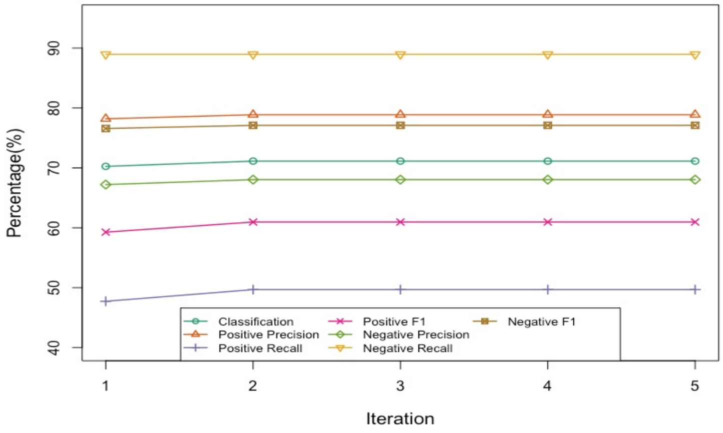

Figure 5-13 Classification accuracy, precision, and recall for the iterative algorithm on protein-protein interaction network with selective settings

5.7 Conclusions and Future work

Predicting the links in a network helps to give an indication of the future state of the network. This provides an opportunity to use knowledge of new link formations to improve the performance of prediction models. In this work, we propose a link-propagation-based prediction methodology that propagates the predicted knowledge of links by introducing the predicted links into the network. Our method uses several features extracted from the network to predict links. We propose two settings, selective and complete, based on how the test data is used in every iteration. Results show an increase in accuracy of 6-7% over single pass link prediction and a 10% increase in the prediction of relevant links. Our experiments show that the performance of model in the selective setting is influenced by the confidence level of links added to the graph. By contrast, the complete setting showed stable performance regardless of the confidence level of added links.

We also tested the ability of the supervised link prediction model to predict the disappearance of links in a network. We used disappearing links as well as predicted links to update the graph in every iteration of our algorithm. Results show that considering link disappearance does not contribute to any improvement in the accuracy of the prediction model. We also applied SVM and C4.5 decision tree machine learning algorithms on our link-propagation-based method to compare the performance.

We studied our methodology using two other datasets: co-authorship and PPI networks. For the co-authorship data, we observed about a 2-3% increase in overall accuracy compared to the typical link prediction approach. We did not find significant change in accuracy for the PPI network data. For future research, we intend to examine the characteristics of the new links formed using an iterative approach.

Whereas we currently use class membership probability of predicted links as a measure of the confidence level for adding links to the network, we will explore the possibility of using interestingness of predicted links as confidence level measures for selecting links to add in each iteration.

6 . Conclusions and Future Work

6.1 Conclusions

In the field of link analysis, link prediction plays a vital role in analyzing complex networks. Most link prediction algorithms use topological and statistical information about the network to predict the likelihood of future node associations. Earlier studies have proposed link prediction methods based on analysis of snapshots, which is not suitable for real-world networks, in which data is highly dynamic. In this work, we make several advances in bringing existing link prediction methods to work on real-time applications. We also propose a link-propagation-based link prediction method for improving the performance of the link prediction method proposed by Katakuri et al. [7].

In Chapter 2, we studied the supervised link prediction methodology from a temporal perspective to address the following questions:

- Does the performance of the link prediction model deteriorate over time?
- How would the link prediction perform in a sliding window setting?

For this study, we considered automating hypothesis discovery from biomedical literature. We also discussed various temporal aspects that had to be considered when evaluating link prediction models on a time-varying network. We designed several experiments to study the performance of models with regard to temporal aspects. Based on the experiment results, we observed the following:

- When using a sliding window model, the durational size of the testing window should be commensurate with the durational size of the training window.

- Although the model’s performance begins to decline when temporal distance is increased, it is reasonably reliable for a short period of time, which eliminates the need to update the model at every instance of new data arrival.

We Investigated three levels of semantic-type-based pruning: comprehensive, selective and precise. We studied the effects of pruning on the overall performance of the predictive model. We observed that the selective and precise settings yielded better results than the comprehensive setting. We also evaluated the link prediction approach by rediscovering the known de facto study cases. Using a comprehensive concept network, we able to discover only one out of six links from case studies. By contrast, the link prediction algorithm was able to uncover four of these links out of six using precise and selective pruning.

While the proposed link prediction model predicts a significant number of possible links in a network at any given time, not all the predicted links are interesting. In several cases, it is vital to identify the most interesting links from among these predicted links. In real-world applications, it is advisable to suggest only the most interesting predictions. To achieve this, the predictions need to be ordered and ranked based on their interestingness.

In Chapter 3, we propose a supervised method for ranking the predicted links using interestingness measures. This method uses several interestingness measures that are common in the field of data mining to identifying the interestingness of predicted links. We consider 13 different interestingness measures and studied the correlations between them. We observed that some of the measures ranked links similarly. We used a supervised method that uses the network features to predict scores and rankings. Our results show that methods for predicting the rankings of predicted links using interestingness measures are more precise than those for predicting the actual scores.

As previously stated, a link prediction model deteriorates over the time and needs to be regularly updated to accommodate recent data, while also preserving the historical patterns. In Chapter 4, we propose incremental learning approaches to the link prediction problem for use in predicting hypotheses on a concept graph. We used incremental support vector machine (SVM) learning with three different methods: 1) the accumulation method, 2) the ensemble method and 3) the hybrid method. For the accumulation method, information from the old model is preserved as support vectors and is used with the new data to update the model. For the ensemble method, a series of models is created at different durations and links are predicted based on a voting system using these models. We studied the ensemble method using different weighting functions: the uniform, linear decay and exponential decay functions. The hybrid method combines the benefits of the two previous methods in which the support vectors are sampled and accumulated for updating the SVM model. Sampling the support vectors in the hybrid method employs either the linear decay or exponential decay weighting function to represent the historical information so that recent data have a greater effect on link prediction.

The experimental results showed that the accuracy of all incremental SVM methods was comparable to the computationally expensive model trained by the conventional SVM. Among the incremental methods proposed, the ensemble method showed the least time complexity.

In Chapter 5, we proposed an iterative link classification methodology that propagates knowledge of predicted links by introducing the predicted links into the network. We proposed two methods, namely selective and complete, based on how the test data is used in every iteration. We observed a significant increase in accuracy over the baseline link prediction method and a significant increase in the number of relevant links predicted. Our experiments show that the performance of the selective method is influenced by the confidence level of links that are added to the graph, whereas the complete setting exhibits a more stable performance regardless of the confidence level of added links.

We also introduced a supervised link prediction model to predict the links likely to disappear in a network. That is, we used the predicted disappearing links along with the predicted links to update the graph in every iteration of our algorithm. We found that incorporating disappearing links does not increase the accuracy of the iterative link prediction model. We also applied SVM and C4.5 decision tree algorithms on our link-propagation-based method to compare performance.

We evaluated our iterative link prediction method on three domains: a concept network, a co-authorship network and a protein-protein interaction network. We observed an increase in accuracy compared to the traditional link prediction approach.

6.2 Future Directions

In this section, we describe future directions for our research. Although this dissertation has addressed several challenges in advancing link prediction methods for real-time link prediction, our future research will extend the work on link prediction in the following directions:

- We will continue to evaluate how frequently the model needs to be updated. Additionally, we will focus on the incremental approach for updating the model instead of retraining the entire model.

- We will explore adopting the ensemble approach, which uses multiple methodologies, to make better predictions instead of relying on a single method.
- We will further examine the possibility of employing incremental learning approaches for large streaming datasets.
- We intend to use the lessons learned from this research to develop a real-time distributed link prediction framework to handle graph stream data.
- We will investigate the topological and temporal characteristics of additional links predicted using our iterative method.
- We are currently using the class membership probability of predicted links as confidence measures to add links to the network. In the future, we will explore the possibility of using the interestingness of predicted links as confidence measures.

Bibliography

[1] D. Liben-Nowell and J. Kleinberg, “The Link Prediction Problem for Social Networks,” *Proc. Twelfth Annu. ACM Int. Conf. Inf. Knowl. Manag.*, no. November 2003, pp. 556–559, 2003.

[2] H. Jeong, S. P. Mason, a L. Barabási, and Z. N. Oltvai, “Lethality and centrality in protein networks.,” *Nature*, vol. 411, no. 6833, pp. 41–42, 2001.

[3] H. Mannila, “Link Prediction on the Semantic MEDLINE Network,” *Discov. Sci.*, vol. 5255, no. OCTOBER, pp. 16–25, 2008.

[4] L. Wang, K. Hu, and Y. Tang, “Robustness of Link-Prediction Algorithm Based on Similarity and Application to Biological Networks,” *Curr. Bioinform.*, vol. 9, no. 3, pp. 246–252, 2014.

[5] C. Wang, V. Satuluri, and S. Parthasarathy, “Local probabilistic models for link prediction,” in *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, 2007, pp. 322–331.

[6] H. Kashima and N. Abe, “A Parameterized Probabilistic Model of Network Evolution for Supervised Link Prediction,” *Sixth Int. Conf. Data Min.*, pp. 340–349, 2006.

[7] J. R. Katukuri, Y. Xie, V. V Raghavan, and A. Gupta, “Hypotheses generation as supervised link discovery with automated class labeling on large-scale biomedical concept networks,” *BMC Genomics*, vol. 13, no. Suppl 3, p. S5, 2012.

[8] Q. Liu, J. Zhang, J. Xiao, H. Zhu, and Q. Zhao, “A Supervised Feature Selection Algorithm through Minimum Spanning Tree Clustering,” in *IEEE 26th International Conference on Tools with Artificial Intelligence*, 2014.

[9] W. Cukierski, B. Hamner, and B. Yang, “Graph-based features for supervised link prediction,” *Proc. Int. Jt. Conf. Neural Networks*, pp. 1237–1244, 2011.

[10] H. H. Song, T. W. Cho, V. Dave, Y. Zhang, and L. Qiu, “Scalable proximity estimation and link prediction in online social networks,” in *IMC 2009: Proceedings of the 9th ACM Internet Measurement Conference SE - IMC ’09*, 2009, pp. 322–335.

[11] P. Zhao, C. C. Aggarwal, and G. He, “Link Prediction in Graph Streams.”

[12] C. C. Aggarwal and K. Subbian, “Evolutionary Network Analysis,” *ACM Comput. Surv.*, vol. 47, no. 1, pp. 1–36, 2014.

[13] Medline, “Medline/MedPub,” 2014. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/. [Accessed: 15-Mar-2014].

[14] Medline, "MEDLINE Fact sheet," 2014. [Online]. Available: https://www.nlm.nih.gov/pubs/factsheets/medline.html. [Accessed: 15-Mar-2014].

[15] D. R. Swanson, "Fish oil, Raynaud's syndrome, and undiscovered public knowledge," *Perspect. Biol. Med.*, vol. 30, no. 1, pp. 7–18, 1986.

[16] R. N. Kostoff and M. B. Briggs, "Literature-Related Discovery (LRD): Potential treatments for Parkinson's Disease," *Technol. Forecast. Soc. Change*, vol. 75, no. 2, pp. 226–238, Feb. 2008.

[17] M. Al Hasan, V. Chaoji, S. Salem, and M. Zaki, "Link prediction using supervised learning," *SDM'06 Work. link Anal. counter-terrorism Secur.*, 2006.

[18] A. Özgür, T. Vu, G. Erkan, and D. R. Radev, "Identifying gene-disease associations using centrality on a literature mined gene-interaction network," *Bioinformatics*, vol. 24, no. 13, pp. 277–285, 2008.

[19] A. Kastrin, T. C. Rindflesch, and D. Hristovski, "Link Prediction in a MeSH Co - occurrence Network : Preliminary Results," 2014.

[20] D. R. Swanson, "Migraine and magnesium: eleven neglected connections," *Perspect. Biol. Med.*, vol. 31, no. 4, pp. 526–557, 1988.

[21] M. D. Gordon and R. K. Lindsay, "Toward discovery support systems: A replication, re-examination, and extension of Swanson's work on literature-based discovery of a connection between Raynaud's and fish oil," *J. Am. Soc. Inf. Sci.*, vol. 47, no. 2, pp. 116–128, 1996.

[22] N. R. Smalheiser and D. R. Swanson, "Using ARROWSMITH: A computer-assisted approach to formulating and assessing scientific hypotheses," *Comput. Methods Programs Biomed.*, vol. 57, no. 3, pp. 149–153, 1998.

[23] M. D. Gordon and S. Dumais, "Using latent semantic indexing for literature based discovery," *J. Am. Soc. Inf. Sci. Technol.*, vol. 49, no. 8, pp. 674–685, 1998.

[24] W. Pratt and M. Y. Yildiz, "LitLinker: capturing connections across the biomedical literature," *Proc. 2nd Int. Conf. Knowl. capture*, pp. 105–112, 2003.

[25] R. K. Lindsay and M. D. Gordon, "Literature-based discovery by lexical statistics," *J. Am. Soc. Inf. Sci.*, vol. 50, no. 7, pp. 574–587, 1999.

[26] A. Kastrin and D. Hristovski, "Learning Links in MeSH Co-occurrence Network Literature-Based Discovery," pp. 1–15, 2014.

[27] Y. Yang, R. N. Lichtenwalter, and N. V. Chawla, "Evaluating link prediction methods," *Knowl. Inf. Syst.*, pp. 1–35, 2014.

[28] MeSH, "MeSH (Medical Subject Headings)," 2014. [Online]. Available:

http://www.ncbi.nlm.nih.gov/mesh. [Accessed: 15-Mar-2014].

[29] M. E. Newman, “Clustering and preferential attachment in growing networks.,” *Phys. Rev. E. Stat. Nonlin. Soft Matter Phys.*, vol. 64, no. 2 Pt 2, p. 025102, 2001.

[30] L. A. Adamic and E. Adar, “Friends and neighbors on the Web,” *Soc. Networks*, vol. 25, no. 3, pp. 211–230, 2003.

[31] P. Jaccard, “Étude comparative de la distribution florale dans une portion des Alpes et des Jura,” *Bull. del la Société Vaudoise des Sci. Nat.*, vol. 37, pp. 547–579, 1901.

[32] Y. Koren, S. C. North, and C. Volinsky, “Measuring and extracting proximity graphs in networks,” *ACM Trans. Knowl. Discov. Data*, vol. 1, no. 3, p. 12–es, 2007.

[33] T. Zhou, L. Lü, and Y. C. Zhang, “Predicting missing links via local information,” *Eur. Phys. J. B*, vol. 71, no. 4, pp. 623–630, 2009.

[34] C. Lei and J. Ruan, “A novel link prediction algorithm for reconstructing protein-protein interaction networks by topological similarity,” *Bioinformatics*, vol. 29, no. 3, pp. 355–364, 2013.

[35] J. O’Madadhain, J. Hutchins, and P. Smyth, “Prediction and ranking algorithms for event-based network data,” *ACM SIGKDD Explor. Newsl.*, vol. 7, no. 2, pp. 23–30, 2005.

[36] I. Kahanda and J. Neville, “Using transactional information to predict link strength in online social networks,” *Proc. 3rd Int. AAAI Conf. Weblogs Soc. Media*, pp. 74–81, 2009.

[37] R. Xiang, J. Neville, and M. Rogati, “Modeling relationship strength in online social networks,” *Proc. 19th Int. Conf. World wide web - WWW ’10*, p. 981, 2010.

[38] K. Kamath, A. Sharma, D. Wang, and Z. Yin, “Realgraph: User interaction prediction at twitter,” in *User Engagement Optimization Workshop@ KDD*, 2014, no. ii.

[39] H. Zhang and R. Dantu, “Predicting social ties in mobile phone networks,” in *Intelligence and Security Informatics (ISI), 2010 IEEE International Conference on*, 2010, pp. 25–30.

[40] M. Zignani *et al.*, “Predicting the Link Strength of " Newborn " Links,” *Proc. 25th Int. Conf. Companion World Wide Web*, pp. 147–148, 2016.

[41] B. Liu, W. Hsu, L. F. Mun, and H. Y. Lee, “Finding interesting patterns using user expectations,” *IEEE Trans. Knowl. Data Eng.*, vol. 11, no. 6, pp. 817–832, 1999.

[42] M. Ohsaki, S. Kitaguchi, K. Okamoto, H. Yokoi, and T. Yamaguchi, “Evaluation of rule interestingness measures with a clinical dataset on hepatitis,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and*

Lecture Notes in Bioinformatics), vol. 3202. pp. 362–373, 2004.

[43] P.-N. Tan, V. Kumar, and J. Srivastava, “Selecting the Right Interestingness Measure for Association Patterns,” *Proc. eighth ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. - KDD ’02*, vol. 2, pp. 32–41, 2002.

[44] P. Lenca, P. Meyer, B. Vaillant, and S. Lallich, “A multicriteria decision aid for interestingness measure selection,” May 2004.

[45] W. A. Aljandal, “Itemset size-sensitive interestingness measures for association rule mining and link prediction,” 2009.

[46] P. Lenca, P. Meyer, B. B. Vaillant, and S. Lallich, “On selecting interestingness measures for association rules: User oriented description and multiple criteria decision aid,” *Eur. J. Oper. Res.*, vol. 184, no. 2, pp. 610–626, 2008.

[47] L. Geng and H. J. Hamilton, “Interestingness measures for data mining: A Survey,” *ACM Comput. Surv.*, vol. 38, no. 3, pp. 1–32, 2006.

[48] B. Vo and B. Le, “Interestingness measures for association rules: Combination between lattice and hash tables,” *Expert Syst. Appl.*, vol. 38, no. 9, pp. 11630–11640, 2011.

[49] R. Fagin, R. Kumar, and D. Sivakumar, “Comparing top-k lists,” *Inf. Syst.*, vol. 38, no. 6, pp. 820–834, 2013.

[50] Y. Wang and I. H. Witten, “Induction of model trees for predicting continuous classes,” 1996.

[51] J. Ben Schafer, J. A. Konstan, and J. Riedl, “E-commerce recommendation applications,” in *Applications of Data Mining to Electronic Commerce*, Springer, 2001, pp. 115–153.

[52] J. Kunegis, E. W. De Luca, and S. Albayrak, “The Link Prediction Problem in Bipartite Networks,” *Comput. Intell. KnowledgeBased Syst. Des.*, vol. abs/1006.5, p. 10, 2010.

[53] M. R. R. Ade and D. P. R. Deshmukh, “Methods for Incremental Learning: a Survey,” *Int. J. Data Min. Knowl. Manag. Process*, vol. 3, no. 4, pp. 119–125, 2013.

[54] R. Roscher, W. Förstner, and B. Waske, “I2VM: Incremental import vector machines,” *Image Vis. Comput.*, vol. 30, no. 4–5, pp. 263–278, 2012.

[55] R. Polikar, L. Udpa, S. S. Udpa, and V. Honavar, “Learn++: An incremental learning algorithm for supervised neural networks,” *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 31, no. 4, pp. 497–508, 2001.

[56] S. U. Guan and F. Zhu, “An incremental approach to genetic-algorithms-based classification,” *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 35, no. 2, pp.

227–239, 2005.

[57] N. Syed, H. Liu, and K. Sung, “Incremental learning with support vector machines,” in *International Joint Conference on Artificial Intelligence*, 1999.

[58] S. Ruping, “Incremental learning with support vector machines,” *Proc. 2001 IEEE Int. Conf. Data Min.*, pp. 0–1, 2001.

[59] R. Xiao, J. Wang, and F. Zhang, “An approach to incremental SVM learning algorithm,” in *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, 2000, vol. 2000–Janua, pp. 268–273.

[60] H. Yang, H. Wei, and F. Lei, “An Incremental Learning Algorithm for SVM based on Voting Principle,” *Int. J. Inf. Process. Manag.*, vol. 2, no. 2, pp. 8–14, 2011.

[61] L. Landau and C. M. C. C. M. Bishop, *Pattern Recognition and Machine Learning*, vol. 4, no. 4. Springer, 2006.

[62] A. Abdiansah, “Time Complexity Analysis of Support Vector Machines (SVM) in LibSVM,” pp. 1–7, 2015.

[63] G. Blanchard, O. Bousquet, and P. Massart, “Statistical performance of support vector machines,” *Ann. Stat.*, vol. 36, no. 2, pp. 489–531, 2008.

[64] Y. Wang, F. Zhang, and L. Chen, “An approach to incremental SVM learning algorithm,” in *2008 ISECS International Colloquium on Computing, Communication, Control, and Management*, 2008, vol. 1, pp. 352–354.

[65] O. L. Mangasarian and D. R. Musicant, “Lagrangian Support Vector Machines,” *J. Mach. Learn. Res.*, vol. 1, pp. 161–177, 2001.

[66] Y. Lee and O. Mangasarian, “RSVM: Reduced support vector machines,” *Sdm*, vol. 1, pp. 1–17, 2001.

[67] V. Martínez, C. Cano, and A. Blanco, “ProphNet: a generic prioritization method through propagation of information.,” *BMC Bioinformatics*, vol. 15 Suppl 1, no. Suppl 1, p. S5, 2014.

[68] C. von Mering *et al.*, “Comparative assessment of large-scale data sets of\n protein–protein interactions,” *Nature*, vol. 417, no. 6887, pp. 399–403, 2002.

[69] C. T. Butts, “Network inference, error, and informant (in) accuracy: a Bayesian approach,” *Soc. Networks*, vol. 25, no. 2, pp. 103–140, 2003.

[70] Z. Huang, X. Li, and H. Chen, “Link Prediction Approach to Collaborative Filtering,” *Artif. Intell.*, pp. 141–142, 2005.

[71] S. Chakrabarti, B. Dom, and P. Indyk, “Enhanced hypertext categorization using

hyperlinks," *ACM SIGMOD Rec.*, vol. 27, no. 2, pp. 307–318, 1998.

[72] J. Neville and D. Jensen, "Iterative classification in relational data," *Learn. Stat. Model. from Relational Data*, pp. 42–49, 2000.

[73] a Heß and N. Kushmerick, "Iterative ensemble classification for relational data: A case study of semantic web services," *Mach. Learn. ECML 2004*, vol. 3, 2004.

[74] A. Galstyan and P. R. Cohen, "Identifying Covert Sub-Networks Through Iterative Node Classification," *Proc. First Int. Conf. Intell. Anal.*, 2005.

[75] B. Taskar, M.-F. Wong, P. Abbeel, and D. Koller, "Link prediction in relational data," *Learn. Stat. Patterns Relational Data Using Probabilistic Relational Model.*, p. 7, 2004.

[76] J. Katukuri, T. Konik, R. Mukherjee, and S. Kolay, "Recommending similar items in large-scale online marketplaces," in *2014 IEEE International Conference on Big Data (Big Data)*, 2014, pp. 868–876.

[77] S. Scellato, A. Noulas, and C. Mascolo, "Exploiting place features in link prediction on location--based social networks," *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discov. data Min.*, pp. 1046–1054, 2011.

[78] J. Scripps and A.-H. Esfahanian, "A matrix alignment approach for link prediction," *2008 19th Int. Conf. Pattern Recognit.*, pp. 1–4, 2008.

[79] R. Guimerà and M. Sales-Pardo, "Missing and spurious interactions and the reconstruction of complex networks," *Proc. Natl. Acad. Sci.*, vol. 106, no. 52, pp. 22073–22078, 2009.

[80] H. Kashima, T. Kato, Y. Yamanishi, M. Sugiyama, and K. Tsuda, "Link propagation: a fast semi-supervised learning algorithm for link prediction," *Siam Icdm 2009*, pp. 1100–1111, 2009.

Pusala, Murali Krishna. Bachelor of Electrical and Electronics Engineering, JNTU Hyderabad, India, Spring 2007; Master of Computer Science in University of Louisiana at Lafayette, Fall 2010; Doctor of Philosophy, University of Louisiana at Lafayette, Summer 2018

Major: Computer Science
Title of Dissertation: Link Discovery Through Iterative Link Classification — Towards a Real-Time Analysis of Graph Evolution
Dissertation Director: Dr. Vijay V. Raghavan
Pages in Dissertation: 116; Words in Abstract: 345

Abstract

In recent years, link prediction has been applied to a wide range of real-world applications which often generate massive dynamic networks that require an effective real-time approach to predicting the formation of future links. Traditionally, link prediction approaches utilize a single snapshot of a network to predict future links. However, real-world network data often evolves dynamically at a rapid pace by adding and removing links. Therefore, there is a need for a dynamic and online link prediction framework. This dissertation focuses on challenges and solutions with the aim of advancing a link prediction framework for use in real-time analytics.

For real-time link prediction, the framework should 1) be reliable and accurate, 2) maintain learning models, and 3) calculate node similarities in real time. In a real-world application that deals with time-varying networks, it is important to understand predictive models in a time-varying context. In this work, we develop several guidelines for using prediction models in a dynamic network. We also propose an incremental support vector machine method for link prediction, which updates the model using the latest data available as well as historical information.

While being able to forecast future links accurately is vital, another equally important problem is to identify the most important and relevant links among large numbers of future

links. To address this problem, we propose a domain-independent, supervised method that predicts the rank of future links using objective interestingness measures.

We also propose an iterative link classification method, which updates the network using only predicted links with a high confidence level at each iteration. Using this method, we observed a significant improvement in accuracy and recall over the baseline link prediction method.

Our proposed solutions address two out of the three requirements defined above, by focusing on maintaining the learning models and increasing the reliability and accuracy of link prediction in a dynamic network. In our future work, we plan to extend this research to address the final requirement by developing the approximation algorithms for computing similarity measures in large dynamic and streaming networks, in real time, using distributed computing frameworks.

Biographical Sketch

Murali Krishna Pusala received his Bachelor of Technology degree in spring 2007 in electrical and electronic engineering from Jawaharlal Nehru Technological University, Hyderabad in Andhra Pradesh, India. Murali joined the master's program in computer science at the University of Louisiana at Lafayette in spring 2008. Originally intending only to complete the master's program, Murali applied for the Doctor of Philosophy in computer science from the University of Louisiana at Lafayette in Fall 2010 and received his degree in summer 2018.