Graph-Based Diffusion Methods for Node Classification and Link Prediction

Mohan Timilsina

Submitted in fulfillment of the requirements for the degree of
Doctor of Philosophy (Computer Sciences)
Supervisor:
Prof. Mathieu d'Aquin
Co-Supervisor:
Dr. Haixuan Yang
Internal Examiner:
Prof. John Breslin
External Examiner:
Dr. Nicola Perra

Data Science Institute, The Insight Centre for Data Analytics, National University of Ireland Galway

Abstract

Graphs are common representation tools to organize information from heterogeneous sources. They have been applied in various domains such as the scientific, engineering, political, and business sectors. The large volume of graph data is now becoming a surging research challenge for building highly accurate predictive models. One of the main motivations behind the interest in graphs is due to their structure, since it describes how information spreads through the nodes. In turn, this natural property of graphs makes them ideal for designing prediction models based on spreading functions, which can activate changes in the connections and its nodes.

Link Prediction (LP) is one of the fundamental problems in graphs. It is the problem of predicting associations between a pair of nodes. Many LP algorithms addressed in the scientific literature evaluate them as a classification task or a ranking problem. Only a few studies have considered the impacts of diffusion in graphs, and none of the works on diffusion-based methods has investigated link prediction in graphs with heterogeneous nodes. Here, we proposed a 2-layered graph framework to combine two different graphs and analyzed diffusion algorithms such as *PageRank, Katz*, and heat diffusion model. We further proposed a novel diffusion method in a 2-layered graph framework by integrating matrix factorization with heat diffusion. Our results show the effectiveness of this integration by computing weighted (i.e., ranked) predictions of initially unknown links between two disjoint nodes.

A prominent problem in the study of diffusion in graphs is that not all kinds of diffusion are the same. Some network structures support long-range diffusion, whereas others support short-range diffusion. Long-range diffusion has been largely explored in applications such as viral marketing, influence maximization, political campaigning, or ordinary label propagation, especially through graph-based semi-supervised machine learning. Conversely, short-range diffusion is less explored. If we consider the case of a real-world network where nodes are shared across different layers i.e., *multiplex network*, long-range diffusion might not work and result in node miss-classification. We proposed a novel boundary-based heat diffusion (BHD) method, which has the flexibility to diffuse the information step by step due to its time-dependent property and guarantees a closed-form solution.

We evaluated BHD models on different types of real-world networks for a node classification task. We took the same inspiration from a semi-supervised machine learning approach by using a small amount of labeled data and a large number of unlabeled data to classify the nodes. For this task, we applied BHD in (i) multiplex network; (ii) homogeneous network with homophilic labels; (iii) homogeneous network with heterophilic labels; and (iv) homogeneous network with mixed labels. Furthermore, we extended our BHD approach in complex data for semi-supervised graph regression problems to predict the real values of the node labels using fewer labeled data. Experiments from business, biomedical, physical, and social domain data show that the boundary-based heat diffusion method can effectively outperform the top state of the art methods.

Communication

Journals

- **Mohan Timilsina**, Mathieu d'Aquin, Haixuan Yang. *Heat Diffusion Approach for Scientific Impact Analysis in Social Media.*
  Social Network Analysis Mining 9(1): 16:1-16:13 (2019).

- **Mohan Timilsina**, Meera Tandan, Mathieu d'Aquin, Haixuan Yang. *Discovering Links Between Side Effects and Drugs Using a Diffusion Based Method.*
  Nature Scientific Reports: 2019.

- **Mohan Timilsina**, Haixuan Yang, Ratnesh Sahay and Dietrich Rebholz-Schuhmann. *Predicting Links Between Tumor Samples and Genes using 2-Layered graph based diffusion approach.*
  BMC Bioinformatics: 2019.

Conference

- **Mohan Timilsina**, Mathieu d'Aquin, Haixuan Yang. *A 2-Layered Graph Based Diffusion Approach for Altmetric Analysis.*
  IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2018

Acknowledgements

I want to express my sincere gratitude to Prof. Mathieu d'Aquin, my research advisor for his valuable suggestions, intuitive ideas, and encouragement during the period of this research study. He always gave me the freedom to choose the problems of my interest and supported me during this research work.

I am grateful to Dr. Haixuan Yang, my co-supervisor, who taught me two courses on applied regression modeling and statistical inferences. He guided me in scientific publications and taught me countless aspects of doing research, and how to write and think like a scientist. I also learned a lot from him while working as a teaching assistant for the courses he taught.

I have had the opportunity to learn about biomedical informatics from Prof. Dr. Dietrich Rebholz Schuhmann. Dietrich guided me as a supervisor during the initial phases of my Ph.D. I am grateful to him for the scientific discussions we had on biomedical data analysis.

Huge credit goes to my parents Gopal Timilsina, Shanta Timilsina, Yam Bahadur Tandan, and Shakuntala Tandan for their continuous encouragement and inspiration towards my study. I am grateful to my dear wife, Meera, for her patience for accepting many weekends full of work. She has been considerate for many days, where I had to go home very late from the lab.

Last but not least, my special thanks go to my friend Pablo Torres Tramon, Zia Ush Shamszaman and Alejandro Figueroa. I always enjoyed our coffee talks and endless chats about research and life outside research.

I want to acknowledge to all Irish taxpayers who have supported my research. In particular, I have been supported by the Science Foundation of Ireland (SFI/12/RC/2289).

Contents

Abstract ii

Acknowledgements v

Contents vi

List of Figures x

List of Tables xii

1 Introduction 1
- 1.1 What can we learn using graphs? 1
- 1.2 Motivation 3
- 1.3 Research Questions 5
  - 1.3.1 How can we predict the links between dissimilar nodes in graphs? 5
  - 1.3.2 Can we integrate heterogeneous information for link prediction between dissimilar nodes in graphs using diffusion methods? 6
  - 1.3.3 Do graph-based diffusion methods adapt to different variants of graphs (multiplex/homogeneous/manifold) for node classification? 6
    - 1.3.3.1 Can graph-based diffusion methods be applied to regression problems using few labeled data points? 7
- 1.4 Contributions 7
- 1.5 Thesis Outline 8

2 Foundations 10
- 2.1 Essential Concepts 10
  - 2.1.1 Essential Concepts of Graph 10
- 2.2 Graph Metrics 11
  - 2.2.1 Order 11
  - 2.2.2 Size 11
  - 2.2.3 Neighborhood 12
  - 2.2.4 Node Degree 12
  - 2.2.5 Indegree 12
  - 2.2.6 Outdegree 12
- 2.3 Graph Based Link Prediction 12
  - 2.3.1 Node-Based 13
  - 2.3.2 Path-based 14
- 2.4 Node Classification 15

2.4.1 Network Matrices and Diffusion Functions . . . 16
2.5 Supervised Classification Model . . . 17
2.5.1 Support Vector Machine . . . 17
2.6 Factorization methods in Graphs . . . 18
2.6.1 Non negative Matrix Factorization (NMF) . . . 18
2.7 Performance Metrics . . . 19
2.7.1 Precision . . . 19
2.7.2 Recall . . . 19
2.7.3 F1 Score . . . 19
2.7.4 Accuracy . . . 19
2.7.5 Receiver Operating Characteristics Curve (ROC) . . . 20
2.7.6 Precision-Recall Curve (PR) . . . 20
2.7.7 Root Mean Square Error (RMSE) . . . 20

3 Related Work 21

3.1 Related Work on Link Prediction and Node Classification in Graphs . . . 21
3.2 Link Prediction - The Big Picture . . . 22
3.2.1 Node-Based Methods . . . 23
3.2.1.1 Common Neighbor (CN) . . . 24
3.2.1.2 Jaccard Coefficient (JC) . . . 24
3.2.1.3 Adamic Adar (AA) . . . 24
3.2.1.4 Preferential Attachment (PA) . . . 25
3.2.1.5 Resource Allocation (RA) . . . 25
3.2.1.6 Leicht–Holme–Newman Index (LHNI) . . . 25
3.2.1.7 Hub Promoted Index (HPI)) . . . 26
3.2.1.8 Hub Depressed Index (HDI) . . . 26
3.2.1.9 Sorenson Index (SO) . . . 26
3.2.1.10 Salton Index (SI) . . . 26
3.2.2 Comparisons of Different Node-Based Algorithms . . . 27
3.2.3 Path-Based Methods . . . 28
3.2.3.1 Katz . . . 28
3.2.3.2 Hitting Time (HT) . . . 29
3.2.3.3 Compute Time (CT) . . . 29
3.2.3.4 Cosine Similarity (CS) . . . 30
3.2.3.5 Spreading Activation (SA) . . . 30
3.2.3.6 SimRank (SR) . . . 30
3.2.3.7 Random Walk With Restart (RWR) . . . 30
3.2.3.8 PropFlow (PF) . . . 31
3.2.4 Comparisons of Different Path-Based Algorithms . . . 31
3.2.5 Supervised Learning . . . 32
3.2.6 Comparisons of Different Supervised Learning Methods in Link Prediction . . . 33
3.2.7 Factorization-Based . . . 34
3.2.7.1 Singular Value Decomposition (SVD) . . . 35
3.2.7.2 Non Negative Matrix Factorization (NMF) . . . 35
3.2.7.3 Tensor Factorization (TF) . . . 36

3.2.8 Comparisons of Different Factorization-Based Methods in Link Prediction . . . 36
3.2.9 Summary of the Link Prediction Methods . . . 37
3.3 Related Work on Node Classification . . . 37
3.3.1 Mincut . . . 38
3.3.2 Harmonic Functions (HMN) . . . 39
3.3.3 Local and Global Consistency (LGC) . . . 40
3.3.4 Heat Diffusion (HD) . . . 40
3.3.5 Dissimilar Label Propagation . . . 41
3.3.6 OMNI-Prop . . . 42
3.3.7 CAMLP: Confidence-Aware Modulated Label Propagation . . . 43
3.3.8 Belief Propagation (BP) . . . 43
3.3.9 Semi Supervised Regression (SSR) . . . 44
3.4 Comparison of Node Classification Algorithm . . . 44

4 Diffusion Based Method for the Link Prediction Task 48
4.1 Introduction . . . 49
4.2 Graph Data Model . . . 49
4.3 Solution Approach . . . 50
4.4 Model Description . . . 51
4.5 Results and Evaluations . . . 55
4.5.1 Link Prediction between Tumor Samples and Genes . . . 56
4.5.2 Link Prediction between Academic Entities and Social Media . . . 59
4.5.3 Selection of Parameters . . . 61
4.5.4 Evaluation Metrics . . . 63
4.6 Discussion . . . 68
4.7 Conclusion . . . 70

5 Discovering Links in a 2-Layered Graph Using a Combination of Matrix Factorization and Diffusion 71
5.1 Introduction . . . 71
5.2 Methodology . . . 72
5.3 Results and Evaluations . . . 77
5.4 Discussion . . . 81
5.5 Conclusion . . . 82

6 A Boundary Based Heat-Diffusion Method for Node Classification in a Multiplex Network 84
6.1 Introduction . . . 84
6.2 Problem Formulation . . . 87
6.3 Methodology . . . 88
6.3.1 Network Aggregation . . . 88
6.3.2 Heat Diffusion in a Boundary Condition in Graph (BHD) . . . 88
6.3.3 Computational Complexity . . . 92
6.3.4 Initial Temperature Setting . . . 92
6.4 Algorithm . . . 93
6.4.1 Parameter t . . . 93
6.5 Experiments . . . 94

6.6 Conclusion 99

7 Semi-Supervised Regression Using Diffusion on Graphs 101
7.1 Introduction 101
7.2 Problem Formulation 103
7.3 Methodology 104
7.3.1 Graph Construction 104
7.3.2 Heat Diffusion in a Boundary Condition in Graph (BHD) 105
7.3.3 Computational Complexity 106
7.3.4 Temperature Setting for initial conditions: 106
7.4 Experiments 107
7.5 Conclusion 113

8 Conclusions 115
8.1 Discussion 115
8.2 Summary of the Chapters of the Thesis 117
8.3 Limitations of the Thesis and Directions for Future Research 118

Bibliography 121

List of Figures

1.1 An example of node classification problem for music genre preference prediction. 2
1.2 Diffusion applications in graphs. 4
1.3 Diffusion framework and thesis contributions outline. 8

3.1 Overview of Link Prediction techniques. 23
3.2 *Manifold of handwritten digits as a 2D representations* 38

4.1 2-Layered graph data model. 50
4.2 Solution approach for link prediction in a 2-layered graph. 51
4.3 2-layered toy networks of tumor and gene. The red arrow signifies the absence of link and the black arrows shown known links. 56
4.4 Visualization of the genes shared across the different anatomical locations on the cosmic methylation data. The size of the ring represents the number of genes in the dataset that are the members of the anatomical location of the body. An arc indicates how often these genes are shares across the connected segments. 57
4.5 Impact of parameter α. The prediction accuracy assessed by varying diffusion parameter ranging from 0 to 1 with the step of 0.1 in BioGRID and STRING Network to predict tumor samples and genes. Blue curve represents BioGRID and green curve represents String network. 62
4.6 Impact of Parameter M. The prediction accuracy assessed by varying iterations in BioGRID and STRING Network to predict tumor samples and genes. Blue curve represents BioGRID and green curve represents the String network. 62
4.7 Result of 10 Fold Cross Validation in nine different gene interaction channels for predicting tumor samples and genes. 64
4.8 Histogram of p-values from permutation test. 69

5.1 An example of side effect-drug bipartite graph and drug-drug similarity graph. 74
5.2 *Side Effects Propagation in a Drug-Drug Similarity Network.* 77
5.3 Histogram of p-values from permutation test. The X-axis is in log base 10 scale. 81

6.1 (A) Multiplex network of AI scientists talking to each other. Each layer represents an AI conference (ICML, NeurIPS, AAAI). (B) Aggregated multiplex network into a single-layer network. 85
6.2 The error curve of different label propagation algorithms with time increasing from 1 to 600. 91

6.3 Impact of parameter t in Multiplex networks. X-axis is the percentage of data ranked. Y-axis is the Precision at each percentage of the data. 98

7.2 Impact of parameter t in regression datasets. X-axis is the percentage of labeled data. Y-axis is the RMSE score. 111

List of Tables

2.1 Matrices derived from networks. 16

2.2 Different variants of diffusion functions. 17

3.1 Qualitative comparison between different node based Link Prediction algorithms. 28

3.2 Qualitative comparison between different path based Link Prediction algorithms. 32

3.3 Qualitative comparison of the supervised classification models for Link Prediction. 34

3.4 Qualitative comparison of the tensor factorization based Link Prediction algorithms. 37

3.5 Quantitative comparison between different label propagation algorithms for node classification. 45

4.1 Network summary of 2-layered tumor-gene graph. 59

4.2 Network summary of 2-layered scientific authors and web graph. 60

4.3 Network summary of 2-layered scientific publications and web graph. . . . 61

4.4 Results of our heat diffusion algorithm with baseline prediction in a 10-fold cross-validation settings. 67

5.1 Network Summary of 2-layered Scientific Publications and Web Graph . . 73

5.2 Network Summary of 2-layered Scientific Authors and Web Graph 73

5.3 Summary of side effect-Drug and Drug-Drug Network. 74

5.4 Link prediction results for the different state of the art methods. Each score shows the mean AUC-PR score for predicting links. The figure after ± is the standard deviation obtained from the 10 fold cross validation. . . 79

5.5 P-values of the t-test at significance level $\alpha = 0.05$, *** indicates high significance. 80

6.1 Multiplex Network Datasets. 94

6.2 The Result of 10 fold cross validation of boundary based heat diffusion with state of the art label prediction algorithm in a multiplex network. The figure behind ± sign is the standard deviation. 97

6.3 Network Datasets. 99

6.4 Summary of results on networks with different label types. 99

7.1 Symbols and Definitions. 104

7.2 Regression Datasets. 108

7.3 The average RMSE score of 10 fold cross validation from state of the art methods. The error is the standard deviation obtained from the 10-fold cross validation. 112

7.4 P-values of the t-test at significance level $\alpha = 0.05$. The bold figures indicate statistical significance. 112

Chapter 1

Introduction

1.1 What can we learn using graphs?

To understand a complex system, one needs to know how components interact with each other. This interaction naturally leads to a graph or network representation. Graph[1] representation offers a common language to study various systems, which might be physical, chemical, biological, or social. The study of interconnected systems evolved to be referred to as "**Network science**," which borrows theories from different scientific disciplines. The network science field is the intersection of various disciplines: (i) discrete mathematics, (ii) statistics, (iii) computer science, and (iv) sociology.

Over the last few years, the mathematical and algorithmic study of networks has risen in attention towards predictive modeling in graphs. At the same time, the computational technologies have developed to store a massive amount of data consisting of entities as well as their relationships. Graphs are mathematical representations which apply to many datasets, including real-world and synthetic datasets. These networked datasets can exhibit topological and graph-theoretic properties that can be exploited for predictive analytics. Thus, it seems crucial to develop computational methods to analyze and effectively use the data held in large networks. One significant aspects of this regards the spread or propagation of information between nodes. Examples from the real-world include web users navigating from site to site by clicking hyperlinks in a browser, new products becoming popular based on word-of-mouth and interactions between people, and disease spreading among human populations based on direct contact [Tsiatas, 2009].

A related problem in graph-based predictive analytics is **Link Prediction (LP)** between nodes. LP is used every day, as it is typically at the foundation of recommender systems,

[1] The words graph and network are used interchangeably in this thesis.

which can be seen as predicting new links between users and items on platforms such as Amazon[2] and Netflix[3] [Gomez-Uribe and Hunt, 2016]. In particular, the prediction of future links is helpful for the understanding of the network and communication evolution [Weng et al., 2013b]. In social media platforms, predicting links that do not exist can promote engagement and interaction among users [Weng et al., 2013b], due to information diffusion [Weng et al., 2013a].

Node Classification is the problem of inferring missing attribute values of nodes. The missing attributes of the nodes are inferred from the given attribute values of other nodes in the network. The node attribute inference provides the ability to bring node-neighborhood information into the predictions. For instance, in a friendship network, we might want to predict the music genre preferences of a user. In Figure 1.1, we saw that Alex has four friends, and three of his friends have a preference for "Rock," and one of his friends has a preference for "Classical" music.

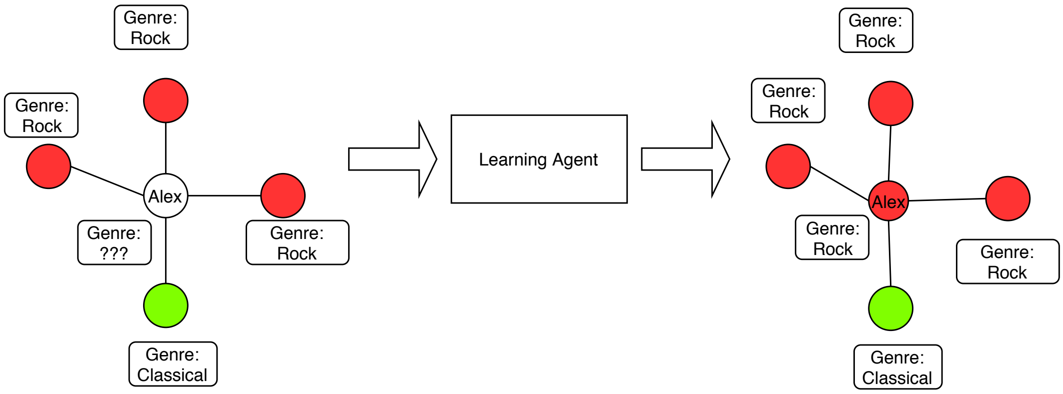


FIGURE 1.1: An example of node classification problem for music genre preference prediction.

Since most of Alex's friends like the "Rock" genre music, a simple machine learning agent might learn from this neighborhood information and predict that it is most likely that Alex will also like "Rock" music.

While other tasks are often associated with graph analytics, such as community detection and graph classification, this thesis focuses on link prediction and node classification in graphs. The main objective behind the research presented in this thesis is to understand how diffusion connects those two crucial problems in graphs. In a typical example from online social networks, when a user finds that his neighbors share or forward a piece of information, the user will be influenced to consider whether to share or forward the information, which leads to information diffusion. Information diffusion permits users to receive or observe information that is beyond the scope of their social cycles. Can

[2] https://www.amazon.com/
[3] https://www.netflix.com

these traces of diffusion influence the creation of new links? Similarly, can we classify nodes based on the information propagated thoughout the graph? By answering these questions, we can understand how the diffusion phenomenon can effectively be used for predictive modeling in graphs.

All the diffusion processes in various networks, from biological, social to computer networks, share the same network-based theoretical and modeling framework. A substantial part of this thesis deals with this diffusion process in graphs, which we study from three perspectives:

- First, we research a diffusion approach applied in various kinds of networks for link prediction tasks. We explore the available algorithms on different graphs such as biological and social media, and analyze if they accurately perform link prediction tasks.
- By using the intuition discovered in part 1, we proposed a novel diffusion approach that is suitable for combining two different graphs, called 2-layered graphs. Its novelty lies in labeling a graph in the first layer and diffusing the information in the second layer for a link prediction task. For this, we developed the diffusion model by integrating matrix factorization with heat diffusion.
- Since we observed the potential of the heat diffusion method in link prediction tasks, we proposed another variant of heat diffusion called a boundary-based heat diffusion kernel that can integrate both harmonic function and heat diffusion. We evaluated this method in various kinds of graphs, such as multiplex, manifold, and real-world homogeneous graphs for node classification problems. We further test our novel approach in a semi-supervised regression problem using graphs.

The diffusion in the network opens opportunities for performing machine learning tasks in networks. We explore this opportunity in three different tasks (i) link prediction, (ii) node classification, and (iii) regression problems.

1.2 Motivation

Why diffusion?

Diffusion on a graph is viewed as an extension of a one-dimensional stochastic process, whose behavior at some nodes is specified. These stochastic processes were introduced first by G. Lumer [Lumer, 1984]. The diffusion process on graphs comes from the natural process of physical and mathematical methods. These methods make the diffusion process in graphs flexible, easy to tune for time-scales, easy to incorporate labeling information, and very fast to compute. The fundamental overview of the diffusion process in graphs is provided by [Cowen et al., 2017], as shown in Figure 1.2.

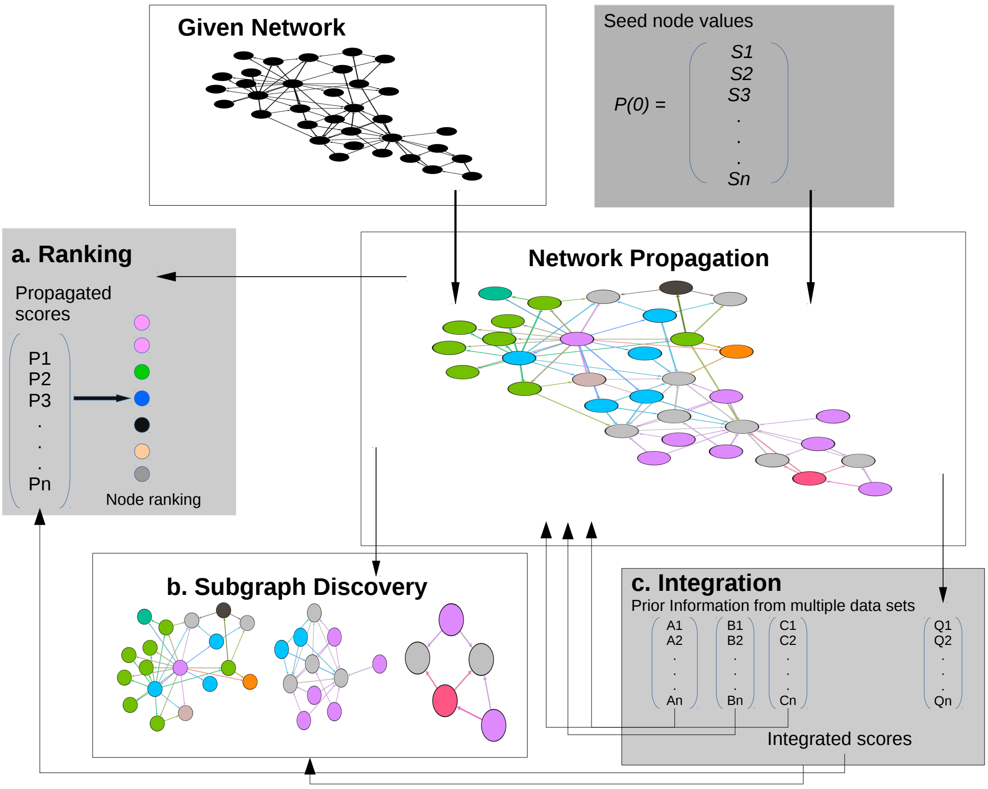


FIGURE 1.2: Diffusion applications in graphs.

The diffusion process in graphs has three broad categories. The first is ranking the nodes based on the influence or impact score in nodes. For example, ranking the users based on the movie preferences by user interaction graphs. The second is discovering the network modules. One typical example of this is classifying users based on movie genres by computing user-user similarities. The third is integrative approaches that combine multiple diffusion scores from multiple data sources to improve prediction performance. Not all the graph-based diffusion methods obtain the same accuracy for all three tasks mentioned. The similarity between all the graph-based diffusion models is that they propagate information in the graph, but they possess differences in terms of computation and the nature of the diffusion. For example, heat-style diffusion propagates faster than PageRank style diffusion by heavily weighting shorter paths in the graphs [Kloster and Gleich, 2014]. This behaviour makes a difference in the performance of the diffusion method for tasks such as node classification and link prediction. Similarly, the diffusion kernels [Kondor and Lafferty, 2002] used in semi-supervised machine learning tasks for label propagation algorithms capture the long-range relationships (global information) between nodes in the network. Due to this, long-range diffusion puts more emphasis on random walks that explore more of the network. However, there are specific real-world networks that tend to be linked by the shortest diffusion paths. For instance, proteins that have similar functions link by shortest paths [Zhou et al., 2002].

Link prediction and node classification is the fundamental problem in predictive analytics in graphs. Previous studies in this area are only applicable for homogenous networks where nodes and edges are of the same type, e.g., friendship prediction in social network analysis, protein-protein interaction in bioinformatics, or similar user identification in an e-commerce application. In the real world setting, the networks are heterogeneous, where the nodes and edges are of different types. For example, (i) links between users and items in a recommender system, (ii) disease, and gene association in a biological network, (iii) scientists and publications in a bibliographic network. Thus, the general diffusion-based method designed for homogenous graphs might not be suitable for the heterogeneous network setting.

Similarly, most of the graph-based diffusion-based method for node classification designed for the homophilic labels meaning similar nodes have the same labels. Some networks can exhibit heterophilic labels indicating similar nodes might have different labels. Moreover, the real-world networks might have interconnected (a network of a network) structures where one node can participate in many layers or networks. In such a situation, the current label propagation algorithm can misclassify the nodes by spreading one layer's information to another layer. Thus, we need a diffusion method which is flexible and can control the range of diffusion from short-range to long-range and can be applied in various label types.

This thesis mainly deals with predictive analytics in the graphs. The section below summarizes the research questions that are designed explicitly as a use-case for addressing the problem of (i) link prediction, (ii) node classifications and (iii) graph regression.

1.3 Research Questions

1.3.1 How can we predict the links between dissimilar nodes in graphs?

In most of the link prediction studies, graph-based prediction models follow the presumption that the network is homogeneous, i.e., nodes are of the same type, and links connect them with the same semantic meanings. For example, in a product diffusion setting using word of mouth, the nodes in the network are users, and the links in the network are friendship relationships. However, in the real-world, nodes are connected via different relationships. One typical example is the bibliographic network where scientists and publications are two different nodes connected by the “author” relationship, and publications are connected by the “citation” relationship. In this example, we observe that the node publication is shared between 2-layers, one is a bipartite graph between scientists and publications, and another is a homogeneous graph between publications. In such settings, what will be the performance of ordinary diffusion algorithms? While previous research focused on predicting the links between similar entities (nodes) using a homogeneous graph using a diffusion approach, there is a gap in how diffusion can be achieved using two different graphs, one for labeling the nodes and another for propagation.

1.3.2 Can we integrate heterogeneous information for link prediction between dissimilar nodes in graphs using diffusion methods?

When multiple graphs are available, it is natural to treat them as supplementary information to strengthen the link prediction task. For example, in a graph of two entities in a 2-layered setting: “user-[purchase]-item-[similar]-item”, there are two graphs: one for “user-[purchase]-item” and the other for “item-[similar]-item”. If the task is to predict links between “user” and “item,” the traditional approach is to follow one of the following methods. The first option is to directly use a matrix factorization method to predict the links between users and items. The second option is to use diffusion or propagation approach of the user information in the “item-[similar]-item” graph to predict new links between user and items. Our intuition is, therefore, to combine these kinds of graphs by combining the methods for a link prediction task. In this thesis, we search for possible diffusion methods that can be combined with matrix factorization for the link prediction task.

1.3.3 Do graph-based diffusion methods adapt to different variants of graphs (multiplex/homogeneous/manifold) for node classification?

Classical diffusion-based methods in graphs follow the assumption that similar nodes have similar labels. This principle is also known as homophily. Later, this idea is expanded to heterophily, which means that different nodes are more likely to attach than similar nodes. Along with the two popular cases, there is also the case of a mixture of homophily and heterophily nodes. For example, in bibliographic scientific author networks, computer scientists co-authors papers with fellow computer scientists but some computer scientists do interdisciplinary research and co-author papers with biologist. Prior work mostly only deals with one of these labels to propagate in the graph. Although label propagation algorithms work fairly well in the majority of networks with a single layer, we do not explicitly know how diffusion algorithms behave in a multiplex network. In such a network, where nodes overlap between the layers, there is a high possibility of node misclassification by using ordinary label propagation algorithms. In this thesis, we search for possible graph diffusion methods for the above-described problems.

1.3.3.1 Can graph-based diffusion methods be applied to regression problems using few labeled data points?

Semi-supervised classification (SSC) using graphs is popular due to its ability to solve pattern recognition in machine learning problems. Most studies deal with the application of SSC techniques in many real-world problems in contrast to Semi-Supervised Regression (SSR). In SSC, the independent variable Y_i is constrained to have only a finite number of possible values, whereas in SSR, Y_i is assumed to be continuous. Hence, SSC algorithms designed for graph partitioning do not apply to the more general SSR problem. In this thesis, we extended our graph-based diffusion approach to the graph regression problem in manifold data.

1.4 Contributions

The main contributions of this thesis are as follows:

- We proposed a 2-layered graph to model the diffusion process for predicting links between dissimilar nodes. We showed how we could use the labeling information of one network layer to propagate to another network layer. During this process, we demonstrated the use of standard diffusion methods for such tasks in biomedical and social media network data.
- We proposed the novel combination of a matrix factorization method and a diffusion-based method in 2-layered graphs. Our approach shows the effectiveness of this integration by computing ranked prediction of initially unknown links between dissimilar nodes.
- We introduced the new variant of heat diffusion algorithms called boundary-based heat diffusion (BHD) algorithm. This algorithm can perform long-range (global) and short-range (local) diffusion in a network with different labeled properties. The algorithm is applied to node classification and also extended to regression using manifold data.

1.5 Thesis Outline

Figure 1.3 shows our approach, which concerns each of the core contribution chapters of the thesis.

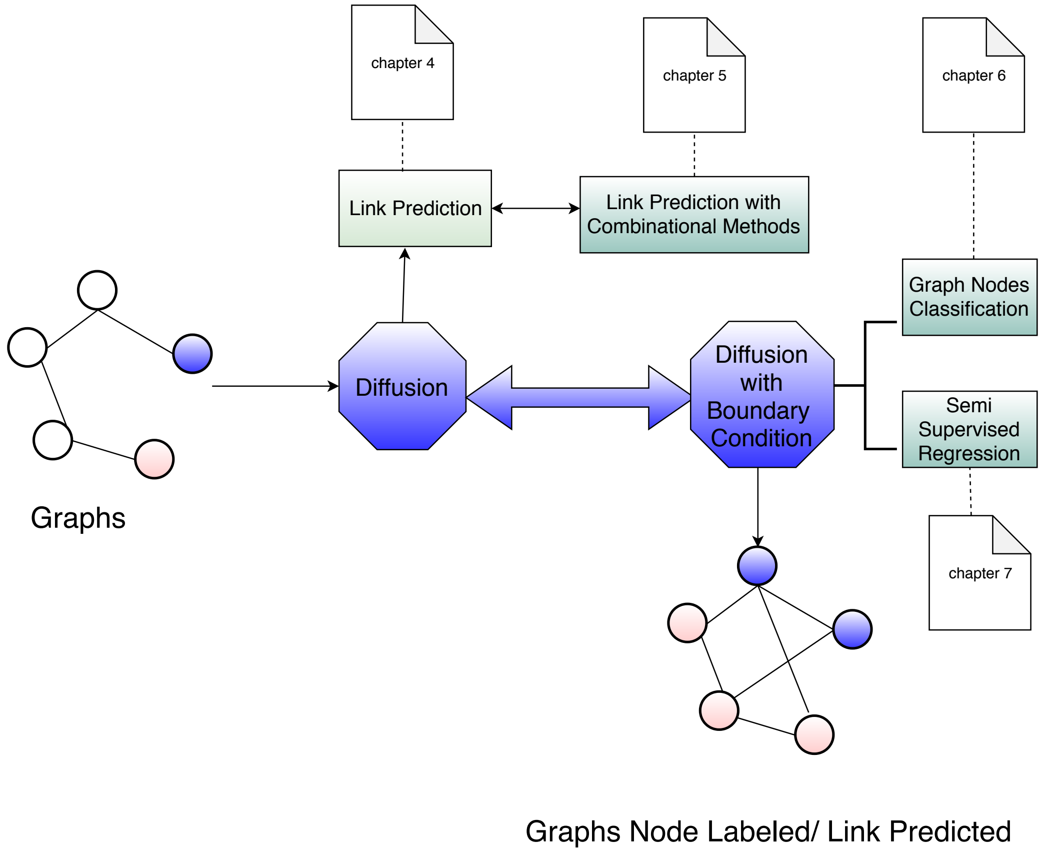

chapter 4
chapter 5
chapter 6
Link Prediction
Link Prediction with Combinational Methods
Graph Nodes Classification
Diffusion
Diffusion with Boundary Condition
Semi Supervised Regression
chapter 7
Graphs
Graphs Node Labeled/ Link Predicted

FIGURE 1.3: Diffusion framework and thesis contributions outline.

The first part of this thesis has three chapters. The current chapter (1) is an introduction. Chapter 2 provide the background of the work, which contains the research foundations. Chapter 3 contains the literature survey and describes the related work on diffusion in graphs for link prediction and node classification tasks, focusing on the approaches that fall in the same category as ours.

The second part is the core of the thesis. It contains the four contribution chapters. Chapter 4 discusses link prediction using a diffusion-based framework in a novel 2-layered graph applied to biomedical and social media domains. Chapter 5 presents our novel approach to combine matrix factorization and diffusion methods in a 2-layered graph to strengthen the link prediction task. Chapter 6 contains our novel diffusion method with boundary conditions, which is a variant of heat diffusion methods. In chapter 7, we expand this idea to semi-supervised graph-based regression problems in manifold data.

Chapter 8 discuss the main findings and limitations of the study and provide future research direction that this work has opened.

Chapter 2

Foundations

This chapter provides the fundamental concepts and techniques that are used in this thesis. We start by the basic notation of graph based analysis. Then we introduce predictive models in graphs, which are able to predict the likelihood of link creation and node classification.

2.1 Essential Concepts

2.1.1 Essential Concepts of Graph

Graph A graph is an ordered pair $G = (V, E)$, which consist of a set V of vertices or nodes and a set E of edges or links. A typical way to represent a graph is by *adjacency matrix* A, of dimension $n \times n$ whose element A_{ij} is equal to 1 if there is an edge in E between nodes i and j, and 0 otherwise.

Homogeneous Graph A graph is called a *homogeneous graph* if the set of the nodes and the set of edges are of the same type.

Heterogeneous Graph A graph is called a *heterogeneous graph* if the set of the nodes and set of edges can have different types.

Directed Graph A directed graph $G = (V, E)$ is a graph where $E {\subseteq} VXV$ is a set of ordered pairs from V.

Undirected Graph An undirected graph $G = (V, E)$ is a graph where E are unordered pairs of elements of V

Weighted Graph With each edge E of G let there be associated a real number $w(E)$, called its weight. Then G, together with these weights on its edges, is called a **weighted graph**.

Bipartite Graph A graph $G = (V, E)$ is called *bipartite* if its vertex set V can be partitioned into two disjoint subsets $V = V_1 \cup V_2$, such that every edge has the form e = (a,b) where $a \in V1$ and $b \in V2$.

Multiplex Graph A multiplex graph is a graph:

$$G =< V, E_1, E_2, ..., E_\alpha : E_k \subseteq V \times V \; \forall k \in \{1, 2, ..., \alpha\} >$$

where V is a set of nodes, E_k is a set of edges of type k and α denotes the number of layers.

Subgraph A *subgraph* is a graph $G' = (V', E')$ of $G = (V, E)$ if and only if $V' \subseteq V$ and $E' \subseteq E$.

Path A *path* through $G = (V, E)$ is a progression of subset of edges in E that joins a progression of nodes in V. The nodes on the path are unique, i.e. any node v of E can only appear once in a path.

2.2 Graph Metrics

Graph metrics are measures that describe graphs. There exist various types of measures from simple metrics, such as the number of nodes and edges, to more complex ones, such as node degrees, which tell how each node is connected. In this section, we describe some of those metrics that will be used in the rest of this thesis. In the next section, we will describe the graph based influence metrics.

2.2.1 Order

The *order* of the graph $G = (V, E)$, is the count of its nodes, i.e. $|V|$.

2.2.2 Size

The *size* of the graph $G = (V, E)$, is the count of its edges, i.e. $|E|$.

2.2.3 Neighborhood

The *neighborhood* of a node v in a graph $G = (V, E)$, is the set of nodes adjacent to v denoted by $N(v)$.

2.2.4 Node Degree

The *degree* of a node in a graph $G = (V, E)$, is the total number of nodes neighboring to the node. Formally the degree of a node is defined as the number nodes of its neighborhood for any node $v \in V$, $deg(v) = |N(v)|$.

2.2.5 Indegree

The *indegree* of a node $v_i \in V$ in a directed graph $G = (V, E)$, is the number of nodes that are directed to v_i. The indegree of node v_i is equal to the number of edges of the form $e_k =< v_j, v_i >$, for all $e_k \in E$, and all $v_j \in V$.

In social network analysis, indegree is also interpreted as **popularity** [Wasserman and Faust, 1994].

2.2.6 Outdegree

The *outdegree* of a node $v_i \in V$ in a directed graph $G = (V, E)$, is the number of nodes that are directed from v_i. The outdegree of node v_i is equal to the number of edges of the form $e_k =< v_i, v_j >$, for all $e_k \in E$, and all $v_j \in V$.

In social network analysis, outdegree is also interpreted as **gregariousness** [Wasserman and Faust, 1994].

2.3 Graph Based Link Prediction

In this thesis we have used two variants of link prediction in graphs (i) Node-based and (ii) Path-based.

2.3.1 Node-Based

Common Neighbors

Common neighbors is used for node-based link prediction. It is computed as [Liben-Nowell and Kleinberg, 2007]:

$$score(x, y) = |N(x) \cap N(y)|$$

where $N(x)$ is the set of nodes adjacent to node x, and $N(y)$ is the set of nodes adjacent to node y.

Jaccard's Coefficient

Jaccard Coefficient of nodes x and y is defined as [Liben-Nowell and Kleinberg, 2007]:

$$score(x, y) = \frac{|N(x) \cap N(y)|}{|N(x) \cup N(y)|}$$

Adamic/Adar

Adamic-Adar index of nodes x and y is defined as [Liben-Nowell and Kleinberg, 2007]:

$$score(x, y) = \sum_{z \in N(x) \cap N(y)} \frac{1}{log|N(z)|}$$

where $N(z)$ is the set of nodes adjacent to z.

Preferential Attachment

Preferential Attachment of nodes x and y is defined as [Liben-Nowell and Kleinberg, 2007]:

$$score(x, y) = |N(x)| \cdot |N(y)|$$

Resource Allocation

Resource Allocation index of nodes x and y uses graph community information to compute the resources between the nodes. It is computed as [Soundarajan and Hopcroft, 2012]

$$score(x,y) = \sum_{z \in |N(x) \cap N(y)|} \frac{1}{|N(z)|}$$

2.3.2 Path-based

Katz Centrality

Katz centrality gives the relative influence of the nodes in the network. The measure of this centrality depends upon the number of immediate and distant neighbors [Katz, 1953]. The connection made to distant neighbors are penalized by an attenuation factor α. If the distance between the two nodes is k then the attenuation factor is given by α^k. In the case of directed graphs, the Katz centrality measure quantifies the ability of node n to send out information along the directed links. The Katz centrality measure is given by:

$$c_i = \textstyle\sum_{k=0}^{\infty} \sum_{j=1}^{n} \alpha (A^k)_{ij}$$

where A is the adjacency matrix: if node i and j are connected then $A_{ij} = 1$, else $A_{ij} = 0$. α is the attenuation factor and k is the distance between the nodes.

PageRank

The PageRank is a link analysis algorithm that provides the relative importance of web pages, which was originally developed by Brin and Page [Page et al., 1999]. The pagerank of a page or node i is the sum of the contributions from the incoming edges or the links. There is a constant damping factor f which is the probability at each page that a “random surfer” will get bored and jump into another page. $(1 - f)$ is the factor that is added to each of the nodes, because if the node has an outdegree of 0, then its page rank will be 0. In order to avoid this situation this constant value is added to the PageRank. Formally, PageRank is defined as:

$$PR(p_i) = \tfrac{1-f}{N} + f \textstyle\sum_{p_j \in M(p_i)} \tfrac{PR(p_j)}{L(p_j)}$$

where p_1, p_2, $...p_N$ are the pages and $M(p_i)$ is the set of the source pages that link p_i, $L(p_j)$ outgoing links on page p_j, and N is the number of pages. The damping factor is often set to value of 0.85 [Brin and Page, 2012].

Rooted PageRank (RPR)

Rooted PageRank (RPR) [Liben-Nowell and Kleinberg, 2007] is a variant of PageRank, which is the algorithm used by search engine for ranking the search results. The rank of a node in the graph is proportional to the probability that the node will be reached through a random walk on the graph. Further, there is a parameter ϵ which specifies how likely the algorithm is to visit the node's neighbors rather than starting over. Let D be a diagonal matrix with $D_{i,i} = \sum_j A_{i,j}$, A the adjacency matrix of the graph,v the teleportation vectors of the nodes, f is the damping factor also known as the restart probability, $(0 \leq f \leq 1)$ normally chosen as 0.85 [Page et al., 1999] and I the identity matrix. The measure is defined as:

$$RPR = (1 - f)(I - fD^{-1}A)^{-1}v$$

Heat Kernel Diffusion Ranking

For a graph G, the transition probability matrix W is defined as $W = D^{-1}A$. Let us define $L = I - W$, where I is the identity matrix [Chung, 2007], diffusion rate α and a preference vector f_0, then the influence score of each node f_α is given by:

$$f_\alpha = \sum_{t=0}^{\infty} \frac{(-\alpha)^t}{t!} f_0 L^t = f_0 e^{-\alpha t L}$$

Where t is a non negative value, also called the timestamp.

2.4 Node Classification

Node classification, also known as node attribute inference, is the problem of inferring missing or incomplete attribute values of some nodes, given attribute values of other nodes in the network.

In the case where there is not an available graph, then one can construct similarity graph by calculating similarities among all the training data features for node classification. Given training data $\{(x_i, y_i)\}_{i=1}^{l}, \{x_j\}_{j=l+1}^{l+u}$, the vertices are the labeled and unlabeled instances $\{(x_i)\}_{i=1}^{l} \cup \{x_j\}_{j=l+1}^{l+u}$. l is the labeled data, u is the unlabeled data, x is the instance of the data, and y is the label of the data. The graph's edges are usually undirected. An edge between two vertices x_i and x_j represents the similarity of the two instances. The edge weights can be computed from one of the following methods:

- Gaussian Kernel. This kernel is also called the Radial Basis Function (RBF) kernel. Every node pairs x_i, x_j in the graph is connected by an edge. The function for computing the edge weight is given by,

$$w_{ij} = exp\left(- \frac{||x_i - x_j||}{2\sigma^2} \right)$$

  where σ is known as the bandwidth parameter which controls the edge weight. The weight is 1 when $x_i = x_j$, and 0 when $||x_i - x_j||$ approaches to ∞.

- kNN. kNN graph refers to the structure that keeps top-k nearest neighbors for each nodes. If node x_i is among k-NN of node x_j then nodes x_i and x_j are connected. If nodes are connected, the edge weight w_{ij} is either the constant 1, in the case of an unweighted graph or a function of the distance. If nodes x_i, x_j are not connected then $w_{ij} = 0$.

2.4.1 Network Matrices and Diffusion Functions

Once the graph is constructed, it can be viewed through the lens of matrices. Table 2.1 shows the matrices that can be derived from graphs for the diffusion process.

Matrix Type	**Symbol**	**Representation**
Adjacency	A	–
Degree	D	$d_{ii} = degree$
Random Walk	P	$A^T D^{-1}$
Laplacian	L	$D - A$
Normalized Laplacian	$\mathcal{L}$	$I - D^{\frac{-1}{2}} A D^{\frac{-1}{2}}$
Normalized Adjacency	$\mathcal{A}$	$D^{\frac{-1}{2}} A D^{\frac{-1}{2}}$

Table 2.1: Matrices derived from networks.

The popular variants of diffusion functions are shown in Table 2.2:

PageRank	**Katz scores**	**Heat Kernel**
$f = \sum_{k=0}^{\infty}(1-\alpha)\alpha^k P^k s$	$f = \sum_{k=0}^{\infty}(1-\alpha)\alpha^k A^k s$	$f = \sum_{t=0}^{\infty}\frac{(-\alpha)^t}{t!}L^t s$
$(I-\alpha P)f = (1-\alpha)s$	$(I-\alpha A)f = (1-\alpha)s$	$f = e^{-\alpha t L}s$

*s: seed vector, k: the number of steps, α : the diffusion coefficients, I: identity matrix, P: transition matrix, t: time stamp, L: laplacian matrix.

Table 2.2: Different variants of diffusion functions.

In the next section, we demonstrate the supervised classification model that we use in this thesis for evaluation purposes.

2.5 Supervised Classification Model

Classification models are widely used to classify objects into one of several predefined categories. More formally, classification is the task of learning a **target function** f that maps each feature vector x to one or more predefined class labels y [Tan, 2018].

2.5.1 Support Vector Machine

A **Support Vector Machine (SVM)** is a supervised machine learning algorithm which is based on the idea of structural risk minimization [Vapnik and Vapnik, 1998] to find the hypothesis which is likely to assure the lowest prediction error. The major benefit of using SVM for classification is that it can be applied to linearly inseparable data. In SVM, non-linear solutions can be conveniently found using a **kernel trick**. This trick is by casting the features to a high dimensional space or **support vectors** in which the problem becomes linearly separable.

SVM aims to find the following linear function:

$$f(x) = w^T x + b,$$

where $f : \mathbb{R}^{|x|} \rightarrow \mathbb{R}$ maps a vector to a real value, $w \in \mathbb{R}^{|x|}$ is a weight vector and $b \in \mathbb{R}$ is called the bias. The label of the classification is determined based on the the following equation:

$$y_i = \begin{cases} +1, \text{if } w^T x + b > 0 \\ -1, \text{if } w^T x + b < 0 \end{cases}$$

where y_i is the predicted label for the data point i.

2.6 Factorization methods in Graphs

Factorization methods are among the popular techniques for machine learning in graphs. In this thesis, we have used non-negative matrix factorization methods.

2.6.1 Non negative Matrix Factorization (NMF)

For a given bi-adjacency matrix $Y = [y_{ij}] \in \mathbb{R}^{mxn}$, where rows and column represent nodes, and non-zero elements represent known links, the goal is to complete this matrix for any node pairs. In the matrix Y, each element y_{ij} $(1 \leq i \leq m, 1 \leq j \leq n)$ belongs to boolean values of $[0, 1]$. Here $y_{ij} = 0$ means that no weight is provided between the node pairs i and j, while $y_{ij} = 1$, is the weight between the node pairs i and j.

To predict the initial weights, the NMF approach uses all the known weights to decompose the matrix Y into the product of two low-rank, latent feature matrices, one for the node type S, $S_{m \times r}$, and the other for node type, $D_{n \times r}$, so that:

$$Y \approx \hat{Y} = SD^T = \underbrace{\begin{bmatrix} s_1^T \\ s_2^T \\ . \\ . \\ . \\ s_m^T \end{bmatrix}}_{m \times r} \underbrace{\begin{bmatrix} d_1 & d_2 & ... & d_3 \end{bmatrix}}_{r \times n} \tag{2.1}$$

The latent feature vectors for node type s and node type d are r dimensional, where $r \ll min\{m, n\}$. The predicted weights for the node pair (s,d) is given by $\hat{y} = s^T d$. The NMF factorization problem in Equation 2.1 can be resolved by solving the optimization problem,

$$\min_{S \in \mathbb{R}^{m \times r}, D \in \mathbb{R}^{r \times n}} \| \left(Y - SD^T\right) \|_F^2 \quad \text{such that} \quad S, D \geq 0 \tag{2.2}$$

where F is the Frobenius norm.

2.7 Performance Metrics

Performance metrics assess the efficiency of algorithms for a classification task. We used the following metrics in this thesis:

2.7.1 Precision

In a classification problem, **Precision** is defined as the number of true positives (t_p) over the number of true positives plus the number of false positives (f_p). Formally:

$$Precision = \frac{t_p}{t_p+f_p}$$

where t_p is the number of True Positives (correctly identified), f_p is the number of False Positives (incorrectly identified).

2.7.2 Recall

In a classification problem, **Recall** is defined as the number of true positives (t_p) over the number of true positives plus the number of false negatives (f_n). Formally:

$$Recall = \frac{t_p}{t_p+f_n}$$

where f_n is the number of False Negatives (incorrectly rejected).

2.7.3 F1 Score

F1 score captures the trade off between precision and recall of a classifier model. It is a combined metric of both precision and recall and computed as the **harmonic mean** between these metrics.

$$F_1 = \frac{2XPrecisionXRecall}{Precision+Recall}$$

2.7.4 Accuracy

Accuracy is the metric for evaluating classification models and it can be calculated as follows:

$$Accuracy = \frac{t_p+t_n}{t_p+t_n+f_p+f_n}$$

where t_p is the number of True Positives, t_n is the number of True Negatives, f_p is the number of False Positives, and f_n is the number of False Negatives.

2.7.5 Receiver Operating Characteristics Curve (ROC)

ROC curve plots the rate of true positives as a function of the rate of false positives with different decision thresholds. The ROC score is the area under this curve (AUC). A perfect classifier which places all positive examples above all negative examples, receives a ROC score of 1, and a random classifier receives a score of approximately 0.5.

2.7.6 Precision-Recall Curve (PR)

A precision-recall curve is a plot of the precision and the recall for different thresholds. It defines how good a model is at predicting the positive example or correctly classifying classes. The PR curve does not make use of the true negatives (correctly rejected) cases. It only takes account of the correct prediction of the positive class.

2.7.7 Root Mean Square Error (RMSE)

RMSE is used for measuring the accuracy of a numeric prediction model. It uses the root of the average of all the errors. RMSE is computed as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})}$$

where y_i is the observed value, $\hat{y_i}$ is the predicted value and n is the number of elements for prediction.

Chapter 3

Related Work

In this chapter, we review the work that tackles the same problems as this thesis, the methods used, and their strengths and weaknesses. In the first part, we focus on the Link Prediction task, while in the second part, we focus on node classification. We consider these two tasks from a graph-based perspective.

3.1 Related Work on Link Prediction and Node Classification in Graphs

For the first scenario, we focus on one of the fundamental problems in graph-based predictive analysis known as Link Prediction. There are different algorithms based on Markov Chains and statistical models proposed by the artificial intelligence (AI) community. However, their works have not caught up with the current progress of the study of networks, especially, combining different methods available for Link Prediction such as Matrix Factorization [Menon and Elkan, 2011], and diffusion-based method [Kashima et al., 2009], considered as established methods for Link Prediction. In chapters 4 and 5, we propose an approach to diffusion framework that overcomes some of these limitations.

For the second scenario, we focus on graph-based semi-supervised methods. Graph-based methods start with a graph where a few nodes are labeled. Furthermore, a large number of nodes are unlabeled. The (weighted) edges reflect the similarity of nodes. The presumption is that nodes connected by large-weight edges tend to have the same label, and labels can propagate throughout the graph. This kind of methods is also called semi-supervised classification (SSC). The goal of SSC is to train a classifier f from both the labeled and unlabeled data, such that it is better than the supervised classifier trained on the labeled data alone. The strength of semi-supervised learning is that it can attain the same level of performance as supervised learning, but with fewer labeled instances for some cases [Zhu and Goldberg, 2009]. It ultimately reduces the labeling effort, which leads to cost reduction. All, the graph-based label propagation relies on the diffusion framework, but they possess differences in the style of propagation. Another limitation is that these algorithms are only applied in a single layer of homogeneous graphs. It affects the classification performances of the algorithms if applied in multiple layers. In chapter 6, we show how we could achieve better performance by a boundary-based diffusion model for a label propagation task in multiple layers. later in chapter 7, we extended a boundary based diffusion approach that can address the limitations of the previous approach, which only performs classification by applying it in regression problems.

3.2 Link Prediction - The Big Picture

A link is a connection between two nodes in a network. Many social, biological, financial, technological, and information systems can be described by networks, where nodes represent individuals, biological elements (proteins, genes, drugs), computers, web users, financial institutions, documents, and links denote the interactions or relations between nodes. The interdisciplinary aspects of the study of complex networks have, therefore, become a typical focus of many branches of science. Great efforts have been made to understand the evolution of networks [Albert and Barabási, 2002, Dorogovtsev and Mendes, 2002], the relations between structures and functions [Boccaletti et al., 2006, Newman, 2003], and the network characteristics [Costa et al., 2007]. An important scientific issue relevant to network analysis is information retrieval [Salton and McGill, 1983], the field of computer science, which aims at finding relevant information from large corpora [Manning et al., 2010].

The Link Prediction task is prevalent due to its applicability and usefulness in a great variety of contexts. For example, in a biomedical context, it requires a vast amount of laboratory experiments to determine associations playing a role in protein interaction [Shin et al., 2006]. This process is time-consuming and expensive. Due to this, computational techniques are desirable for performing link association. In the context of social networks, there is a problem of missing data [Kossinets, 2006], where Link Prediction techniques can be useful. Link Prediction techniques are widely prevalent in our daily lives too, for instance, recommending people we may know, but we are not yet connected with in our social networks [Liben-Nowell and Kleinberg, 2007]. Another example is product recommendation, where we could be interested in electronic commerce [Chen et al., 2005].

The whole Link Prediction process can be categorized into four broad parts, shown in Figure 3.1:

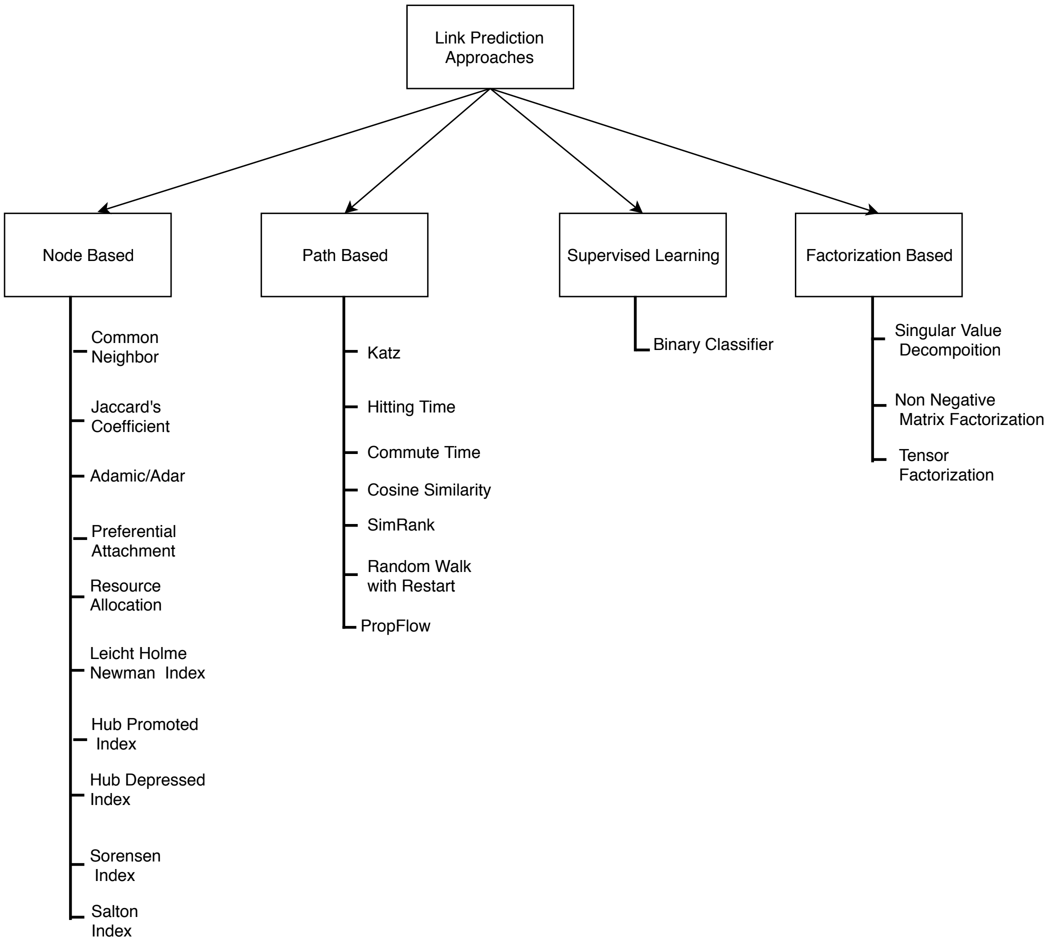


Figure 3.1: Overview of Link Prediction techniques.

3.2.1 Node-Based Methods

Node-based methods are a straightforward way to carry out the Link Prediction task. These methods assume that similar nodes tend to form links with other similar nodes. These approaches define a similarity measurement function $s(x, y)$, which assigns a score known as a similarity for each pair of nodes x and y. This score is computed between the unseen node pairs. Thus, unseen links rank according to their scores. The node pairs with higher scores are more likely to be connected than those with lower scores. Although similarity-based algorithms are popular due to their simplicity, the definition of node similarity is an important challenge. Another important observation is about the impact of the difference in prediction results based on similarity measurements.

3.2.1.1 Common Neighbor (CN)

In this case, the similarity between two nodes is the number of shared neighbors between both nodes [Liben-Nowell and Kleinberg, 2007]. It is safe to assume that, if two nodes share many common neighbors, they are more likely to link with each other. Different studies have confirmed this hypothesis by observing a correlation between the number of shared neighbors between pairs of nodes and the probability of being linked [Newman, 2001]. The similarity function is then:

$$score(x, y) = |N(x) \cap N(y)|$$

where $N(x)$ is the set of nodes adjacent to node x, and $N(y)$ is the set of nodes adjacent to node y. This approach is simple but performs surprisingly well on most real-world networks and beats very sophisticated approaches.

3.2.1.2 Jaccard Coefficient (JC)

The JC method is a widely used metric in information retrieval systems initially proposed by Paul Jaccard (1868-1944). These metrics capture the similarity and diversity of sample sets [Jaccard, 1901]. The similarity function is the ratio of shared neighbors in the complete set of neighbors for two nodes:

$$score(x, y) = \frac{|N(x) \cap N(y)|}{|N(x) \cup N(y)|}$$

This metric is the variant of the CN method with a penalization for each non-shared neighbor.

3.2.1.3 Adamic Adar (AA)

This metric refines the counting of common neighbors by assigning the less-connected neighbors more weights. Each counted neighbor is logarithmically penalized by its appearance frequency [Adamic and Adar, 2003]. The similarity is calculated as:

$$score(x, y) = \sum_{z \in N(x) \cap N(y)} \frac{1}{log|N(z)|}$$

3.2.1.4 Preferential Attachment (PA)

Many real-world network node degrees follow the power-law distribution [Adamic and Huberman, 2000] resulting in representing scale-free networks [Barabási and Albert, 1999]. Preferential attachment is the mechanism that can generate evolving scale-free networks, where the probability that a new link is connected to the node is proportional to the degree of the nodes. This concept in social networks can also be viewed as that users with many friends tend to create more connections in the future. This model leads to the concept of "the rich get richer" in a scale-free network. The similarity between two nodes, according to this model can be estimated as

$$score(x, y) = |N(x)| \cdot |N(y)|$$

This measure does not rely on shared neighbors so that it can also apply in a nonlocal context. However, applying this idea to a global context might lead to poor prediction performance.

3.2.1.5 Resource Allocation (RA)

The foundation of resource allocation is complex networks where every node sends the unit of resource to its neighbors [Zhou et al., 2009]. For any two nodes, x and y, node x sends some resources to node y via its common neighbors, which play the role of resource transmitters.

The similarity between two nodes, according to this model can be estimated as

$$score(x, y) = \sum_{z \in |N(x) \cap N(y)|} \frac{1}{|N(z)|}$$

3.2.1.6 Leicht–Holme–Newman Index (LHNI)

This metric gives high similarity to node pairs that have many common neighbors compared to the expected number of such neighbors [Leicht et al., 2006]. It is computed as

$$score(x, y) = \frac{|N(x) \cap N(y)|}{|N(x)| \cdot |N(y)|}$$

This method is a more sensitive measure of structural equivalence than other methods such as Salton [Salton and McGill, 1983] or the Jaccard metrics.

3.2.1.7 Hub Promoted Index (HPI))

This index was identified by studying community structures in metabolic networks [Ravasz et al., 2002]. These networks exhibit a hierarchical structure with small, highly internally connected modules that are also highly separated from each other. The property of this index is that the links adjacent to hubs are likely to obtain a higher similarity score. This index is then calculated as:

$$score(x,y) = \frac{|N(x) \cap N(y)|}{min(|N(x)|, |N(y)|)}$$

3.2.1.8 Hub Depressed Index (HDI)

This is similar metrics to HPI but with the opposite goal [Ravasz et al., 2002]. It gives links adjacent to hub a lower score. This index is computed as:

$$score(x,y) = \frac{|N(x) \cap N(y)|}{max(|N(x)|, |N(y)|)}$$

3.2.1.9 Sorenson Index (SO)

This index is used for ecological community network data [Sørensen, 1948]. The similarity between the nodes is computed as:

$$score(x,y) = \frac{2|N(x) \cap N(y)|}{|N(x)| + |N(y)|}$$

3.2.1.10 Salton Index (SI)

This index is also termed as the cosine similarity [Salton and McGill, 1983] metric and is closely related to the Jaccard similarity metrics. In practical settings, the Salton index yields a value that is approximately twice the Jaccard index [Hamers et al., 1989]. This metric is computed as:

$$score(x, y) = \frac{|N(x) \cap N(y)|}{\sqrt{|N(x)||N(y)|}}$$

3.2.2 Comparisons of Different Node-Based Algorithms

Table 3.1 shows a qualitative comparison among the node-based algorithms. These algorithms do not require parameters for optimization. We saw that the four metrics: CN, AA, PA, and RA, are not normalized. It implies that the similarities given by these metrics cannot be used for ranking of the node pairs. They do not represent the likelihood of the formation of links. The node-based metrics are intuitive and have a good reference in terms of theoretical foundations. These approaches are faster than other approaches to Link Prediction. Due to their simplicity, these algorithms support handling Link Prediction in dynamic networks, such as online social networks.

Another important aspect of theses algorithms is their computational complexity. If n is the average number of neighbors in a network, for two nodes x and y, then the total time to find the neighbors of the node is $\mathcal{O}(n)$. Therefore, to calculate the intersection between the two nodes is $\mathcal{O}(n^2)$. From Table 3.1, we observe that for methods CN, SO, SI, HPI, HDI, and LHNI have a $\mathcal{O}(n^2)$ complexity due to the calculation of the set intersections. For JC, the time complexity is $\mathcal{O}(2n^2)$ due to computation of the union and intersection of the two sets. Similarly, AA and RA also need to compute the intersection of two sets and find neighbors of common neighbors; therefore, their time complexities is $\mathcal{O}(2n^2)$. The only metric which has lower computational complexity is PA, which is $\mathcal{O}(2n)$. This metric only needs to find neighbors of node x and y but do not need to find the shared neighbors. Thus this measure can also be applied in non-local contexts. Though this metric is fast to compute its prediction accuracy is usually poor when applied as a global measure.

Most of the node-based Link Prediction metrics work for unweighted graphs. Only the SI metric supports weighted graphs. The study by [Sarkar et al., 2011] demonstrated that carefully using a weighted count of common neighbors often outperforms the unweighted count. Similarly, the node-based metrics are applied to predict the links between similar nodes. None of these metrics can be directly applied to predict the links between nodes in heterogeneous networks.

Methods	Time	Simple to	Normalization	Support	Weighted	Parameters	Dissimilar	References
	Complexity	compute		Directions	Edge		Nodes	
CN	$\mathcal{O}(n^2)$	✓	×	✓	×	×	×	[Liben-Nowell and Kleinberg, 2007]
JC	$\mathcal{O}(2n^2)$	✓	✓	✓	×	×	×	[Jaccard, 1901]
AA	$\mathcal{O}(2n^2)$	✓	×	✓	×	×	×	[Adamic and Adar, 2003]
PA	$\mathcal{O}(2n)$	✓	×	✓	×	×	×	[Barabási and Albert, 1999]
RA	$\mathcal{O}(2n^2)$	✓	×	✓	×	×	×	[Zhou et al., 2009]
LHNI	$\mathcal{O}(n^2)$	✓	✓	✓	×	×	×	[Leicht et al., 2006]
HPI	$\mathcal{O}(n^2)$	✓	✓	✓	×	×	×	[Ravasz et al., 2002]
HDI	$\mathcal{O}(n^2)$	✓	✓	✓	×	×	×	[Ravasz et al., 2002]
SO	$\mathcal{O}(n^2)$	✓	✓	✓	×	×	×	[Sørensen, 1948]
SI	$\mathcal{O}(n^2)$	✓	✓	✓	✓	×	×	[Hamers et al., 1989]

TABLE 3.1: Qualitative comparison between different node based Link Prediction algorithms.

Although there are many existing neighbor-based metrics, in practical applications, one should recognize the right metrics according to the characteristics of the network, because many experiment evaluation results have shown that there is no one metric that can beat all the metrics in terms of prediction accuracy. The major drawback of the node-based approaches is that these metrics only use local information, which restricts the set of nodes similarity computation from neighbors of neighbor nodes. It can be a big drawback as many links are formed at distances greater than two hops in many real-world networks, especially in non-small-world networks [Liben-Nowell and Kleinberg, 2007]. Thus, path-based methods are proposed.

3.2.3 Path-Based Methods

Path-based methods are also called global similarity-based metrics, and use the entire network's topological information for computing links. These methods are not restricted to only computing similarities between two nodes. The path-based methods are computationally more expensive than the neighborhood-based approaches. It is due to the computation of link scores between every node pairs of the graphs. Below are known algorithms for Link Prediction using path-based approaches.

3.2.3.1 Katz

This metric is based on the summation of all the possible paths between the two node pairs. The sum of the paths is exponentially damped by their length, meaning that the shorter paths have more weights than the longer paths [Katz, 1953]. Mathematically, it is computed as:

$$s(x,y) = \sum_{l=1}^{\infty} \beta^l.|paths_{xy}^{(l)}| = \beta A_{xy} + \beta^2(A^2)_{xy} + \beta^3(A^3)_{xy} + ...,$$

where $paths_{xy}^{(l)}$ is the set of all paths with length l which connect nodes x and y. β is a free parameter, known as the attenuation or damping parameters, which has a role in controlling the path weights. A minimal value of β yields scores close to CN, making the long paths to have little or no contribution. The similarity between all pairs of nodes can be directly computed using a matrix form and has a closed-form solution:

$$S = (I - \beta A)^{-1} - I$$

3.2.3.2 Hitting Time (HT)

HT is defined as the expected number of steps required for a random walk from node x to hit node y. If $P = D_A^{-1}A$ where diagonal matrix D_A of A has value $(D_A)_{i,i} = \sum_i A_{ii}$ and $P_{i,j}$ is the transition probability of walking on node j from node i, then it is computed as [Pirotte et al., 2007]:

$$HT(x,y) = 1 + \sum_{\omega \in N(x)} P_{x,\omega} HT(\omega, y)$$

3.2.3.3 Compute Time (CT)

HT is not symmetric thus CT is used to count the expected number of steps both from node x to node y and from node y to node x. This metric is computed as [Pirotte et al., 2007]:

$$CT(x,y) = HT(x,y) + HT(y,x) = m(L_{x,x}^{-1} + L_{y,y}^{-1}) - 2L_{x,y}^{-1}),$$

where L^{-1} is the pseudo inverse of the Laplacian matrix $L = D - A$ and m is the number of edges in the network.

3.2.3.4 Cosine Similarity (CS)

This metric is based on L^{-1}, which is the is the pseudo inverse of the Laplacian matrix and the similarity between the nodes x and y is then given as:

$$CS(x,y) = \frac{L^{-1}_{x,y}}{\sqrt{L^{-1}_{x,x} L^{-1}_{y,y}}}$$

3.2.3.5 Spreading Activation (SA)

SA is a node ranking algorithm [Crestani, 1997, Huang et al., 2004], which operates like the PageRank method for searching processes in networks. The search process is initiated by labeling a set of source nodes with weights or "activation" and then repetitively propagating or "spreading" the activation to other nodes connected to the source nodes. SA algorithms are a popular tool to determine the mutual relevance of nodes, particularly in a semantic network [Hartig and Karbe, 2017]. In the context of entity retrieval in knowledge graphs, the SA approach is applied to infer the relevant entities, for example, extracting user interest profiles in a hierarchical knowledge base [Piao and Breslin, 2017, 2018].

3.2.3.6 SimRank (SR)

Here, the two nodes are similar if they are connected to similar nodes. This metric relies on a parameter γ which controls how fast the weight of connected nodes decreases as they get farther away from the original nodes [Jeh and Widom, 2002]. It is computed as:

$$SR(x,y) = \begin{cases} 1, \mathrm{x = y} \\ \gamma . \frac{\sum_{a \in N(x)} \sum_{b \in N(y) SR(a,b)}}{|N(x)||N(y)|}, \text{otherwise} \end{cases}$$

This metric is computationally expensive, which restricts its usage in a large scale network.

3.2.3.7 Random Walk With Restart (RWR)

RWR is the slight modification from the core algorithm used by a web search engine called PageRank [Page et al., 1999]. The rank of a node in a graph is proportional to the probability that the node will be reachable through a random walk [Tong et al., 2006]. There is a parameter in the metric called ϵ, which specifies random jumps to the neighboring node than starting over. If D is the diagonal matrix with $D_{i,i} = \sum_j A_{ij}$. The measure is defined as [Tong et al., 2006] :

$$RWR(x, y) = (1 - \epsilon)(I - \epsilon D^{-1} A)^{-1} v$$

Where v is the preference vector or starting vector.

3.2.3.8 PropFlow (PF)

This metric is similar to RWR. It is concerned with the localized concept of random walk that the probability of a restricted random walk starting at node x and ending at node y in no more than l steps [Vanunu and Sharan, 2008]. The restricted walk allows links for selection based on weights and ends when it reaches node y or revisits any node. If node x and node y are directly linked, then $PF(x, y)$ is computed as

$$PF(x, y) = PF(a, x) \frac{\omega_{xy}}{\sum_{k \in N(x)} \omega_{xk}},$$

where k is x's neighbor, ω_{xy} refers to the weight of the link between node x and node y and a is the previous node of x on a random walk path. If x is the initial node then $PF_{a,x} = 1$. If nodes x and y are indirectly connected, then $PF(x, y)$ is the ensemble of all the shortest paths from x to y.

3.2.4 Comparisons of Different Path-Based Algorithms

Table 3.2 shows the qualitative comparison among the Path-Based algorithms. We saw that the four metrics: HT, CT, CS, and PF, do not require parameter tuning in the graph. Whereas, Katz, SR, and RWR require parameters for the Link Prediction task. From the computational point of view, the models inspired by random walks are faster to compute than other methods because they can compute linearly using the power iteration method. We observed that the SR method has very high computational complexity in comparison to other methods. It will limit its usage for a large scale network. Unlike other random walk processes, PF does not require walk restarts or convergence. This metric employs a modified breadth-first search limited to a height of l. Due to this property, it is faster than RWR, and other random walk inspired models. One of the vital properties of RWR is that using this metric; one can rank the nodes based on overall “importance” (the core of PageRank algorithms) or can bias the random resets towards the relevant nodes of interest.

Methods	**Time Complexity**	**Convergence**	**Parameters**	**Dissimilar Nodes**	**References**
Katz	$\mathcal{O}(n^3)$	✓	✓	×	[Katz, 1953]
HT	$\mathcal{O}(cn^2k)$	✓	×	×	[Pirotte et al., 2007]
CT	$\mathcal{O}(cn^2k)$	✓	×	×	[Pirotte et al., 2007]
CS	$\mathcal{O}(nk^3)$	×	×	×	[Salton and McGill, 1983]
SR	$\mathcal{O}(v^2k^{2l+2})$	✓	✓	×	[Jeh and Widom, 2002]
RWR	$\mathcal{O}(cn^2k)$	✓	✓	×	[Tong et al., 2006]
PF	$\mathcal{O}(cn^2k)$	×	×	×	[Vanunu and Sharan, 2008]

c: time step when convergence is reached. k: sparsity of the graph.

l: number of steps in a random walk.

TABLE 3.2: Qualitative comparison between different path based Link Prediction algorithms.

3.2.5 Supervised Learning

Link Prediction is also approached from the binary classification perspective. For example, predicting "link" and "no link" classes between the node pairs. Let us assume that u, v are nodes that belong to the set of nodes V in the graph, E is the set of edges in the graph $G(V, E)$ and $l^{(u,v)}$ is the label of the node pair instance (u, v). In Link Prediction using classification methods, each non-linked pair of nodes corresponds to an instance that includes the class label and corresponding features that describe the pair of nodes. Therefore, a pair of nodes can be labeled as "link" if there is a link connecting the nodes. Otherwise, the pair labeled as "no link." The label of x and y is defined as follows:

$$l^{(u,v)} = \begin{cases} +1, \text{if (u,v)} \in E \\ -1, \text{if (u,v)} \notin E \end{cases}$$

,

Where +1 means the "link" class, and −1 means the "no link" class. This problem can be approached by a supervised learning method. This method is a powerful technique since it can use any topological property and measure or even any other Link Prediction measure as a feature. The caveat of Link Prediction using a supervised machine learning approach is that it has to deal with the class imbalance problems [Kotsiantis et al., 2006], since almost all real networks are sparse; that is, the number of missing links is extremely high compared to the number of links.

There are various supervised machine learning algorithms such as Decision Tree, Support Vector Machine, Naive Bayes, k-nearest neighbors, multilayer perceptrons, radial basis function networks, and different ensembles of these classifiers applied for Link Prediction tasks [Hasan et al., 2006]. The popular variant of a supervised classification algorithm called random forest obtained good classification accuracy for a Link Prediction task [Cukierski et al., 2011]. The classifier-based methods do not rank probable links, unlike similarity-based methods. This property makes the comparison between theses methods harder since the number of predicted links in each class cannot be ranked in this case.

For Link Prediction, supervised learning algorithms allow us to leverage advances in machine learning for accurate Link Prediction. Further, the algorithms applied in this problem seem to often use the same models with default hyperparameters on datamining software like WEKA[1]. Thus some of their feature sets were more favorable to different models or hyperparameters. Another important observation is that supervised learning does not naturally take into account the network structure exhibited by the data. The model relies on handpicked network statistics as features for prediction. The best choice of features is a difficult problem in a prediction task. If there is a feature between the node pairs, then this approach can be applied to any networks, whether directed, undirected, homogeneous, and heterogeneous.

3.2.6 Comparisons of Different Supervised Learning Methods in Link Prediction

Table 3.3 shows the comparison of some of the supervised classification for Link Prediction.

We observe the following characteristic of supervised Link Prediction:

- For the Link Prediction task, the well-known classification models are used by tuning parameters.
- Most of the supervised models are used to understand the importance of features for Link Prediction.

[1] https://www.cs.waikato.ac.nz/~ml/weka/

References	Features	Supervised Learning Models	Network Types	Strength or Weakness
[Lichtenwalter and Chawla, 2012]	Vertex Collocation Profile (VCP) describing the relationship between vertices in terms of their common membership.	Classification models from WEKA using Bagging and Random Subspaces.	Directed, Weighted, temporal and multi-relational networks.	Can preserve topological information between nodes.
[Li and Chen, 2013]	Graph based features based on random walk paths and node features.	One class SVM kernel.	Bipartite networks of User-Item.	No explicit features generation. Performance is based on kernel function selection.
[Leskovec et al., 2010]	Degree of the nodes and triads in the graph.	Logistic Regression.	Signed Networks.	High accuracy for the sign link prediction.
[Chiang et al., 2011]	Features called longer cycles derived from the graphs .	Logistic Regression.	Signed Networks.	High accuracy for the sign link prediction. This approach is also successful in lowering the false positive rate.
[Pujari and Kanawati, 2012]	Topological features from graph.	KNN, Naive Bayes, Decision Trees, Supervised Rank Aggregation.	Bipartite Networks.	Can aggregate many features but high time complexity.
[De Sá and Prudêncio, 2011]	Features from topology based metrics	J48, LibSVM, LibLinear, Naive Bayes	Weighted Networks.	Weights can improve Link Prediction.
[Scripps et al., 2008]	Node attributes, neighborhood topological feature	Discriminative classifiers.	Directed and Un-directed networks.	Determines the most predictive features for Link Prediction.
[Lu et al., 2010]	Features extracted from the path of the network.	Logistic Regression.	Citation networks.	High prediction accuracy but does not consider other network features.
[Zhang et al., 2014]	Features extracted from the path of the network.	SVM with linear kernel.	Multi-relational network.	Prediction of link with high accuracy.

Table 3.3: Qualitative comparison of the supervised classification models for Link Prediction.

- For the generic node-Link Prediction, the topological or node-based features are used. Whereas for specific Link Prediction, the domain knowledge is also integrated to improve the prediction performance.

- Supervised machine learning is a powerful technique because different studies show increasing accuracy using these methods. At the same time, these models can become computationally expensive in model training and feature selection [Lichtenwalter and Chawla, 2012, Pujari and Kanawati, 2012].

3.2.7 Factorization-Based

Link Prediction is closely related to the problem of matrix completion, where the input is a partially observed matrix of node association scores, and the aim is to complete the matrix. This idea has been used in recommender systems to recommend items for the users. The problem of recommending an item to a user can be modeled into a bipartite weighted Link Prediction problem, where nodes represent users and items, and edges between nodes are weighted according to the preference score.

In addition to the topological information of the graph, there are extra information or covariates for the nodes [Menon and Elkan, 2011]. It can be explained from the network example of protein-protein interactions on top of the topology of the protein interaction graph. There might be features describing the biological properties of each protein. Encoding these features can improve Link Prediction between protein.

There are several matrix factorization based methods that are proposed to do this, which are inspired by the collaborative filtering perspective [Koren et al., 2009]. In the context of a multi-relational graph, the matrix cannot capture the various relationships because the matrix works only in two-dimensions; thus, tensor factorization is used because tensors can capture more than two dimensions in the relational data and be used in a multi-relational Link Predictions [Nickel et al., 2011]. In the next section, we will describe factorization-based methods for Link Prediction.

3.2.7.1 Singular Value Decomposition (SVD)

If A is the adjacency matrix of the graph, to analyze the matrix A is to compute the low rank matrix A_k of rank k which approximates A. The SVD factorization [Van Loan and Golub, 1983] of A for a Link Prediction task [Menon and Elkan, 2011] is then given by :

$$A = U \Sigma V^T,$$

where Σ is a diagonal matrix containing the singular values of A, U and V are the orthogonal matrices of size $m \times k$ and k is the rank of the adjacency matrix A. If Σ is substituted by Σ_k which contains the k largest singular values of A then the resulting matrix $A_k = U\Sigma_k V^T$ will be the good approximation of A.

3.2.7.2 Non Negative Matrix Factorization (NMF)

The adjacency matrix A is a non-negative characteristic matrix where each column represents the characteristic vector of a node and the goal of NMF is to obtain two non-negative matrices U and V such that U is a $m \times k$ matrix and V is $k \times m$ matrix and $k < m$. The product of matrix U and V is very close to matrix A [Chen et al., 2017]:

$$A = U_{m \times k} V_{k \times m}$$

Here, k is the dimension of the latent space (k < n). U consists of the bases of the latent space and is called the base matrix. V represents the combination coefficients of the bases for reconstructing matrix A and is called the coefficient matrix. Generally, this decomposition problem model as the following Frobenius norm optimization problem:

$$\min_{u,v} \| (A - UV) \|_F^2 \quad \text{such that} \quad U, V \geq 0$$

3.2.7.3 Tensor Factorization (TF)

In a heterogeneous graph, there are multiple relationships. A single matrix cannot represent multiple relationships without information loss. The two modes, as in the matrix case, are not descriptive enough to model heterogeneous graphs. Therefore, *tensors*, which are n-modal generalizations of matrices, are employed. The similar intuition of matrix factorization inspires tensor factorization. A partially observed matrix or tensor is decomposed into a product of embedding matrices with a much smaller rank, resulting in fixed-dimensional vector representations for each nodes and relationships in the graph. For a given relationship $r(u, v)$ in which node u is linked to node v through relation r, the score can then be recovered as a multi-linear product between the embedding vectors of u, r and v [Nickel et al., 2011, 2015].

There are literature published around tensor factorization methods which are explored for learning from knowledge graphs and multi-relational data [Franz et al., 2009, Kolda et al., 2005]. The tensor factorization method proved to be successful in analyzing the link structure of Web pages and Semantic Web data, respectively [Drumond et al., 2012], and to predict triples in a knowledge graph [Rendle and Schmidt-Thieme, 2010].

3.2.8 Comparisons of Different Factorization-Based Methods in Link Prediction

Table 3.4 shows the qualitative comparison of the tensor factorization based Link Prediction algorithms. The advantage of the tensor factorization model is that it can capture different patterns from the heterogeneous relationships and also provide higher accuracy in a Link Prediction task. Tensor factorization is known to be efficient compared to other methods. The drawbacks of this method include huge computational time and space constraints [Wang et al., 2015]. We observed from Table 3.4 that the recent progress in tensor factorization algorithms has improved in the computational cost in comparison to the previous models, but still, the model relies on the rank of the tensors. The ranks are parameters that need to determine using an internal cross-validation procedure by a grid search. Thus this process is time-consuming and computationally expensive.

Algorithms	Computational Complexity	Parameters Tuning
RESCAL [Nickel et al., 2011]	$\mathcal{O}(K^2)$	✓
TransE [Bordes et al., 2013]	$\mathcal{O}(K^2)$	✓
NTN [Socher et al., 2013]	$\mathcal{O}(K^2D)$	✓
DistMulT [Yang et al., 2014]	$\mathcal{O}(K)$	✓
HolE [Nickel et al., 2016]	$\mathcal{O}(KlogK)$	✓
ComplEx [Trouillon et al., 2016]	$\mathcal{O}(K)$	✓

D is an additional latent dimension of NTN model. K is the low rank of the tensors.

Table 3.4: Qualitative comparison of the tensor factorization based Link Prediction algorithms.

3.2.9 Summary of the Link Prediction Methods

There are numerous efforts made in Link Prediction from topology-based to learning-based methods. All the Link Prediction methods have their strengths and weaknesses, which we pointed out in this section. Factorization methods handle any graphs; for instance, homogeneous or heterogeneous, this property makes this method appealing. However, diffusion-based methods are simple, linear, and interpretable and easily scales to large graphs. Most of the diffusion frameworks proposed in graphs have an iterative scheme to predict links, which makes the process faster. The problem with the diffusion-based method is that it can be only used in homogeneous graphs to predict the links between similar nodes. In chapter 5, we combined both factorization and diffusion methods to handle two different kinds of graphs for Link Prediction.

3.3 Related Work on Node Classification

Node classification is one of the active areas of research in semi-supervised machine learning. The idea of semi-supervised node classification is inspired by label propagation or diffusion in graphs. The known labels propagates information through the graph in order to label all the nodes. Among the numerous semi-supervised techniques, the graph-based semi-supervised technique is gaining a lot of attention due to its ability to achieve a decent classification accuracy in manifold data. Figure 3.2 shows the manifold data of handwritten digits in a 2D representations[2].

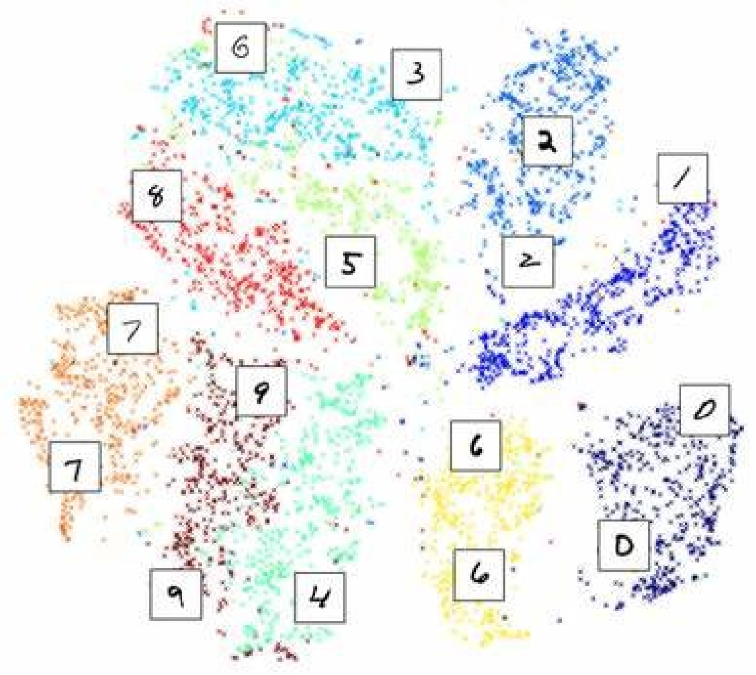


Figure 3.2: *Manifold of handwritten digits as a 2D representations*

These methods used a graph Laplacian regularizer to smooth the classification function concerning the data manifold [Belkin et al., 2006]. The effect of Laplacian regularizer depends upon how the graph is constructed. Most of the graph-based methods are similar to each other because all of them use propagation techniques to predict the unlabelled nodes. However, they differ in the particular choice of the loss function and the regularizer. More importantly, the graph-based label propagation works well if the graph is of good quality. However, graph construction is also an open research problem in semi-supervised machine learning [Zhu, 2005]. In this section, we show different label propagation algorithms along with their strengths and weaknesses.

3.3.1 Mincut

The first graph-based semi-supervised learning algorithm is inspired by the graph cut perspective. Blum et al. [Blum et al., 2004] proposed semi-supervised learning as a graph mincut problem. In the binary label classification, there are two labels (i) positive and (ii) negative. The positive labels act as sources, and negative labels act as sinks. This method aims to remove the minimum set of edges, which will inhibit all flow from the

[2] https://towardsdatascience.com/manifolds-in-data-science-a-brief-overview-2e9dde9437e5

sources to the sinks. The nodes connecting to the sources are labeled positive, and those to the sinks are labeled negative. One limitation with mincut is that it only gives hard classification without marginal probabilities of the nodes falling into the labels. It will lead to multiple possibilities of the nodes belonging to either of the two labels. Since the focus of the algorithm is only the partition of the graph.

3.3.2 Harmonic Functions (HMN)

A harmonic function is a function that has the same values as given labels on the labeled data, and satisfies the weighted average property on the unlabeled data [Zhu et al., 2003]. Let us assume a weighted graph G with n nodes indexed as $1, ..., n$. A symmetric weight matrix, denoted as W, represents the strength of linkage. All weights are non-negative ($w_{ij} \geq 0$), and if $w_{ij} = 0$, there is no edge between nodes i and j. We assume that the first l training nodes have binary label, $y_1, y_2, ..., y_l$, where $y_i \in \{-1, 1\}$. The remaining is the unlabeled nodes given as $u = n - l$ also known as test nodes. Thus the goal here is to predict the label for the unlabeled nodes for $y_{l+1}, y_{l+2}, ..., y_n$. The underlying assumption used here is that the label of an unlabeled node is likely to be similar to the label of its neighboring nodes. Mathematically, the goal is to find a function $f(x) \in \{-1, 1\}$ on the vertices, such that $f(x_i) = y_i$. In the graph context, a harmonic function is a function that has the same values as given a label on the labeled data, and satisfies the weighted average property on the unlabeled data: $f(x_i) = y_i, i = 1, ...l$;

$$f(x_j) = \frac{\sum_{k=1}^{l+u} w_{jk} f(x_k)}{\sum_{k=1}^{l+u} w_{jk}}, j = l + 1...l + u.$$

This iterative procedure will converge to a harmonic function, regardless of the initial values on the unlabeled vertices. The unnormalized graph Laplacian matrix L is $L = D - W$, where D is the degree matrix, and W is the weight of edges between the nodes. The Laplacian matrix can be subdivided into 4 submatrix as L is an $(l + u)$ x $(l + u)$ matrix with labeled ones are listed first.

$$L = \begin{bmatrix} L_{ll} & L_{lu} \\ L_{ul} & L_{uu} \end{bmatrix}$$

The function f can be partitioned into functions of labeled and unlabeled nodes (fl, fu), and let $y_l = (y_1, ..., y_l)$. Then solving the constrained optimization problem using Lagrange multipliers with matrix algebra, the harmonic solution is $f_l = yl$ and,

$$f_u = -L_{uu}^{-1} L_{ul} y_l \tag{3.1}$$

The score f_u is the predictive values of the unlabeled nodes. The harmonic function has closed-form solutions. It provides the node marginal probabilities and solves the limitation of the graph Mincut problems.

3.3.3 Local and Global Consistency (LGC)

The LGC method is slightly influenced by the HMN method. The algorithm uses every point iteratively to spread its label information to its neighbors until a global stable state is achieved [Zhou et al., 2004]. LGC method uses the normalized graph Laplacian and a classifying function. At each step the nodes receive a contribution from its neighbors which are weighted by the normalized weight of the edges between the nodes, and an additional small contribution given by its initial value of the nodes. The LGC algorithms converges to the closed-form solution given by:

$$f(1) = (1-\alpha)(\mathbf{I} - \mathcal{L})^{-1} f(0) \tag{3.2}$$

where I is the identity matrix, f(0) is the initial values of the nodes, α is the parameter that lies in $0 < \alpha < 1$ and f(1) is the final values of the labels of the nodes. The normalized Laplacians $\mathcal{R}$ is computed as:

$$\mathcal{L} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2} \tag{3.3}$$

W is the weight adjacency matrix of the graph G. The degree matrix **D** is defined as a diagonal matrix with $d_1, d_2, d_3, ...d_n$ as diagonal entities.

3.3.4 Heat Diffusion (HD)

Heat diffusion also known as exponential kernels used in a node classification problems. This idea was initially proposed by [Kondor and Lafferty, 2002]. The exponential kernel is defined as

$$e^{-\alpha L} = \lim_{N \to \infty} \left(1 + \frac{-\alpha L}{N}\right)^N \tag{3.4}$$

Where L is the graph Laplacian, I is the identity matrix, α is the diffusion parameter which determines the degree of diffusion.

The equation 3.4 is derived from the special initial condition of heat. For a known graph structure, the heat flow with initial conditions can be defined by the following second order differential equation:

$$\frac{\partial f(x,t)}{\partial t} - \Delta f(x,t) = 0 \tag{3.5}$$

where $f(x,t)$ is the heat at location x at time t, and Δf is the Laplace-Beltrami operator on a function f. The heat diffusion kernel $K_t(x,y)$ is a special solution to the heat equation with a special initial condition which is a unit heat source at position y when there is no heat in another end.

Yang et al [Yang et al., 2007] proposed a discrete approximation to compute the heat diffusion in a graph. Using, this method heat diffusion can be computed iteratively,

$$f(1) = f(0)\left(I + \frac{-\alpha}{N}L\right)^N \tag{3.6}$$

Where $f(0)$ is the initial heat scores of the nodes and $f(1)$ is the final heat scores after the diffusion process. In a practical setting, keeping the value of $\alpha = 1$ and $N = 30$ works in most of the cases [Yang et al., 2007]. The $f(1)$ value is used for the node label classification.

3.3.5 Dissimilar Label Propagation

Most of the label propagation works on the principle of consistency, i.e. (1) nearby nodes are likely to have the same label, and (2) points on the same structure (typically referred to as a cluster or a manifold) are likely to have the same label [Zhou et al., 2004]. However, we might also have dissimilarity information between two nodes that should have different labels. This property is called heterophily (love for the different ones).

Goldberg et al. [Goldberg et al., 2007] define a different graph energy function for dissimilarity edges. In particular, if node i and node j are dissimilar, one minimizes energy functions $w_{ij}(f(i) + f(j))^2$. The major change to note here in comparison to homogeneous label propagation is the choice of sign. The "+" sign is chosen instead of a "-" sign. This enables node $f(i)$ and $f(j)$ to have different labels. The energy function can be easily computed using linear algebra. Tong et al. [Tong and Jin, 2007] adopted other objective functions based on minimizing the ratio between inconsistency and consistency between the labels to solve heterophily label propagation.

3.3.6 OMNI-Prop

OMNI-Prop is a seamless node classification algorithm that can be classified in any label correlations such as homophily, heterophily, or mix (homophily + heterophily) without any assumptions about it [Yamaguchi et al., 2015]. This algorithm is based on the follower[3] and followee[4] relationship. If the nodes with most of the followers have the same labels, than rest also have the same label. The algorithm has two important parts (i) Self scores and (ii) Follower scores. These scores in matrix form are computed as:

$$S_U \longleftarrow (D_U + \lambda I)^{-1}(A_U T + \lambda \mathbf{1}_U \mathbf{b}^T) \tag{3.7}$$

$$T \longleftarrow (F + \lambda I)^{-1}(A^T S + \lambda \mathbf{1}_N \mathbf{b}^T) \tag{3.8}$$

In Equation 3.7, S_U is the self score matrix of the unlabeled nodes. D_U is the degree matrix of the unlabeled nodes, λ is prior strength parameter, I is the identity matrix, A_U is the adjacency matrix of the unlabeled nodes, 1_U is the unlabeled column vectors where each component is 1, b is uniform prior (i.e, $b_K = 1/K$ K is a number of labels) $\mathbf{b}^T = (b_1, b_2, b_3, ...b_K)$.

In Equation 3.8, T is a follower score matrix, F is $U \times U$ diagonal matrices defined as $F_{ii} = \sum_{j=1}^{N} A_{ji}$. Other symbols are similar as Equation 3.7.

OMNI-prop has a closed form solution which is computed as:

$$S = (I - Q_{UU})^{-1}(Q_{UL} S_L + \mathbf{r}\mathbf{b}^{\mathbf{T}}) \tag{3.9}$$

,

where,
$Q = (D_U + \lambda I)^{-1} A (F + \lambda I)^{-1} A^T$,
$r = (D_U + \lambda I_U)^{-1}(\lambda 1_U + \lambda A_U (F + \lambda I_N)^{-1} \mathbf{1_N})$

[3] followers: indegrees of the nodes
[4] followee: outdegree of the nodes

3.3.7 CAMLP: Confidence-Aware Modulated Label Propagation

CAMLP [Yamaguchi et al., 2016] uses the evidence from its neighbors for the propagation process. The algorithm is also flexible in all kinds of label correlations, such as homophily, heterophily, and mix label propagation. To incorporate the confidence, CAMLP considers that every node determines its label based on both its prior beliefs and signals from neighbors. The basic idea here is that if a node receives many signals from neighbors, it will determine its label based on the (enough) signals. The label propagation is computed by iterative approach,

$$F^r \longleftarrow Z^{-1}(Y + \beta A F^{r-1} H) \tag{3.10}$$

,

Where F is the label probability matrix between nodes and labels, β is the influence parameter which has the values $(0, \infty]$, A is the adjacency matrix, $Z = I + \beta D$ is the normalization matrix, $D = \{d_i\}$ is the diagonal matrix of the node degrees, Y is the prior belief matrix, H is the modulation matrix of size $K \times K$, and K is the number of labels.

CAMLP has a closed-form solution and converges in any graphs regardless of the graph structure and initial values of F and Y.

3.3.8 Belief Propagation (BP)

The BP algorithm [Pearl, 1982] estimates marginal probabilities in Bayesian networks, Markov random fields, graphical models, and factor models. The algorithm has been successfully applied in node classification. The caveats of BP is that there is no guarantee for the convergence of the algorithms on general graphs. Koutra et al. [Koutra et al., 2011] developed a linear version of BP in a binary class settings (FaBP). Later, [Gatterbauer et al., 2015] developed a linearized BP in multi-class settings (LinBP) with two simple linear approximations. These algorithms are considered fast and are guaranteed to converge with specific criteria. Additionally, both algorithms offer closed-form solutions based on simple matrix inversion. Although these two variants of BP can handle heterophily networks.

3.3.9 Semi Supervised Regression (SSR)

Theoretically, all graph-based Semi Supervised Classification (SSC) methods are function estimators. SSC estimates “soft labels”, which are the probabilities associated with labels attached to each node. The classification function tries to be close to the targets y in the labeled set, and at the same time, be smooth on the graph [Zhu, 2005]. Thus most of the graph-based SSC methods can also naturally perform regression. The co-training style algorithm developed by Zhou and Li [Zhou and Li, 2005] demonstrated useful in SSR using graphs. Similarly, [Wang et al., 2006] proposed an algorithm, which is about the kernel regression framework exploiting both labeled and unlabeled examples. Sindhwani et al. [Sindhwani et al., 2005] and Brefeld et al. [Brefeld et al., 2006] perform multi-view regression, where a regularization term depends on the disagreement among regressors on different views. Cortes er al. [Cortes and Mohri, 2007] proposed a transductive regression model for SSR tasks. Similarly, [Guo and Uehara, 2015] proposed an inductive semi-supervised regression model by incorporating prior graph information into the standard Gaussian process (GP) regression. This method builds an adjacent graph over all the labeled and unlabeled data and applies a graph-based co-variance function to derive the marginal likelihood and the prediction distribution of semi-supervised GP regression for prediction. All the above mentioned methods work with the same analogy of SSC; only the label nodes treated as real values instead of classes.

3.4 Comparison of Node Classification Algorithm

Table 3.5 shows the qualitative comparisons of the graph-based node classification algorithms. We observed that OMNI Prop is a robust graph-based algorithm because it can adapt to any label correlated graphs, and there is no need for parameters to tune. OMNI is shown to be very useful in a single layer when the node is not shared across different layers. We are not sure how this algorithm performs in a network of layers where the nodes are shared across different layers.

	Mincut	**HMN**	**LGC**	**HD**	**OMNI**	**CAMLP**	**BP**
Homophily	✓	✓	✓	✓	✓	✓	✓
Heterophily					✓	✓	✓
Mix					✓	✓	
Parameters Tuning			✓	✓		✓	
Closed Form Solution		✓	✓	✓	✓	✓	✓
Convergence	✓	✓	✓	✓	✓	✓	?
Manifold	✓	✓	✓	✓	✓	✓	✓
Multi-layer Graphs							✓

Table 3.5: Quantitative comparison between different label propagation algorithms for node classification.

Similarly, BP is another powerful algorithms and primarily designed for probabilistic graphical models [Pearl, 1982]. This model does not require parameter tuning and scales to any label propagation. Although BP is a handy algorithm, due to its recursive calculations, it offers no guarantee for convergence in arbitrary graphs. The label propagation algorithms like Mincut, HMN, LGC, and HD perform better in a robust homophilic label propagation. The problem with these algorithms is that they suffer from high classification errors in heterophilic and mix label propagation.

Tang et al. [Tang and Liu, 2009a,b] and Perozzi et al. [Perozzi et al., 2014] extracted latent social features by proposing DeepWalk, based on a deep learning framework [Bengio et al., 2013]. Aggarwal et al. [Aggarwal and Li, 2011] explored node classification on evolving networks. Chaudhari et al. [Chaudhari et al., 2014] showed a way to classify nodes with high precision at the sacrifice of recall. Moore et al. [Moore et al., 2011] used active learning [Settles, 2010] to select the most informative nodes to perform node classification in assortative and disassortative networks better. These models demonstrated that they could deal with heterophily networks. However, they can only handle small scale networks ($<$ 1000 nodes). Talukdar et al. [Talukdar and Crammer, 2009] proposed an algorithm called MADDL that handles the correlation between labels for node classification, but MADDL cannot handle heterophily networks. All of the above algorithms can only work in a single layer network.

Our proposed work in label propagation is similar to harmonic and heat diffusion in terms of methodology. However, the caveat of both heat and harmonic methods is that they follow continuous or long-range diffusion. Not all the networks are suitable for long-range diffusion. In chapter 6 we show that a multiplex network which is called network of networks requires a special diffusion kernel because information propagates very quickly within it. Thus, we proposed a novel heat diffusion with boundary conditions which has the property of both heat diffusion and harmonic function. This method adapts to a long-range or short-range parameter based on a special parameter called time. We extended the boundary-based heat diffusion also to homophily, heterophily, mixed, and SSR problems.

In the next chapters, we present the core of the thesis. Chapter 4 discusses diffusion in a 2-layered graph framework for the link prediction between dissimilar nodes. In chapter 5, we show how combining methods enhances the performances of link prediction between dissimilar nodes in the 2-layered framework. In chapter 6, we demonstrated our novel Boundary Heat Diffusion (BHD) method to apply in (i) homophily, (ii) heterophily, (iii) mix, and (iv) multiplex network for node classification problems. We further extended our BHD algorithm to a slightly relevant problem as classification known as regression in a semi-supervised setting in chapter 7. Chapter 8 presents the conclusion and future work opened by this thesis.

Part II

Core

Chapter 4

Diffusion Based Method for the Link Prediction Task

Information diffusion is the communication of information in a social network system. For example, in online social networks like Facebook[1] or Twitter[2], when a user discovers that his friend circle share, repost, retweet a piece of information, the user will be influenced to review whether to share, repost or retweet the information. This sharing/reposting/retweeting behavior leads to information diffusion. Information diffusion is the process that allows users to receive or observe information that is beyond the scope of their social cycles. Furthermore, this property makes it desirable to create new links. Similarly, in a biomedical problem, diffusion-based methods are drivers to identify the genes and underlying diseases. The network diffusion approaches model the information flow in molecular networks to perform various computational biomedical tasks such as biomedical link prediction.

This chapter presents a novel approach to link prediction between two dissimilar nodes based on diffusion in graphs. It starts by formalizing the problem, and then by describing the proposed approach. We test our approach in two different use-cases: (i) computational biomedical problem and (ii) social media problems. Parts of the research reported in this chapter have been published in [Timilsina et al., 2018, 2019b].

[1] https://www.facebook.com

[2] https://www.twitter.com

4.1 Introduction

In this chapter, we explore various link prediction task in predicting links between different nodes. The contribution of the work presented in this chapter is that we proposed a novel 2-layered graph data model, where the first layer is for labeling the nodes, while the second is the diffusion layer. The main motivation behind using a 2-layered network for diffusion is that the link prediction between two different kinds of nodes is a complex process. For example, in the context of disease and gene links prediction, disease propagates among genes, and genes interact with each other. From this example, we already notice there are two different types of information available. When multiple pieces of information are available, it is natural to incorporate them as additional information sources. These kinds of complex mechanisms are not possible to capture in a single graph layer.

We check the performance of our approach in two datasets: (i) biomedical problems for tumor sample and gene link prediction and (ii) social media data for predicting links between academic entities (scientist/publications) and social media (weblogs/mainstream news). Each of the datasets is considered as its own research problem. The former is called disease-gene identification in biomedicine, and the latter is called altmetrics, which is an alternative way of measuring scientific impacts on social media. We can model these problems in 2-layered graphs and use a heat diffusion-based method. Our results show that the heat diffusion-based link prediction method improves over the current state of the art methods. One of the essential characteristics of heat diffusion in a link prediction task is that it is computationally cheaper [Bourigault et al., 2014, Carlin et al., 2017] to compute and robust in-memory usage [Al-Mohy and Higham, 2011].

4.2 Graph Data Model

As we discussed above, we have two layers in the graph. The first layer is the bipartite graph of dissimilar entities and the second layer is the graph between similar entities as shown in Figure 4.1.

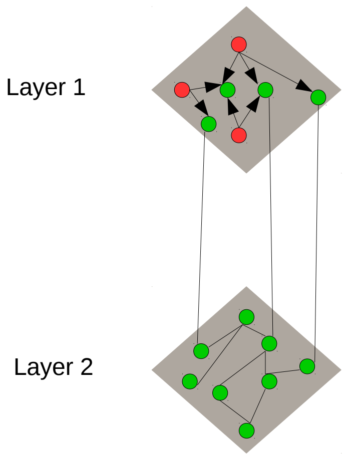


Figure 4.1: 2-Layered graph data model.

In the first layer, we model the two entities as a bipartite graph. The bipartite network consists of two disjoint sets of nodes: for example, a bipartite graph between disease and genes where disease and genes are two dissimilar entities or nodes. The "hasGene" connects the disease and gene nodes.

The second layer is the graph between similar entities, such as gene interaction networks. For example, the genes are connected by the "co-expression" relationship. The graph in the second layer is the homogeneous graph for the diffusion process.

4.3 Solution Approach

Our main aim is to predict the links between dissimilar nodes. For this, our first step is to use the 2-layered graph as input. This input graph has missing links between nodes. In Figure 4.2, red arrows are the missing links that we want to predict.

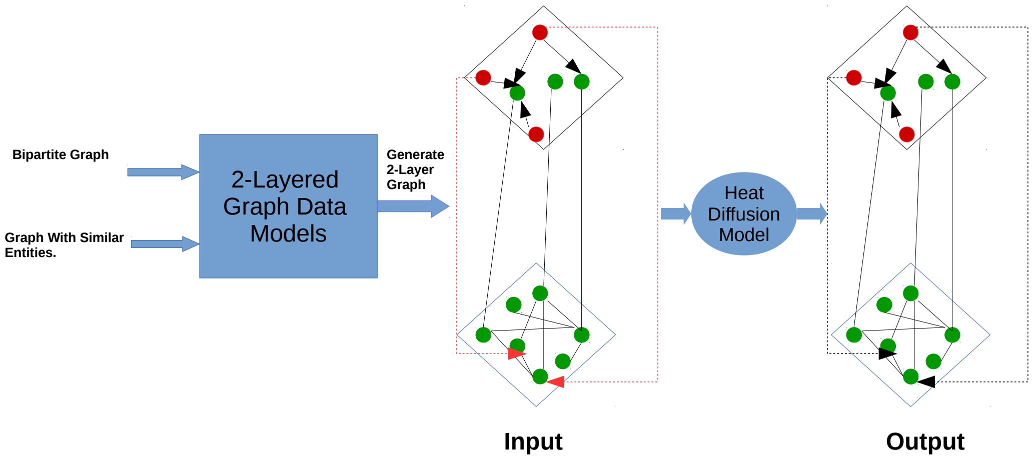


FIGURE 4.2: Solution approach for link prediction in a 2-layered graph.

Our second step is to apply the heat diffusion algorithm. For the execution of this algorithm, we need seed nodes. This information is available from the first network layer and diffused to the second network layer. Once the diffusion process is over, we get the association score between every dissimilar node pairs. These association scores are the likelihood of links.

4.4 Model Description

Heat Diffusion Model

Heat diffusion is a common physical phenomenon. In a medium, heat always flows from a high to a low temperature region. The heat diffusion-based method already showed its usefulness in various domains such as web spamming in web graph analysis [Yang et al., 2007], recommender systems [Ma et al., 2008a,b] and disease-gene prioritization [Nitsch et al., 2010]. To make it self-contained, we will briefly introduce the heat diffusion model for weighted and unweighted undirected graphs, which is adapted from [Yang et al., 2007].

Heat Flow on an Undirected and Unweighted Graph

In the case of undirected and unweighted graphs, the edge (g_i, g_j) is considered as a pipe from where the heat flows and connects nodes g_i and g_j.

In an undirected graph, the heat can be modeled as follows. For instance, at time t, every node g_i obtains $M(i, j, t, \Delta t)$ amount of heat from its neighbor node g_j for a time of Δt. We have two assumptions here:

- The heat obtained $M(i, j, t, \Delta t)$ is proportional to the time period Δt.
- The heat obtained $M(i, j, t, \Delta t)$ is proportional to the heat difference $f_j(t) - f_i(t)$.
- $d(g_j)$ is the degree of the node g_j.
- $f(0)$ is the initial heat vectors of the nodes.
- $f(1)$ is the final heat vectors of the nodes.

Furthermore, based on these assumption the amount of heat transfers between nodes is expressed as:

$$\mathbf{f}(1) = e^{\alpha \mathbf{H}^{**}} \mathbf{f}(0) \tag{4.1}$$

Where $\mathbf{H}^{**}$ is the heat matrix for the undirected unweighted graph

$$H^{**}_{ij} = \begin{cases} -d(g_j), & \text{if } j = i \\ 1, & (g_j, g_i) \in E, \\ 0, & \text{otherwise,} \end{cases} \tag{4.2}$$

Heat Flow on an Undirected and Weighted Graph

In the case of the weighted links between the nodes, we need to modify the heat diffusion model. Consider a weighted graph such that $G = (V, E, W)$ where V is the list of all the nodes such that $V = \{g_1, g_2, g_3, ..., g_n\}$. On a weighted graph, in the pipe (g_i, g_j). $W = w_{ij} \mid \textit{weight}$ score associated with edge $(g_i,\ q_j)$. Suppose, at time t, each node g_i receives $RH = RH(i, j, t; \Delta t)$ amount of heat from g_j during a period of Δt. We made four assumptions as follows:

- RH is proportional to the time period Δt.
- RH should be proportional to the weight w_{ji} of the undirected edge $(g_j,\ g_i)$.
- RH should be proportional to the heat at node g_j.

- RH is zero if there is no link between g_j to g_i. As a result, g_i will receive $\sum_{j:(g_j,g_i)\in E} \sigma_j w_{ji} f_j(t)\Delta t$ amount of heat from its neighbors that are connected to it.
- $\sigma_j = \frac{\alpha}{d(g_j)}$ where $d(g_j)$ is the out degree of the node g_j and α is the thermal conductivity.

Simultaneously, node g_j diffuses DH(i,t,Δ t) amount of heat to its neighboring nodes. We consider that:

- The heat $DH(i,t,\Delta t)$ should be proportional to the time-period Δt.
- The heat $DH(i,t,\Delta t)$ should be proportional to the heat at node g_i.
- Each node has the same ability to diffuse the heat.
- The heat $DH(i,t,\Delta t)$ should be distributed to its neighboring nodes proportional to the weight on each edge.
- τ is the flag to check whether the node has any outgoing links. If there is any outgoing links then $\tau = 1$ else $\tau = 0$

Thus heat diffusion between gene nodes is given by,

$$\mathbf{f}(1) = e^{\alpha \mathbf{H^*}} \mathbf{f}(0), \tag{4.3}$$

The $\mathbf{H}^*$ which is the heat matrix for the undirected weighted graph model as,

$$H^*_{ij} = \begin{cases} -(\frac{\tau_i}{d_i}) \sum_{k:(i,k)\in E} w_{ik}, & \text{if } j = i \\ \frac{w_{ji}}{d_j}, & (g_j, g_i) \in E, \\ 0, & \text{otherwise,} \end{cases} \tag{4.4}$$

The matrix $e^{\alpha H^*}$ is known as the diffusion kernel. The heat diffusion process continues infinitely many times from the initial heat diffusion. The parameter α is called thermal conductivity. The higher the value of α, the faster is the spread of heat in the network. If α is infinitely large, then heat diffuse from one node to another quickly.

In the context of some networks such as gene-gene interactions or hyperlink web graphs, there are random relations among different nodes [Brin and Page, 1998, Luo et al., 2007, Ma et al., 2008a]. Thus, in order to capture that behavior, we add uniform random relations among different nodes. Let γ denotes the probability of not forming random interactions, and (1- γ) is the probability of taking a "random jump". This behavior is also called "teleport" operation in the computation of PageRank [Page et al., 1999] in web graph. The real-world application considers the random edges [Yang et al., 2007], so we followed the same setting of $\gamma = 0.85$ as in PageRank in all of our experiments.

Without any prior knowledge, we set $g = \frac{1}{n}1$ where g is a uniform stochastic distribution vector, 1 is the vector of all ones, and n is the number of nodes. We employed the above information and adapted our model as:

$$f(1) = e^{\alpha R} f(0), R = \gamma H + (1 - \gamma) g 1^T \tag{4.5}$$

Where H can be replaced either H^* or H^{**} depending upon the kind of graph used.

Computational Complexity

When the graph is large, then the direct computations of $e^{\alpha R}$ is time-consuming so we adopted the discrete approximations by [Yang et al., 2007]:

$$f(1) = \left(I + \frac{\alpha}{M} R \right)^M f(0), \tag{4.6}$$

Where M is a positive integer, and I is the identity matrix. In order to reduce the computational complexity, we apply three methods: (1) Since $f(0)$ is a vector, we iteratively calculate $(I + \frac{\alpha}{M} R)^M f(0)$ by applying the operator $(I + \frac{\alpha}{M} R)^M$ to $f(0)$. (2) For the matrix R, we apply a data structure which only stores information of non-zero entries, since it is a sparse matrix. (3) For every heat source, which is tumor samples or academic entities in our cases, we bind it by diffusing heat to its neighbors. The selection of α and M parameters is described in details in the experimental section. Specifically, after using the discrete formalization, the complexity of the heat diffusion algorithm in our model is given by $O(M|E|T)$, where M is the number of iterations, T is the number of labeled nodes from layer 1 and $|E|$ is the number of edges in the graph in layer 2.

In the next section, we focus on how we use the diffusion model to predict a link between hypothetical 2-layered toy networks of Disease (Tumor samples) and Gene relationships.

Tumor Gene Predictions in a Toy Network

With the diffusion model described in the above section, we can now make the prediction by the following approach:

Let us consider a toy network as shown in Figure 4.3. Network layer 1 is a Tumor sample and Gene layer and the network layer 2 is a gene-gene interaction layer. The initial temperature from **Tumor X** to **Gene A, B, C** and **D** in first layer at $t = 0$ is given by the vector $f(0)$:
The initial values of the vector $f(0)$ are given by: $f(0) = [1, 1, 1, 0]^T$. We see in this vector the position of **Gene D** is 0 because there is no connection from **Tumor X** to **Gene D**.

The network layer 2 is an unweighted network, so we model the heat matrix using equation 4.2. Thus, our heat matrix is:

$$H = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

Then heat diffusion at $t = 1$ with $\alpha = 1$, is given by:

$$f(1) = e^{\alpha H} f(0), \tag{4.7}$$

Thus the computed $f(1)$ vector is given by $f(1) = [1.0, 1.0, 0.5, 0.43]$. Now normalizing each vector in $f(1)$ by sum of all the numbers in $f(1)$ then $f(1)$ = [0.34, 0.34, 0.17, 0.14]. Here the interesting thing to observe is at the position of **Gene D**. This value was initially **0** after diffusion and normalization we saw the value to be **0.14**. This value is the likelihood of **Tumor X** to form link with **Gene D**.

4.5 Results and Evaluations

For the evaluation of the heat diffusion approach, we performed two experiments. The first experiment predicts the links between tumor samples and genes using biological data. The second experiment predicts the links between academic and social media entities using social media data.

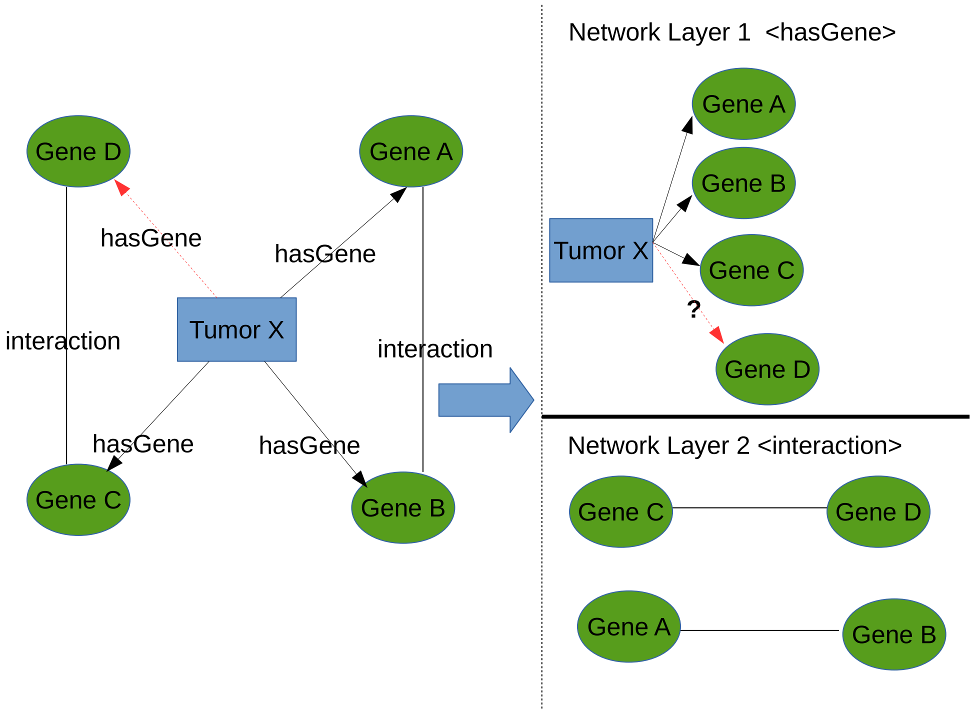


Figure 4.3: 2-layered toy networks of tumor and gene. The red arrow signifies the absence of link and the black arrows shown known links.

4.5.1 Link Prediction between Tumor Samples and Genes

Datasets

The data for this experiment is from COSMIC (Catalogue Of Somatic Mutations In Cancer) Methylation RDF Data[3]. We use the COSMIC database because it uses the expert-curated information of somatic mutations in human cancers [Forbes et al., 2014]. COSMIC has divided the datasets into logical categories, namely “Complete Mutation Data”, “Non-Coding Variants” and “DNA Methylation Data”. The COSMIC RDF methylation data has the properties, IDs, sample names, locations, gene names, and methylation status. Each sample name is a tumor sample of the patient from different locations in the body. For example, “TCGA-CV-A6JN01” is a tumor sample and its location is “Upper Aerodigestive Tract”.

Figure 4.4 shows the presence of genes across the different anatomical locations from COSMIC Methylation RDF data. The tumor samples in the datasets are from ten

[3] http://bioopenerproject.insight-centre.org/dataset

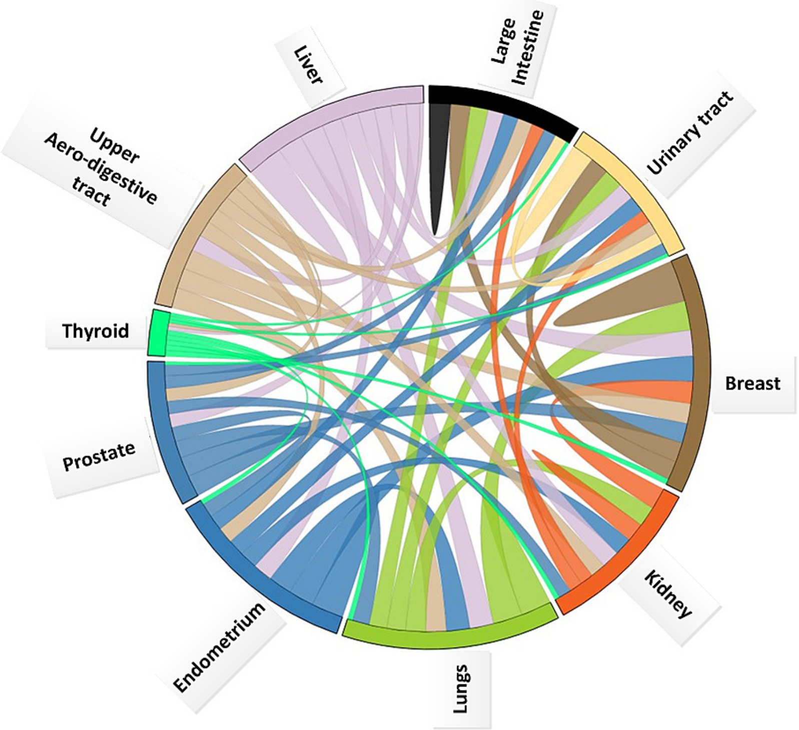


Figure 4.4: Visualization of the genes shared across the different anatomical locations on the cosmic methylation data. The size of the ring represents the number of genes in the dataset that are the members of the anatomical location of the body. An arc indicates how often these genes are shares across the connected segments.

different anatomical locations. The *gene name* in the datasets is the accepted HGNC[4] (HUGO Nomenclature Committee) identifier which provides a unique gene symbol and name for human loci.

Construction of the 2-layered graph between Tumor Samples and Genes

The first layer, which is the bipartite network, consists of two disjoint sets of nodes: one set corresponds to the tumor samples; the other set corresponds to all the genes in each tumor sample. The edges between the tumor samples and genes are based on the facts reported in the data. For instance, if "TCGA-B6-A0RG-01" is a tumor sample, and "HOXC4" is a gene associated with it, then we link the two corresponding nodes by the "hasGene" edge: $\left[\text{TCGA-B6-A0RG-01} \xrightarrow{hasGene} \text{HOXC4}\right]$.

The second layer is the interaction graph between genes, as shown in Figure 4.1. For the construction of network layer 2, we used nine different variants of protein interaction channels. Out of nine different variants of protein interaction channels, eight are from

[4] https://www.genenames.org/

STRING[5] and one is from the BioGRID[6] database. These two databases are publicly available.

Prior studies by [Köhler et al., 2008, Nitsch et al., 2010] modeled the gene interaction graph as an undirected graph. We took the same approach.

We used the protein-protein interaction links with weights for the **homosapiens** class from the latest STRING version 10.5 database. There are eight different weighted channels of the protein-protein interaction networks available in STRING, which are as follows: *co-expression, co-occurrence, database, experimental, fusion, neighborhood, text mining*, and *combined*.

From the BioGRID API[7], we constructed the Gene-Gene physical network. The BioGRID database provides protein interactions curated from the biomedical literature [Chatr-Aryamontri et al., 2014] and has provided well-validated *physical* interactions. The previous studies by [Li and Patra, 2010, Navlakha and Kingsford, 2010] have shown the potential of prioritizing the genes based on physical properties. This network is unweighted.

To transform the original protein interaction network into a gene interaction network for both STRING and BioGRID data, we took the following approaches: (i) Protein names were mapped to their encoding genes by parsing the EnsEMBL files [Aken et al., 2016]. (ii) In the case of genes encoding multiple proteins, we took the edge of maximum (integrated) weight connecting any pair of proteins encoded by such genes. The prior studies [Gonçalves et al., 2012], have also used a similar technique for protein to gene mapping.

Table 4.1 shows a detailed summary of the nodes and edges used in the construction of the network.

[5] `https://string-db.org/`

[6] `https://thebiogrid.org/`

[7] `https://wiki.thebiogrid.org/doku.php/biogridrest`

Property	Value
Number of Tumor Samples	4086
Number of Genes	4071
Number of relations between tumor samples and genes (hasGene)	222252
Co-occurrence	1166
Co-expression	208470
Database	23169
Experimental	170642
Fusion	98
Neighborhood	18929
Text mining	322883
Combined	358627
Physical	18395

Table 4.1: Network summary of 2-layered tumor-gene graph.

4.5.2 Link Prediction between Academic Entities and Social Media

For this task, we used the Spinn3r[8] data, which is a crawl of social media data for the time of 2010 November to 2011 July. From this data, we only used mainstream news and weblogs about Avian Influenza epidemics that happened during that time. Avian influenza was a newsworthy topic during that period. In our experiments, the term "academic entities" refers to scientists and publications.

Construction of the 2-layered graph between scientist and web documents

The scientists are identified from the mainstream news and weblogs using the Natural Language Processing (NLP) pipeline called GATE[9], which is open-source software for text mining problems. To solve the disambiguation problem on the identified name of the scientist, we used the Scopus API[10] and then manually verified the scientists' publications related to avian influenza.

For the second layer, we constructed the hyperlink web document graph from the weblogs and mainstream news data. The weblogs and mainstream news data item from the first

[8] https://www.programmableweb.com/api/spinn3r

[9] https://gate.ac.uk/

[10] https://dev.elsevier.com/sc_apis.html

layer has the source URL and content. In the content section of every item, we searched for the hyperlinks, and from each hyperlink, we extracted the URL, which is the target URL. We constructed a directed graph with source nodes as a source URL and target nodes as a target URL.

Table 4.2 shows the detailed summary of the nodes and edges used in the construction of the 2-layered Scientific Authors and Web graph.

Property	**Value**
Number of scientific authors	375
Number of web documents	8,124
Number of "mention" relationships	845
Number of "hyperlink" relationships	204,095

Table 4.2: Network summary of 2-layered scientific authors and web graph.

Construction of the 2-layered Graph between Scientific Publications and Web Documents

The straightforward way to identify scientific publications on the web is by their Digital Object Identifier (DOI)[11] or by their PMID, which is the unique identifier used in PubMed Citations[12]. In the case of DOI, it can be found as a unique persistent identifier, for example, http://dx.doi.org/10.1371/journal.pmed.1000388. Then DOI of this publication is 10.1371/journal.pmed.1000388. Similarly, for PMID if the URL is https://www.ncbi.nlm.nih.gov/pubmed/3472723 then is 3472723. We checked the URLs with the pattern **dx.doi.org** and **ncbi.nlm.nih.gov/pubmed/** in our graph and extracted DOIs and PMIDs. We checked each of the identified DOIs and PMIDs in the Scopus database using the Scopus API. This API directly provides the flexibility to search scientific publications using DOIs and PMIDs.

Table 4.3 shows a detailed summary of the nodes and edges used in the construction of the 2-layered Scientific Publications and Web graph.

[11] http://www.apastyle.org/learn/faqs/what-is-doi.aspx

[12] http://answers.library.curtin.edu.au/faq/121100

Property	**Value**
Number of scientific publications	1,344
Number of web documents	948,238
Number of "publication" relationships	508
Number of "hyperlink" relationships	5,406,939

Table 4.3: Network summary of 2-layered scientific publications and web graph.

4.5.3 Selection of Parameters

From Equation 4.6, we observed that the heat diffusion equation has two important parameters: α, which is the diffusion rate, and M the number of iterations. There is no straightforward way to estimate these parameters, so we estimated these values from our data. The parameter α, also known as thermal conductivity, plays an important role in a heat diffusion process. If α set to high, then heat diffuses faster. Following the [Yang et al., 2007] simulation study for a web graph setting, $\alpha = 1$ and $M = 30$ work in most of the cases. We took the same approach for the hyperlink graph.

For the biological network, the parameters alpha α and iterations M are estimated from 10-fold cross-validation on the training sets and applied the learned parameter in the test sets. Thus, AUC-ROC reported in the test set is the average score across the ten folds. We used the AUC-ROC evaluation metric, which is commonly used in machine learning communities for quantifying the accuracy of link prediction algorithm algorithms [Clauset et al., 2008].

To demonstrate the impact of α in both STRING and BioGRID graph, we monitored the AUC-ROC score at different values of α. As shown in Figure 4.5, we observed the increasing trend of the AUC-ROC score with a change in the diffusion parameter α. The high AUC-ROC score of 0.74 for BioGRID (unweighted network) and 0.85 for STRING using fusion (weighted) channel are observed. We used the Fusion channel in STRING data because this network gave better results in comparison to other network channels in STRING data. After $\alpha \geq 1$, there is no change in the AUC-ROC scores.

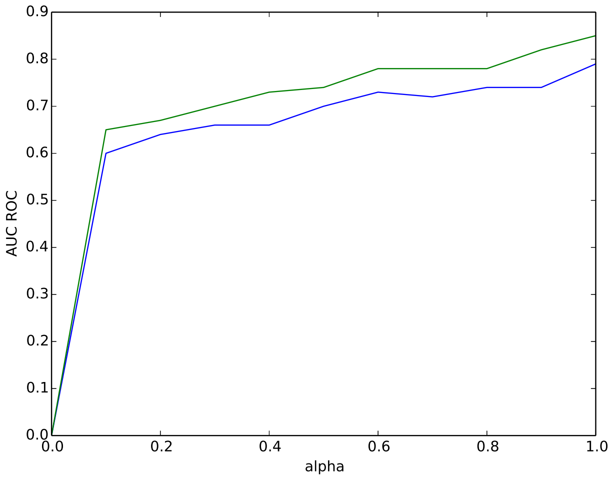


Figure 4.5: Impact of parameter α. The prediction accuracy assessed by varying diffusion parameter ranging from 0 to 1 with the step of 0.1 in BioGRID and STRING Network to predict tumor samples and genes. Blue curve represents BioGRID and green curve represents String network.

We applied a similar approach for the parameter M as shown in Figure 4.6. We detected that when $M = 5$ for BioGRID and $M = 6$ for STRING graphs, the heat diffusion algorithm attains better performance. After that, in both the graph, the AUC-ROC score is constant. It means the heat diffusion algorithm has converged.

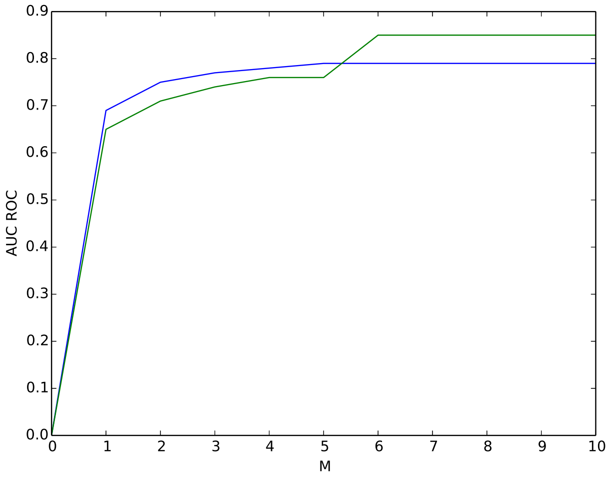


Figure 4.6: Impact of Parameter M. The prediction accuracy assessed by varying iterations in BioGRID and STRING Network to predict tumor samples and genes. Blue curve represents BioGRID and green curve represents the String network.

Physical Meaning of Parameters

We determined the value for α and the number of iterations M in the cross-validation mode on the training sets for the biomedical network. These two parameters are quite hard to determine in advance without accessing any data. The mechanistic meaning of alpha is that if it approaches infinity, then the diffusion reaches equilibrium, thus all the connected nodes receive the same diffusion contribution, which is similar to PageRank. If we set alpha as 0, then there is no diffusion. The optimum α balances the extent of heat, which diffuses from genes to its immediate neighbors and the rest of the network.

Similarly, for M, which is the number of iterations, the impact of this parameter relates to how far the heat diffuses from the seed nodes. The physical meaning of the scalar parameter M is the total time of diffusion, which controls the amount of heat to which the initial signal is allowed to spread over the network. The probabilistic interpretation for these computations is that if the input values are preference binary vector, which is our case (1 means genes having an association with tumor samples whereas 0 means no association), of starting positions for heat diffusion across the edges of the graph, the final value is the position distribution after M iterations. If M tends to infinity, then the probability distribution approaches to a uniform distribution over all the genes.

4.5.4 Evaluation Metrics

We conducted cross-validation by partitioning all the links in the network layer 1 into ten folds and deleting the relationship of the links in the test set. We then computed heat diffusion scores and ranked all reconstructed node pair relationships and recorded the AUC-ROC. The AUC-ROC metric can be understood as the probability that a randomly chosen missing link is given a higher score than a randomly chosen non-existing link [Wang et al., 2014].

To implement the AUC-ROC in the link prediction context, we took the following approach.

- The observed links E is randomly split into two parts: the training set E_{train} is treated as known information, while the test set E_{test} is used for testing and no information in the test set is allowed to be used for prediction. Thus the total existing edge set is, $E = E_{train} \cup E_{test}$ and $E_{train} \cap E_{test} = \emptyset$.
- Theoretically, this metric is computed as : $AUC = (n' + 0.5n'')/n$. Where,

1. n': Number of times the missing links (links in E_{test}) have a higher score than the non-existing links.
2. n'': Number of times the scores of missing links is equal to a number of times the score of non-existing links (links in $U - E$), where U is the universal set.
3. n: Number of independent comparisons between missing and non-existing links.

There is an extensive discussion about this technique in the link prediction literature [Clauset et al., 2008, Wang et al., 2014]. If the AUC score exceeds 0.5, it means that the algorithm performs better than pure chance.

Performance of the Heat Diffusion Algorithms across different Channels in a Biomedical Data of Tumor samples and Gene Link Prediction

For tumor samples and gene data, we have nine different networks in network layer 2. So, we performed the experiments in all of these networks. Figure 4.7 shows the performance of the algorithm in 10-fold cross-validation across different channels.

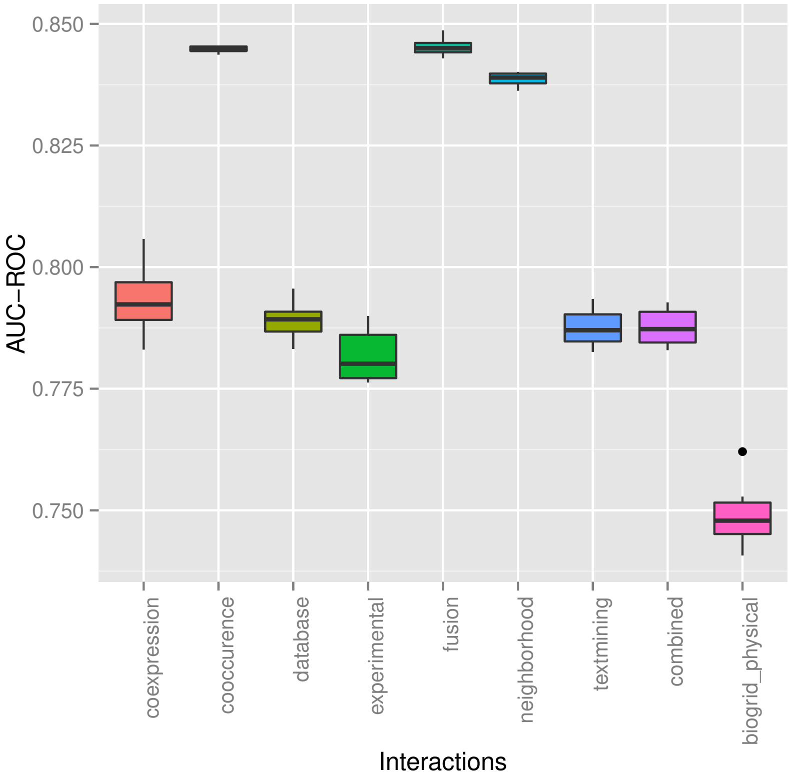


Figure 4.7: Result of 10 Fold Cross Validation in nine different gene interaction channels for predicting tumor samples and genes.

We observed that the mean AUC-ROC score is 0.84 for fusion and co-occurrence channels. Similarly, the neighborhood channel has a mean AUC-ROC score of 0.83. These three channels do not have a significant difference in mean AUC-ROC scores. In STRING, all three interactions aim to identify pairs of genes which appear to be under normal selective pressures during evolution (more so than expected by chance), and which are therefore thought to be functionally associated [Von Mering et al., 2005]. A candidate gene fusion pair with a high score is more likely to be a driver gene fusion of tumor progression [Zhao et al., 2016]. From here onward, we used the fusion channel for the rest of our experiments.

Consecutively, the other three genetic interaction approaches –(i) text mining, (ii) database, and (iii) combined channels– each have mean AUC-ROC scores of 0.78. One reason for the combined channel to perform similarly as text mining and database channel is that it contains scores of all the channels as the text mining approach inherently rely on noisy data, which leads to comparatively lower AUC-ROC scores.

For the BioGRID physical interaction channel without any weights between the gene pairs, the heat diffusion algorithm showed mean AUC-ROC scores of 0.74. It shows the potential of network propagation methods by only using network topology to predict the gene associations [Cowen et al., 2017, Navlakha and Kingsford, 2010].

Comparison with Baseline Algorithms

We compare our results from heat diffusion algorithms with link prediction baseline. To do this, we applied several algorithms for predicting links, such as scores based on similarity metrics, namely Common Neighbors, Jaccard similarity, Adamic/Adar, Preferential Attachment, and Resource Allocations (See Chapter 2, Section 2.3.1). These algorithms are also called node-based topological similarity algorithms because they can be viewed as computing a measure of “proximity" or “similarity" between nodes [Liben-Nowell and Kleinberg, 2007].

We also compared the results from the heat diffusion algorithm with the two widely used path-based similarity algorithms called Katz and Personalized PageRank. The path-based approaches are also known as the information diffusion approaches. We used those algorithms in our 2-layered graph.

- **Random Baseline** This assigns each candidate edge a random score. This score is meant for the benchmark to compare other algorithms.

- **Node Based** The link prediction metrics assigns a score of each candidate edges. These metrics presented by [Liben-Nowell and Kleinberg, 2007] are used extensively in the link prediction problem. However, using two different node sets cannot be directly applied in the context of the bipartite graph because the neighbors of nodes on opposite sides of the network do not intersect. The bipartite graph in network layer 1 is a directed graph. If we consider two disjoint nodes x and y then, the terms used in the equations can be described as:

  – $N_{out}(x)$ denotes outgoing neighbors of node x.

  – $N^{'}_{out,in}(y)$ is interpreted as follows: (i) Set of all the incoming neighbors of node y. (ii) From the list of a neighbor of node y, get all the list of outgoing neighbors.

  1. Common Neighbors: $score(x, y) = |N_{out}(x) \cap N^{'}_{out,in}(y)|$
  2. Jaccard's Coefficient: $score(x, y) = \frac{N_{out}(x) \cap N^{'}_{out,in}(y)|}{N_{out}(x) \cup N^{'}_{out,in}(y)|}$
  3. Adamic/Adar:
     $score(x, y) = \sum_{z \in N_{out}(x) \cap N^{'}_{out,in}(y)} \frac{1}{log|N_{out} z|}$
  4. Preferential Attachment:
     $score(x, y) = |N_{out}(x)| \cdot |N^{'}_{out,in}(y)|$
  5. Resource Allocation :
     $score(x, y) = \sum_{z \in N_{out}(x) \cap N^{'}_{out,in}(y)} \frac{1}{|N_{out} z|}$

- **Path Based** Path based link prediction is based on the paths from one node to another. The two nodes are likely to be connected if there exist more paths between them. We employed the following metrics to compute the score between two sets of nodes:

  1. Katz: $score(x, y) = \sum_{i=1}^{\infty} \beta^{l} \cdot |paths^{<l>}_{x,y}|$
  2. Personalized PageRank: $score(x, y)$ is the probability of node y in a random walk that returns to node x with a probability α at each step, moving to a random neighbor with probability $1 - \alpha$

Those link prediction algorithms are evaluated using both the AUC-ROC and the AUC-PR metrics. The AUC-PR [see 2.7.6] metric is more informative with massive class imbalance problems such as link prediction [Garcia-Gasulla et al., 2016, Lichtnwalter and Chawla, 2012] and can perform robustly in a noisy environment [Zhang et al., 2016]. Hence, in this study, we showed both AUC-ROC and AUC-PR evaluation metrics for link prediction by the heat diffusion algorithm and the baselines.

The 10 Fold cross-validation results of the prediction are shown in Table 4.4:

	Tumor-Gene Link Prediction		**Scientist-Social Media Link Prediction**		**Scientific Publication-Social Media Link Prediction**	
	AUC-ROC	AUC-PR	AUC-ROC	AUC-PR	AUC-ROC	AUC-PR
Methods						
Random Baseline	0.50 ± 0.000	0.0125 ± 0.024	0.5 ± 0.013	0.00002 ± 0.013	0.5 ± 0.017	0.00017 ± 0.012
Common neighbor approach (CN)	0.72 ± 0.025	0.038 ± 0.012	0.55 ± 0.023	0.02 ± 0.019	0.59 ± 0.223	0.012 ± 0.024
Jaccard similarity (JS)	0.76 ± 0.015	0.054 ± 0.112	0.58 ± 0.071	0.05 ± 0.091	0.61 ± 0.547	0.041 ± 0.024
Preferential attachment (PA)	0.73 ± 0.015	0.042 ± 0.322	0.57 ± 0.189	0.03 ± 0.257	0.60 ± 0.084	0.040 ± 0.025
Resource allocation (RA)	0.78 ± 0.001	0.061 ± 0.025	0.62 ± 0.015	0.042 ± 0.004	0.65 ± 0.002	0.045 ± 0.007
Adamic Adar index (AAI)	0.79 ± 0.011	0.064 ± 0.071	0.63 ± 0.089	0.049 ± 0.018	0.66 ± 0.012	0.047 ± 0.002
Katz	0.72± 0.061	0.053 ± 0.041	0.66 ± 0.039	0.042 ± 0.085	0.63 ± 0.012	0.049 ± 0.015
Personalized pageRank algorithm (PPR)	0.81 ± 0.062	0.031 ± 0.026	0.67 ± 0.040	0.051 ± 0.011	**0.75 ± 0.034**	**0.054 ± 0.085**
Heat Diffusion (HD)	**0.85 ± 0.067**	**0.083 ± 0.069**	**0.69 ± 0.037**	**0.056 ± 0.011**	**0.75 ± 0.018**	0.050 ± 0.031

Table 4.4: Results of our heat diffusion algorithm with baseline prediction in a 10-fold cross-validation settings.

The heat diffusion algorithm outperformed other methods in predicting links between (i) tumor samples and genes (ii) scientist-social media and (iii) scientific publications-social media. Only in the scientific publications-social media link prediction, there is a marginal improvement using PPR of AUC-PR. One of the reasons for this might be there are some weblogs or mainstream news farther away from the publication nodes which are still relevant. Thus, PPR captures the global region of the graph whereas HD captures sub-region of the graph.

From Table 4.4, we see the disagreement between AUC-ROC and AUC-PR scores in a link prediction task. AUC-PR considers only the prediction of positive examples and is generally used for problems common in Information Retrieval, where negatives dominate the positives and are not considered important. For the link prediction problem, AUC-PR gives credit for correctly predicting edges but do not give credit for correctly predicting non-edges. We believe this is one of the reasons for the discrepancies between both metrics.

Statistical Significance of the Link Prediction in a 2-layered Graph Using Heat Diffusion Method

To check the significance of the link prediction results, we performed a permutation test. To run this test, we need to carry out a graph randomization process by preserving the degree distribution and perform the diffusion in a random graph for K times. We partitioned the random graph data into training (75%), and testing (25%) sets. We recorded the heat diffusion scores in a test set.

The result of this test is the estimated p-value. If K is big then, it should be close to the p-value obtained by considering all the possible shuffles. In our case, we chose $K = 10,000$. For a large web graph with millions of nodes, performing a randomization test for 10,000 times is very time consuming and computationally expensive. So, we only present here the significance test of link prediction between tumor samples and gene links.

The p-value of the link prediction score is:

$$p - value(M, N) = \frac{\Omega}{K} \tag{4.8}$$

Where N and M are two different nodes, Ω is the number of randomly produced links between N and M, which obtained a higher diffusion scores than its actual predicted one. K is the total number of times the test is performed. The pair between (M, N) receiving higher p-values will be less likely to be an actual link because this pair will have a strong association with many randomly produced heat scores. The histogram of the p-values of our test is shown in Figure 4.8.

We observe from the histogram that a large proportion of links are statistically significant (p-value are near to zero). However, for some links, the p-values are large. Thus there is a risk of reporting false positives for a small proportion of links. This problem may be caused by the poor quality or incomplete network used in link prediction.

4.6 Discussion

We considered two different types of baselines to compare the results of the heat diffusion algorithm. One is node-based, and another is path-based algorithms. Out of the Node-based link prediction metrics, Adamic/Adar and Resource Allocation methods performed the best, and the Common Neighbor approach performed the worst for the considered datasets. In terms of AUC-ROC and AUC-PR, both Adamic/Adar and Resource Allocation have similar scores. The heat diffusion algorithm has produced more accurate predictions, surpassing Adamic/Adar and Resource Allocation by up to 7.05% with regard to AUC-ROC. The integration of gene interaction data in the diffusion model has proved to have a significant influence on the performance of the tumor samples and gene link prediction. One thing also important to observe is that only exploiting the

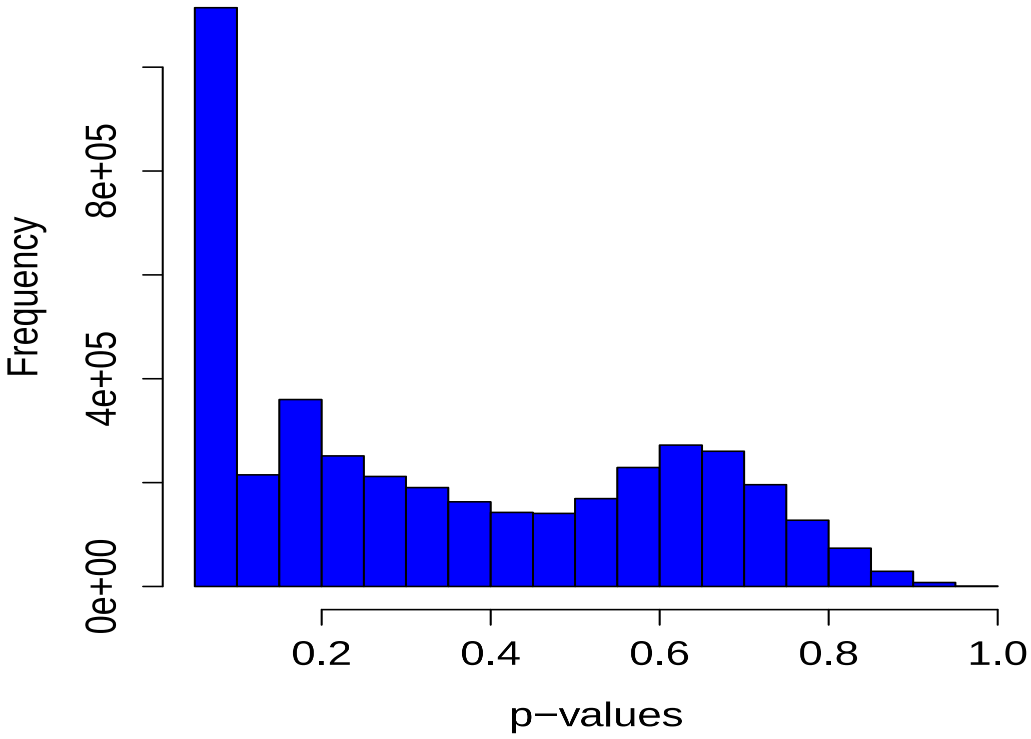


Figure 4.8: Histogram of p-values from permutation test.

link structures, the Personalized PageRank algorithm outperforms several link prediction algorithms in the link prediction task, as shown in Table 4.4.

We observed that the heat diffusion algorithm outperforms Personalized PageRank in predicting tumor samples and gene links. There is a gain of 4% in AUC-ROC scores using the heat diffusion algorithm over the Personalized PageRank algorithm, which means that the genetic interaction scores in heat diffusion are contributing to the improvement of prediction quality. However, in predicting the scientists and social media links, we saw that there is a slight improvement of 2% in AUC-ROC scores using the heat diffusion algorithm over the Personalized PageRank algorithm. The heat diffusion algorithm also gives the missing links which are not in the training set. In this work, we did not further investigate the missing links because that evidence was not reported in the database, and we are unsure whether those are spurious or meaningful links.

4.7 Conclusion

We presented the heat diffusion algorithm to predict links between dissimilar nodes using a 2-layer network. We used a heat diffusion algorithm for this task in two different datasets (i) biomedical data and (ii) social media data. In biomedical data for link prediction between tumor samples and genes, we noted that heat diffusion gave the highest AUC-ROC scores in a fusion, co-occurrence, and neighborhood channels using STRING data. The heat diffusion-based method results in decent predictions even if no knowledge is available about the disease or phenotype. It outperformed some of the baseline such as Common Neighbors, Preferential Attachment, and Katz methods. We also observed the similar behavior in a social media and academic entities link prediction.

The other reason to choose heat diffusion is that it is less memory intensive and easier to compute than alternative. The algorithm can be computed in linear time. In both the experiments, we observed that Personalized PageRank Algorithm also gave comparable results. However, heat diffusion uses exponential sum, which converges quickly while personalized PageRank uses geometric sum.

In chapter 5, we will demonstrate how we can take advantage of a 2-layered graph. A 2-layered graph has two different network information. From the first layer, we learn the weights of the seed nodes using a matrix completion algorithm. These learned weights can be used in the second layer for diffusion. Using this approach, we can strengthen the link prediction between the dissimilar nodes.

Chapter 5

Discovering Links in a 2-Layered Graph Using a Combination of Matrix Factorization and Diffusion

This chapter proposes a novel approach to discover links in a 2-layered graph. Our method combines matrix factorization with diffusion in a 2-layered graph for predicting links. Using a matrix factorization method, we can learn the weights of the seed nodes in network layer 1 and diffuse this information in network layer 2. In this way, we can make use of the 2 different graphs and 2 different methods. The experiments show that our approach achieves better results than diffusion- or factorization-based methods alone. Parts of the research reported in this chapter have been published in [Timilsina et al., 2019a].

5.1 Introduction

In a link prediction problem, matrix factorization is one of the widely used methods because the network exhibits matrix representation. The cells of the matrix represent the relationships between the nodes. Therefore, according to Menon et al. [Menon and Elkan, 2011], link prediction can be treated as a problem of matrix completion. Singular Value Decomposition (SVD) [Tang et al., 2015] uses low-rank matrix decomposition for this purpose. Non-negative Matrix Factorization (NMF) [Sra and Dhillon, 2006] is another variant of matrix factorization applied in link prediction tasks [Chen et al., 2017]. One of the advantages of using NMF-based matrix factorization is that it can easily integrate heterogeneous information [Wang et al., 2018]. For multi-relational link prediction, tensor-based factorization is a popular method. The strength of tensors is that the multi-relational graph can be expressed in higher-order tensors, which can be easily factorized. These models do not require a priori knowledge inference from the data, unlike graphical models such as Markov Logic Networks (MLN) or Bayesian Networks [Nickel et al., 2011]. In recent studies, a node2vec approach was used to analyze different network neighborhoods to embed nodes based on the assumption of homophily as well as structural equivalence for link prediction in a homogeneous network for the same edge type [Grover and Leskovec, 2016]. Due to their high accuracy, node embedding techniques [Zitnik and Leskovec, 2017] are popular methods, but they also have some limitations. These methods require learning steps that might be unfeasible for large-scale networks that have millions of nodes [Zhou and Jia, 2017]. Similarity-based propagation methods have been well studied in predicting the links in bipartite networks. such as predicting most relevant products for users in a recommender systems [Zhang et al., 2007b].

Our approach has two advantages. First, the diffusion-based methods usually only apply in a homogeneous network, where nodes and edges are of the same type. As in the previous chapter, we consider heterogeneous networks and effectively integrate them using a 2-layered approach. Second, for these two graphs, we integrated matrix factorization and heat diffusion methods to handle the two different networks effectively.

5.2 Methodology

Datasets

To test our method, we chose to apply it on (i) biomedical and (ii) social media data. The reason to chose biomedical data is that biology as a subject makes wide use of various biological databases to tackle different challenges such as understanding the treatment for diseases and cellular function. Thus large biological datasets are now available to the scientific community to carry out experiments and discover non-obvious patterns in datasets and to make reliable statistical predictions about similar new data [Chicco, 2017]. We applied our method in a computational biomedical problems for identifying unintended side effects of drugs.

For social media data, we used the same data for predicting links between academic entities and social media, as described in Section 4.5.2. Tables 5.1 and 5.2 show the detailed summary of the nodes and edges used in the construction of 2-layered academic entities and social media web graph.

Property	**Value**
Number of scientific publications	1,344
Number of web documents	948,238
Number of "publication" relationships	508
Number of "hyperlink" relationships	5,406,939

Table 5.1: Network Summary of 2-layered Scientific Publications and Web Graph

Property	**Value**
Number of scientific authors	375
Number of web documents	8,124
Number of "mention" relationships	845
Number of "hyperlink" relationships	204,095

Table 5.2: Network Summary of 2-layered Scientific Authors and Web Graph

Construction of the Bipartite Graph between Side Effects and Drugs for Network Layer 1

The datasets we used are publicly available databases: (i) DrugBank[1] for drugs, (ii) SIDER[2] for drug side-effects, (iii) PubChem[3] for compound IDs which are used to link drugs in DrugBank to the ones in SIDER. After the linking of drugs and side effects, there are 1020 unique Drugs and 5598 side effects. The edges between side effects and drugs are the facts reported in the SIDER database.

Construction of the Semantic Graph between Drugs for Network Layer 2

We constructed the semantic similarity between the drugs using a trained word2vec models. For this task, we used the open-source skip-gram model provided by NLPLab[4]. This model is trained on all PubMed abstracts and PMC full texts (4.08 million distinct words)

[1] `https://www.drugbank.ca`

[2] `http://sideeffects.embl.de/`

[3] `https://pubchem.ncbi.nlm.nih.gov/`

[4] `http://evexdb.org/pmresources/vec-space-models/wikipedia-pubmed-and-PMC-w2v.bin`

with 200 dimensions in combination with the latest Wikipedia dump. This model has also been used to extract chemical-induced disease relations from the scientific literature [Le et al., 2016]. The summary of the network is shown in Table 5.3.

Node Types	**Property**
Number of Drug Nodes	1020
Number of Side Effect Nodes	5598
Number of Side effect and Drug relationships	133750
Number of Drug-Drug relationships	519690

TABLE 5.3: Summary of side effect-Drug and Drug-Drug Network.

An example of drug-drugs semantic similarity graph is shown in Figure 5.1. The right side rectangle in the dashed line contains the drug-drug similarity graph and the left side rectangle in the dashed line contains the side effects-drugs bipartite graph.

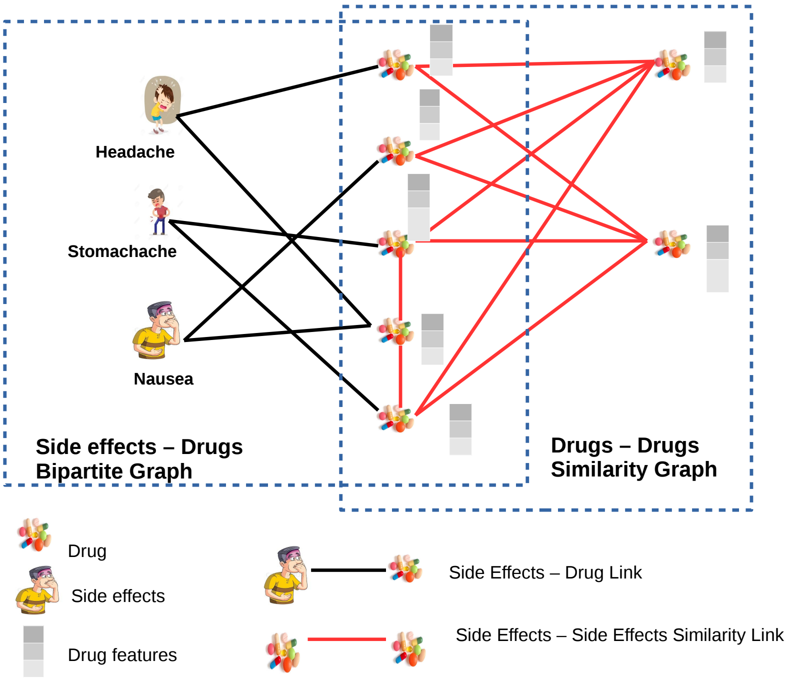

Headache
Stomachache
Nausea
Side effects – Drugs Bipartite Graph
Drugs – Drugs Similarity Graph
Drug
Side effects
Drug features
Side Effects – Drug Link
Side Effects – Side Effects Similarity Link

FIGURE 5.1: An example of side effect-drug bipartite graph and drug-drug similarity graph.

Heat Diffusion Model

We model the diffusion of side effects in the drug network as a process of heat diffusion. In a drug-drug network, the drugs which are linked with side effects information act as heat sources, and have a very high amount of heat. These drugs initiate to influence other drugs and diffuse their influence on their neighbors. In this chapter, we used heat diffusion models applied in an undirected graph where the links are weighted. We used the same formalism for heat diffusion model provided in Chapter 4, Section 4.4.

NMF based Matrix Completion

The motivation for the matrix completion problem is to discover an unknown real matrix from a small subset of its entries. This problem comes up in many application areas and has received wide attention in recommender systems, particularly after the Netflix[5] challenge to predict user ratings for movies. Taking inspiration from this idea, we made an initial prediction for unobserved values using the bipartite graph network layer 1. We expect that there exists an ideal matrix that encodes the weights of relationships between all the node pairs.

For a given bi-adjacency matrix $Y = [y_{ij}] \in \mathbb{R}^{mxn}$, where rows and columns represent disjoint nodes, and non-zero elements represent known links, the goal is to complete this matrix for node pairs. In the matrix Y, each element y_{ij} $(1 \leq i \leq m, 1 \leq j \leq n)$ belongs to boolean values of $[0, 1]$. Here $y_{ij} = 0$ means that no weight is provided by node i for node j, while $y_{ij} = s, 0 \leq s \leq 1$, is the diffusion weight given by node i for node j. Among different matrix completion algorithms, NMF is a powerful tool for completing sparse matrix [Zhang et al., 2006] and performs best compared to other states of the art approaches such as Singular Value Decomposition (SVD) [Cai et al., 2011].

The NMF approach uses all the known weights to decompose the matrix Y into the product of two low-rank, latent feature matrices, one for the nodes in the column, $S_{m\times r}$,

[5] https://www.netflix.com

and other for nodes in row, $D_{n\times r}$, so that:

$$Y \approx \hat{Y} = SD^T = \underbrace{\begin{bmatrix} s_1^T \\ s_2^T \\ . \\ . \\ . \\ s_m^T \end{bmatrix}}_{m\times r} \underbrace{\begin{bmatrix} d_1 & d_2 & ... & d_3 \end{bmatrix}}_{r\times n} \tag{5.1}$$

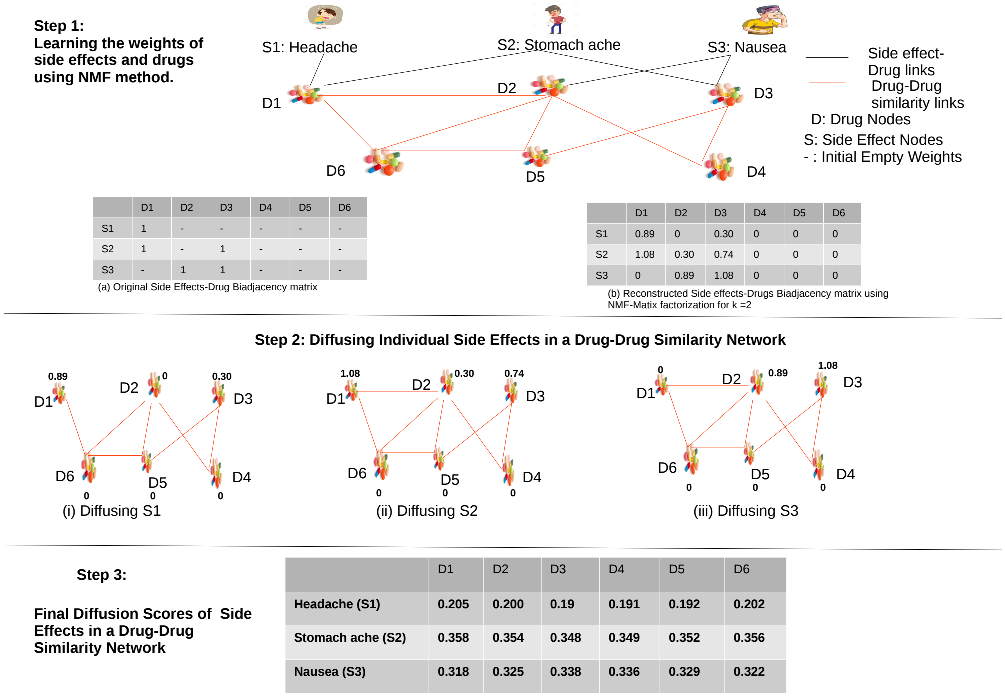
The latent feature vectors for nodes in row s and nodes in column d are r dimensional, where $r \ll min\{m,n\}$. The predicted weights for the node pair (s,d) is given by $\hat{y} = s^T d$. The NMF factorization in Equation 2.1 problem can be achieved by solving the optimization problem,

$$\min_{S\in\mathbb{R}^{m\times r}, D\in\mathbb{R}^{r\times n}} \| \left(Y - SD^T\right)\|_F^2 \quad \text{such that} \quad S, D \geq 0 \tag{5.2}$$

where F is the Frobenius norm.

NMF and HD Combined in a 2-layer Toy Network

To demonstrate the solution approach of our method, we demonstrate an intuitive example of propagation of the side effects in a 2-layer toy network. In Figure 5.2, step 1, there are 2 types of nodes and relationships. The nodes are Side effects and Drugs in the first layer. The relationship between Side effects and Drugs association is shown by the black line, and the semantic relation between drugs is shown by the red line. The bipartite graph of Side effects and Drugs is shown as a biadjacency matrix (Figure 5.2 step 1(a)). The cells with 1 mean that connections exist, and empty cells mean no connection. To fill the empty cells, we applied the NMF matrix factorization algorithm with $k = 2$. Once the matrix is full or completed, we obtain association scores of each side effects with the respective drugs, as shown in the matrix (Figure 5.2 step 1(b)). These scores are heat weights for each side effect.


Step 1:
Learning the weights of side effects and drugs using NMF method.
S1: Headache
S2: Stomach ache
S3: Nausea
D1
D2
D3
D4
D5
D6
Side effect-Drug links
Drug-Drug similarity links
D: Drug Nodes
S: Side Effect Nodes
- : Initial Empty Weights
Step 2: Diffusing Individual Side Effects in a Drug-Drug Similarity Network
0.89
D1
D2
0
0.30
D3
D6
0
D5
0
D4
0
1.08
D1
D2
0.30
0.74
D3
D6
0
D5
0
D4
0
0
D1
D2
0.89
1.08
D3
D6
0
D5
0
D4
0
Step 3:
Final Diffusion Scores of Side Effects in a Drug-Drug Similarity Network

	D1	D2	D3	D4	D5	D6
S1	1	-	-	-	-	-
S2	1	-	1	-	-	-
S3	-	1	1	-	-	-

(a) Original Side Effects-Drug Biadjacency matrix

	D1	D2	D3	D4	D5	D6
S1	0.89	0	0.30	0	0	0
S2	1.08	0.30	0.74	0	0	0
S3	0	0.89	1.08	0	0	0

(b) Reconstructed Side effects-Drugs Biadjacency matrix using NMF-Matix factorization for k =2

(i) Diffusing S1

(ii) Diffusing S2

(iii) Diffusing S3

	D1	D2	D3	D4	D5	D6
Headache (S1)	**0.205**	**0.200**	**0.19**	**0.191**	**0.192**	**0.202**
Stomach ache (S2)	**0.358**	**0.354**	**0.348**	**0.349**	**0.352**	**0.356**
Nausea (S3)	**0.318**	**0.325**	**0.338**	**0.336**	**0.329**	**0.322**

FIGURE 5.2: *Side Effects Propagation in a Drug-Drug Similarity Network.*

In step 2, we diffuse the weights of each side effects in the drug-drug similarity network in a second layer. For instance, the initial heat weight vectors of side effect **Headache** is given by $f(0) = [0.89, 0, 0.30, 0, 0, 0]$. Now with this weight vectors diffusing the side effects in a drug-drug similarity network with $\alpha = 1$ and applying heat diffusion from Equation 4.3, the new diffusion scores of side effect **Headache** are given by $f(1) = [0.20, 0.20, 0.19, 0.19, 0.19, 0.20]$. From this computation, we observed that the pair (Headache, Drug: D1) and (Headache, Drug: D6) are the first and second-highest ranked pairs. In Figure 5.2 at step 3, the drugs D2, D3, D4,D5 and D6 have new weights for the side effect **Headache** which initially had no weights. This final diffusion score vector can be considered as the impact of side effects on the corresponding drugs. With the same process, the other side effects **Stomachache** and **Nausea** are calculated.

5.3 Results and Evaluations

In this section, we present the results of applying the method described above on our two use cases.

Computation of Heat Diffusion Scores with NMF Method in a 2-layered Graph

To fill the missing weights between node pairs in the first layer, we first applied the NMF algorithm. To choose the optimal number of latent factors in NMF we use a cross-validation method in the training sets. After the weights are learned, these are used as the initial temperature for the heat diffusion process. We diffuse this in network layer 2. There are other two important parameters: α (diffusion rate) and M (number of iterations). The parameter α, also known as thermal conductivity, plays an important role in a heat diffusion process. If α is set too high, heat diffuses too fast. [Yang et al., 2007] found that in practical settings, $\alpha = 1$ and $M = 30$ is optimal in most of the cases. If α is set to 0, this means there is no thermal conductivity, and therefore no diffusion occurs.

Evaluation

To evaluate the prediction ability of the NMF based Heat diffusion method, we conducted a 10 fold cross-validation by partitioning all links in network layer 1 into ten folds and removing the links in the test set. We computed the heat diffusion score and ranked all node pair scores in network layer 1 and recorded the area under the Precision-Recall curve (AUPR) score. The AUPR score is a standard performance metric in machine learning communities for quantifying the accuracy of the link prediction algorithm [Yang et al., 2015]. This metric is more informative and robust with heavy class imbalance problems such as link prediction [Garcia-Gasulla et al., 2016].

Comparison with Baseline Link Prediction Algorithms

We compare our results from heat diffusion with NMF methods to baseline algorithms. The details of these algorithms have been already described in chapter 4, Section 4.5.4. The results of our comparisons are shown in Table 5.4

	Side Effects -Drug Prediction	**Scientist-Social Media Link Prediction**	**Scientific Publication-Social Media Link Prediction**
	AUC-PR	AUC-PR	AUC-PR
Methods			
Random Baseline	0.12 ± 0.025	0.00002 ± 0.013	0.00017 ± 0.01
Common Neighbors (CN)	0.35 ± 0.075	0.02 ± 0.019	0.012 ± 0.02
Resource Allocation (RA)	0.36 ± 0.061	0.042 ± 0.004	0.045 ± 0.007
Adamic Adar (AA)	0.35 ± 0.011	0.049 ± 0.018	0.047 ± 0.002
Katz	0.35 ± 0.001	0.042 ± 0.085	0.049±0.015
Personalized PageRank (PPR)	0.37 ± 0.011	0.051 ± 0.011	0.054 ± 0.085
Heat Diffusion (HD)	0.39 ± 0.021	0.054 ± 0.034	0.050 ± 0.031
Non negative matrix Factorization (NMF)	0.41 ± 0.011	0.26 ± 0.011	0.14 ± 0.002
NMF + HD	**0.44 ± 0.014**	**0.30 ± 0.041**	**0.19 ± 0.032**

Table 5.4: Link prediction results for the different state of the art methods. Each score shows the mean AUC-PR score for predicting links. The figure after ± is the standard deviation obtained from the 10 fold cross validation.

We observed that our combined approach of NMF with heat diffusion outperforms other states of the art link prediction methods. We performed the t-test to find out if there is a significant difference in the ten-fold cross-validation results between our NMF-based heat diffusion and the other link prediction methods using significance (α) level 0.05. In the 10-fold cross-validated paired t-test procedure, we divide the test set into ten parts of equal size. Each of these parts is then used for testing, while the remaining nine parts which are joined together are used for training. For each 10-fold cross-validation iteration, we compute the difference in performance between the NMF-based heat diffusion and baseline link prediction algorithms. By assuming that these ten differences were independently drawn and follow an approximately normal distribution, we can compute the t-statistic with 9 degrees of freedom according to Student's t-test, under the null hypothesis that the NMF-based heat diffusion and baseline link prediction algorithms have equal performance. After the t-statistic is computed, we can compute the p-value and compare it to our chosen significance (α) level 0.05. If the p-value is smaller than (α), we reject the null hypothesis. The p-values of the test are in Table 5.5.

We found that there is a significant difference between the prediction performed by NMF-based heat diffusion with the other state of the art methods. All p-values are lower than the 0.05 significance level suggesting a significant difference in the prediction performed by NMF+HD methods with the other methods.

T-test for predicting links between side effects and drugs	**P-value**
Heat Diffusion With NMF Vs NMF	7.247e-12***
Heat Diffusion With NMF Vs Heat Diffusion	1.253e-15***
Heat Diffusion With NMF Vs Personalized Page Rank	2.2e-16***
Heat Diffusion With NMF Vs Katz	1.685e-14***
Heat Diffusion With NMF Vs Adamic Adar	2.2e-16***
Heat Diffusion With NMF Vs Resource Allocation	5.925e-10***
Heat Diffusion With NMF Vs Common Neighbors	2.2e-16***
Heat Diffusion With NMF Vs Random	2.2e-16***
T-test for predicting links between scientist and social media	**P-value**
Heat Diffusion With NMF Vs NMF	2.247e-12***
Heat Diffusion With NMF Vs Heat Diffusion	1.247e-16***
Heat Diffusion With NMF Vs Personalized Page Rank	1.356e-12***
Heat Diffusion With NMF Vs Katz	1.17e-12***
Heat Diffusion With NMF Vs Adamic Adar	1.236e-7***
Heat Diffusion With NMF Vs Resource Allocation	2.2e-17***
Heat Diffusion With NMF Vs Common Neighbors	1.236e-7***
Heat Diffusion With NMF Vs Random	2.133e-10***
T-test for predicting links between scientific publications and social media	**P-value**
Heat Diffusion With NMF Vs NMF	1.246e-12***
Heat Diffusion With NMF Vs Heat Diffusion	2.247e-12***
Heat Diffusion With NMF Vs Personalized Page Rank	1.346e-16***
Heat Diffusion With NMF Vs Katz	4.251e-10***
Heat Diffusion With NMF Vs Adamic Adar	2.21e-12***
Heat Diffusion With NMF Vs Resource Allocation	2.23e-10***
Heat Diffusion With NMF Vs Common Neighbors	2.221e-15***
Heat Diffusion With NMF Vs Random	1.223e-14***

TABLE 5.5: P-values of the t-test at significance level $\alpha = 0.05$, *** indicates high significance.

Statistical Significance of the Link Prediction in a 2-layered Graph Using the NMF and HD method

We also performed the same test as discussed in Section 4.5.4 of Chapter 4. For a large web graph with millions of nodes, and performing a randomization test for 10,000 times is very time consuming and computationally expensive. So, we only present here the significance test of link prediction between side-effects and drugs. We partitioned the random graph data constructed by preserving the degree distribution of drug-drug network into training (75%), and testing (25%) sets. We recorded the heat diffusion with NMF scores in a test set.

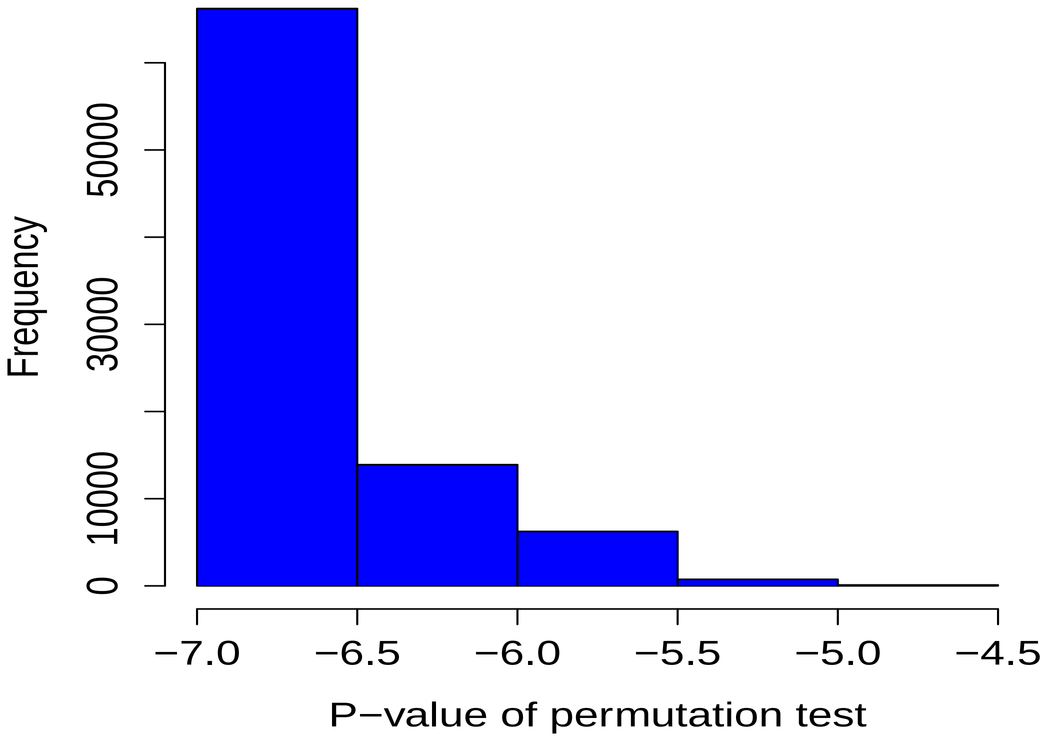


Figure 5.3: Histogram of p-values from permutation test. The X-axis is in log base 10 scale.

The histogram of the p-values of our test is shown in Figure 5.3. We observed from the histogram that the large proportion of predicted links are statistically significant (p-value are near to zero) and a minimal amount of links have p-values > 0.05.

5.4 Discussion

Our results clearly show the combination of matrix factorization and heat diffusion-based technique outperformed (i) node-based (ii) path-based and (iii) NMF method. In terms of AUC-PR results, heat diffusion with NMF brings an improvement of 6.81% over the NMF method alone and of 11.36% over the heat diffusion method in a side effect and drug link prediction. For scientists and social media link prediction, the heat diffusion with NMF has a significant improvement of 82% over heat diffusion and 13% over the NMF method. Similarly, in scientific publications and social media link prediction, the heat diffusion with NMF has a significant improvement of 73% over the heat diffusion and 26% over the NMF method. One of the reasons for the high difference in accuracy between heat diffusion and NMF with heat diffusion in social media graphs is due to the low quality of the social media networks. Indeed, we consider that we cannot take much advantage of heat diffusion alone in this data due to noise.

In terms of path-based link prediction in a side effect and drug link prediction, heat diffusion has a marginal improvement over Personalized PageRank. The important features of heat diffusion is that it represents an exponential sum which converges quickly in most cases than the geometric sum for path-based diffusion models like Personalized PageRank [Chung, 2007]. It can be advantageous in large graphs to get the desired results faster.

The main contribution of our method over the related state of the art is the combination of the following factors: (1) The presented method achieves better results in combining NMF with the heat diffusion method; (2) For a side effect and drug link prediction task, our method does not use any laboratory data like other previous studies such as Drug-Drug interaction or other information from biomedical experiments, such as patients, or report data. In this work, we took advantage of the NMF method to make an initial prediction between node pairs in network layer 1. We borrowed the concept of NMF based matrix factorization, which is successful in missing value imputation from large and sparse matrices. The same concept has been applied in recommender systems for predicting ratings for movies. To the best of our knowledge, it is the first time NMF is incorporated with a heat diffusion method for link prediction. This concept enabled us to integrate two different graphs. We learned the weights using matrix factorization in a bipartite graph from the first layer. The learned weights are then propagated in the second layer. From the result, we observed that the combination methods performed better than independent methods such as NMF or heat diffusion alone, and also that they beat the other state of the art link prediction algorithms.

5.5 Conclusion

This work proposed a novel method incorporating matrix factorization and diffusion and was applied for side effects and drug prediction, as well as for academic entities and social medial. The performance of the combined NMF and heat diffusion model was compared with a state-of-the-art method, where our experimental results showed that the proposed method significantly outperforms others in predicting links between them. For the side effects and drug link prediction, there might be a limitation in the construction of semantic similarity between drugs. In this work, we did not apply any semantic similarity threshold. It might have introduced some noise in the graph, which might have influenced the prediction performance. Also, social media graphs might contain noise which can explain the lower accuracy obtained, but we showed that the combination of different graphs and methods can still improve the results in comparison to state of the art methods.

In the next chapter, we will introduce the novel diffusion algorithm called Boundary Heat Diffusion (BHD). BHD has the advantage of doing long-range (global) and short-range (local) diffusion. In the same chapter, we will introduce the node classification in a multiplex network using a few labeled and large unlabeled data. We further demonstrate how BHD can adapt to various label correlation problems such as heterophily, homophily, and mixed node classification.

Chapter 6

A Boundary Based Heat-Diffusion Method for Node Classification in a Multiplex Network

This chapter proposes a novel approach to node classification in a graph. Node classification is one of the vital graph learning problems that predict the labels of a node. A node of a graph represents any real-world entity, such as a user in a social network, a publication in a citation network or a protein in a protein-protein interaction network. Real-world networks exhibit a network of network structure, which is also called multiplex networks. In this work, we proposed a novel boundary-based heat diffusion algorithm that guarantees a closed-form solution. Experiments on five real-world multiplex network datasets related to political, social, co-authorship, and biological, genetic interaction networks demonstrate the benefits of the proposed algorithm, where boundary-based heat diffusion outperforms the top state of the art methods. Our approach also adapts to the label prediction problem in any labels (i.e., homophily, heterophily, and a mixture of them). Parts of the research reported in this chapter have been submitted in the journal of Applied Soft Computing[1].

6.1 Introduction

Real-world networks often demonstrate a layered structure in which links in each layer reflect the function of nodes in different environments [Buono et al., 2014]. These interconnected networks are often called *networks of networks* [Gao et al., 2011] or

[1] https://www.journals.elsevier.com/applied-soft-computing

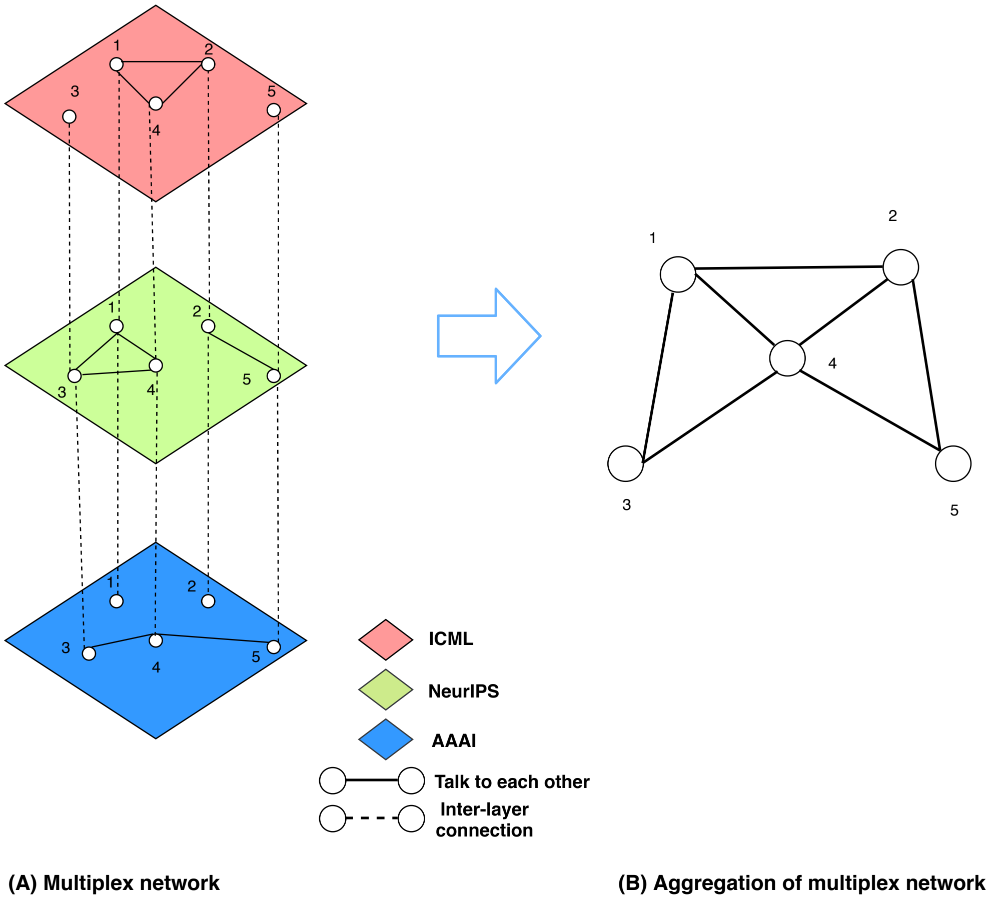


Figure 6.1: (A) Multiplex network of AI scientists talking to each other. Each layer represents an AI conference (ICML, NeurIPS, AAAI). (B) Aggregated multiplex network into a single-layer network.

multiplex networks [Battiston et al., 2014, Liu et al., 2018, Mucha et al., 2010]. In a multiplex network, the same nodes are linked by different networks (layers). For instance, multiplexes are good descriptions of a scientist's social network, where the nodes represent the scientist, and the different layers correspond to different scientific conferences they attend and in which they form different connections. For illustration purposes, Figure 6.1(A) shows scientists talking to each other in three different Artificial Intelligence (AI) conferences (ICML, NeurIPS, AAAI), represented by different colors. These three different conferences are the three different layers in the multiplex networks. The important observation in this example is that the same scientists in every layer or conference can be captured by interlayer edges connecting each scientist to its copies in other layers. Figure 6.1(B) exhibits the aggregated multiplex network without interlayer edges. In a practical setting, such interlayer edges are omitted for the sake of simplicity [Newman, 2018] because they represent self-edges.

The social and technological innovation brought by, for example, the worldwide web, biomedicine, and social networks has exposed the need to consider that networks might be made up of many different layers of interactions. Compared to single-layer networks, the topological and dynamic properties of a multiplex network are different [Cozzo et al., 2013, Wang et al., 2013]. The study of propagation processes in multiplex networks is a rapidly evolving research area [Buono et al., 2014]. For instance, a diffusion process can have an enhanced-diffusive behavior on a multiplex network, which means that the time scale associated with it is shorter than that occurring on a single layer network [Gomez et al., 2013]. Due to this, it is quite intriguing to consider how label propagation algorithms work in a multiplex network.

Although label propagation algorithms work reasonably well in the majority of networks with a single layer, we do not explicitly know how these algorithms behave in a multiplex network. In such a network, where nodes overlap between the layers, there is a high possibility of node misclassification by using ordinary label propagation algorithms [Fortunato, 2010]. The diffusion kernels [Kondor and Lafferty, 2002] used in the label propagation algorithm capture the long-range relationships (global information) between nodes in the network. Due to this, long-range diffusion puts more emphasis on random walks that explore more of the multiplex network, which eventually leads to misclassification. However, there are specific real-world multiplex networks that tend to be linked by shortest diffusion paths, such as proteins that have similar functions [Zhou et al., 2002].

Another example is in image segmentation, which is one of the critical areas to extract information from the images in a computer vision problem. Currently, the use of a multiplex network [Hu et al., 2012] has improved the precision of image segmentation because it allows the analysis of networks with multiple resolutions. The multiplex networks applied in image segmentation [Browet et al., 2011, Raghavan et al., 2007] problems use classical label propagation algorithm applicable for a single layer network. Using such label propagation algorithms in multiple layers might misclassify the images due to long-range diffusion. Therefore, we need a diffusion kernel which can control the propagation in the layers and efficiently classify the nodes in a multiplex network.

In this chapter, we propose a node classification algorithm that handles the concerns mentioned above. Our algorithm uses the intuitive and natural model of a physical heat diffusion system with boundary conditions. The heat flow can be captured by measuring the heat between points in the network, and the heat amount that is added and removed from the system. Here, the points at which heat is measured can be represented by nodes in a graph, and edges are associated with heat flows between those points. The injection and extraction points can be viewed as the boundaries of the system. Due to this idea, heat diffusion with boundary condition will control the heat flow, which makes it ideal for node classification in a multiplex network.

Our contributions to heat diffusion with boundary condition (BHD) are summarized as follows:

1. We develop a closed-form solution to BHD.
2. Theoretically, we show how the popular harmonic function is the limit case for BHD.
3. We develop an iterative method to BHD, whose computation complexity is linear in the number of edges.

Consequently, BHD has the following advantages for its applications:

1. Accuracy: Our algorithm achieves relatively good accuracy on different label propagation tasks in a multiplex network in comparison with state of the art methods.
2. Scalable: it can be applied to a large graph as BHD is linear in the number of edges.
3. Parameter estimation: BHD has just one parameter which can be chosen using cross-validation from the training set.
4. Adaptability: Our algorithm also adapts the diffusion in a single-layer network with homophily, heterophily, and mixed labels for node classification.

Moreover, we performed extensive experiments using five different labeled multiplex networks: a political membership network, a coauthorship network, diffusion innovation in physician network, and drug mechanism of action and tumor location prediction. The results demonstrated that our algorithm often outperforms state of the art label propagation algorithm in terms of top p% label prediction and classification accuracy. To the best of our knowledge, our algorithm is the first solution to handle node classification using label propagation in a multiplex network relying only on the graph structure.

6.2 Problem Formulation

This section details terms and introduces the node classification problem in a multiplex network. Suppose $\mathcal{N}$ is the list of nodes and there are K different types of edges which are expressed as $G_1, \ldots, G_K$. For every edge-based network $G_k = (\mathcal{N}_k, E_k)$, we have $E_k \subseteq \mathcal{N}_k \times \mathcal{N}_k$, and $\mathcal{N}_k \subseteq \mathcal{N}$. The set of nodes is composed of two types of components

$\mathcal{N} = L \cup U$ where $L = \{n_1, \ldots, n_l\}$ is a set of l labelled nodes and $U = \{n_{l+1}, \ldots, n_{l+u}\}$ is the list of unlabeled nodes.

Given a set C of c possible labels, let $f = [f_{ip}]$ be a matrix, where $f_{ip} = 1$ if node $i \in \mathcal{N}$ has label $p \in C$, and 0 otherwise. We can observe f_L, a part of f, where nodes are restricted to the set L. The problem is to predict the other part f_U of f, where nodes are restricted to the set U. Thus the **node classification problem** in multiplex networks is expressed as follows:

- **Input:** A partially labeled multiplex network. That is, $G_1, \ldots, G_K$, and f_L.
- **Score:** Find a score S_{ip} for each unlabeled node i and each $p \in C$. A good method will have the property that larger values for S_{ip} imply more probably that i takes the label p. In an evaluation, the precision can be produced based on such scores.
- **Decision:** Assess through checking whether the classification function $\arg\max_p S_{ip}$ results in the same label as the ground truth, and produce accuracy in the evaluation.

6.3 Methodology

6.3.1 Network Aggregation

We took the classical approach to transform a multiplex network into a single-layer network for BHD application. For this, we aggregate data from the different layers of a multiplex network. We assume that the component networks in a multiplex network complement each other, and to integrate the available information, we average their adjacency matrices to generate a weight matrix $W = [W_{ij}]$ for a single-layer network. This aggregation approach is similar to the one proposed by DeFord et al. [DeFord and Pauls, 2017]. Also, in multiplex networks, the inter-layer edges disappear as a result of aggregation processes because self-edges (nodes having links to itself) cannot account for information propagation to other nodes [Holme, 2005].

6.3.2 Heat Diffusion in a Boundary Condition in Graph (BHD)

In the context of our work, we are considering diffusion in a boundary condition. By boundary condition, we mean that we have some information about the solution at the endpoints.

Let us suppose that there are l labeled and u unlabeled nodes and $N = l + u$ be the total nodes in the multiplex graph. Then $L = \{1, 2, ..., l\}$ corresponds to labeled nodes with labels $f_1, ..., f_l$, and nodes $U = \{l + 1, l + 2, ..., l + u\}$ refers to the unlabeled points. Our job here is to assign the labels for the nodes U. The edge of the graphs is a $n \times n$ weight matrix W also known as adjacency matrix.

Now to formulate our model, let us assume that, at time t, each node $i \in U$, receives a certain amount of heat $M(i, j, t, \Delta t)$ from its neighbor j during a period of Δt. The heat $M(i, j, t, \Delta t)$ is proportional to the time Δt and the heat difference $f_j(t)$ - $f_i(t)$. Due to this, the heat difference at node i between time $t + \Delta t$ and time t will be equal to the sum of the heat that it receives from all of its neighbors. This is expressed as:

$$f_i(t + \Delta t) - f_i(t) = \sum_{j=1}^{n} (f_j(t) - f_i(t)) W_{ij} \Delta t \tag{6.1}$$

Dividing Eq. 6.1 by Δt into both sides, and let $\Delta t \to 0$, we have

$$\frac{df_i}{dt} = W_{i,:} f - d_i f_i \tag{6.2}$$

In terms of matrix operations, we split the weight matrix W into four blocks after the L^{th} row and column:

$$W = \begin{bmatrix} W_{LL} & W_{LU} \\ W_{UL} & W_{UU} \end{bmatrix} \tag{6.3}$$

Note that $W_{U,:} f = [W_{UL}\, W_{UU}] \begin{bmatrix} f_L \\ f_U \end{bmatrix}$, and $\Delta_{UU} = D_{UU} - W_{UU}$. Here Δ is the the combinatorial Laplacian which is given in the matrix form as $\Delta = D - W$ where $D = diag(d_i)$. The $diag(d_i)$ is the diagonal matrix with entries $d_i = \sum_j w_{ij}$ and $W = [w_{ij}]$ is the weight matrix.

We have a matrix form:

$$\begin{aligned}\frac{df_U}{dt} &= W_{U,:}f - D_{UU}f_U \\ &= W_{UL}f_L + W_{UU}f_U - D_{UU}f_U \\ &= W_{UL}f_L - \Delta_{UU}f_U\end{aligned} \tag{6.4}$$

Solving this linear differential equation which is the form of $dy/dx + Py = Q$ to find the closed form solution we have:

$$\frac{df_U}{dt} = W_{UL}f_L - \Delta_{UU}f_U \tag{6.5}$$

Here $P = \Delta_{UU}$ and $Q = W_{UL}f_L$

$$Integrating\ Factor\ (IF) = e^{\int Pdt} = e^{\int \Delta_{UU}dt} = e^{\Delta_{UU}t}$$

$$e^{\Delta_{UU}t}\frac{df_U}{dt} + e^{\Delta_{UU}t}\Delta_{UU}f_U = e^{\Delta_{UU}t}W_{UL}f_L$$

$$\frac{d}{dt}(e^{\Delta_{UU}t}f_U) = e^{\Delta_{UU}t}W_{UL}f_L$$

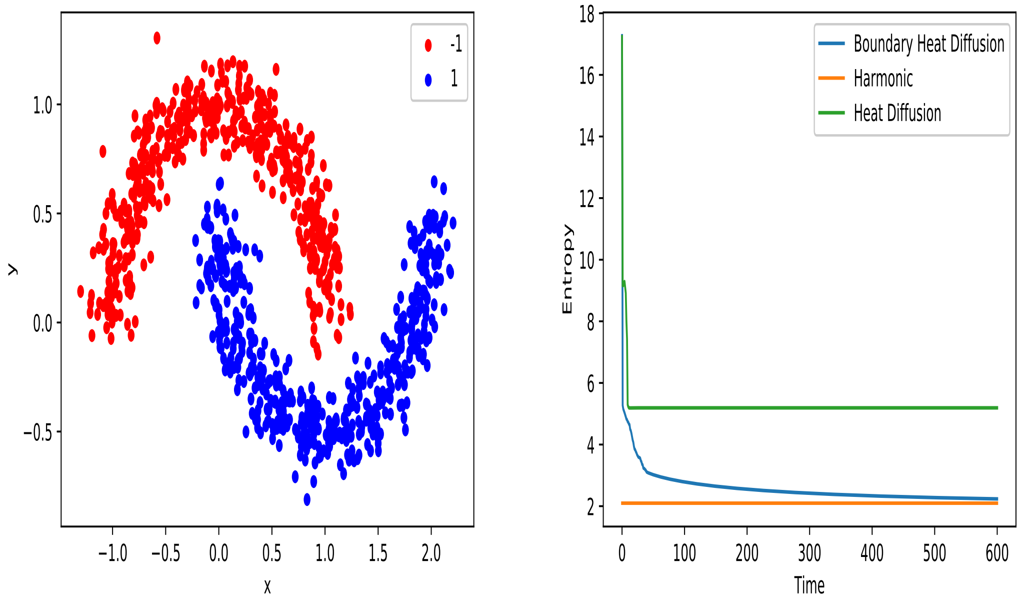
$$\begin{aligned}\int d(e^{\Delta_{UU}t}f_U) &= \int e^{\Delta_{UU}t}W_{UL}f_L dt \\ e^{\Delta_{UU}t}f_U &= e^{\Delta_{UU}t}\Delta_{UU}^{-1}W_{UL}f_L + C\end{aligned}$$

$$f_U = e^{-\Delta_{UU}t}e^{\Delta_{UU}t}\Delta_{UU}^{-1}W_{UL}f_L + e^{-\Delta_{UU}t}C$$

$$f_U = \Delta_{UU}^{-1}W_{UL}f_L + e^{-\Delta_{UU}t}C \tag{6.6}$$

This is the temperature distribution on the unlabelled nodes at time t, given the boundary condition f_L. This function is used to predict the labels for the unlabelled node. Given the initial condition $f_U|_{t=0} = f_U(0)$, $C = f_U(0) - \Delta_{UU}^{-1}W_{UL}f_L$. Note that, in the limit $t \to \infty$, f$_U = \Delta_{UU}^{-1}W_{UL}f_L$, which is the harmonic function.

In order to interpret Equation 6.6 and the heat diffusion with the boundary process more intuitively, we constructed two moon-shaped simulated data from 1000 points with a standard deviation of 0.1. The pattern of 2 moons shaped data is shown in the Figure 6.2[a]. The red data points have label -1, and the blue data points have label +1. There are two visibly separate clusters in the data. We employed the Gaussian RBF Kernel $w_{ij} = \exp\big(-\frac{||x_i - x_j||^2}{2\sigma^2}\big)$ to construct the graph between these points and randomly choose two points from each of the labels (-1,1) and consider the rest of the points as unlabeled. We apply the closed-form equations for heat diffusion, harmonic function and boundary heat diffusion. Figure 7.1b[b] shows the performance of these algorithms. The y-axis is the cross-entropy loss, and the x-axis is the time. The harmonic function does not have the time component in its equation, but HD and BHD both have the time component. We can see from the curve that when time equals to 0, both algorithms have the highest cross-entropy loss. As time increases, HD and BHD both started to have a low cross-entropy loss. At time equals to 15, HD started to converge. There is no further reduction of the entropy, whereas the BHD kept decreasing till to the point it has a similar entropy to harmonic function. It means for a higher value of time; the BHD will be the same as harmonic function favoring continuous or long-range propagation.



(A) 2 Moon-Shaped Data.

(B) Error Curves.

FIGURE 6.2: The error curve of different label propagation algorithms with time increasing from 1 to 600.

6.3.3 Computational Complexity

In the solution provided by Equation 6.6 we have to consider two parts: (i) the harmonic part and (ii) the exponential part. When the graph is large, the computation will be time-consuming because both of the terms have a $O(n^3)$ complexity. To solve this, we took an iterative approach to compute the harmonic part provided by Zhu et.al [Zhu and Ghahramani, 2002], which is the same as a random walk with restart (RWR) [Tong et al., 2006]. For the exponential part we took the discrete approximations by Yang et.al [Yang et al., 2007]: $f(t) = \left(I - \frac{t}{M}\Delta_{UU}\right)^M f(0)$, where M is the number of iterations chosen as $M = 30$, as in [Yang et al., 2007], and I is the identity matrix. t is from the cross-validation in the training set ranging from $[0.05, ..., 5]$ with each iteration increased by 0.05. $f(0)$ is the initial temperature and $f(t)$ is the temperature at timestamp t. Specifically, after the discrete formalization of the complexity of exponential kernel in our model is given by $O(M|E|n)$ where M is the number of iterations, n is the number nodes and $|E|$ is the number of edges in the graph.

6.3.4 Initial Temperature Setting

The heat diffusion with the boundary condition process needs the initial temperature in the boundary to propagate heat in the network. We set the initial temperature at time zero for the labeled nodes as 1.

Initial Temperature Setting on a Test Set: If we do not have the network data or the quality of the network is poor, the ideal way to make inference about the probability that a node has label 1 in the test set is to use the sample mean. It can be assumed that nodes without any links are from a population that is binomially distributed with an unknown success probability p. We can then make an inference of p by a sample mean [Miller et al., 1999], which is used to set the initial temperature for the unlabeled node in a test set.

However, when the network contains valuable information for making the prediction, this initial estimation should be combined with diffusion along the edges in the network. Our method has this property shown in Equation 6.6, while the harmonic function ignores this initial estimation because of the long-range diffusion. Moreover, if the network is of poor quality, our model has the freedom of choosing a small value of t.

6.4 Algorithm

As already mentioned, heat diffusion with boundary condition has two-part (harmonic and heat diffusion). For the harmonic part, we took the iterative approach provided by Zhu et al. [Zhu and Ghahramani, 2002]. This algorithm requires $n \times n$ transition matrix, $n \times c$ label matrix where c is the number of labels, $n \times n$ Laplacian matrix Δ, and M is the number of iterations. Once we calculate the harmonic score, we need to calculate the constant C, as shown in Equation 6.6. C is obtained by subtracting an initial label matrix from a harmonic score. The initial label matrix has an initial temperature for each node. We imputed the values for the unlabeled nodes as the means of the labeled nodes. This C is the initial condition of state matrix ($n \times c$) for heat diffusion with boundary conditions. Formally the algorithm is described in Algorithm 1

Algorithm 1: Heat Diffusion with Boundary Condition

```
Input  : The transition matrix $T$ of size $n \times n$; initial label matrix Y of size $n \times c$;
         Laplacian matrix $\Delta$; $M$ is the number of iteration chosen as 30; $I$ is the
         identity matrix of size $n \times n$
Output : State matrix of size $n \times c$
Initialize U = Y
repeat
    $Y^{k+1} \leftarrow TY^k$
    Row Normalize: $Y^{k+1}$
    $Y^{k+1} \leftarrow Y^{k+1} + U$
    $Y^k = Y^{k+1}$
    $k = k + 1$
until error between $Y^{k+1}$ and $Y^k$ becomes sufficiently small
Initial_Temperature: Impute mean value for unlabelled nodes using labeled value
  from column of matrix U
C = Initial_temperature - $Y^K$
State_Matrix = C
$t$ is a parameter in (0,5);
for $b = 1$ to $M$ do
    $State\_Matrix = Y^K + \left(I - \frac{t}{M}\Delta\right) State\_Matrix$
end
return $State\_Matrix$
```

6.4.1 Parameter t

Parameter t has a vital role in the diffusion process. If t has a high value, heat will diffuse very quickly. From Equation 6.6, we see that if t tends to ∞, then the heat diffusion with boundary condition will become a harmonic function. It means that the heat will travel deeper into the graph, also known as long-range or global diffusion. If t is small, then heat will diffuse slowly, favoring short-range or local diffusion. Different networks have different values of t. For instance, rumor or fake news propagate faster in a social network than true stories [Vosoughi et al., 2018]. In that case, t is high because heat immediately transfers to the rest of the neighbors, making the diffusion process faster.

6.5 Experiments

In this experiment we answer the following questions:

- Q1: *Parameter:* Does the parameter t affect the performance of heat diffusion with boundary conditions in a multiplex network?
- Q2: *Accuracy:* Do existing label propagation algorithms that work successfully in a single layer network can have similar performance in a multiplex network?
- Q3: *Adaptability:* Can heat diffusion with boundary condition adapt to various labels such as (i) homophily, (ii) heterophily, and (iii) mixed?

Datsets: We used five different multiplex network datasets from the different domains with ground truth labels in our experiments. The Table 6.1 shows the description of our datasets.

Datasets	**#Layers**	**#Nodes**	**#Edges**	**#Labels**	**Directed**
US President Party Membership RDF data	3	71	101	14	✓
CKM Physicians	3	246	1,551	3	✓
NG_Leskovec Collaboration Data	4	191	1,836	2	✓
Drugs Mechanism of Actions	4	4,054	354,566	7	
Tumor Location	4	4,054	354,566	10	

Table 6.1: Multiplex Network Datasets.

Among the five datasets, three datasets were of public access collected from scientific studies and a publicly available multiplex network repository[2]. We curated the drug's mechanism of action and tumor samples-gene datasets for our experimental study from publicly available drug and genomics databases.

[2] https://comunelab.fbk.eu/data.php

1. **US President Party Membership RDF Data:** This data was reused from the study by [Nickel et al., 2011]. This dataset was created for the US president's example, by retrieving the names of all presidents and vice presidents of the United States from DBpedia[3], in combination with their party membership and the presidentOf/vicePresidentOf information.

2. **CKM Data:** This data was of the physicians in four towns in Illinois, Peoria, Bloomington, Quincy, and Galesburg by [Coleman et al., 1957]. This multiplex network is about the impact of network ties on the physicians' adoption of a new drug. There are three layers in this network, and the labels are the researchers associated with their original companies.

3. **NG and Leskovec Co-Authorship Data:** This data [Chen and Hero, 2017] contains the coauthors of Prof. Andrew Ng and Jure Leskovec at Stanford University from the year 1995 to the year 2014. This multiplex graph is a 4-layer temporal graph. For each layer, there is an edge between two researchers if they coauthored at least one paper in the 5-year interval.

4. **Drugs Mechanism of Action Data:** This is a genetic data where each gene has a label with drug actions. There are seven types of mechanisms labeled for the target of the drugs, namely: (i) blocker, (ii) antagonist, (iii) agonist, (iv) activator, (v) inhibitor, (vi) channel blocker and (vii) binder. The gene-gene interaction data has four different layers, text mining, experimental, coexpression, and co-occurrence. This dataset is from the publicly available DGIdb database[4] that annotates the genes concerning drug-gene interactions and potential druggability. The gene-gene interaction data is used from the publicly available STRING version 10.5 protein-protein interaction database[5].

5. **Tumor Location:** This is a cancer genetic data where each gene has a label with the anatomical location of the human body. There are ten location labeled for the target genes namely: (i) liver, (ii) large intestine, (iii) urinary tract, (iv) breast, (v) kidney, (vi) lungs, (vii) endometrium, (viii) prostate, (ix) thyroid and (vii) upper aero-digestive tract. The gene-gene interaction data is the same as the drug's mechanism of action data.

Evaluation: We hide 90% of labeled nodes on each network. Then we applied the algorithms to infer the hidden labels. We reported (i) precision@p and (ii) accuracy metric because both of these metrics assess the performance of the label propagation

[3] `https://wiki.dbpedia.org/`

[4] `http://www.dgidb.org/`

[5] https://string-db.org/

algorithm. Precision@p is the precision of top p% nodes ordered by their maximum score of $max_p S_{ip}$. The accuracy score computes the subset accuracy in a multilabel classification task. All of the evaluation metrics reported are on ten-fold cross-validation.

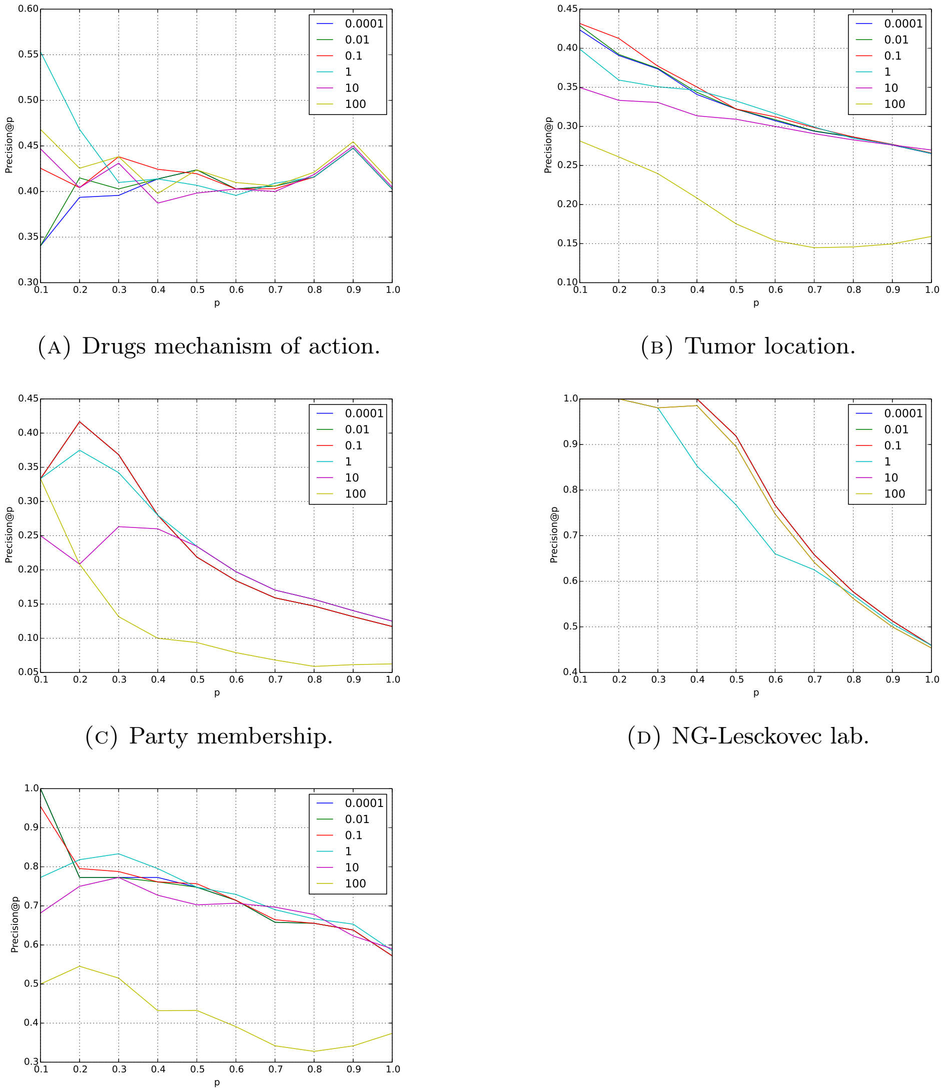
Q1: Parameter We wanted to assess node classification ability of our approach by varying the parameter t from 0.01 to 100. We report the result of all the multiplex network data that we used in our studies. From Figure 6.3, we see that the largest value of the parameter t leads to low precision in comparison to smallest t. The values of t less than or equal to 1 lead to high precision for small p and also perform better than large t. Although for all p, we do not achieve the best results with the small value of t. In figures 6.3b, 6.3c and 6.3e we saw that setting t as 100, the precision quickly dropped, this means that as t increases, heat will quickly transfer over all the network leading to miss-classifications. From our observation in this data, the multiplex networks for node classification tends to favor small values of t. It means that in this kind of network, short-range diffusion is supported. We can say that setting $t < 1$ achieves high precision overall in all networks.

Q2:Accuracy We compared our approach with several state of the art label propagation algorithms namely: (i) harmonic function [Zhu and Ghahramani, 2002], (ii) Local and Global Consistency Method (LGC) [Zhou et al., 2004], (iii) OmniProp [Yamaguchi et al., 2015], (iv) CAMLP [Yamaguchi et al., 2016] and (v) Heat Diffusion [Yang et al., 2007] by 10 folds cross-validation using randomly constructed 10 test sets for all multiplex networks to compute precison@10%, precision@100% and accuracy. The LGC algorithm requires the value for parameter α which lies in the range $[0, 1]$. We estimated the value for α from 10 fold cross-validation in the training sets. Similarly, OmniProp and CAMLP has one parameter with effective default value 1 [Yamaguchi et al., 2015, 2016]. We estimated the parameter t for BHD by cross validation on the training datasets from the range of 0.05 to 5 with a step of 0.05.

Table 6.2 shows the results of all the algorithms in the Multiplex network data. We can see that BHD performs either equaling or surpassing the state of the art algorithms in terms of precision and accuracy. In terms of accuracy and precison@100, our method has always performed better in comparison to other label propagation algorithms. Moreover, BHD showed significant improvements over all the state of the art label propagation algorithms for US-President party membership prediction and Drugs Mechanism of action prediction in terms of precision@10, precision@100, and accuracy metrics. In the CKM Physician multiplex network, Omniprop performed better in terms of precison@10 and precison@100 than all of the algorithms because Omniprop is claimed to work better with the precision for top small p% nodes.

	CKM Physican			**NG-Leskovec**			**US President**			**Drugs** **Mechanism of Action**			**Tumor Location**		
	Prec@10	Prec@100	Accuracy	Prec@10	Prec@100	Accuracy	Prec@10	Prec@100	Accuracy	Prec@10	Prec@100	Accuracy	Prec@10	Prec@100	Accuracy
HMN	0.69 ± 0.31	0.66 ± 0.08	0.63 ± 0.10	**1.0 ± 0.00**	**0.71 ± 0.15**	0.99 ± 0.07	0.26 ± 0.24	0.11 ± 0.04	0.12 ± 0.03	0.42 ± 0.05	0.331 ± 0.01	0.29 ± 0.06	0.41 ± 0.04	0.32 ± 0.08	0.29 ± 0.01
LGC	0.69 ± 0.31	**0.67 ± 0.09**	0.64 ± 0.11	**1.0 ± 0.00**	**0.71 ± 0.15**	0.99 ± 0.01	0.25 ± 0.251	0.11 ± 0.04	0.12 ± 0.05	0.43 ± 0.06	0.33 ± 0.01	0.30 ± 0.08	0.47 ± 0.04	0.32 ±0.09	0.29 ± 0.01
CAMLP	0.75 ± 0.10	0.65 ± 0.09	0.62 ± 0.11	0.97 ± 0.07	0.70 ± 0.15	0.99 ± 0.01	0.25 ± 0.26	0.11 ± 0.05	0.12 ± 0.05	0.48 ± 0.04	0.34 ± 0.01	0.31 ± 0.06	**0.49 ± 0.02**	0.33 ± 0.05	0.30 ±0.03
OMNI	**0.81 ± 0.11**	**0.67 ± 0.07**	0.64 ± 0.10	**1.0 ± 0.00**	0.70 ± 0.15	0.99 ± 0.01	0.36 ± 0.26	0.12 ± 0.05	0.11 ± 0.052	0.48 ± 0.02	0.33 ± 0.02	0.29 ± 0.03	0.44 ± 0.05	0.31 ± 0.09	0.27 ± 0.010
HD	0.79 ± 0.08	0.65 ± 0.09	0.63 ±0.11	0.97 ± 0.07	0.70 ± 0.15	0.99 ± 0.01	0.25± 0.29	0.11 ± 0.05	0.12 ± 0.07	0.49 ± 0.025	0.32 ±0.010	0.30 ± 0.01	0.47 ± 0.03	0.31 ± 0.07	0.30 ± 0.05
BHD	0.73 ± 0.14	0.65 ± 0.11	**0.74 ± 0.11**	**1.0 ± 0.00**	**0.71 ± 0.15**	0.99 ± 0.07	**0.43 ± 0.30**	**0.22 ±0.16**	**0.44 ± 0.33**	**0.51 ± 0.051**	**0.44 ± 0.09**	**0.48 ±0.05**	0.47 ± 0.04	**0.35 ± 0.01**	**0.34 ± 0.01**

TABLE 6.2: The Result of 10 fold cross validation of boundary based heat diffusion with state of the art label prediction algorithm in a multiplex network. The figure behind ± sign is the standard deviation.



(A) Drugs mechanism of action.

(B) Tumor location.

(C) Party membership.

(D) NG-Lesckovec lab.

(E) CKM Physician.

FIGURE 6.3: Impact of parameter t in Multiplex networks. X-axis is the percentage of data ranked. Y-axis is the Precision at each percentage of the data.

Q3: Adaptability In this experiment, we assess our boundary-based heat diffusion algorithm in homophily, heterophily, and mixed (homphily+heterophily) labeled networks. For this task, we took three real-world network datasets described in Table 6.3. PolBlogs is a blog citation network where the labels are political orientations of blogs. Facebook and POKEC-G are the social network sites where the labels are the genders of the users. All the datasets are available on the web[6].

[6] https://snap.stanford.edu/data/

Datasets	**Nodes**	**Edges**	**Directed**	**Label Types**	**Labels**
PolBlogs [Adamic and Glance, 2005]	1,490	19,090	✓	homophily	2
Facebook [Traud et al., 2012]	6,637	249,967		mixture slight homophily	3
POKEC-G [Takac and Zabovsky, 2012]	1,632,803	30,622,564	✓	mixture slight heterophily	2

Table 6.3: Network Datasets.

Table 6.4 is a summary of the results. Overall, we observe that BHD performed either equaling or outperforming the state of the art methods. In the homophily (PolBlogs) and heterophily (Facebook) networks, the BHD outperformed all the methods. In the mixed labeled (POKEC-G) network, the OMNI-prop method marginally outperformed BHD in the precision@10 metric. As OMNI-prop propagates in a Bayesian manner, the amount of evidence it receives from neighboring nodes strengthen its performance [Yamaguchi et al., 2015]. However, in precision@100 and overall accuracy of the classification, BHD performed better. These results demonstrate that our BHD method performs well on various types of labels.

	PolBlogs			**Facebook**			**POKEC-G**		
	Prec@10	Prec@100	Accuracy	Prec@10	Prec@100	Accuracy	Prec@10	Prec@100	Accuracy
HMN	0.97 ± 0.02	0.79 ± 0.01	0.82 ± 0.01	0.53 ± 0.01	**0.53 ± 0.04**	**0.54 ± 0.03**	0.50 ± 0.01	0.50 ± 0.01	0.50 ± 0.01
LGC	0.97 ± 0.02	0.79 ± 0.01	0.82 ± 0.01	0.52 ± 0.01	0.52 ± 0.01	0.51 ± 0.01	0.50 ± 0.01	0.50 ± 0.01	0.50 ± 0.01
CAMLP	0.90 ± 0.01	0.78 ± 0.01	0.80 ± 0.01	0.53 ± 0.01	0.52 ± 0.01	**0.54 ± 0.01**	0.50 ± 0.01	0.50 ± 0.01	0.50 ± 0.04
OMNI	**0.99 ± 0.01**	0.80 ± 0.01	0.83 ± 0.02	0.55 ± 0.02	0.53 ± 0.02	0.53 ± 0.01	**0.92 ± 0.02**	0.70 ± 0.01	0.74 ± 0.01
HD	0.97 ± 0.01	0.79 ± 0.06	0.81 ± 0.04	0.51 ± 0.01	0.49 ± 0.04	0.49 ± 0.06	0.51 ± 0.01	0.50 ± 0.03	0.50 ± 0.01
BHD	**0.99 ± 0.01**	**0.90 ± 0.04**	**0.95 ± 0.04**	**0.56 ± 0.03**	**0.53 ± 0.04**	**0.54 ± 0.03**	0.90 ± 0.01	**0.82 ± 0.01**	**0.92 ± 0.01**

Table 6.4: Summary of results on networks with different label types.

6.6 Conclusion

We presented a novel heat diffusion method with the boundary condition, which addresses the node classification problem on multiplex networks. Our method shows that it can adapt to other kinds of networks with varieties of labels (homophily, heterophily, and mixed). BHD assigns a node with the initial temperatures which act as the boundary and diffuse the heat in the network. This model requires a time parameter, which controls the range of propagation in the network. The advantages of our algorithm are:

1. Accuracy: it outperforms or equals the state of the art algorithms in label propagation on node classification tasks in multiplex networks (Table 6.2). BHD never loses in terms of accuracy to methods like HMN because in the worst-case scenario, when the time parameter tends to infinity, BHD becomes HMN.
2. Adaptability: BHD adapts to various networks with different label types and outperforms or equals the state of the art algorithms (Table 6.4).
3. Linear: Our algorithm has a closed-form solution that can be evaluated in a finite number of steps (Algorithm 1).
4. Parameter: It has only one timestamp parameter t, which controls the heat flow for long or short-range diffusion (Figure 6.2).

We believe that our boundary-based heat diffusion method is a simple but effective method for node classification in various types of networks with different label properties.

In chapter 7, we will demonstrate how we can take advantage of BHD in regression problems. Most of the graph-based methods are applied only in a classification problem, and regression is a slightly ignored case. The graph-based regression is closely related to node classification problems, where instead of discrete labels, we are predicting continuous labels. We took advantage of a few labeled and large unlabeled data using graph-approach in manifold data to predict the continuous values of the nodes in the graph.

Chapter 7

Semi-Supervised Regression Using Diffusion on Graphs

In chapter 6, we demonstrated the boundary-based heat diffusion (BHD) approach for node classification in a multiplex network and its adaptability across different kinds of networks. In this chapter, we extend the ability of BHD to tackle semi-supervised regression problems. Most of the graph-based approaches focused on classification problems, and graph-based regression has received less attention. In this chapter, we explored our BHD approach applied to manifold data for a regression problem. Experiments from business, biomedical, physical, and social domain data show that the boundary-based heat diffusion method can effectively outperform the top state of the art methods. Parts of the research reported in this chapter have been submitted to the journal of Applied Soft Computing[1].

7.1 Introduction

Labeling data is often laborious, or expensive, as it requires the effort of human experts for annotation. However, there are large amounts of unlabeled data available in real-world machine learning applications. A typical example is speech recognition. It costs almost nothing to record huge amounts of speech, but labeling it requires humans to listen and transcribe. This process is burdensome and time-consuming. In such a case, “semi-supervised learning (SSL)" becomes useful to tackle the few labeled data and large unlabeled data. SSL uses the combined information of labeled and unlabeled data to improve classification performance.

[1] https://www.journals.elsevier.com/applied-soft-computing

Semi-supervised classification (SSC) is commonly used in pattern recognition and other real-world problems [Sugiyama et al., 2010, Zhang et al., 2007a]. However, Semi-supervised regression (SSR), is less explored. In SSC, the independent variable Y_i is constrained to have only a finite number of possible values, whereas, in SSR, Y_i is assumed to be continuous. Hence, SSC algorithms designed for graph min-cut [Blum and Chawla, 2001] do not apply to the more general SSR problem. Other algorithms, such as Gaussian Fields [Zhu, 2005], apply to both SSR and SSC by using graphs.

The co-training style algorithm developed by [Zhou and Li, 2005] demonstrated useful in SSR using graphs. Similarly, [Wang et al., 2006] proposed an algorithm, which is about a kernel regression framework exploiting both labeled and unlabeled examples. However, these algorithms require parameter tuning and for a large graph it will be computationally expensive.

The above discussion justifies the development of SSR using a graph-based propagation methods. Different graph-based diffusion approaches have different spreading mechanisms. For instance, PageRank uses a geometrically weighted sum of random walks; Heat diffusion uses an exponentially weighted sum of random walks [Chung, 2007]. This kind of diffusion impacts in the performance. [Yang et al., 2007] showed that heat style diffusion is robust to web spamming in comparison to PageRank style diffusion. Another important observation is that many graph-based label propagation algorithms suffer from the problem of continuous diffusion. It means that the label density is infinitely propagated in the network until the convergence is guaranteed [Kondor and Lafferty, 2002, Zhu and Goldberg, 2006, Zhu et al., 2003]. While performing a random walk with continuous diffusion, the algorithms explore more of the network. This behavior eventually leads to errors in prediction. Therefore, we need a diffusion function which can control the propagation in the network and efficiently predict the values for nodes in a network.

In this chapter, we applied the BHD algorithm in a semi-supervised regression problem. This algorithm is based on a physical heat diffusion systems. The heat flow between points in the network is captured by measuring the amount of heat added or removed from the system. The points represent nodes in a graph, and heat flow between the points is the edges. The injection and extraction points of the heat are the boundary of the system, which controls the heat flow. The final temperature distribution of the nodes after the heat diffusion process makes this technique ideal for a regression problem.

Our contributions in developing BHD for SSR are summarized as follows:

1. Accuracy: Our algorithm achieves relatively good prediction accuracy on different label propagation in regression datasets.

2. Parameter estimation: Heat diffusion with boundary condition has just one parameter with a default value of 1. It means there is no need for complex paramter tuning.

Moreover, we performed extensive experiments using seven different regression datasets from different domains: (i) Sales prediction using TV advertisement (ii) Boston housing price prediction, (iii) White wine alcohol volume prediction, (iv) Red wine alcohol volume prediction, (v) Parkinson's patient sound level prediction, (vi) Airfoil self noise prediction, and (vii) Bike-sharing rental count prediction. All the datasets used in the experiments are publicly available. Out of 7 datasets, 5 of the datasets (Parkinson, White wine, Red wine, Airfoil self-noise, Bike-sharing) are from a UCI machine learning data repository[2]. The Boston housing data is from the python open source scikit data repository[3]. Advertisement data is collected from the data repository [4] of the book "Elements of Statistical Learning" [Hastie et al., 2005]. Results demonstrated that our algorithm often outperforms state of the art label propagation algorithm in terms of prediction accuracy. To the best of our knowledge, our algorithm is the first solution to handle the graph-based regression problem using boundary-based heat diffusion, relying only on the graph structure.

7.2 Problem Formulation

This section details some terms and introduces the graph regression problem. Suppose $\mathcal{N}$ is the list of nodes and E is the number of edges. For undirected graph $G = (\mathcal{N}, E)$, we have $E \subseteq \mathcal{N} \times \mathcal{N}$ also $\mathcal{N}_i \subseteq \mathcal{N}$. The set of nodes is composed of two types of components $\mathcal{N} = \mathcal{N}^L \cup \mathcal{N}^U$ where $\mathcal{N}^L = \{n_1, ...n_L\}$ is a set of L labelled nodes and $\mathcal{N}^N = \{n_{L+1}, ...n_{L+U}\}$ is the list of unlabeled nodes. Let Y be the set of possible labels and $Y_L = \{y_1, ...y_L\}$ are the labels assigned to the nodes in $\mathcal{N}^L$. Thus, the graph regression problem is expressed as follows:

Problem (Graph regression)

- **Available:** A partially labeled graph.

[2] https://archive.ics.uci.edu/ml/index.php

[3] https://scikit-learn.org/stable/datasets/index.html

[4] https://web.stanford.edu/~hastie/ElemStatLearn/data.html

- **Score:** Find the score $S_{i,j}$ which corresponds to the value of the unlabeled node i through labeled node j.

- **Estimates:** The function estimates of the response variable:

$$\hat{S} = MS$$

where $\hat{S}$ are the new estimates, S are the observations, and M is a matrix which may be constructed based on the data.

7.3 Methodology

We are given an undirected graph constructed from data features by applying a distance similarity metric. We follow the same smoothness assumption made by [Zhu et al., 2003] that nodes close to each other have similar values. This idea also applies to the regression problem [Rwebangira and Lafferty, 2009]. Table 7.1 shows the list of symbols we used in this chapter.

Symbols	**Definitions**
W	Adjacency matrix
N,E	# of nodes, # of edges
L	# of labeled nodes
U	#of unlabeled nodes
t	time
D	Degree matrix
f_u	Temperature distribution of the unlabelled node
f_l	Temperature distribution of the labelled node

TABLE 7.1: Symbols and Definitions.

7.3.1 Graph Construction

We use a fully connected graph, where every pair of vertices x_i, x_j is connected by an edge. An edge between two vertices x_i, x_j represents the similarity of the two instances. One popular weight w_{ij} function used in a semi-supervised machine learning task is given by:

$$w_{ij} = \exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right) \tag{7.1}$$

This function is also called a Gaussian kernel or a Radial Basis Function (RBF) kernel [Zhu, 2005]. The edge weight decreases as the Euclidean distance $||x_i - x_j||$ increases, σ is known as the bandwidth parameter and chosen as $\frac{1}{n}$ where n is the number of features. The weight $w_{ij} = 1$ when $x_i = x_j$, and 0 when $x_i - x_j$ approaches to ∞.

7.3.2 Heat Diffusion in a Boundary Condition in Graph (BHD)

In Chapter 6 Section 6.3.2 we demonstrated the derivation of BHD. In this chapter, to make it self contained, we will briefly introduce BHD. The labels in this setting are the real values of the nodes, and unlabeled nodes are nodes with no values.

Let us suppose that there are l labeled and u unlabeled nodes and $N = l + u$ be the total nodes in the graph. Then $L = \{1, 2, ..., l\}$ corresponds to labeled nodes with labels $f_1, ..., f_l$, and nodes $U = \{l + 1, l + 2, ..., l + u\}$ refers to the unlabeled points. Our job here is to assign the labels for the nodes U. The edge of the graphs is a n x n weight matrix W also known as adjacency matrix. The final solution for the unlabeled nodes using BHD is given by,

$$f_U = \Delta_{UU}^{-1} W_{UL} f_L + e^{-t\Delta_{UU}} C \tag{7.2}$$

This is the temperature distribution on the unlabelled nodes at time t, given the boundary condition f_L. This function is used to predict the labels for the unlabelled node. Given the initial condition $f_U|_{t=0} = f_U(0)$, $C = f_U(0) - \Delta_{UU}^{-1} W_{UL} f_L$. Note that, in the limit $t \to \infty$, f$_U = \Delta_{UU}^{-1} W_{UL} f_L$, which is the harmonic function.

In order to interpret Equation 7.2 and the heat diffusion with the boundary condition intuitively, we simulated the regression datasets with 1000 data points in two different data shapes one linear (standard deviation (σ)= 40) and another spiral. Both datasets contain two features and one target variable. The pattern of data is shown in Figures 7.1a and 7.1c. We labeled 2 data points which are in a red triangle and orange star, and the rest of the data are unlabeled, which is in black. We employed the Gaussian RBF Kernel $w_{ij} = \exp\big(-\frac{||x_i - x_j||^2}{2\sigma^2}\big)$ to construct the graph between these points and applied the closed-form equations for heat diffusion, harmonic function and boundary heat diffusion. Figures 7.1b and 7.1d show the performance of these algorithms. The y-axis is the Root Mean Square Error (RMSE), and the x-axis is the time. The harmonic function does not have the time component in its equation, but HD and BHD both have the time component. We can see from the curve that when time equals to 10^{-4}, both HD and BHD algorithms have the highest RMSE (see Figure 7.1b and 7.1d).

In the case of linear-shaped data, as time increases, HD and BHD both started to have a low RMSE. At time equals to 10^{-1}, HD started to converge Fig 7.1b. There is no further reduction of the RMSE, whereas the BHD kept decreasing till to the range between 10^{-1} to 10^{0}. Beyond 10^{0} RMSE started to increase, and after the timestamp, 10^{2}, the BHD has a similar RMSE to harmonic function.

In the case of spiral-shaped data, RMSE for BHD rapidly decreases when time equals 10^{0} after that RMSE of BHD is similar to a harmonic function. HD does not change much in RMSE in these datasets and converges faster. In this dataset, we also observe that for a higher value of time, the BHD will be the same as harmonic function favoring continuous or long-range propagation. Thus, BHD never looses to harmonic function because, with an infinite time stamp, BHD will ultimately become harmonic function inferring that harmonic function is the limiting case for BHD.

7.3.3 Computational Complexity

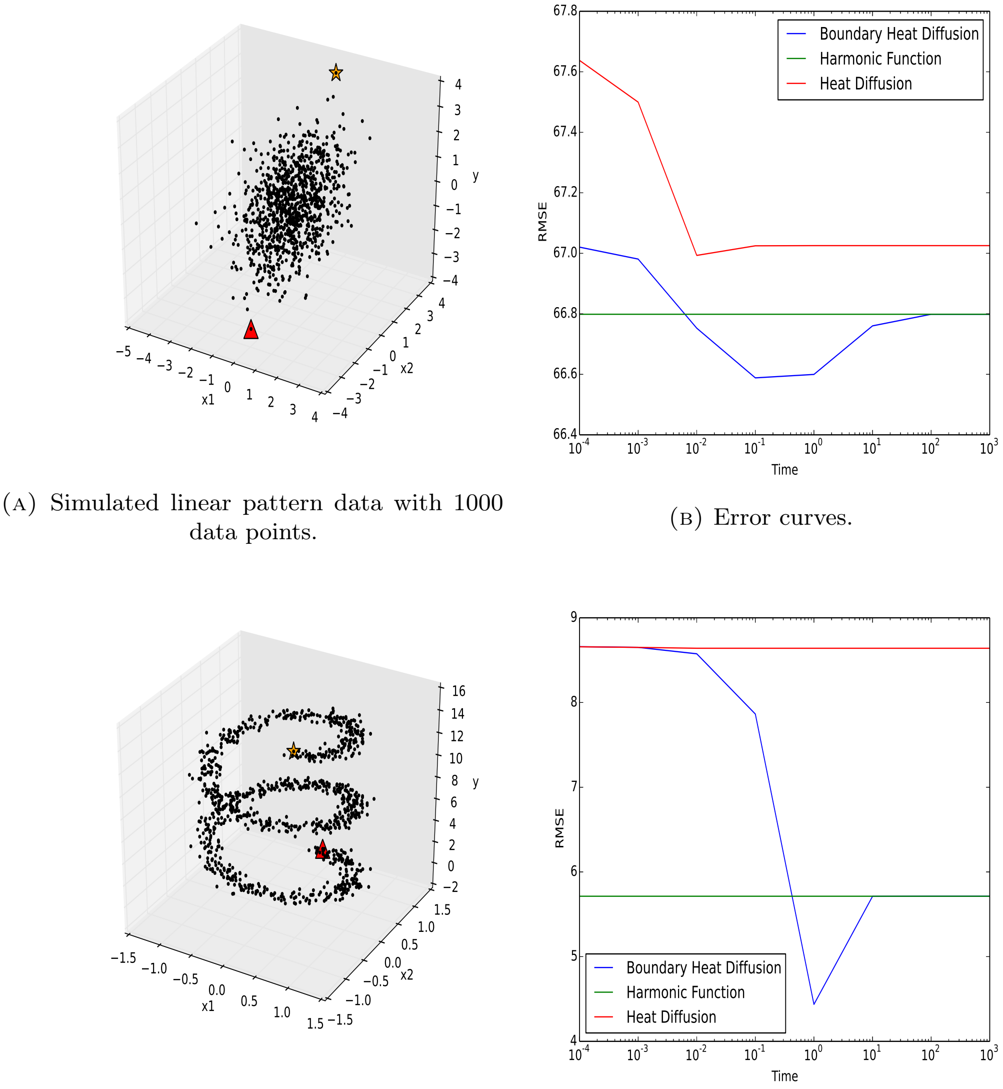
We adopted the discrete approximation of BHD as discussed in Chapter 6 Section 6.3.3. Specifically, after the discrete formalization, the complexity of exponential kernel of BHD will be reduced from $O(n)^3$ to $O(M|E|n)$ where M is the number of iterations, n is the number nodes and $|E|$ is the number of edges in the graph.

7.3.4 Temperature Setting for initial conditions:

For setting the initial temperature in a graph, we took a similar approach as for the node classification problem, as previously discussed in Chapter 6 Section 6.3.4. To propagate heat, we need to set the initial temperature. The initial temperature of the node is the labeled real values of the nodes. These labeled nodes are the training nodes.

Initial Temperature Setting in test set: If the quality of the network is poor, the ideal way to make inferences about the prediction of the node label values in the test set is to use the sample mean. It can be assumed that the nodes without any links are from a population with a mean μ. We can then make an inference of μ by a sample mean Miller et al. [1999], which is used to set the initial temperature for an unlabeled node in a test set.

If the network contains valuable information for making the prediction, this initial estimation should be combined with diffusion along the edges in the network. Our method supports this property, as shown in Equation 7.2, while the harmonic function ignores this initial estimation because of the continuous or global diffusion. Additionally, if the



(A) Simulated linear pattern data with 1000 data points.

(B) Error curves.

(C) Simulated spiral pattern data with 1000 data points.

(D) Error curves.

network is of poor quality, our boundary heat diffusion model has the freedom of choosing a small value of t.

7.4 Experiments

In this experiment, we answer the following questions:

- Q1: *Parameter:* Does the parameter t affect the prediction of heat diffusion with boundary condition?

- Q2: *Accuracy:* How accurate is BHD in comparison to the state of the art label propagation algorithm?

Datsets: We used seven regression datasets from different domains in our experiments. The datasets used in the experiments are shown in Table 7.2.

Datasets	**Domain**	**Number of Features**	**Number of Data Points**
Advertisement	Business	1	200
Boston Housing	Business	13	506
Parkinsons Telemonitoring	Biomedical	16	5875
White Wine Quality	Business	10	1599
Red Wine Quality	Business	10	4898
Airfoil Self-Noise	Physical	5	1503
Bike Sharing	Social	16	17389

Table 7.2: Regression Datasets.

A brief description of the data is as follows:

1. **Advertisement:** It contains the advertising data sales (in thousands of units) for a particular product, including advertising budgets (in thousands of dollars) for TV, radio, and newspaper media. We use TV budgets to predict advertising sales.
2. **Boston Housing:** It contains information collected by the U.S Census Service concerning housing in the area of Boston Mass [Freeman, 1981]. This data has been used extensively throughout the literature to benchmark algorithms. We used this data to predict the housing prices.
3. **Parkinson Telemonitoring:** It is composed of a range of biomedical voice measurements from 42 people with early-stage Parkinson's disease [Tsanas et al., 2009]. From this data, we predicted the UPDRS (Unified Parkinson's Disease Rating Scale) score of each patient.
4. **Red and White Wine Quality:** It is composed of two different wines, i.e., red and white. These two datasets are related to red and white variants of the

Portuguese “Vinho Verde” wine [Cortez et al., 2009]. From this data, we predicted the alcohol level in the wine.

5. **Airfoil Self-Noise:** This data is from NASA[5], which comprises different size airfoils at various wind tunnel speeds and angles of attack. From this data, we predicted the scaled sound pressure level [Brooks et al., 1989].

6. **Bike Sharing:** Bike-sharing [Fanaee-T and Gama, 2014] is an automated bike rental system. A user can rent a bike from a particular location and return to another location by using these systems. These systems are getting popular due to their impact on traffic, health, and environmental issues. The bike-sharing systems generate data that make these systems attractive for artificial intelligence (AI) based research. The bike-sharing systems records the duration of travel, departure, and arrival position. This property turns the bike-sharing system into a virtual sensor network. The data collected from these sensors are useful for identifying mobility in the city. From this data, we predicted the bike rental demand in the Capital Bikeshare program in Washington, D.C [6].

Metric: We chose the root mean square error (RMSE) to evaluate the performance of the algorithm. RMSE is a quadratic scoring rule that measures the average magnitude of the error. It is the square root of the average of squared differences between prediction and actual observation. RMSE score is then:

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{N}(y_i - \hat{y_i})^2} \tag{7.3}$$

where y_i is the observed value and $\hat{y_i}$ is the predicted value and n is the number of observations.

Q1: Parameter In our experiment, we assessed the prediction ability of our approach by varying the parameter t. The parameter t is varied from 0.0001 to 1. We report the results of all regression data from our experiments.

We applied the RBF kernel in the data points to construct the graph from these data. We hid the node labels from 10% to 90% and performed the SSR algorithm to predict the scores of the nodes and recorded the RMSE.

From Figure 7.2, we observed that when the percentage of the labeled node increases, the RMSE score decreases in all the values for the parameter t. It is because a majority of the nodes were already labeled so less heat was required to label the remaining nodes. We

[5] https://www.nasa.gov/

[6] https://www.kaggle.com/c/bike-sharing-demand

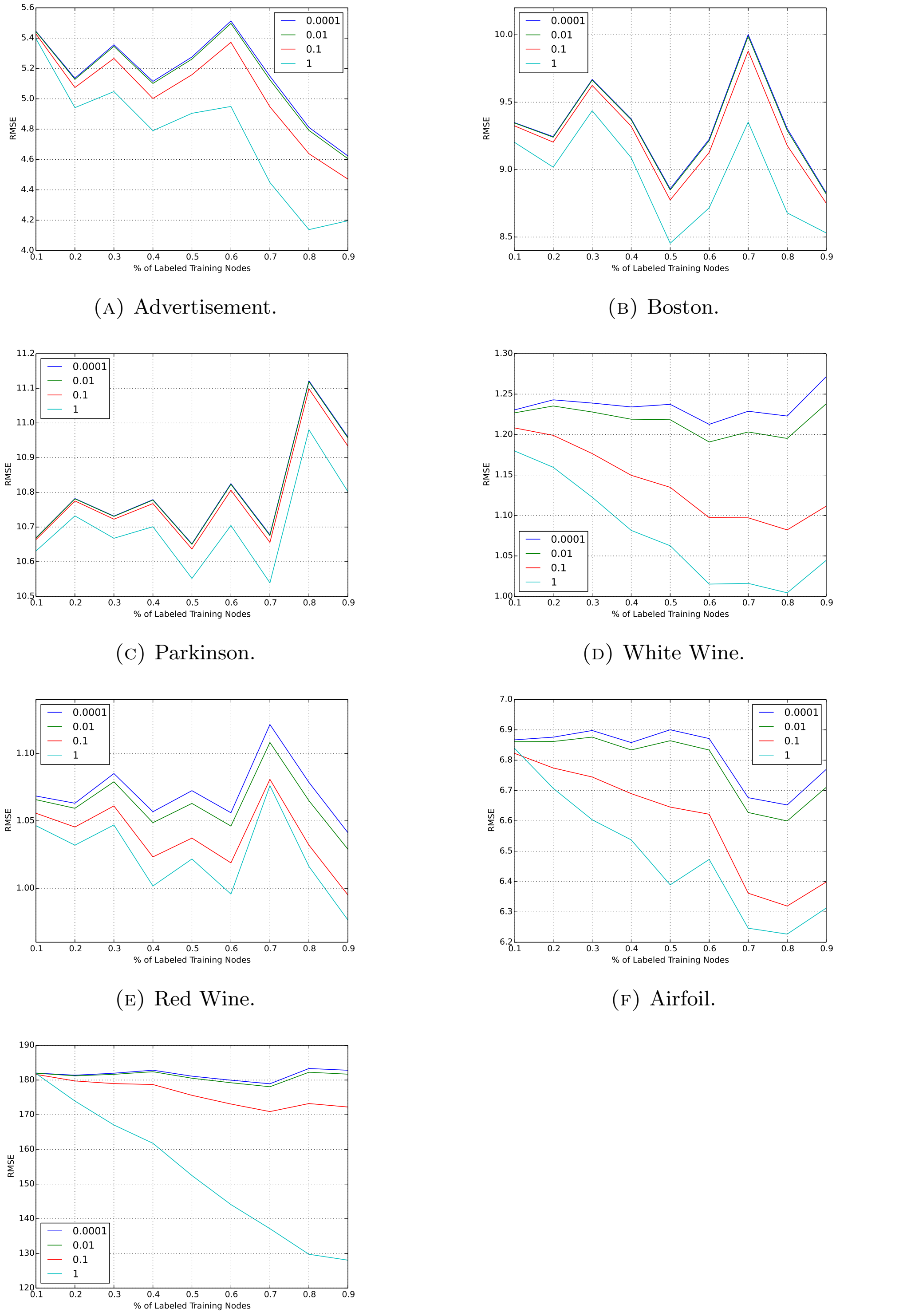
observed that at $t = 1$, the RMSE score is minimum in comparison to other parameters, as shown by the light-blue curve. It means that we need maximum heat to perform diffusion in this kind of network, which in this case, is $t = 1$. This observation also means that we do not need to tune the parameter when using this algorithm. Hence, we use this default value $t = 1$ for all experiments.

Q2: Accuracy We compared our approach with state of the art methods namely: (i) harmonic function (HMN) [Zhu and Ghahramani, 2002], (ii) Local and Global Consistency Method (LGC) [Zhou et al., 2004], (iii) heat diffusion (HD) [Yang et al., 2009] and Support Vector Regression(SVR) using linear kernel [Zhu and Goldberg, 2006].

We split data as follows 10% for training, 90% testing, and apply the algorithms in a 10 Folds cross-validation setting to record the average RMSE score.

From Figure 7.3, we observe that the boundary-based heat diffusion method has performed either at least equaling or exceeding the four state of the art methods. LGC and HD performed poorly in the datasets. BHD has performed significantly better than the HMN, LGC, and HD in predicting the outcome values for Advertisement, Boston Housing, White Wine, Red Wine, and Airfoil Self Noise and Bike Sharing datasets. However, in Parkinson's data, BHD has a marginal improvement over HMN. One of the reasons for this might be the nature of diffusion. In HMN diffusion, the information propagates infinitely favoring long-range interactions, and BHD also has a similar property. For a long-range diffusion, HMN equals to BHD, which is one of the vital properties of BHD. In the Advertisement and Boston Housing datasets, the SVR method performed marginally better than BHD. As this data has a strong linear association with outcome variable and SVR with linear kernel captures this better than BHD so it might have performed better.

We noted that the problem of SSR is a more general problem than SSC. In the latter case, the outcome variable is constrained to have only a finite number of possible values, whereas, in regression, the outcome variable is assumed to be continuous. Hence, LGC and HD algorithms might be more suitable for SSC tasks, whereas HMN and BHD can be applicable for SSR tasks, as shown by our results.



(A) Advertisement.

(B) Boston.

(C) Parkinson.

(D) White Wine.

(E) Red Wine.

(F) Airfoil.

(G) Bike Sharing.

FIGURE 7.2: Impact of parameter t in regression datasets. X-axis is the percentage of labeled data. Y-axis is the RMSE score.

	Advertisement	Boston Housing	Parkinson	White Wine	Red Wine	Airfoil Self Noise	Bike Sharing
HMN	12.78 ± 0.39	22.94 ± 0.38	10.68 ± 0.05	2.64 ± 0.01	2.50 ± 0.17	66.84 ± 2.64	200.48 ± 2.78
LGC	14:03 ± 0.17	23:64 ± 0.20	28:30 ± 0.08	9:62 ± 0.01	9:55 ± 0.01	114:29 ± 0.12	246.40 ± 1.42
HD	14.41 ± 0.13	23.50 ± 0.23	29.21 ± 0.06	9.96 ± 0.01	9.86 ± 0.01	118.22 ± 0.09	251.04 ± 1.28
SVR	5.65 ± 2.14	10.96 ± 3.22	10.85 ± 0:08	1.76 ± 0:62	1.64 ± 0:58	139.69 ± 0.51	185.68 ± 1.61
BHD	5.94 ± 0:18	11.12 ± 0:33	**10.61 ± 0.03**	**1.52 ± 0.01**	**1.59 ± 0.01**	**11.33 ± 0.1**	**179.98 ± 0.95**

Table 7.3: The average RMSE score of 10 fold cross validation from state of the art methods. The error is the standard deviation obtained from the 10-fold cross validation.

Different algorithms shared the same random trials. So, we could perform statistical tests. We applied the paired t-test to find out if there is a significant difference in the 10-fold cross-validation results between BHD and other states of the art methods using a significance (α) level of 0.05. The p-values of the t-test are reported in Table 7.4. We found that there is a significant difference between the predictions performed by BHD with other states of the art methods (p-values <0.05). In Boston housing, white wine, and red wine dataset, we observed the p-value higher than 0.05 between BHD and SVR method. It suggests that there is no significant difference in the ten folds predictions between these methods.

	HMN	HD	LGC	SVR
BHD (Advertisement)	**2.347e-12**	**2.56e-16**	**7.438e-16**	**1.213e-05**
BHD (Boston Housing)	**4.883e-14**	**<2.2e-16**	**3.319e-16**	0.6139
BHD (Parkinson)	**0.0002416**	**<2.2e-16**	**<2.2e-16**	**5.012e-06**
BHD (White Wine)	**4.24e-11**	**<2.2e-16**	**<2.2e-16**	0.2748
BHD (Red Wine)	**8.03e-08**	**<2.2e-16**	**<2.2e-16**	0.78
BHD (Airfoil Self Noise)	**4.116e-13**	**<2.2e-16**	**<2.2e-16**	**0.006906**
BHD (Bike Sharing)	**3.516e-13**	**<2.2e-16**	**<2.2e-16**	**0.009806**

Table 7.4: P-values of the t-test at significance level $\alpha = 0.05$. The bold figures indicate statistical significance.

7.5 Conclusion

We presented the application of boundary heat diffusion in an SSR problem. We applied these algorithms in different domains, such as business, biomedical, physical, and social domain data. The main idea of our method is to assign a node with the initial temperatures. The initialized temperatures act as the boundary and diffuse the heat in the network. The advantages of our algorithm are:

1. Accuracy: it outperforms or equals the state of the art algorithms in label propagation on graph-based semi-supervised regression tasks: [Table 7.3].
2. Parameter Free: It has just one parameter with the effective default value 1 (Figure 7.2).

We employed BHD for the real-valued labels in a graph constructed from manifold data. The method outperformed some of the states of the art methods, but in some data, the support vector regression (SVR) method performed better. One of the reasons for this might be a strong linear association between an outcome variable and predictors, and SVR with linear kernel captures this better than BHD. One way to handle this problem is to construct a better graph from the manifold data. We used the Gaussian kernel, which provided us with a fully connected graph. Of course, better graphs can be constructed if one can define better distance functions, connectivity, and edge weights. It is another critical challenge in a graph-based semi-supervised regression problem.

One of the significant strengths of heat diffusion with boundary condition is computational complexity. We showed that the boundary-based heat diffusion could be computed using a discrete approximation. It will have an advantage in the scalability issues in a large graphs because the complexity is linear for the number of edges in the graph. This property makes the method suitable for bigger graph regression problems. We believe that our proposed approach provides a simple but effective method to estimate the real values for performing semi-supervised graph-based regression.

Finally, in chapter 8, we will present the conclusion of this thesis. We further show the potential future work and new research direction that stems out from this work.

Part III

Conclusion

Chapter 8

Conclusions

8.1 Discussion

This thesis presents an investigation into how diffusion methods can be applied in different tasks of the graph-based predictive analysis. There are different types of diffusion reported in the literature, from information diffusion to physical energy propagation like heat. These propagation models are successfully implemented in graphs to solve the tasks of like node-link prediction and classification.

With Research Question 1.3.1, we asked whether we can predict the links between dissimilar nodes in the graphs. To answer this question, we conducted the experiments for predicting links in a 2 layered graph. We chose to predict links between two different types of nodes because prior studies have already assessed the evaluation of link prediction between similar nodes. We conducted the experiments using social media and biomedical network data. We found that there is a difference in the link prediction abilities in the diffusion-based algorithms in terms of accuracy. In biomedical data, that the links were weighted helped heat diffusion for better link prediction, whereas in unweighted networks, random walk with restart (RWR) performed slightly better. In social network data, we also saw heat diffusion outperforming the other methods, but the difference with other methods was not very high. The main similarity between all diffusion methods is the process of spreading information originating from one or more nodes to the rest of the nodes in the graph. The difference is in the way the spread is modeled. In an information diffusion style propagation, the source nodes never lose information, whereas in heat style diffusion the source node loses information but energy is conserved in the graph. This behavior has an impact on link prediction. Some network data fit better to the information diffusion models, whereas some are more appropriate for the heat-style diffusion model. This opens up further research questions about effectively selecting a model for diffusion.

With research question 1.3.2, we asked whether integrating two different graphs can improve link prediction performance. To answer this question, we provided a novel method to combine the information from the two graphs using the diffusion model and a matrix factorization method. Through using the matrix factorization method, we learned the seed node information in a first layer and diffused this information in a second layer. This approach combined not only 2-different graphs but also provided good link prediction results in comparison to different state-of-the-art methods. The caveat of this method is that the matrix factorization method operates on the hidden components. It is quite hard to estimate those components in advance.

Diffusion based frameworks have been actively used in semi-supervised machine learning tasks in graphs. In the graph with few-labeled nodes, the diffusion allows predicting the labels for unlabeled nodes. Most of the graph-based diffusion method work on the homophilic labeled network where the nearby nodes also share similar labels. In a real-world setting, networks exhibit complex layers, which are called multiplex networks, also known as “networks of networks”. In multiplex networks, the same node is shared among different layers, and ordinary diffusion has a high possibility of node miss-classification. Thus, we investigated through Research Question 1.3.3 the design of a graph-based diffusion method that can adapt to different types of a graphs such as multiplex, homogeneous, and also graphs from manifold data. To provide such a model, we investigated a variant of the heat diffusion model called the heat diffusion with boundary condition (BHD). The time-dependent feature of the BHD model can control long-range (global) or short-range (local) propagation. This property makes it adaptive to different graphs with homophilic, heterophilic, and mixed labels. We further extended the same idea to answer our Research Question 1.3.3.1 about a semi-supervised regression (SSR) task. We employed BHD for the real-valued labels in a graph constructed from manifold data. The method outperformed some of the states-of-the-art methods, but with some data, the support vector regression (SVR) method performed better. One of the reasons for this might be a strong linear association between an outcome variable and predictors, and SVR with linear kernel captures this better than BHD. One way to handle this problem is to construct a better graph from the manifold data. We used the Gaussian kernel, which provided us with a fully connected graph. Of course, better graphs can be constructed if one can define better distance functions, connectivity, and edge weights. It is another critical challenge in graph-based semi-supervised machine learning.

One of the major strengths of the diffusion model in our approach is computational complexity. The diffusion kernel of the heat diffusion is $e^{-\alpha t L}$, where t is the time, and

L is the Laplacian matrix. The direct computation of this kernel has cubic complexity $O(n^3)$. We employ the discrete approximation of this kernel which is given by $f(t) = (I + \frac{-tL}{N})^N f(0)$, where $f(0)$ is the initial preference vector and $f(t)$ is the final computed vector. If we suppose the graph has a M number of edges, the complexity of executing the heat diffusion process is $O(MN)$, which means the number of iteration N times the number of edges M. In a practical setting $\alpha = 1$ and $N = 30$ is sufficient for approximating the heat diffusion equation. It has a big advantage in addressing the scalability issues in a large graph because the complexity $O(MN)$ is linear for the number of edges in the graph. Similarly, the boundary-based heat diffusion method also has two components: heat diffusion and harmonic function. Both of these functions have a linear complexities of $O(MN)$. It means boundary-based heat diffusion scales linearly for large graphs.

8.2 Summary of the Chapters of the Thesis

We have developed a computational model based on diffusion. We evaluated this idea in two relevant use-cases in graph-based machine learning task: (i) link prediction and (ii) node classification. Our idea is inspired by the heat diffusion methods, which is initially studied in Physics. We demonstrated the flexibility and efficacy of our approach to a range of real-world networks and manifold data.

Link Prediction between Two Dissimilar Nodes

A 2 layered graph enabled us to model the two different types of graphs. The first layer is the bipartite graph, and the second layer is the homogeneous graph. Using this modeling method, we tested our approach in two different graph data (i) biomedical graphs of tumor samples and genetic interaction network, (ii) social media graphs of academic entities and web-pages linked by hyperlink network. We applied the heat diffusion method in these 2-layered graphs, and our method showed better performance in comparison to other states of the art methods.

Link Prediction between Two Dissimilar Nodes using Combinational Methods

When we have a 2-layered graph, we have two different types of information. Thus using these, we can perform two different computations and integrate their results. Thus, an application of a 2-layered graph is to combine two different approaches. For this task, we proposed a novel method combining matrix factorization and heat diffusion for predicting links. Using a matrix factorization approach, we can learn the weights of the seed nodes in network layer 1 and diffuse this information in network layer 2. In this way, we can make use of 2 different graphs and two different methods. Our experiments show that our approach achieves better results than diffusion or factorization alone.

Node Label Classification

Another application of heat diffusion is to enable efficient label classification. Real-world networks often exhibit a network of networks structure, which is also called multiplex network. In such a network, where nodes overlap between layers, there is a high possibility of node misclassification by using ordinary label propagation algorithms [Fortunato, 2010]. Thus, for this task, we proposed a novel boundary-based heat diffusion (BHD) algorithm that guarantees a closed-form solution. Experiments on five real-world multiplex network datasets related to the political, social, co-authorship, and biological (genetic interaction) domains demonstrate the benefits of the proposed algorithm, where boundary-based heat diffusion outperforms the top state of the art methods.

Semi Supervised Graph Regression

Although we primarily focused on the node classification task using BHD, we demonstrated the extensibility of BHD to regression problems in manifold data, since most of the graph-based approaches focus on classification problems, and graph-based regression is a less explored case. In this work, we demonstrated our BHD approach in different domains from business, biomedical, physical, and social data, and showed that the boundary-based heat diffusion method could effectively outperform top state of the art approaches.

8.3 Limitations of the Thesis and Directions for Future Research

Graph Quality

The core of graph-based machine learning relies on the data captured in the graphs. A good graph should reflect our prior knowledge about the domain. Its construction is more of an art than science [Zhu, 2005]. It is the practitioner's responsibility to feed a good graph to a graph-based diffusion algorithm in order to expect useful output. From our experiment in link prediction using a web graph of social media data, we observed poor performance by the algorithm. One of the reasons for this is the noise in the graph. Thus it is crucial to have a quality graph for the diffusion algorithm in order to get better performance. In this thesis, we did not focus on the construction of quality graphs, especially when we construct the graph from manifold data. It is still an open research problem in the semi-supervised graph-based machine learning community.

Latent Components

In a 2-layered graph-based diffusion, we used a matrix factorization methods to learn initial weights of seed nodes for diffusion. To learn the weights, we need latent components for matrix factorization. In our current approach we estimated latent components using a grid search in cross-validation in a training set. This approach is time-consuming and computationally expensive. So, a key future work stemming from our research relates to automatically identifying the number of latent components for learning the initial weights for diffusion. One way to achieve this is to use automatic relevance determination (ARD), which is successfully implemented in Bayesian Principal Component Analysis (PCA). The study by [Tan and Févotte, 2012] demonstrated the usability of the approach by the automatic recovery of latent components from real datasets. We believe that the same approach can be applied to complete the bi-adjacency matrix in network layer 1 to diffuse in network layer 2 for the link prediction tasks.

Parameter Sensitivity

We demonstrated that our parameter time t has a special function in label propagation. It can adapt to long-range (global) and short-range (local) diffusion. Based on these properties, our approach showed better performances in predicting labels in different variants of labels. The limitation of the study is that we have a free parameter t, which we estimated from cross-validation mode in the training set. For this, extra computational time is required to find the optimum t. One direction for future research is to save the computational time for identifying the best t. The possible way to solve this is by learning the parameter based on the network structure. For this task, identifying the structural node position of the labeled node is important. If we can examine the local neighborhood around each labeled node, we could only propagate whose neighborhoods which have a similar structure by rotating the graph structure to avoid inter-class edges.

Information Loss

Another caveat of this research is an aggregation of the multiplex graph into a single graph for node classification. During the projection into a single layer graph, we might lose the information about the layers. We believe that preserving this information about layers can improve the label classification in such networks. Different layers mean different dimensions in the network, thus keeping track of the layers to preserve information loss is an essential task. We speculate that the tensor-based models can be useful [Nickel et al., 2011] because tensor can preserve the layered information, but the challenge will be identifying appropriate diffusion parameters for each layer.

Semi Supervised Label Prediction in Knowledge Graphs

Most of the techniques for label prediction in knowledge graphs are using embedding-based methods. The embedding techniques are useful and proven to be highly accurate, but due to their high computational cost and there requirement to label data, these methods can be cumbersome. For such a task, we can use the diffusion-based method, which is linear and give result faster and works with the fewer labeled data. The work by [Torres-Tramón et al., 2019] demonstrated how one could take advantage of entity search in knowledge graphs using diffusion-based methods. One of the immediate future work is to assess our boundary-based heat diffusion approach in various knowledge graphs for label classification.

Bibliography

Lada A Adamic and Eytan Adar. Friends and neighbors on the web. *Social networks*, 25 (3):211–230, 2003.

Lada A Adamic and Natalie Glance. The political blogosphere and the 2004 us election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pages 36–43. ACM, 2005.

Lada A Adamic and Bernardo A Huberman. Power-law distribution of the world wide web. *science*, 287(5461):2115–2115, 2000.

Charu C Aggarwal and Nan Li. On node classification in dynamic content-based networks. In *Proceedings of the 2011 SIAM International Conference on Data Mining*, pages 355–366. SIAM, 2011.

Bronwen L Aken, Premanand Achuthan, Wasiu Akanni, M Ridwan Amode, Friederike Bernsdorff, Jyothish Bhai, Konstantinos Billis, Denise Carvalho-Silva, Carla Cummins, Peter Clapham, et al. Ensembl 2017. *Nucleic acids research*, 45(D1):D635–D642, 2016.

Awad H Al-Mohy and Nicholas J Higham. Computing the action of the matrix exponential, with an application to exponential integrators. *SIAM journal on scientific computing*, 33(2):488–511, 2011.

Réka Albert and Albert-Lászlό Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

Federico Battiston, Vincenzo Nicosia, and Vito Latora. Structural measures for multiplex networks. *Physical Review E*, 89(3):032804, 2014.

Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res.*, 7:2399–2434, December 2006. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1248547.1248632.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

A. Blum, J. Lafferty, M.R. Rwebangira, and R. Reddy. Semi-supervised learning using randomized mincuts. pages 97–104, 2004. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-14344266781&partnerID=40&md5=ad465c7d2faea55c324c3313ef55ad28.

Avrim Blum and Shuchi Chawla. Learning from labeled and unlabeled data using graph mincuts. 2001.

Stefano Boccaletti, Vito Latora, Yamir Moreno, Martin Chavez, and D-U Hwang. Complex networks: Structure and dynamics. *Physics reports*, 424(4-5):175–308, 2006.

A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. Neural information processing systems foundation, 2013.

Simon Bourigault, Cedric Lagnier, Sylvain Lamprier, Ludovic Denoyer, and Patrick Gallinari. Learning social network embeddings for predicting information diffusion. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 393–402. ACM, 2014.

Ulf Brefeld, Thomas Gärtner, Tobias Scheffer, and Stefan Wrobel. Efficient co-regularised least squares regression. In *Proceedings of the 23rd international conference on Machine learning*, pages 137–144. ACM, 2006.

Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

Sergey Brin and Lawrence Page. Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer networks*, 56(18):3825–3833, 2012.

Thomas F Brooks, D Stuart Pope, and Michael A Marcolini. Airfoil self-noise and prediction. 1989.

Arnaud Browet, P-A Absil, and Paul Van Dooren. Community detection for hierarchical image segmentation. In *International Workshop on Combinatorial Image Analysis*, pages 358–371. Springer, 2011.

Camila Buono, Lucila G Alvarez-Zuzek, Pablo A Macri, and Lidia A Braunstein. Epidemics in partially overlapped multiplex networks. *PloS one*, 9(3):e92200, 2014.

Deng Cai, Xiaofei He, Jiawei Han, and Thomas S Huang. Graph regularized nonnegative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1548–1560, 2011.

Daniel E Carlin, Barry Demchak, Dexter Pratt, Eric Sage, and Trey Ideker. Network propagation in the cytoscape cyberinfrastructure. *PLoS computational biology*, 13(10): e1005598, 2017.

Andrew Chatr-Aryamontri, Bobby-Joe Breitkreutz, Rose Oughtred, Lorrie Boucher, Sven Heinicke, Daici Chen, Chris Stark, Ashton Breitkreutz, Nadine Kolas, Lara O'donnell, et al. The biogrid interaction database: 2015 update. *Nucleic acids research*, 43(D1): D470–D478, 2014.

Gaurish Chaudhari, Vashist Avadhanula, and Sunita Sarawagi. A few good predictions: selective node labeling in a social network. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 353–362. ACM, 2014.

Bolun Chen, Fenfen Li, Senbo Chen, Ronglin Hu, and Ling Chen. Link prediction based on non-negative matrix factorization. *PloS one*, 12(8):e0182968, 2017.

Hsinchun Chen, Xin Li, and Zan Huang. Link prediction approach to collaborative filtering. In *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'05)*, pages 141–142. IEEE, 2005.

Pin-Yu Chen and Alfred O Hero. Multilayer spectral graph clustering via convex layer aggregation: Theory and algorithms. *IEEE Transactions on Signal and Information Processing over Networks*, 3(3):553–567, 2017.

Kai-Yang Chiang, Nagarajan Natarajan, Ambuj Tewari, and Inderjit S Dhillon. Exploiting longer cycles for link prediction in signed networks. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1157–1162. ACM, 2011.

Davide Chicco. Ten quick tips for machine learning in computational biology. *BioData mining*, 10(1):35, 2017.

Fan Chung. The heat kernel as the pagerank of a graph. *Proceedings of the National Academy of Sciences*, 104(50):19735–19740, 2007.

Aaron Clauset, Cristopher Moore, and Mark EJ Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98, 2008.

James Coleman, Elihu Katz, and Herbert Menzel. The diffusion of an innovation among physicians. *Sociometry*, 20(4):253–270, 1957.

Corinna Cortes and Mehryar Mohri. On transductive regression. In *Advances in Neural Information Processing Systems*, pages 305–312, 2007.

Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, 2009.

L da F Costa, Francisco A Rodrigues, Gonzalo Travieso, and Paulino Ribeiro Villas Boas. Characterization of complex networks: A survey of measurements. *Advances in physics*, 56(1):167–242, 2007.

Lenore Cowen, Trey Ideker, Benjamin J Raphael, and Roded Sharan. Network propagation: a universal amplifier of genetic associations. *Nature Reviews Genetics*, 18(9):551, 2017.

Emanuele Cozzo, Raquel A Banos, Sandro Meloni, and Yamir Moreno. Contact-based social contagion in multiplex networks. *Physical Review E*, 88(5):050801, 2013.

Fabio Crestani. Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review*, 11(6):453–482, 1997.

William Cukierski, Benjamin Hamner, and Bo Yang. Graph-based features for supervised link prediction. In *The 2011 International Joint Conference on Neural Networks*, pages 1237–1244. IEEE, 2011.

Hially Rodrigues De Sá and Ricardo BC Prudêncio. Supervised link prediction in weighted networks. In *The 2011 international joint conference on neural networks*, pages 2281–2288. IEEE, 2011.

Daryl R DeFord and Scott D Pauls. A new framework for dynamical models on multiplex networks. *Journal of Complex Networks*, 6(3):353–381, 2017.

Sergey N Dorogovtsev and Jose FF Mendes. Evolution of networks. *Advances in physics*, 51(4):1079–1187, 2002.

Lucas Drumond, Steffen Rendle, and Lars Schmidt-Thieme. Predicting rdf triples in incomplete knowledge bases with tensor factorization. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing - SAC '12*, the 27th Annual ACM Symposium, pages 326–331. ACM Press, 2012. ISBN 9781450308571.

Hadi Fanaee-T and Joao Gama. Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, 2(2-3):113–127, 2014.

Simon A Forbes, David Beare, Prasad Gunasekaran, Kenric Leung, Nidhi Bindal, Harry Boutselakis, Minjie Ding, Sally Bamford, Charlotte Cole, Sari Ward, et al. Cosmic: exploring the world's knowledge of somatic mutations in human cancer. *Nucleic acids research*, 43(D1):D805–D811, 2014.

Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3-5):75–174, 2010.

T. Franz, A. Schultz, S. Sizov, and S. Staab. Triplerank: Ranking semantic web data by tensor decomposition. volume 5823, pages 213–228, 2009. ISBN 364204929X.

A Myrick Freeman. Hedonic prices, property values and measuring environmental benefits: a survey of the issues. In *Measurement in public choice*, pages 13–32. Springer, 1981.

Jianxi Gao, Sergey V Buldyrev, Shlomo Havlin, and H Eugene Stanley. Robustness of a network of networks. *Physical Review Letters*, 107(19):195701, 2011.

Dario Garcia-Gasulla, Eduard Ayguadé, Jesús Labarta, and Ulises Cortés. Limitations and alternatives for the evaluation of large-scale link prediction. *arXiv preprint arXiv:1611.00547*, 2016.

Wolfgang Gatterbauer, Stephan Günnemann, Danai Koutra, and Christos Faloutsos. Linearized and single-pass belief propagation. *Proceedings of the VLDB Endowment*, 8 (5):581–592, 2015.

Andrew B Goldberg, Xiaojin Zhu, and Stephen Wright. Dissimilarity in graph-based semi-supervised classification. In *Artificial Intelligence and Statistics*, pages 155–162, 2007.

Sergio Gomez, Albert Diaz-Guilera, Jesus Gomez-Gardenes, Conrad J Perez-Vicente, Yamir Moreno, and Alex Arenas. Diffusion dynamics on multiplex networks. *Physical review letters*, 110(2):028701, 2013.

Carlos A Gomez-Uribe and Neil Hunt. The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)*, 6(4):13, 2016.

Joana P Gonçalves, Alexandre P Francisco, Yves Moreau, and Sara C Madeira. Interactogeneous: disease gene prioritization using heterogeneous networks and full topology scores. *PloS one*, 7(11):e49634, 2012.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016.

Xinlu Guo and Kuniaki Uehara. Graph-based semi-supervised regression and its extensions. *International Journal of Advanced Computer Science and Applications*, 6(6):260–269, 2015.

Lieve Hamers et al. Similarity measures in scientometric research: The jaccard index versus salton's cosine formula. *Information Processing and Management*, 25(3):315–18, 1989.

Kerstin Hartig and Thomas Karbe. Spreading activation simulation with semantic network skeletons. 2017.

Mohammad Al Hasan, Vineet Chaoji, Saeed Salem, and Mohammed Zaki. Link prediction using supervised learning. In *In Proc. of SDM 06 workshop on Link Analysis, Counterterrorism and Security*, 2006.

Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

Petter Holme. Network reachability of real-world contact sequences. *Physical Review E*, 71(4):046119, 2005.

Huiyi Hu, Yves van Gennip, Blake Hunter, Andrea L Bertozzi, and Mason A Porter. Multislice modularity optimization in community detection and image segmentation. In *2012 IEEE 12th International Conference on Data Mining Workshops*, pages 934–936. IEEE, 2012.

Zan Huang, Hsinchun Chen, and Daniel Zeng. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, 22(1):116–142, 2004.

Paul Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bull Soc Vaudoise Sci Nat*, 37:547–579, 1901.

Glen Jeh and Jennifer Widom. Simrank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 538–543. ACM, 2002.

Hisashi Kashima, Tsuyoshi Kato, Yoshihiro Yamanishi, Masashi Sugiyama, and Koji Tsuda. Link propagation: A fast semi-supervised learning algorithm for link prediction. In *Proceedings of the 2009 SIAM international conference on data mining*, pages 1100–1111. SIAM, 2009.

Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1): 39–43, 1953.

Kyle Kloster and David F Gleich. Heat kernel based community detection. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1386–1395. ACM, 2014.

Sebastian Köhler, Sebastian Bauer, Denise Horn, and Peter N Robinson. Walking the interactome for prioritization of candidate disease genes. *The American Journal of Human Genetics*, 82(4):949–958, 2008.

T. G. Kolda, B. W. Bader, and J. P. Kenny. Higher-order web link analysis using multilinear algebra. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 8 pp.–, Nov 2005. doi: 10.1109/ICDM.2005.77.

Risi Imre Kondor and John Lafferty. Diffusion kernels on graphs and other discrete structures. In *Proceedings of the 19th international conference on machine learning*, volume 2002, pages 315–322, 2002.

Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, Aug 2009. ISSN 0018-9162. doi: 10.1109/MC.2009.263.

Gueorgi Kossinets. Effects of missing data in social networks. *Social networks*, 28(3): 247–268, 2006.

Sotiris Kotsiantis, Dimitris Kanellopoulos, Panayiotis Pintelas, et al. Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering*, 30(1):25–36, 2006.

Danai Koutra, Tai-You Ke, U Kang, Duen Horng Polo Chau, Hsing-Kuo Kenneth Pao, and Christos Faloutsos. Unifying guilt-by-association approaches: Theorems and fast algorithms. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 245–260. Springer, 2011.

Hoang-Quynh Le, Mai-Vu Tran, Thanh Hai Dang, Quang-Thuy Ha, and Nigel Collier. Sieve-based coreference resolution enhances semi-supervised learning model for chemical-induced disease relation extraction. *Database*, 2016, 2016.

Elizabeth A Leicht, Petter Holme, and Mark EJ Newman. Vertex similarity in networks. *Physical Review E*, 73(2):026120, 2006.

Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*, pages 641–650. ACM, 2010.

Xin Li and Hsinchun Chen. Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach. *Decision Support Systems*, 54(2): 880–890, 2013.

Yongjin Li and Jagdish C Patra. Genome-wide inferring gene–phenotype relationship by walking on the heterogeneous network. *Bioinformatics*, 26(9):1219–1224, 2010.

David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007.

Ryan N Lichtenwalter and Nitesh V Chawla. Vertex collocation profiles: subgraph counting for link analysis and prediction. In *Proceedings of the 21st international conference on World Wide Web*, pages 1019–1028. ACM, 2012.

Ryan Lichtnwalter and Nitesh V Chawla. Link prediction: fair and effective evaluation. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, pages 376–383. IEEE Computer Society, 2012.

Quan-Hui Liu, Xinyue Xiong, Qian Zhang, and Nicola Perra. Epidemic spreading on time-varying multiplex networks. *Physical Review E*, 98(6):062303, 2018.

Zhengdong Lu, Berkant Savas, Wei Tang, and Inderjit S Dhillon. Supervised link prediction using multiple sources. In *2010 IEEE international conference on data mining*, pages 923–928. IEEE, 2010.

G Lumer. Equations de diffusion générales sur des réseaux infinis. In *Séminaire de Théorie du Potentiel Paris, No. 7*, pages 230–243. Springer, 1984.

Feng Luo, Yunfeng Yang, Jianxin Zhong, Haichun Gao, Latifur Khan, Dorothea K Thompson, and Jizhong Zhou. Constructing gene co-expression networks and predicting functions of unknown genes by random matrix theory. *BMC bioinformatics*, 8(1):299, 2007.

Hao Ma, Haixuan Yang, Irwin King, and Michael R Lyu. Learning latent semantic relations from clickthrough data for query suggestion. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 709–718. ACM, 2008a.

Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. Mining social networks using heat diffusion processes for marketing candidates selection. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 233–242. ACM, 2008b.

Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103, 2010.

Aditya Krishna Menon and Charles Elkan. Link prediction via matrix factorization. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 437–452. Springer, 2011.

Irwin Miller, Marylees Miller, and John E Freund. *John E. Freund's mathematical statistics*. Prentice Hall, 1999.

Cristopher Moore, Xiaoran Yan, Yaojia Zhu, Jean-Baptiste Rouquier, and Terran Lane. Active learning for node classification in assortative and disassortative networks. In

Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 841–849. ACM, 2011.

Peter J Mucha, Thomas Richardson, Kevin Macon, Mason A Porter, and Jukka-Pekka Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *science*, 328(5980):876–878, 2010.

Saket Navlakha and Carl Kingsford. The power of protein interaction networks for associating genes with diseases. *Bioinformatics*, 26(8), 2010.

Mark Newman. *Networks*. Oxford university press, 2018.

Mark EJ Newman. Clustering and preferential attachment in growing networks. *Physical review E*, 64(2):025102, 2001.

Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2): 167–256, 2003.

M. Nickel, L. Rosasco, and T. Poggio. Holographic embeddings of knowledge graphs. In *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, pages 1955–1961. AAAI press, 2016. ISBN 9781577357605.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 809–816, USA, 2011. Omnipress. ISBN 978-1-4503-0619-5. URL http://dl.acm.org/citation.cfm?id=3104482.3104584.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1): 11–33, 2015.

Daniela Nitsch, Joana P Gonçalves, Fabian Ojeda, Bart De Moor, and Yves Moreau. Candidate gene prioritization by network analysis of differential expression using machine learning approaches. *BMC bioinformatics*, 11(1):460, 2010.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: bringing order to the web. 1999.

Judea Pearl. *Reverend Bayes on inference engines: A distributed hierarchical approach*. Cognitive Systems Laboratory, School of Engineering and Applied Science ..., 1982.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.

Guangyuan Piao and John G Breslin. Inferring user interests for passive users on twitter by leveraging followee biographies. In *European Conference on Information Retrieval*, pages 122–133. Springer, 2017.

Guangyuan Piao and John G Breslin. Inferring user interests in microblogging social networks: a survey. *User Modeling and User-Adapted Interaction*, 28(3):277–329, 2018.

Alain Pirotte, Jean-Michel Renders, Marco Saerens, et al. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on Knowledge & Data Engineering*, 19(3):355–369, 2007.

Manisha Pujari and Rushed Kanawati. Link prediction in complex networks by supervised rank aggregation. In *2012 IEEE 24th International Conference on Tools with Artificial Intelligence*, volume 1, pages 782–789. IEEE, 2012.

Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical review E*, 76(3):036106, 2007.

Erzsébet Ravasz, Anna Lisa Somera, Dale A Mongru, Zoltán N Oltvai, and A-L Barabási. Hierarchical organization of modularity in metabolic networks. *science*, 297(5586): 1551–1555, 2002.

Steffen Rendle and Lars Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the third ACM international conference on web search and data mining*, WSDM '10, pages 81–90. ACM, 2010. ISBN 9781605588896.

Mugizi Robert Rwebangira and John Lafferty. Local linear semi-supervised regression. *School of Computer Science Carnegie Mellon University, Pittsburgh, PA*, 15213, 2009.

Gerard Salton and Michael J McGill. *Introduction to modern information retrieval.* mcgraw-hill, 1983.

Purnamrita Sarkar, Deepayan Chakrabarti, and Andrew W Moore. Theoretical justification of popular link prediction heuristics. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

Jerry Scripps, Pang-Ning Tan, Feilong Chen, and Abdol-Hossein Esfahanian. A matrix alignment approach for link prediction. In *2008 19th International Conference on Pattern Recognition*, pages 1–4. IEEE, 2008.

Burr Settles. Active learning literature survey. university of wisconsin. Technical report, Madison Tech. Report, 2010.

Hyunjung Shin, Koji Tsuda, B Schölkopf, A Zien, et al. Prediction of protein function from networks. In *Semi-supervised learning*, pages 361–376. MIT press, 2006.

Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin. A co-regularization approach to semi-supervised learning with multiple views. In *Proceedings of ICML workshop on learning with multiple views*, volume 2005, pages 74–79. Citeseer, 2005.

R. Socher, D. Chen, C.D. Manning, and A.Y. Ng. Reasoning with neural tensor networks for knowledge base completion. Neural information processing systems foundation, 2013.

Thorvald Julius Sørensen. *A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons.* I kommission hos E. Munksgaard, 1948.

Sucheta Soundarajan and John Hopcroft. Using community information to improve the precision of link prediction methods. In *Proceedings of the 21st International Conference on World Wide Web*, pages 607–608. ACM, 2012.

Suvrit Sra and Inderjit S Dhillon. Generalized nonnegative matrix approximations with bregman divergences. In *Advances in neural information processing systems*, pages 283–290, 2006.

Masashi Sugiyama, Tsuyoshi Idé, Shinichi Nakajima, and Jun Sese. Semi-supervised local fisher discriminant analysis for dimensionality reduction. *Machine learning*, 78(1-2):35, 2010.

Lubos Takac and Michal Zabovsky. Data analysis in public social networks. In *International Scientific Conference and International Workshop Present Day Trends of Innovations*, volume 1, 2012.

Partha Pratim Talukdar and Koby Crammer. New regularized algorithms for transductive learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 442–457. Springer, 2009.

Pang-Ning Tan. *Introduction to data mining.* Pearson Education India, 2018.

Vincent YF Tan and Cédric Févotte. Automatic relevance determination in nonnegative matrix factorization with the/spl beta/-divergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1592–1605, 2012.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077. International World Wide Web Conferences Steering Committee, 2015.

Lei Tang and Huan Liu. Relational learning via latent social dimensions. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 817–826. ACM, 2009a.

Lei Tang and Huan Liu. Scalable learning of collective behavior based on sparse social dimensions. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1107–1116. ACM, 2009b.

Mohan Timilsina, Haixuan Yang, and Dietrich Rebholz-Schuhmann. A 2-layered graph based diffusion approach for altmetric analysis. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 463–466. IEEE, 2018.

Mohan Timilsina, Meera Tandan, Mathieu d'Aquin, and Haixuan Yang. Discovering links between side effects and drugs using a diffusion based method. *Scientific reports*, 9(1): 10436, 2019a.

Mohan Timilsina, Haixuan Yang, Ratnesh Sahay, and Dietrich Rebholz-Schuhmann. Predicting links between tumor samples and genes using 2-layered graph based diffusion approach. *BMC Bioinformatics*, 20(1):1–20, 2019b.

Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Fast random walk with restart and its applications. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 613–622. IEEE, 2006.

Wei Tong and Rong Jin. Semi-supervised learning by mixed label propagation. In *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 1*, AAAI'07, pages 651–656. AAAI Press, 2007. ISBN 978-1-57735-323-2. URL http://dl.acm.org/citation.cfm?id=1619645.1619750.

Pablo Torres-Tramón, Mohan Timilsina, and Conor Hayes. A diffusion-based method for entity search. In *2019 IEEE 13th International Conference on Semantic Computing (ICSC)*, pages 16–23. IEEE, 2019.

Amanda L Traud, Peter J Mucha, and Mason A Porter. Social structure of facebook networks. *Physica A: Statistical Mechanics and its Applications*, 391(16):4165–4180, 2012.

T. Trouillon, J. Welbl, S. Riedel, E. Ciaussier, and G. Bouchard. Complex embeddings for simple link prediction. In *33rd International Conference on Machine Learning, ICML 2016*, volume 5, pages 3021–3032. International Machine Learning Society (IMLS), 2016. ISBN 9781510829008.

Athanasios Tsanas, Max A Little, Patrick E McSharry, and Lorraine O Ramig. Accurate telemonitoring of parkinson's disease progression by noninvasive speech tests. *IEEE transactions on Biomedical Engineering*, 57(4):884–893, 2009.

Alexander Tsiatas. Pagerank and diffusion on large graphs. *UCSD Research Exam*, 14, 2009.

Charles F Van Loan and Gene H Golub. *Matrix computations*. Johns Hopkins University Press, 1983.

Oron Vanunu and Roded Sharan. A propagation-based algorithm for inferring gene-disease associations. In *German Conference on Bioinformatics*. Gesellschaft für Informatik e. V., 2008.

Vladimir Naumovich Vapnik and Vlamimir Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.

Christian Von Mering, Lars J Jensen, Berend Snel, Sean D Hooper, Markus Krupp, Mathilde Foglierini, Nelly Jouffre, Martijn A Huynen, and Peer Bork. String: known and predicted protein–protein associations, integrated and transferred across organisms. *Nucleic acids research*, 33(suppl_1):D433–D437, 2005.

Soroush Vosoughi, Deb Roy, and Sinan Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.

Huijuan Wang, Qian Li, Gregorio D'Agostino, Shlomo Havlin, H Eugene Stanley, and Piet Van Mieghem. Effect of the interconnected network structure on the epidemic threshold. *Physical Review E*, 88(2):022801, 2013.

Liang Wang, Ke Hu, and Yi Tang. Robustness of link-prediction algorithm based on similarity and application to biological networks. *Current Bioinformatics*, 9(3):246–252, 2014.

Meng Wang, Xian-Sheng Hua, Yan Song, Li-Rong Dai, and Hong-Jiang Zhang. Semi-supervised kernel regression. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 1130–1135. IEEE, 2006.

Peng Wang, BaoWen Xu, YuRong Wu, and XiaoYu Zhou. Link prediction in social networks: the state-of-the-art. *Science China Information Sciences*, 58(1):1–38, 2015.

Wenjun Wang, Minghu Tang, and Pengfei Jiao. A unified framework for link prediction based on non-negative matrix factorization with coupling multivariate information. *PloS one*, 13(11):e0208185, 2018.

Stanley Wasserman and Katherine Faust. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.

Lilian Weng, Filippo Menczer, and Yong-Yeol Ahn. Virality prediction and community structure in social networks. *Scientific reports*, 3:2522, 2013a.

Lilian Weng, Jacob Ratkiewicz, Nicola Perra, Bruno Gonçalves, Carlos Castillo, Francesco Bonchi, Rossano Schifanella, Filippo Menczer, and Alessandro Flammini. The role of information diffusion in the evolution of social networks. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 356–364. ACM, 2013b.

Yuto Yamaguchi, Christos Faloutsos, and Hiroyuki Kitagawa. Omni-prop: Seamless node classification on arbitrary label correlation. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Yuto Yamaguchi, Christos Faloutsos, and Hiroyuki Kitagawa. Camlp: Confidence-aware modulated label propagation. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 513–521. SIAM, 2016.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. 2014.

Haixuan Yang, Irwin King, and Michael R Lyu. Diffusionrank: a possible penicillin for web spamming. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 431–438. ACM, 2007.

Haixuan Yang, Michael R Lyu, and Irwin King. A volume-based heat-diffusion classifier. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2): 417–430, 2009.

Yang Yang, Ryan N Lichtenwalter, and Nitesh V Chawla. Evaluating link prediction methods. *Knowledge and Information Systems*, 45(3):751–782, 2015.

Daoqiang Zhang, Zhi-Hua Zhou, and Songcan Chen. Semi-supervised dimensionality reduction. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, pages 629–634. SIAM, 2007a.

Jiawei Zhang, Philip S Yu, and Zhi-Hua Zhou. Meta-path based multi-network collective link prediction. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1286–1295. ACM, 2014.

Peng Zhang, Xiang Wang, Futian Wang, An Zeng, and Jinghua Xiao. Measuring the robustness of link prediction algorithms under noisy environment. *Scientific reports*, 6, 2016.

Sheng Zhang, Weihong Wang, James Ford, and Fillia Makedon. Learning from incomplete ratings using non-negative matrix factorization. In *Proceedings of the 2006 SIAM International Conference on Data Mining*, pages 549–553. SIAM, 2006.

Yi-Cheng Zhang, Marcel Blattner, and Yi-Kuo Yu. Heat conduction process on community networks as a recommendation model. *Physical review letters*, 99(15):154301, 2007b.

Jianmei Zhao, Xuecang Li, Qianlan Yao, Meng Li, Jian Zhang, Bo Ai, Wei Liu, Qiuyu Wang, Chenchen Feng, Yuejuan Liu, et al. Rwcfusion: identifying phenotype-specific cancer driver gene fusions based on fusion pair random walk scoring method. *Oncotarget*, 7(38):61054, 2016.

Dengyong Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Advances in neural information processing systems*, pages 321–328, 2004.

Tao Zhou, Linyuan Lü, and Yi-Cheng Zhang. Predicting missing links via local information. *The European Physical Journal B*, 71(4):623–630, 2009.

Wen Zhou and Yifan Jia. Predicting links based on knowledge dissemination in complex network. *Physica A: Statistical Mechanics and its Applications*, 471:561–568, 2017.

Xianghong Zhou, Ming-Chih J Kao, and Wing Hung Wong. Transitive functional annotation by shortest-path analysis of gene expression data. *Proceedings of the National Academy of Sciences*, 99(20):12783–12788, 2002.

Zhi-Hua Zhou and Ming Li. Semi-supervised regression with co-training. In *IJCAI*, volume 5, pages 908–913, 2005.

Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, Citeseer, 2002.

Xiaojin Zhu and Andrew Goldberg. Semi-supervised regression with order preferences. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2006.

Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.

Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, pages 912–919, 2003.

Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005.

Marinka Zitnik and Jure Leskovec. Predicting multicellular function through multi-layer tissue networks. *Bioinformatics*, 33(14):i190–i198, 2017.