CROSS-DOMAIN LEARNING FOR NETWORK REPRESENTATIONS

Shan Xue

Faculty of Engineering and Information Technology

University of Technology Sydney

A thesis submitted for the Degree of

Doctor of Philosophy

February 2019

CERTIFICATE OF AUTHORSHIP/ORIGINALITY

This thesis is the result of a research candidate conducted jointly with Shanghai University as part of a collaborative Doctoral degree. I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of the requirements for a degree except as part of the collaborative doctoral degree and/or fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This research is supported by the Australian Government Research Training Program.

Production Note:
Signature removed prior to publication.

Shan Xue

February 2019

Acknowledgements

On having completed this thesis, I would like to express my sincere gratitude to my principal supervisor Distinguished Professor Jie Lu. Without her patience and encouragement, I would not have been able to complete this PhD program. Professor Lu led me in a new academic research direction, giving me as much guidance as possible and trusted me to pursue my own research interests. She taught me how to think and study independently, and how to solve difficult scientific problems in a flexible and rigorous way. She instilled me with confidence and motivated me when I felt lost or afraid about the future. Her wisdom, decisiveness and sharp insights inspired me to keep going regardless of any difficulty I encountered. The help that Professor Lu gave me will continue to benefit me throughout my life.

I would also to express my thankfulness to my co-supervisor A./Professor Guangquan Zhang, who has been like a father to me throughout the duration of my four-year PhD candidature. I always knew that, in him, I had someone to whom I could talk about any of my concerns and he would always lead me to a solution.

I would like to express my thankfulness to every member of the Decision Systems & e-Service Intelligence Lab (DeSI) in the Centre for Artificial Intelligence (CAI). It was a wonderful experience to spend four years with these dedicated researchers. I will also have fond memories of UTS and Sydney. I especially thank Dr. Junyu

Xuan who helped me greatly during my PhD candidature, Dr. Vahid Behbood who strictly helped me solve several problems related to my research, Dr. Hua Zuo who was like an older sister to me and Dr. Qian Zhang who studied with me throughout the four years. I would also thank Dr. Fan Dong, Yiliao Song and Feng Liu who helped me and shared my joys and sadness.

I am incredibly grateful to my parents for their generosity and encouragement. This thesis would never have been completed without their constant support and understanding. I am also thankful to my friends who accompanied me and were by my side, listening to me and supporting me through the arduous journey of four years.

Abstract

Network representation aims to learn a latent feature space so that artificial intelligent algorithms can be applied based on the latent features. The set of latent features is obtained from the information hidden behind network structures, which is learned to provide knowledge for traditional machine learning tasks, such as node classification, recommendation and data visualization. Networks, which are a kind of structured data, limit the representation performance in the structure searching process. Therefore, a good node sampling strategy plays an important role in network representation. Recent research has driven significant progress in network representation by employing random walk as the network sampling strategy. However, real-world large-scale information networks naturally have structural sparsity. The existing approaches to random walk-based network representations are in the domain-specific view to represent the nodes in a vector format, which cannot guarantee a good representation by one network knowledge learning.

To address these gaps, this research proposes a framework and develops two algorithms to adapt useful information across relational large-scale information networks and allows the information of the network structure to be transferred from one network to another network to improve the performance of network representation. First, a novel framework of transferring structures across large-scale information

networks (FTLSIN) is proposed. FTLSIN consists of a two-layer random walk to measure the relations between two networks and predict the links across them. Second, a cross-domain network representation algorithm (CDNR) is proposed to demonstrate the knowledge which transfers across domains. CDNR learns the structural information from dense networks to sparse networks and further defines the two-layer random walk in unsupervised feature learning with a cross-domain node mapping procedure and a cross-domain walk mapping procedure. Thirdly, a cross-domain similarity learning algorithm (CDSL) is proposed to acquire the most relevant knowledge from the external network. CDSL is nested in the biased random walk-based node sampling and targets the minimum cost of searching the neighborhood in the biased random walk that considers the first-order and second-order walking; and the neighborhood is described by a dual centrality indicator which consists of closeness centrality and betweenness centrality. The developed framework and the two algorithms are very innovative and significantly contribute to both fields of transfer learning and network representation.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Information networks (Newman, 2010) is a kind of structure-based data, which employ network topology structures to save a part of information. Large-scale information network ranges the size from hundreds of nodes to millions and billions of nodes (Tang et al., 2015). The large volume of nodes make complex connections over the network and contains complex data structures than normal information networks. To fully analyze such kind of information networks is a quite challenging problem especially in machine learning domain.

The large-scale information network datasets, which are extracted from real-world complex systems, are rich in domain-specific information. New techniques are needed to learn the knowledge from network structures, and the aim of network research in the big data age is to learn this knowledge in an intelligent way. Networks categorize the components (nodes), the direct interactions (links), and the connection strengths (weights of the links) to explain the properties of systems in the physical

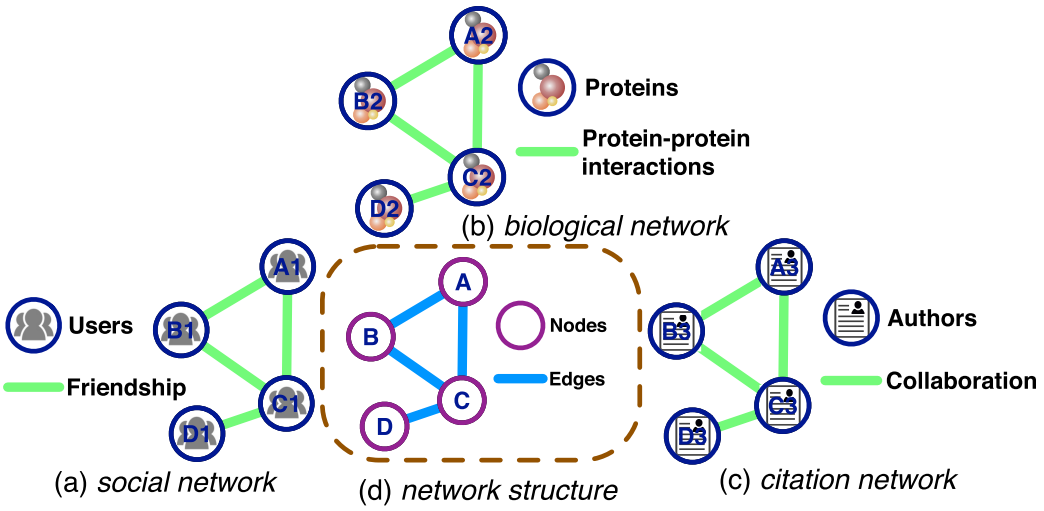


Figure 1.1 An illustration of different networks with the same substructure

world. Therefore, an information network is a type of network in which the information users and suppliers are described as the nodes and the information transfers are represented as the links (Tang et al., 2015). These structures generally exist in real-world systems and are denoted as a type of structure-based data. Meanwhile, the network behaviors are denoted as information and are usually encoded in the network structures, while conversely, the network structures limit or enhance the network behaviors. The best-known information networks are social networks (Chaker et al., 2017; Choi et al., 2012), biological networks (Jancura and Marchiori, 2010) and citation networks (Lu et al., 2018). Social networks, such as online communities, leverage networks to discover deep friendship information among users. Biological networks model the cell by representing different types of information in the shape of a network. For example, the **P**rotein-**P**rotein **I**nteractions (PPI) network is a type of biological network which represents the physical relationships between proteins and reflects information about almost every process in a cell. Citation networks represent information about the collaborations of co-authors. Figure 1.1 illustrates different

information networks that share the same substructures: subset networks of (a) social network formed by users {A1,B1,C1,D1} and user friendships; (b) biological network formed by proteins {A2,B2,C2,D2} and protein-protein interactions; and (c) citation network formed by authors {A3,B3,C3,D3} and collaborations. Figure 1.1 (d) extracts the network structure of (a), (b) and (c) formed by nodes {A,B,C,D} and their links. It is worth noting that although the nature of the nodes and the links differ in various systems, as shown in Figure 1.1 (a)-(c), the sub-structures may be exactly the same, as illustrated in Figure 1.1 (d). It is therefore also a challenging task to acquire useful information that may be hiding behind these different types of networks.

Network representation, also known as network embedding, allows analyzing the network structure and mining the information behind the structure in a machine learning perspective (Huang et al., 2017; Wang et al., 2017). By generating a latent representation space in relatively low dimensions from the interactions in high dimensions, network representation inputs a structured data of graph and outputs the embeddings of the graph in a specific dimensional space. It guarantees the correspondence between community structure in the input graph and its embeddings. Therefore, the main advantage of network representation is that the learned representations encode community structure, so it can be easily exploited by simple and standard classifiers (Bhagat et al., 2011).

Good network representation in a domain-specific network topology structure will ensure that the subsequent learning algorithm is appropriate. Network representation learns a latent feature/vector space by learning the information from the network structures formed by nodes, links and weights (Tang and Liu, 2009). It is particularly useful in non-decomposable systems because it allows the network

structure to be analyzed and the information behind the structure to be mined with the guarantees of the correspondence between community structure in the input network and its representations.

Representation learning in the machine learning domain involves complex but highly structured dependencies; it requires smoothness in learning functions and faces sparsity in latent feature learning (Bengio et al., 2013). The previously used per-node partition function (Bandyopadhyay and Gamarnik, 2008) is expensive to compute, especially for large information networks. To overcome this disadvantage for network representations, a series of sampling strategies have been proposed (Kurant et al., 2011; Lelis et al., 2013) to analyze the statistics within local structures, e.g., communities and sub-networks. These approaches are different from traditional representation learning (Ding et al., 2015; Paalanen et al., 2006; Zhu et al., 2015). The latent feature learning of the network representation captures neighborhood similarity and community membership in topologies (Pan et al., 2016; Tu et al., 2016; Yang et al., 2015). To handle the sparsity of network representations and to improve sampling performance, researchers have been motivated over the last decade to use a stream of short random walks as a basic tool for extracting information from real-world large-scale information networks. The random walk (Noh and Rieger, 2004) is a type of similarity measurement for a variety of problems in community detection (Lai et al., 2010; Liu and Lü, 2010; Noh and Rieger, 2004), which computes the local community structure information sub-linear to the size of the input network (Breitkreutz et al., 2008; Leskovec et al., 2005; Leskovec and Mcauley, 2012; Perozzi et al., 2014; Yang and Leskovec, 2015).

In this research, we focus on the challenge of a lack of connections in a network representation task. The existing network representation methods do not perform

well in domain-specific network representation when facing structural sparsity that is caused by a lack of connections. Furthermore, the performance of domain-specific network representation methods that are developed for one network decreases sharply when they are applied to another network. If we develop a framework that successfully combines the advantages of network representation and domain adaptation in transfer learning, structure transfer will greatly benefit real-world large-scale information network representations. Therefore, we develop a novel **F**ramework of **T**ransferring Structures across **L**arge-**S**cale **I**nformation **N**etwork (FTLSIN) and two algorithms of **C**ross-**D**omain **N**etwork **R**epresentation (CDNR) and **C**ross-**D**omain **S**imilarity **L**earning (CDSL).

1.2 Research Questions and Objectives

This research aims to answer the following four research questions.

1.2.1 Research Questions

QUESTION 1. *(RQ1) How to relate two independent networks that belong to different domains in a transfer learning setting?*

The model, developed for source domain network representation, usually has poor performance if directly applied to the target domain network representation task because the target domain network is not as dense in structural information as the source domain network. Two different networks abstracted from different systems which belong to different domains also differ in distributions or feature spaces. To leverage knowledge from the source domain network to improve the performance of

target domain network representation, a model should be developed for cross-domain knowledge adaptation use which has the function of learning from external data. Therefore, the proposed framework is beyond the existing domain-specific network representation approaches. It is crucial to find a way to relate two independent networks so that knowledge transfer can be undertaken between the two domains.

QUESTION 2. *(RQ2) How to match the unbalanced scales of nodes that belong to different networks in the source domain and in the target domain?*

The sample scales of the source domain and the target domain differ in the following two aspects: the source domain has dense information in structures and the target domain has sparse information in structures; and the node scale of the source domain network is much larger than the node scale of the target domain network. The representation made on network structures is dramatically influenced by structural density. The network representation made on sparse structures might be enriched by employing external related network structures. However, finding the relationships, such as by link prediction, across two networks consumes time and resources and also increases the cost of searching for the target network representation. Direct link prediction on nodes would cause high computational complexity. In addressing the scales of nodes, it is important to match the node scale of the source domain network to the same level of the target domain network, so that computational complexity is kept at the same level as the domain-specific approach which is related to the smaller node scale of the target domain.

QUESTION 3. *(RQ3) What kind of knowledge should be transferred from the source domain network to the target domain network?*

This research focuses on a simple network structure where the network is originally unweighted, the nodes in the network are without attributes and the links in the network are undirected. Therefore, the network representation only depends on such structures. In this case, what kind of knowledge should be transferred from the source domain to the target domain is crucial for cross-domain network representations. Per-node partitioning is expensive for large-scale real-world networks. Therefore, the type of transferred knowledge should exclude the original network structures to avoid the high cost of computations. In other words, network structures need to be mapped into another format while keeping as much information as possible from the original structure.

QUESTION 4. *(RQ4) How to select the most relevant network in the source domain to adapt knowledge to the network in the target domain?*

In the real world, the number of related networks that are selected as candidates for the source domain is very large. To improve the representation performance of the target domain network, the most relevant and helpful source domain network is required to guarantee positive knowledge transfer. Previous works on transfer learning usually do not discuss the selection of the source domain; instead, the source domain selection usually depends on human experience to select the source domain or test the performance on a small sample set. However, in the large-scale information network representation domain, human experience may lead to huge variations in real-world facts and a small sample set is not effective for network-structure data.

Therefore, it is essential to calculate the similarity between the network in the source domain and the network in the target domain.

1.2.2 Research Objectives

This research aims to achieve the following objectives, which are expected to answer the above research questions:

OBJECTIVE 1. *To predict links between nodes in the cross-domain networks by designing FTLSIN.*

This objective corresponds to RQ1. Since the domain-specific random walk-based network representations in this thesis are used as the baseline, the divergence of the source domain and the target domain is discussed from the perspective of random walk-based network representations in cross-domain learning for network representations. To this end, the discrepancies between domain-specific network representations and cross-domain network representations are explored from two aspects. First, the local network structures of the neighborhood in the two domains are compared and the node degrees are measured to determine whether the biased random walks are the same in the two domains. Secondly, if the biased random walks are different across domains, FTLSIN is designed to achieve cross-domain link prediction and to construct relationships between the two domains. FTLSIN includes a two-layer random walk, which provides cross-domain knowledge transfer paths along the predicted links. The top layer and bottom layer load the source domain network and the target domain network, respectively. Knowledge is transferred between layers and the cross-domain network representation finally works on the top layer supported by the biased random walk.

OBJECTIVE 2. *To develop a cross-domain node mapping procedure in the CDNR algorithm to cluster a super node in the source domain network corresponding to the node in the target domain network.*

This objective corresponds to RQ2. To balance the scales of the two networks in the source domain and in the target domain, the source domain network is reshaped into a smaller size but it keeps the same level of structural information. To achieve this goal, the concept of a super node is employed to cluster a group of nodes that share the same property in the source network and which corresponds to the node property in the target network. In this work, node property refers to node importance. This can also be the node degree as the proposed dual centrality. Therefore, the cross-domain node mapping procedure works between the nodes of the target network and the super nodes of the source network; then, the predicted links are constructed based on cross-domain node mapping.

OBJECTIVE 3. *To develop a cross-domain walk mapping procedure in the CDNR algorithm to map the source domain network structure information of random walks to the target domain network.*

This objective corresponds to RQ3. The super links are studied for this objective to describe the structures of the super graph. This cross-domain walk mapping procedure is developed as a key procedure in CDNR and defines the transferred knowledge in two aspects: knowledge objective and knowledge format/value. In FTLSIN, the bottom layer that loads the source domain network pre-learns a set of random walks. The source domain network is reshaped by the super graph so that the random walks are a sequence of super nodes. Then, the cross-domain walk mapping generates weights for the corresponding links in target network. Therefore, the super

graph-shaped random walks in the bottom layer are the transferred knowledge and the weights are the format and the value of the knowledge. These weights guide the random walks in the top layer for the target network that contains the structure information of the source domain network.

OBJECTIVE 4. *To develop a CDSL algorithm to select the source domain network based on the target domain network representations.*

This objective corresponds to RQ4. To address the fact that the relations of two domains is not clear, this study develop the CDSL algorithm to particularly measure the similarity between networks across domains which further supports a source domain selection task. CDSL calculates the domain similarities based on the network patterns of node importance where the information flow is fully observed in this research by node centrality and node power. CDSL learns from the dual centrality which reflects the network pattern. The learning of the dual centrality parameter corresponds to the similarity of the paired networks across domains.

1.3 Research Contributions

The main contributions of this research are summarized as follows:

- A novel FTLSIN is proposed to elaborate on how external network information helps the target domain network representation in unsupervised feature learning. FTLSIN investigates scalable networks and works on a two-layer random walk for both the source domain and the target domain. The knowledge transfer in FTLSIN is theoretically developed by link predictions across

two-layer random walks so that the relationships between two independent networks across domains are constructed.

- Based on FTLSIN, a novel CDNR algorithm is developed. Compared to the domain-specific algorithms, CDNR is more effective for sparse network structures, and the challenge from unbalanced scales of nodes in two networks is overcome by a cross-domain node mapping procedure. The concepts of super node and super graph are introduced for source domain node clustering in this procedure which learns based on the target domain node samples. The experiment results show that CDNR can improve network representation accuracy without developing a new representation model for the target domain.

- CDNR defines the transfer objective, i.e., the knowledge of random walks from the source domain. The knowledge of source domain random walks is specifically sampled from the super graph structure and it is saved as the weights on the super links. This part of the weights is transferred along the predicted links across domains to guide the target domain random walks. This procedure theoretically proves the advances of the sampling strategy in CDNR.

- CDSL addresses the source domain selections based on the network patterns. The novel work of CDSL fully considers the node importance in CDNR and embeds two node importance indicators describing node centrality and node control power on information flows to form a dual centrality indicator, which is parametrized with the domain similarity and learns with the biased random walks to minimize the searching costs within the neighborhoods.

1.4 Research Significance

The theoretical and practical significance of this research is summarized as follows:

Theoretical significance: This study investigates the natural properties of structural sparsity in large-scale information networks and proposes to represent the network structure information by a set of latent features from a cross-domain adaptation view. This work theoretically proves that the knowledge sharing framework of FTLSIN across related but independent domains improves the network representations of the target sparse network. To fulfill the task of cross-domain learning for network representation, the two algorithms of CDNR and CDSL are further proposed to define the objectives and the paths of knowledge transfer, which inherit the advantage of transfer learning in information supplemented by learning to learn. The cross-domain node mapping procedure and the cross-domain walk mapping procedure in the CDNR algorithm theoretically define knowledge transfer paths; and the CDSL algorithm provides the theorem of domain selection based on the target domain network representation task. The framework and the algorithms developed in this research work in the two-layer random walk that enables effective knowledge transfer.

Practical significance: This study starts from the practical research challenges of network-structure data sampling and data sparsity, which are commonly found in a new real-world system. These challenges limit a good network representation which helps artificial intelligent learning task. We propose the solution of cross-domain learning for network representation which will benefit society given the important role transfer learning plays in daily life. The findings help resolve the real-world cold start problem, especially in new areas where the collected data are scarce while

the relational area has a large amount of data that can offer referable information. Newly founded corporations and organizations can utilize data from related systems that abstract as an information network. The data biases of scales, distributions and features in a new system and the related systems are modified by this research.

1.5 Thesis Structure

The logical structure of this thesis (the chapters and the corresponding research questions and research objectives) and the relationship between the chapters are shown in Figure 1.2. The main content of each chapter is summarized as follows:

CHAPTER 2. This chapter studies the literature and discovers the network representation limitations in the domain-specific view, thereby revealing the current research gap. In this chapter, random walk-based network representation for large-scale network information mining is introduced, after which a categorization of the existing algorithms based on network pattern mining and structure sampling is analyzed. A detailed literature review of network pattern mining in node importance to network representations are given. Lastly, the transfer learning approaches which support the cross-domain network representations in the network knowledge transfer aspect are summarized. The limitations of the reviewed approaches and algorithms are discussed in this chapter, which inspires the following chapters and solutions.

CHAPTER 3. The existing domain-specific methods of mining information networks in machine learning represent the nodes of an information network in a vector format. However, a real-world large-scale information network cannot undertake network representation well using only one network. When network structure information is transferred from one network to another, the performance of

network representation decreases sharply. To address this issue, this chapter proposes a novel framework of FTLSIN to transfer useful information across relational large-scale information networks. FTLSIN consists of a two-layer random walk to measure the relations between two networks and predict the links across them. This chapter constitutes the theoretical foundation of the proposed algorithms, and it addresses RQ1 to achieve Objective 1.

CHAPTER 4. To overcome the recent research limitations in the domain-specific network sampling strategy of structural sparsity, this chapter describes FTLSIN and proposes a CDNR algorithm. CDNR works on the two-layer random walks with unsupervised feature learning. The bottom layer prepares the knowledge from the random walk learning of the networks in the source domain. The top layer completes the random walk-based network representation for the target network using the prepared knowledge transferred from the bottom layer. Knowledge transfer works on a cross-domain node mapping procedure and a cross-domain walk mapping procedure. The cross-domain node mapping procedure addresses the issue of unbalanced node scales of networks in different domains (addresses RQ2 to achieve Objective 2), and the cross-domain walk mapping procedure defines the transfer of knowledge from the source domain to the target domain (addresses RQ3 to achieve Objective 3). The work in this chapter accomplishes structural information transfer from a dense network to a sparse network.

CHAPTER 5. The success of adapting structural knowledge from the external network in the source domain for the target domain network discussed in Chapter 3 and in Chapter 4 leads to the important question posed in RQ4. From the numerous network candidates in the source domain, Chapter 5 proposes a CDSL algorithm to acquire the most relevant knowledge from the external network. CDSL is nested in

biased random walk-based node sampling which is also the knowledge that stands for the network structural information for network representations. CDSL targets the minimum cost of searching the neighborhood which considers first- and second-order walking; and network pattern analysis focusing on the neighborhood is described by a dual centrality indicator which consists of closeness centrality and betweenness centrality. This chapter achieves Objective 4.

CHAPTER 6. This chapter summarizes the findings of this thesis and points to directions for future work.

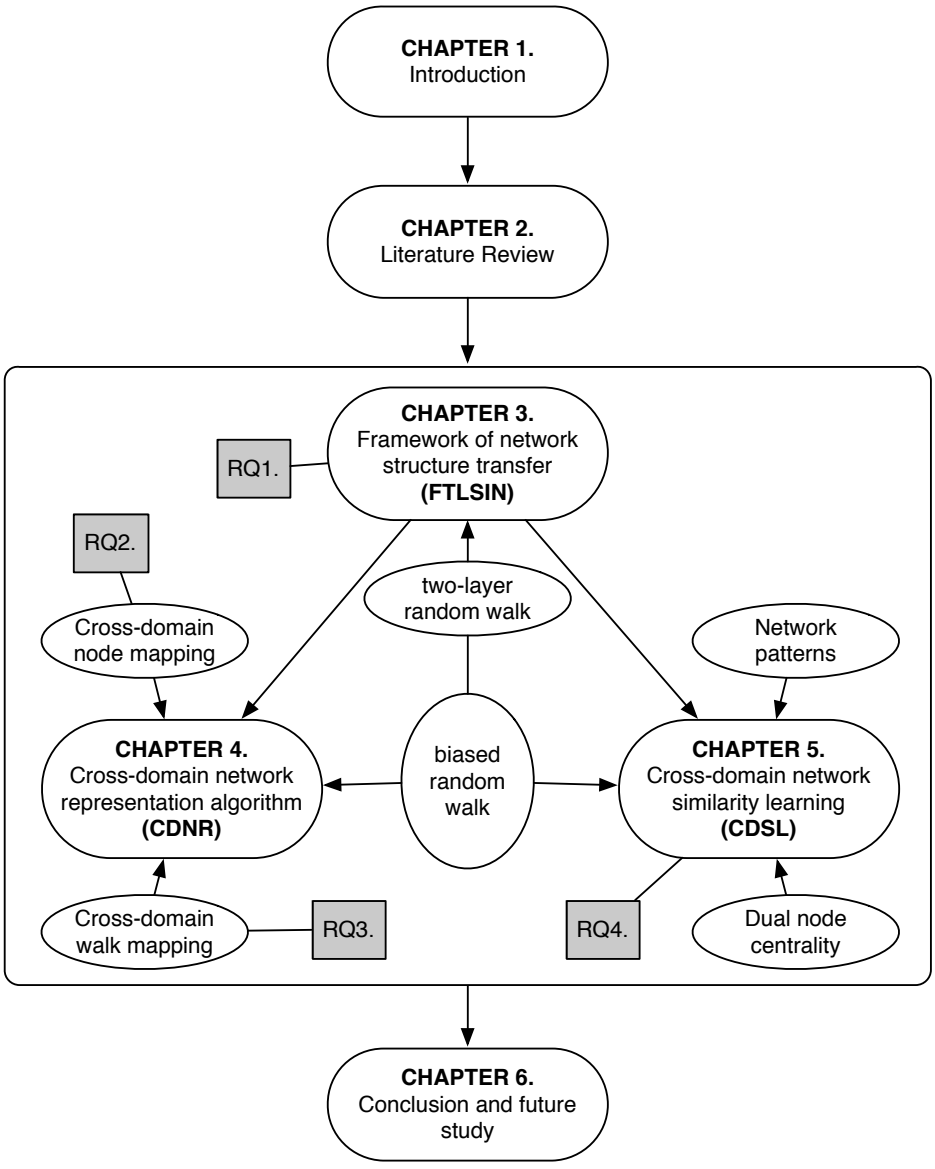


Figure 1.2 Thesis structure and relationship between chapters

1.6 Publications Related to this Thesis

Below is a list of the refereed international journal and conference papers during my PhD research that have been published or currently under review:

Published:

1. **Xue, S.**, Lu, J., Zhang, G., and Xiong, L. (2018). A framework of transferring structures across large-scale information networks. In *Proceedings of the 2018 International Joint Conference on Neural Networks*, pages 1-6, IEEE. (ERA Rank A, Outstanding student paper award)

2. **Xue, S.**, Lu, J., Wu, J., Zhang, G., and Xiong, L. (2016). Multi-instance graphical transfer clustering for traffic data learning. In *Proceedings of the 2016 International Joint Conference on Neural Networks*, pages 4390-4395. IEEE. (ERA Rank A)

3. **Xue, S.**, Lu, J., Zhang, G., and Xiong, L. (2015). SEIR immune strategy for instance weighted Naïve Bayes classification. In *Proceedings of the 22nd International Conference on Neural Information Processing*, pages 283-292. Springer. (ERA Rank A)

4. **Xue, S.**, Lu, J., Zhang, G., and Xiong, L. (2015). Heterogeneous feature space based task selection machine for unsupervised transfer learning. In *Proceedings of the 10th International Conference on Intelligence Systems and Knowledge Engineering*, pages 46-51. IEEE. (ERA Rank B)

5. Zhao, L., Xiong, L., and **Xue, S.*** (2016). Global recursive based node importance evaluation. In *Proceedings of the 12th International Conference*

on Advanced Data Mining and Applications, pages 738-750. Springer. (ERA Rank B)

6. Xiong, L., **Xue, S.***, Yang, S., and Han, C. (2015). Multi-source macro data process based on the idea of sample=overall in big data: An applicability study on influence factors to smart city. In *Proceedings of the 2015 International Conference on Logistics, Informatics and Service Science*, pages 1-6. IEEE.

7. **Xue, S.**, Xiong, L., Zhao, L., and Wu, J. (2017). Graph-theoretic node importance mining in world city networks: Methods and applications. *Information Discovery and Delivery*, 45(2): 57-65. (ERA Rank A)

8. **Xue, S.**, Xiong, L., Yang, S., and Zhao, L. (2016). A self-adaptive multi-view framework for multi- source information service in cloud ITS. *Journal of Ambient Intelligence and Humanized Computing*, 7(2): 205-220.

9. Liu, Z., **Xue, S.***, Zhang, L., Pu, J., and Wang, H. (2017). An improved kernel minimum square error classification algorithm based on $L_{2,1}$-norm regularization. *IEEE Access*, 5: 14133-14140.

10. Lu, J., Behbood, V., Hao, P., Zuo, H., **Xue, S.**, and Zhang, G. (2015). Transfer learning using computational intelligence: A survey. *Knowledge-Based Systems*, 80: 14-23. (ERA Rank B)

Paper Under Review and Working Papers:

1. **Xue, S.**, Lu, J., and Zhang, G. (2018). Cross-domain network representation. *Pattern Recognition*, under review. (ERA Rank A*)

2. **Xue, S.**, Lu, J., and Zhang, G. (2018). Cross-domain similarity learning based on network patterns. *IEEE Transactions on Neural Networks and Learning Systems*, submitted. (ERA Rank A*)

3. **Xue, S.**, Lu, J., and Zhang, G. (2018). Cross-domain node embedding on heterogeneous information networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, will submit. (ERA Rank A*)

4. **Xue, S.**, Lu, J., and Zhang, G. (2018). Complex networks in decision making: A survey. *Decision Support Systems*, will submit. (ERA Rank A*)

Chapter 2

Literature Review

2.1 Random Walk-based Network Representation

The data structure of a network that includes nodes and links is complex for machine learning algorithms. Network representation inputs structural information and learns a latent feature space, which ensures that machine learning models accomplish artificial intelligence tasks (Yang et al., 2015). Both deep mining network structures and developing representation deep learning models achieves improvements in performance in network representation (Wang et al., 2018b). A good network representation reduces the number of machine learning difficulties. Similarly, a good network structure learning reduces representation learning loads (Tang et al., 2015).

2.1.1 Network Representation for Large-scale Information Networks

Large-scale information networks are usually extracted from real-world complex systems and in particular preserve network structures to retain all kinds of information. To acquire this information requires network analysis. The information is usually large in volume, which is good for mining, but is varied and imbalanced in content, which has disadvantages in network representation. The network structure is also naturally sparse, which requires special attention when network representations are conducted.

Attributions have been studied in graph/network representations to improve the network representation performance. Attributed community detection method was leveraged to add attributes on the links (Li et al., 2018a). Rich text information of the node attributes was learned (Yang et al., 2015). However, it introduces the extra challenge in identifying the two different kinds of information for network representations from network structures and node attributes. Therefore, some recent studies have focused on developing deep learning models to enhance the network representation learning which use network structural information only, such as GraphGAN (Wang et al., 2018b); sampling structural information with network properties, such as scale-free observations (Feng et al., 2018); predicting links on a dynamic network in the homogeneous space, such as the DepthLGP model (Ma et al., 2018); and capturing the evolutionary structure properties of a network (Zhou et al., 2018). The above attempts are all conducted on a single domain and learn from unlabeled data.

In contrast to the domain-specific approaches, our aim is to leverage useful information from similar domains by efficient domain adaptation. Previous studies on domain adaptation focused on latent feature space training to overcome the differences in feature distributions or attributes (Lu et al., 2015). They attempted to reduce the distribution discrepancy between similar domains to a minimum by feature matching and instance reweighting (Cao et al., 2018), because traditional machine learning models are only suitable for the uniformed feature distribution of different domains. Most such approaches are based on **M**aximum **M**ean **D**iscrepancy (MMD). For unsupervised domain adaptations, however, it has been proved that subspace-based techniques, such as domain adaptation on statistical manifold (Baktashmotlagh et al., 2014), outperform MMD-based approaches. Furthermore, two scenarios of homogeneous and heterogeneous feature spaces on what to transfer across domains, and how, have been separately studied. Domain adaptations on heterogeneous spaces (Moon and Carbonell, 2017) are more challenging but commonly exist in real-world applications. Augmented features for supervised and semi-supervised heterogeneous domain adaptation have been discussed (Li et al., 2014) to enable a **S**upport **V**ector **M**achine (SVM) classification task. A deep asymmetric transfer network has been proposed for unbalanced domain adaptation which offers deep models for the source domain and the target domains (Wang et al., 2018a). To address the imbalanced problem, Wang et al. (2018c) focused on completely-imbalanced labels for the network embedding using the intra-class similarity and inter-class dissimilarity measurement, where the solution can be introduced to unsupervised target domain network representations in a cross-domain way.

2.1.2 Shortest Path-based Network Patten Mining

Few studies have worked on cross-domain network representation learning, particularly cross-network similarity learning. Network analyses measure topology similarities, such as similarities in nodes (Ribeiro et al., 2017) and neighborhoods (Perozzi et al., 2014), to summarize network patterns. Node centralities evaluate the node position in a network so that nodes with close importance level to the network can be clustered as a group (Kermarrec et al., 2011). Node centrality is measured in two aspects on centrality and power, and each aspect consists of various indicators (Xue et al., 2017). For example, the indicator of closeness describes the node's ability in information spreading (Solé-Ribalta et al., 2016), and the indicator of betweenness calculates the node's control ability on information distribution (Goh et al., 2003; Newman, 2005a). Node centralities also work with random walks to sample up network substructures (Newman, 2001).

A state-of-the-art practice in network representation input to partition the network to a range of shortest paths in a fixed length (Grover and Leskovec, 2016; Mikolov et al., 2013; Perozzi et al., 2014). The shortest paths try to retain as much information as the network. This transformation from a network to a set of paths is essential for large-scale networks, because the computational complexity in the network search process is extremely high, since the representation starts from every node and walks along links. It has been proved that random walks that are rooted at each node and repeatedly search along the shortest paths can represent the structures of the whole network (Perozzi et al., 2014).

2.1.3 Network Representation by Random Walks

For the purpose of using network analysis on structured data, a series of algorithms have been proposed based on DeepWalk (Perozzi et al., 2014), which trains a natural language model on the random walks generated by the network structure. Denote a random walk w_{v_s} that starts from a root node v_s, DeepWalk slides a window in a length of $2w+1$, and maps the central node v_i to its representation $f(v_i)$. Hierarchical Softmax factors out the probability distributions $Pr(v_{i\pm d}|f(v_i))$, where $d = \{1, \cdots, w\}$, corresponding to the paths staring at v_i and going over all other nodes in the random walk. The representation f is updated to maximize the probability of v_i co-occurring with its context $\{v_{i\pm d},\ d = \{1, \cdots, w\}\}$. Random walk based DeepWalk shows promising results on large-scale network representation if the datasets have a satisfying structure.

LINE and Node2Vec are the other two structure-based network representation algorithms that improves the performance of DeepWalk. LINE (Tang et al., 2015) preserves both local and global network structure by first-order proximity and second-order proximity respectively and suitable for all kinds of networks, *i.e.,* directed and undirected networks and weighted and unweighted networks. Node2Vec (Grover and Leskovec, 2016) explores the diverse neighborhoods of nodes in a biased random walk procedure with search bias α.

Above mentioned algorithms are inspired from recent advancements in unsupervised feature learning and the language modeling from sequences of words to vectors or networks. They contributed to the network analysis by modeling a stream of short random walks. Different from traditional representation learning, the latent feature learning of network representation captures neighborhood similarity and community

membership in topologies. The potential of these algorithms in real-world scenarios is their good performance in large heterogeneous networks.

To the best of our acknowledge, a few works have studied cross-domain network adaptations, but most of them are for sentimental purposes (Li et al., 2018b). The cost is also a key aspect of cross-domain representation in large-scale networks, which directly affects the performance of machine learning algorithms (Bousmalis et al., 2016). To meet all the above requirements, we try to find a solution from the domain similarity learning and start from the network pattern of node centralities. Our work can be categorized in a cross-domain network representations based on domain similarities which only focuses on network structure information. It generally works for both homogeneous and heterogeneous domain adaptations. The properties of network patterns also contribute to network representation in network structure sampling in the two domains, therefore the similarity learning algorithm using network patterns is efficient for the cross-domain network representations.

2.2 Graph-theoretic Node Importance Mining in Information Networks

Information networks use graph theory to express nodes and links in perspective of complex networks. The information senders and receivers in a network are denoted as nodes, and the information flows between nodes are denoted as links. In this way, an information network can be extracted as a complex network model and mapped into an adjacent matrix. In this section, we survey the articles that use

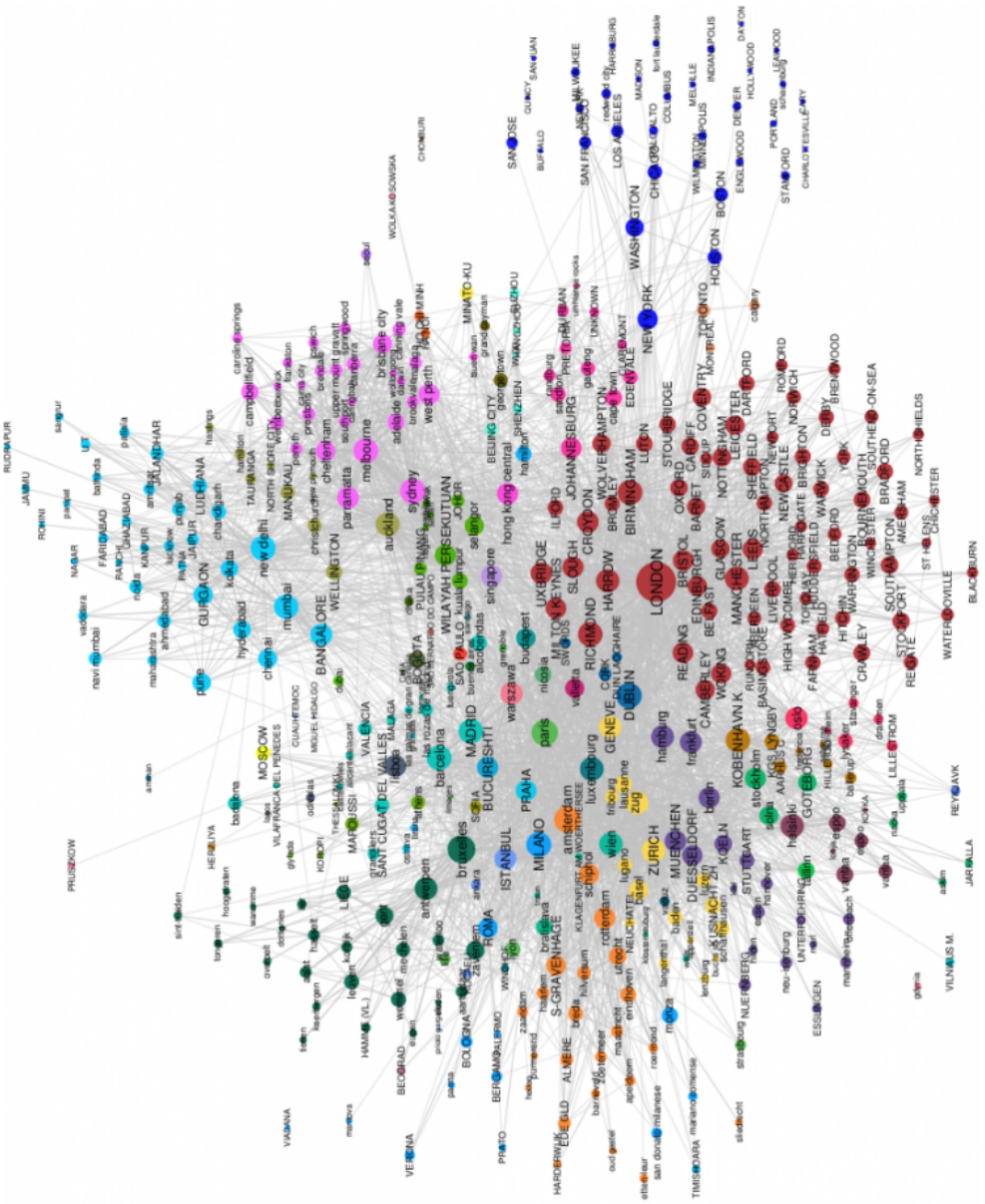


Figure 2.1 An illustration of information network with highlighted important node

complex network theories to extract information networks as a complex system and use graph-theoretic node importance mining methods (Figure 2.1).

In the real world, the rapid development of global finance and technology has led to the inclusion of a new set of importance nodes in information networks. The emergence of developing important nodes and these changes to the information network have attracted attention from both academics and industry. Evaluating the node importance in an information network is a fundamental academic question, but it is also very challenging to detect these changes. One of the phenomena within these network changes is that developing nodes are replacing the positions of other nodes that used to be important in the network. The significance and innovations of graph-theoretic node importance mining in information networks are as follows:

- It represents real-world networks as a part of information network, for example, in document networks, financial networks and cultural networks;
- It prevents the negative influence of information networks, for example, controlling and preventing the spread of rumor over Internet; and
- It improves the robustness of information networks by network optimization.

Take the most complex information network for example, **W**orld **C**ity **N**etwork (WCN) combine two concepts: cities and complex systems. Previous research has defined node importance, and how to evaluate the importance of nodes within a WCN has also been discussed. Within the three perspectives of node importance mining – social network analysis, system science and information searching – Neal (2011) made great contributions to node importance mining in WCN by proposing a differentiation between centrality and power. Because of the development of

large-scale networks and large-scale graph mining, the research in this area has dramatically developed in the past five years. The study of node importance has been expanded in both breadth and depth. For example, Liu et al. (2013) provided an in-depth analysis of node importance models in large city networks. Their work focuses on network topologies and spreading dynamics and explains their advantages, shortcomings and computing complexity.

The study of information networks covers a broad range of fields which includes document networks and information services over resources flows. In the scope of social and behavioral science, node importance mining method implements the process of digital information capture, analysis and deep mining, which acquires knowledge from raw data by intelligent techniques and achieves the goal of digital information supply chain, including transport, flows, tracking, exchanging and sharing. The graph-theoretic mining method benefits to large-scale datasets and heterogeneous datasets, which is a supportive tool of big data in the graph mining domain.

Motivated by the significance and the innovation of graph-theoretic node importance mining in information networks, we survey the graph-theoretic methods for mining node importance and the prospects for future works. In the remaining parts of this section, we review the graph-theoretic indicators of node importance; the graph-theoretic node importance mining methods on network topologies (in the view of static networks) including but not limited to node relevance, centrality measurement, power measurement, measurement with centrality and power and heterogeneous fusion; and graph-theoretic methods for mining node importance that are based on transmission mechanisms (in the view of dynamic networks) which

consist of network evolution, node immunization and robustness in dynamics. The last section presents our analysis and main findings.

2.2.1 Graph-theoretic Indicators of Node Importance

Centrality

Centrality is one of the most popular methods to measure a node's position and its importance in an information network. Centrality requires that nodes are not only independent from each other but also related to other nodes in the network (Neal, 2011). Centrality means the node that links to the most resources in positive linked networks is a central node and can be partially regarded as the important node of an information network evaluated by the centrality analysis. In other words, centrality reflects the resources that are spread and gathered in an information system.

In WCN, Frideman (1986) was the first to propose a world city hypothesis. He asserted that the important nodes in WCN are always those cities that are home to the headquarters of multinational companies, while world cities are those that possess the most social resources. Sassen (2010) extended this world city hypothesis from business into finance, indicating that the Internet gathers information resources, so that the centrality has various styles. However, the two centralities defined by Friedmann and Sassen both reflect only one aspect of resource gathering but do not consider the spread of resources. For example, in the resource flow of city logistic networks (Duan and Lu, 2013) and airline networks (Derudder et al., 2004), the important nodes are not the cities that act as resource-gathering centers, but they are the cities that spread resources. Therefore, Neal (2011) extended the traditional concepts of centrality and indicated that centrality should have both

resource-gathering and resource-spreading characteristics, so does other kinds of information networks.

Power

Power is defined by the node that controls the most information resources in a negative linked network. For example, cities with the headquarters of multinational companies are the command and control centers in global finance markets. However, in actor-network theory (Latour, 2005), all cities can be translated into actors and can, therefore, be included in unified networks to maintain a balanced status. Therefore, a single city cannot control all the power. The power lies within a system and is obtained by network spreading (Allen, 2009). Therefore, Neal (2011) defines power as the influence of resource flows.

Combination of Centrality and Power

As evidenced above, centrality and power are two major indicators for mining node importance from information networks. Neal (2011) first described the difference between centrality and power. The social behavior exchange theory states that when two sides meet their requirements from each other, they become more highly reliant; however, when one side becomes an unavoidable choice for the other, the former can assume control of the exchange processes. Therefore, controlling power is not only influenced by the relationships within information networks but also between related nodes. If one node is strongly connected to other nodes, then that node exerts greater influence and control over the information network. For example, suppose a Central Node A and a Central Node B have the same number of links (as shown in Figure 2.2), but the nodes linked to Node A also link to many other nodes, while the nodes linked to Node B have no other links, then we can say Node B has greater control of the network than Node A.

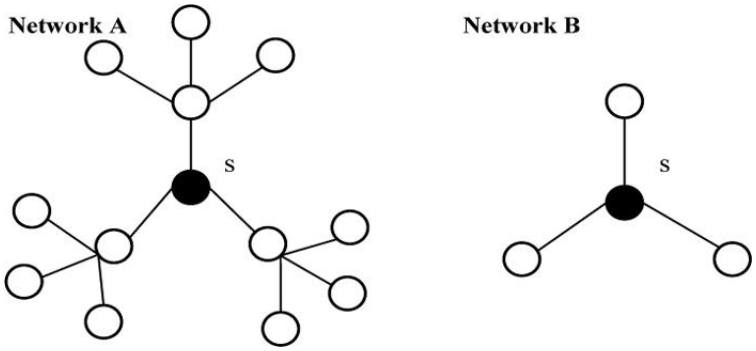


Figure 2.2 Two network examples with different node importances

Neal (2011) proved that power is not equal to centrality. Those nodes with high centrality may not have great power (Cook et al., 1983). To this end, Cook et al. (1983) introduced a multi-dimensional matrix of network positions to better describe node importance in information networks. Using this matrix, they found in WCN that nowadays, cities like New York and London have both high centrality and high power; cities like Washington and Brussels have high centrality and low power; and cities like Miami and Stockholm have low centrality and high power. Moreover, the importance of a node is also influenced by its sub-connected nodes (Neal, 2013a) and the two-mode interlocking networks (Liu and Derudder, 2012).

2.2.2 Graph-theoretic Node Importance Mining on Network Topologies

There are many methods that mine important nodes from information networks. Most of them are based on graph theory and graph mining technology (Xie et al., 2015); however, the most popular methods are based on network topologies. The methods of node importance mining based on network topologies are different from network connectivity mining, as they are a kind of graph-theoretic measures of centrality, and

thus, they consider the structure of networks so that the potential node integrations within information networks are evaluated as well (Neal, 2012). In this section, we survey node relevance, centrality and power-based measurements, heterogeneous fusion and other methods. The subsequent relationships of the methods are shown in Figure 2.3.

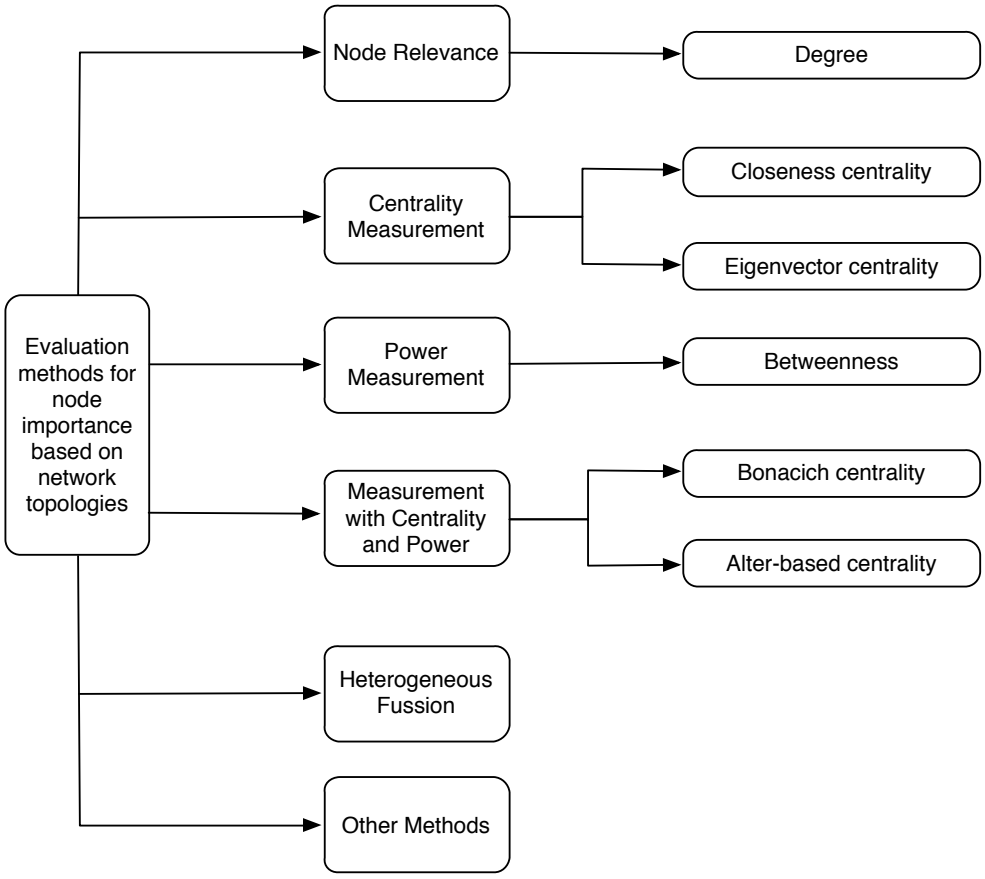


Figure 2.3 Graph-theoretic node importance mining methods on network topology

The graph-theoretic node importance mining methods based on network topologies comprise two main categories: node relevance and shortest path. The method of node relevance is measured by degree analysis. The methods of shortest path that aim

at finding optimal spreading paths are measured by several node importance analyses, for example, betweenness, closeness centrality, eigenvector centrality, Bonacich centrality and alter-based centrality. Betweenness is used particularly for measurements of power, while closeness centrality and eigenvector centrality are used particularly for measurements of centrality. Bonacich centrality is an extension of eigenvector centrality which measures node importance on both centrality and power. The other mining methods for node importance based on network topologies included in this review are via processes such as node deleting, node contraction and data mining and machine learning - embedded techniques. For heterogeneous network structures, fusion methods integrate all the previously mentioned measurements.

Node Relevance

Node relevance focuses on the characteristics of network structures. It evaluates the links in the network and the condition of the links. A classical method is by degree analysis. The degree centrality is one of the most important indicators of power. Some existing research regards centrality and power as the same indicator (Derudder et al., 2004). The degree denotes the number of links that directly connect to a node (Freeman, 1978). Node relevance can also reflect the influence of a node within a information network, but this method only considers the number of neighbors, not the importance of the neighbor nodes. In particular, when networks meet the Matthew effect, the evaluation of the information network is not accurate. Nodes that act as centers can be important nodes in information networks, even though they do not control many information resources.

Centrality Measurement

Closeness centrality

Closeness centrality is the reciprocal of the sum of the shortest paths between all nodes which is used to reflect a node's ability to gather and spread information resources. A node is regarded as the closeness centrality only when it has a short number of paths from other nodes in information networks (Freeman, 1978). The node with the highest closeness centrality score is considered the center of the network. However, this method relies on network topology. The process of interlocking somehow promotes the closeness centrality, as it is common in information network construction and produces dense in node importance mining (Neal, 2012). Therefore, closeness centrality works well on star networks but is not suitable for random networks.

Eigenvector centrality

Eigenvector centrality is an important method for centrality that calculates the reputation of a single node by summing the linear reputations of all other nodes in the network (Borgatti and Everett, 2006). It uses the eigenvectors corresponding to the eigenvalues of linear equations. The eigenvectors comprehensively analyze the importance of their neighbor nodes and assume diverse importance among them. For this reason, it stands out for its accuracy compared to other centrality methods. Poulin et al. (2000) proposed a cumulated nomination indicator based on iteration mapping when solving eigenvectors. The cumulated nomination is suitable for both big and multiple-branch networks because of its rapid convergence speed. Affected by interlocking, however, only in a few situations, the eigenvector centrality method of node importance causes star topologies in information networks which easily indicates eigenvector centrality and shows its robustness (Neal, 2012). Therefore, a

disadvantage of this method is that it is difficult to apply in the real world because eigenvectors are based on linear topologies and most nodes in the real world are not linear.

Power Measurement

Betweenness was first proposed by Freeman in 1977 to evaluate the power of one single node on an entire social network. Betweenness is calculated by the number of shortest paths that include the node (Freeman, 1977). Betweenness is an important indicator for measuring the control of power now (Alderson and Beckfield, 2004), as it can accurately reflect which nodes control the resources in networks and monitor their flow. Cities like Miami, which serve as a transportation link between several other cities, influence entire networks and therefore hold important positions. Hennemann and Derudder (2014) proposed a framework based on traditional betweenness which integrated a randomized baseline model and considered the original degree distribution. This framework distinguished the important links from the overall connectivity. Sun et al. (2016a) proposed a vulnerability evaluation model aimed at detecting weak nodes by introducing methods from graph theory and complex networks. In their experiments, they found that the nodes with a high degree of betweenness also had high influence in urban transportation networks. Such nodes are easily attacked, and nodes with high betweenness offer key links in information networks. However, betweenness cannot be directly applied to big networks.

Measurement with Centrality and Power

Bonacich centrality

Bonacich centrality is a method that calculates eigenvector centrality and measures the characteristics of centrality and power (Bonacich, 1987). It consists a user-defined attenuation parameter and an adjacent matrix of the network. When

the attenuation parameter is positive, Bonacich centrality is used to find the links to central nodes in the network; when the attenuation parameter is negative, it means that the network has few connected nodes, the nodes cannot be replaced and they take control of the resources in the network. Furthermore, Neal (2011) indicated that the eigenvector and Bonacich centrality methods could not find the key nodes from networks when eigenvalues become high. Boyd et al. (2013) extended Neal's theory by introducing a new parameter. In this method, eigenvector centrality is relevant to recurrent centrality, but it is not relevant to power. In previous research, Bonacich centrality proved to be feasible in a few circumstances (Neal, 2013a) because of this attenuation parameter.

Alter-based centrality

Neal (2013a) proposed alter-based centrality to evaluate the positions of nodes in information networks. Alter-based centrality consists of a recursive centrality indicator and a recursive power indicator and considers the degrees of both the nodes and their neighbors. Suppose there are two networks, as shown in Figure 2.2, the degrees of the black nodes s are both 3. In Network A, the recursive centrality of the node s is 9 and its recursive power is 0.75; in Network B, its recursive centrality and recursive power are both 3. The above results show that compared to other methods, alter-based centrality which consists of recursive centrality and recursive power can distinguish nodes by local structures. Although Neal's alter-based centrality considered the relationships between nodes, it still has some shortcomings (Boyd et al., 2013; Neal, 2013a,b) which are as follows: 1) partial recursion is not sufficient to evaluate node importance mining; 2) it does not consider the degrees of all nodes; and 3) the parameters only reflect the degree and connection strengths but ignores their weight. These disadvantages reduce node importance mining accuracy.

Heterogeneous Fusion

The heterogeneous fusion of node importance mining methods on network topologies has been proposed for a few applications. Liu et al. (2014b) compared the centralities of degree, closeness centrality and betweenness centrality on a two-mode WCN of an inter-city network and inter-firm network where the firms have their branches in different cities and different firms in one city produce behaviors such as information sharing, cooperation and innovation diffusion: degree evaluates the direct linkage of individual nodes to other nodes in WCN, closeness centrality evaluates the inverse distances of individual nodes to other nodes in WCN and betweenness centrality detects the nodes that control or facilitate interactions. Jia et al. (2015) suggested the comprehensive consideration of the influence of node characteristics and nodes that are heterogeneous to network topologies. The mining of important nodes is conducted through synthetic evaluation and merging, main eigenvector calculations with a graphical Fourier transform and heterogeneous node integration through graph signal-processing-based centrality. Konstantakis et al. (2015) improved Global VAR to model a network using different types of economic entities and analyze networks by degree centrality, the recursive power of alter-based centrality and Bonacich centrality.

Other Methods

The graph-theoretic node importance mining methods based on network topologies also integrate the data mining and machine learning methods. Take the data mining method integration for example, benchmarking and self-organizing maps were proposed for heterogeneous world cities to which hierarchical analysis was applied in network topological research (Arribas-Bel et al., 2013); Neal (2014) validated the methods of naive, normalized, interlocking and sorting in WCN mea-

surement; and social network analysis studied economic corporation networks (Sigler and Martinus, 2017). Sanderson et al. (2015) used a least squares regression model to mine the relationships between WCN and immigration. This research found that degree centrality was the better choice for this research problem compared to betweenness.

2.2.3 Graph-theoretic Node Importance Mining on Transmission Mechanisms

Node importance also relates to factors in transmissions within fixed network topologies and changing topologies. By introducing network dynamics, researchers have tried to find the driving nodes that control the entire network as another kind of the key methods of node importance mining (De Domenico et al., 2015; Pei and Makse, 2013; Saito et al., 2012). In this section, we review the graph-theoretic node importance mining methods of network evolution, node immunization and robustness in dynamics, based on transmission mechanisms.

Network Evolution

A set of visualization methods, such as alluvial diagrams (Liu et al., 2014a), are popularly used to illustrate the network evolutions over time. Most of the network evolutions are reflected by similarity/dissimilarity patterns on clusters which indicate the major tendencies. Using hierarchical cluster analysis, the temporal evolutions and the regional patterns can also be extracted. To address the great challenge of conducting visual analysis on dynamic networks, Hadlak et al. (2013) developed a new approach to discover the sub-structures involving important nodes and links, which

share similarities over time, jointly using the techniques of intelligence computation, data visualization and network interaction analysis.

Node Immunization

In the epidemic model, the node immunization problem focuses on epidemic dynamic processes. The nodes in such a network can have different states, such as susceptible (S), infected (I), recovered (R) and exposed (E). The transmissions between different states form various epidemic models representing corresponding immune strategies. The well-known epidemic models of **S**usceptible-**I**nfected (SI) (Chen et al., 2016), **S**usceptible-**I**nfected-**S**usceptible (SIS) (Saito et al., 2012) and **S**usceptible-**I**nfected-**R**ecovery (SIR) (Wang, 2014; Zhang et al., 2015) work on the node immunization. Du et al. (2014) first applied **T**echnique for **O**rder of **P**reference by **S**imilarity to **I**deal **S**olution (TOPSIS) to identify node importance in complex network area which was supported by centrality evaluation methods according to network types. Hu et al. (2016) then evaluated node-spreading abilities as a measurement of node importance using modeling techniques, TOPSIS and weighted centrality in degrees, along with closeness centrality and betweenness, simulated in a SIR model, and proved that indicator weightings lead to accurate results. This research was previously supported in the study by Hu et al. (2015) by considering the relationships between a node and all its neighbor and non-neighbor nodes, and its robustness was proven using a deleting method. Sun et al. (2016b) developed an appropriate node importance mining method for the real-world application of energy industry wireless sensor networks by introducing important transmission parameters in an influence transfer-feedback mechanism.

Robustness in Dynamics

The transmission-mechanism-based robustness analysis of node importance in dynamic networks includes node deleting and node contraction. Taking the node deleting method as an example, it evaluated the node importance, linkages and the network's robustness when deleting some nodes. The basic idea of this method is that if the deleted nodes cause massive destruction to the network, the nodes must be important. Corley and David (1982) regarded the distance changes between the source node and the target node as a factor of node importance when nodes are deleted. Based on the study of influential spreaders in complex networks, Niu et al. (2015) analyzed the robustness of the methods of degree centrality, closeness centrality, betweenness, k-shell decomposition and eigenvector centrality. By adding, deleting and resetting links in original networks, node importance may change.

2.2.4 Comprehensive Analysis and Findings

In this section, we have reviewed several current trends of graph-theoretic node importance mining in information networks, which are developed on the bases of graph-theoretic indicators of node importance: centrality and power. According to this review, the graph-theoretic mining methods used in node importance are classified into two main groups based on static and dynamic: statistic mining on network topologies and dynamic mining on transmission mechanisms. The trends of node importance mining include:

- **Centrality versus Power.** As the node importance indicator of power is distinguished from centrality, more studies focus on the characteristic of the power of control over the information network. In the state-of-the-art research

works, many more methods and applications are measured with the indicator of power compared with centrality. The main reason is that power indicates the influence of the node (i.e. information flows), reflects the controlling performance of network spreading and has higher robustness in dynamics.

- **Centrality/Power versus Centrality and power.** Of the methods and applications on node importance indicators, miscellaneous methods with centrality and power measurements have been extensively used for node importance mining in information networks, mainly in the domain of theoretic methods. Nevertheless, miscellaneous methods are more suitable in dealing with complex real-world applications, especially the application problems that are solved by transmission mechanisms.

- **Methods on network topologies versus Transmission mechanisms.** The theoretic research on transmission mechanisms are relatively new compared with the research on network topologies; however, it is playing an increasingly important role in recent applications. As transmission processes rely on resource distribution on information networks, researchers take robustness into account for power. This is why there is limited work on the mining of centrality.

- **Methods versus Applications.** Information networks face various challenges from multiple fields such as economics, politics, cultures and information technology. In recent studies, most application problems are represented by transmission mechanisms. It seems that accurately measuring the node importance in the real world using single indicator could be difficult in the future. Although the research of node importance mining has developed over

more than two decades, it is strongly limited by techniques of graph theory-based complex network and with the constraints of real-world applications and vice versa.

From this survey, we address several important research challenges of graph-theoretic node importance mining and offer the following suggestions for future work. So far, node importance mining methods have been applied to small-scale problems in information networks. Nonetheless, in the era of big data, there are many interesting applications that can exploit where the measurements of degree, eigenvector centrality, Bonacich centrality and alter-based centrality can manage large-scale networks. In the future, the techniques of computational intelligence will pave the way for using data-driven graph mining and case-based learning in complex real-world applications.

2.3 Transfer Learning

Although machine learning technologies have attracted a remarkable level of attention from researchers in different computational fields, most of these technologies work under the common assumption that the training data (source domain) and the test data (target domain) have identical feature spaces with underlying distribution. As a result, once the feature space or the feature distribution of the test data changes, the prediction models cannot be used and must be rebuilt and retrained from scratch using newly- collected training data, which is very expensive and sometimes not practically possible. Similarly, since learning-based models need adequate labeled data for training, it is nearly impossible to establish a learning-based model for a target domain which has very few labeled data available for supervised learning. If

we can transfer and exploit the knowledge from an existing similar but not identical source domain with plenty of labeled data, however, we can pave the way for construction of the learning-based model for the target domain.

Transfer learning has emerged in the computer science literature as a means of transferring knowledge from a source domain to a target domain. Unlike traditional machine learning and semi-supervised algorithms, transfer learning considers that the domains of the training data and the test data may be different (Fung et al., 2006). Traditional machine learning algorithms make predictions on the future data using mathematical models that are trained on previously collected labeled or unlabeled training data which is the same as future data (Baralis et al., 2008; Kuncheva and Rodriguez, 2007; Yin et al., 2006). Transfer learning, in contrast, allows the domains, tasks, and distributions used in training and testing to be different. The study of transfer learning has been inspired by the fact that human beings can utilize previously-acquired knowledge to solve new but similar problems much more quickly and effectively. The fundamental motivation for transfer learning in the field of machine learning focuses on the need for lifelong machine learning methods that retain and reuse previously learned knowledge. Research on transfer learning has been undertaken since 1995 under a variety of names: learning to learn; lifelong learning; knowledge transfer; meta learning; inductive transfer; knowledge consolidation; context sensitive learning and multi-task learning (Pan and Yang, 2010).

2.3.1 Transfer Learning Concepts

Definition 2.1. Domain (Pan and Yang, 2010) A domain, which is denoted by $\mathscr{D} = \{\mathscr{X}, P(X)\}$, consists of two components: 1) Feature space $\mathscr{X}$; and 2) Marginal probability distribution $P(X)$, where $X = \{x_1, \cdots, x_n\} \in \mathscr{X}$.

Definition 2.2. Task (Pan and Yang, 2010) A task, which is denoted by $\mathscr{T} = \{Y, f(\cdot)\}$, consists of two components: 1) A label space $Y = \{y_1, \cdots, y_m\}$; and 2) An objective predictive function $f(\cdot)$ which is not observed and is to be learned by pairs $\{x_i, y_i\}$.

The function $f(\cdot)$ can be used to predict the corresponding label, $f(x_i)$, of a new instance x_i. From a probabilistic viewpoint, $f(x_i)$ can be written as $P(y_i|x_i)$. More specifically, the source domain can be denoted as $\mathscr{D}^s = \{(x_1^s, y_1^s), \cdots, (x_n^s, y_n^s)\}$ where $x_i^s \in \mathscr{X}^s$ is the source instance and $y_i^s \in Y^s$ is the corresponding class label. Similarly, the target domain can be denoted as $\mathscr{D}^t = \{(x_1^t, y_1^t), \cdots, (x_n^t, y_n^t)\}$ where $x_i^t \in \mathscr{X}^t$ is the target instance and $y_i^t \in Y^t$ is the corresponding class label and in most scenarios $t_n \ll s_n$.

Definition 2.3. Transfer learning (Pan and Yang, 2010) Given a source domain $\mathscr{D}^s$ and its learning task $\mathscr{T}^s$, a target domain $\mathscr{D}^t$ and its learning task $\mathscr{T}^t$, transfer learning aims to improve the learning of the target predictive function $f^t(\cdot)$ in $\mathscr{D}^t$ using the knowledge learned from $\mathscr{D}^s$ and $\mathscr{T}^s$ where $\mathscr{D}^s \neq \mathscr{D}^t$ or $\mathscr{T}^s \neq \mathscr{T}^t$.

In the above definition, the condition $\mathscr{D}^s \neq \mathscr{D}^t$ implies that either $\mathscr{X}^s \neq \mathscr{X}^t$ or $P^s(X) \neq P^t(X)$. Similarly, the condition $\mathscr{T}^s \neq \mathscr{T}^t$ implies that either $Y^s \neq Y^t$ or $f^s(\cdot) \neq f^t(\cdot)$. In addition, there are some explicit or implicit relationships between the feature spaces of two domains such that we imply that the source domain and target

domain are related. It should be mentioned that when the target and source domains are the same ($\mathscr{D}^s = \mathscr{D}^t$) and their learning tasks are also the same ($\mathscr{T}^s = \mathscr{T}^t$), the learning problem becomes a traditional machine learning problem.

According to the uniform definition of transfer learning introduced by Definition 2.2, transfer learning techniques can be divided into three main categories: 1) Inductive transfer learning, in which the learning task in the target domain is different from the learning task in the source domain ($\mathscr{T}^s \neq \mathscr{T}^t$); 2) Unsupervised transfer learning which is similar to inductive transfer learning but focuses on solving unsupervised learning tasks in the target domain such as clustering, dimensionality reduction and density estimation ($\mathscr{T}^s \neq \mathscr{T}^t$); and 3) Transductive transfer learning, in which the learning tasks are the same in both domains, while the source and target domains are different ($\mathscr{T}^s = \mathscr{T}^t$, $\mathscr{D}^s \neq \mathscr{D}^t$). When the method aims to optimize the performance on multiple tasks or domains simultaneously, it is considered to be multi-task learning. If it optimizes performance on one domain, given training data that is from a different but related domain, it is considered to be transductive transfer learning or domain adaptation. Transfer learning and transductive transfer learning have often been used interchangeably with domain adaptation. In addition, unsupervised domain adaptation can be considered as a form of semi-supervised learning, but it assumes that the labeled training data and the unlabeled test data are drawn from different distributions. The existing techniques and methods, which have thus far been used to handle the domain adaptation problem, can be divided into four main classes:

- Instance weighting for covariate shift methods which weight samples in the source domain to match the target domain. The covariate shift scenario might

arise in cases where the training data has been biased toward one region of the input space or is selected in a **non**-**i**ndependent and **i**dentically **d**istributed (non-i.i.d.) manner. It is closely related to the idea of sample-selection bias which has long been studied in statistics (Heckman et al., 2013) and in recent years it has been explored for machine learning. Huang et al. (2007) proposed a novel procedure called **K**ernel **M**ean **M**atching (KMM) to estimate weights on each instance in the source domain, based on the goal of making the weighted distribution of the source domain look similar to the distribution of the target domain. Sugiyama et al. (2008) and Tsuboi et al. (2009) proposed a similar idea called **K**ullback-**L**eibler **I**mportance **E**stimation **P**rocedure (KLIEP). Here too the goal is to estimate weights to maximize similarity between the target and weight-corrected source distributions.

- Self-labeling methods which include unlabeled target domain samples in the training process and initialize their labels and then iteratively refine the labels. Self-training has a close relationship with **E**xpectation **M**aximization (EM) algorithm, which has hard and soft versions. The hard version adds samples with single certain labels while the soft version assigns label confidences when fitting the model. Tan et al. (2009) modified the relative contributions of the source and target domains in EM. They increased the weight on the target data at each iteration, while Dai et al. (2007) specified the trade-off between the source and target data terms by estimating **K**ullback-**L**eibler (KL) divergence between the source and target distributions, placing more weight on the target data as KL divergence increases. Self-training methods have been applied to domain adaptation on **N**atural **L**anguage **P**rocessing

(NLP) tasks including parsing (Sagae, 2010); **P**art-**o**f-**S**peech (POS) tagging (Jiang and Zhai, 2007a); conversation summarization (Sandu et al., 2010); entity recognition (Ciaramita and Chapelle, 2010; Jiang and Zhai, 2007b); sentiment classification (Tan et al., 2008); spam detection (Jiang and Zhai, 2007a); cross-language document classification (Shi et al., 2010); and speech act classification (Jeong et al., 2009).

- Feature representation methods which try to find a new feature representation of the data, either to make the target and source distributions look similar, or to find an abstracted representation for domain-specific features. The feature representation approaches can be categorized into two classes: 1) Distribution similarity approaches aim explicitly to make the source and target domain sample distributions similar, either by penalizing or removing features whose statistics vary between domains (Arnold et al., 2007; Jiang and Zhai, 2007b) or by learning a feature space projection in which a distribution divergence statistic is minimized (Chen et al., 2009; Pan et al., 2011); 2) Latent feature approaches aim to construct new features by analyzing large amounts of unlabeled source and target domain data (Ben-David et al., 2010; Ciaramita and Chapelle, 2010; Pan et al., 2010).

- Cluster-based learning methods rely on the assumption that samples connected by high-density paths are likely to have the same label if there is a high density path between them (Gao et al., 2008). These methods aim to construct a graph in which the labeled and unlabeled samples are the nodes, with the link weights among samples based on their similarity. Dai et al. (2007) proposed a co-clustering based algorithm to propagate the label information across

domains for document classification. Xue et al. (2008) proposed a cross-domain text classification algorithm to integrate labeled and unlabeled data from different but related domains.

2.3.2 Transfer Learning using Naïve Bayes

Bayesian techniques refer to methods that are related to statistical inference and are developed based on Bayesian theorem. A Bayesian classifier is a probabilistic methodology for solving classification problems. Since probability is a useful tool for modeling the uncertainty in the real world and is adequate for quantifying the certainty degree of an uncertain truth, Bayesian classifier is popular in the machine learning community. When it comes to the transfer learning setting, the distribution of the training data and test data is not identical, so a Bayesian classifier trained on training data may not be predictive for the test data. To address this challenging problem, Bayesian-based transfer learning algorithms have been developed in recent years.

The naïve Bayes classifiers (Lewis, 1992) are among the most popular classifiers in real world application. They pose a simple but strong assumption that there is independence between each pair of features given the class variables. Though this assumption is not suitable in most real scenarios, naïve Bayes classifiers have nevertheless been proved to work quite well in some complicated applications, especially automatic medical diagnosis (Kononenko, 1993), spam filtering (Androutsopoulos et al., 2000) and text categorization (Sebastiani, 2002) , in which they may even outperform more advanced algorithms, such as support vector machine, or random forests. Normally, the probabilistic model for a classifier is

$$P(C|F_1,\cdots,F_n) = \frac{p(C)p(F_1,\cdots,F_n|C)}{p(F_1,\cdots,F_n)} \tag{2.1}$$

where $p(F_1,\cdots,F_n)$ indicates a posteriori probability of class variable C, conditional on feature variables F_1 through F_n. Since $p(F_1,\cdots,F_n)$ has no relation with the class variable and the value of $F_i(i=1,\cdots,n)$ is observable, the above equation can be expressed as

$$P(C|F_1,\cdots,F_n) \propto p(C)p(F_1,\cdots,F_n|C) \tag{2.2}$$

Under the independence assumption adopted by naïve Bayes classifier, which means

$$P(C|F_1,\cdots,F_n) \propto p(C)\prod_{i=1}^{n} p(F_i|C) \tag{2.3}$$

From Eq. (2.3) we find that a prediction made by a classifier depends on the prior probability of the class variable and the product of the likelihood of each feature variable given a specific class variable. To estimate each feature's distribution, it is necessary to make parameter estimation, assuming a predefined distribution (i.e., multinomial distribution or multivariate Bernoulli distribution) or generating a non-parametric model for a feature that comes from training data. However, if the test data (new-domain data) follow a different distribution from the training data (old-domain data), we cannot obtain an accurate feature distribution estimation for the new-domain data based on the parameter learned from the old-domain data, which leads to bad prediction performance in the result. Estimating the feature distribution

for new-domain unlabeled data limits the application of the naïve Bayes classifier in the transfer learning setting.

To adapt the naïve Bayes classifier from the training data to the test data, Dai et al. (2007) proposed a novel **N**aïve **B**ayes **T**ransfer **L**earning (NBTL) classification algorithm for text categorization. NBTL first trains a naïve Bayes classifier on the training data and applies the learned classifier on the test data to obtain a pseudo label for the test data during learning, thereby providing an initial model estimation for the test data under target distribution. The EM algorithm is then applied in iteration to find a local optimal model only for fitting the target distribution, meaning that the naïve Bayes classifier trained on the training data is adapted to the test data. To measure the difference between the different distributions, KL divergence is used to estimate a trade-off parameter in the NBTL model, and the experiment results show that the performance of NBTL increases when the distribution between the training data and the test data is significantly different. The main disadvantage of NBTL lies in the fact that the influence of new-domain specific features is ignored. Instead of treating both old-domain and new-domain data equally, an adaptive naïve Bayes is proposed in (Tan et al., 2009). It uses a weighted EM algorithm to dynamically increase the importance of new-domain data and decrease the weight of old data, while at the same time emphasizing the usage of both generalizable features drawn from the old-domain data and all the features from the new-domain data for tackling the cross-domain sentiment classification problem. Roy and Kaelbling (2007) developed an alternative method of transferring the naïve Bayes classifier. They first partition the dataset into a number of clusters, such that the data for each cluster for all tasks has the same distribution. Then they train one classifier for each partition; all classifiers are then combined using a Dirichlet process.

In addition to text classification, Ma et al. (2012) developed a **T**ransfer **N**aïve **B**ayes (TNB) algorithm to predict cross-company software defects. The implementation can be summarized in three steps: it first collects maximum and minimum value vectors of the target feature from test data, then each feature of a training sample is compared with the corresponding part of those two vectors to calculate the number of similar attributes and the weight of that training instance is computed through a gravitational analogy. After obtaining all the weights for the training data, a prediction model can be built with those weighted training data to classify the test dataset.

2.3.3 Comprehensive Analysis and Findings

In this section, we have reviewed several current trends of computational intelligence-based transfer learning. From the summary of transfer learning, it is concluded that transfer learning with the use of computational intelligence, as an emerging research topic, starts playing an important role in almost all kinds of application. In the future, several important research challenges in the field of computational intelligence-based transfer learning need to be addressed.

- First, the computational complexity is a crucial issue in computational intelligence-based transfer learning. Almost, all reviewed studies have focused on accuracy as a measurement for model performance. However, comparing with the statistical transfer learning methods, computational intelligence techniques usually gain more computational complexity which should be handled.

- In addition, how to avoid negative transfer is an open problem in not only the classical transfer learning but also in computational intelligence-based transfer

learning. The transferability among source and target domains needs to be studied profoundly and a comprehensive and accurate transferability measures to be implemented that can guarantee the negative learning will not happens.

- Moreover, all reviewed studies have assumed that the feature spaces between the source and target domains are the same. However, in many applications, which we wish to transfer knowledge among domains, this assumption cannot be held. This type of transfer learning which is referred as the heterogeneous transfer learning has not been addressed in computational intelligence-based transfer learning literature.

- Finally, so far the computational intelligence techniques are applied for small scale transfer learning problems. Nonetheless, in the era of big data, there are many interesting applications such as social network analysis and web-based recommender systems that can exploit transfer learning and computational intelligence techniques. The capability of computational intelligence to handle non-i.i.d. noisy data can pave the way to use these techniques in big scale real world applications.

Chapter 3

Framework of Transferring Structures across Large-scale Information Networks

3.1 Introduction

In this chapter, we consider the following challenges of developing a framework for transferring the network structures across large-scale information networks.

- **Challenge 1**: How to effectively predict links between nodes across relational networks for the purpose of improving performance of network representation in the target network?
- **Challenge 2**: How to transfer the random walks in the source network to the target network based on the similarity measurement achieved in Challenge 1?

To this end, we propose a FTLSIN which implements an unsupervised feature learning for scalable networks. FTLSIN is built by a two-layer random walk to generate a neighborhood of nodes in the target network with a secondary from the learned walks in the source network which measures the similarity and predicts links across networks. Experiment results on real-world datasets empirically demonstrate that FTLSIN achieves better performance compared to the state-of-the-art network representation algorithms.

3.2 Problem Statement

The problem of FTLSIN is formulated as follows. Suppose we have a source domain ($\mathscr{D}^s$) and a target domain ($\mathscr{D}^t$), where the source domain has a source network $G^s = (V^s, E^s)$ with its corresponding label space Y^s, and the target domain has a target network $G^t = (V^t, E^t)$ with its label space Y^t. Both networks are unweighted. For a cross-domain classification problem $< \mathscr{T}_t, \mathscr{T}_s, (\mathbf{x}^t_{test}, y^t_{test}) >$, we firstly implement a latent feature learning procedure from topology structures of G^s and G^t as a maximum likelihood optimization problem and then learn the labels Y^t in the target domain with standard classifiers as an evaluations of the cross-domain network representations.

In this chapter, let $f : V \to \mathbb{R}^d$ be the mapping function from nodes to feature representation, where d refers to the lower-dimensions of our representation, fs are specially designed for the source network and the target network respectively, i.e., $f^s : V^s \to \mathbb{R}^d$ and $f^t : V^t \to \mathbb{R}^d$. As a sampling strategy, we define a neighborhood of nodes $N_S(u) \subset V$ for every node in the source network and in the target network, where $N_S(u^s) \subset V^s$, $u^s \in V^s$, $N_S(u^t) \subset V^t$ and $u^t \in V^t$. By predicting the latent feature

space $\mathbb{R}^d$, our proposed framework of FTLSIN can be applied to any (un)directed and (un)weighted network across domains.

3.3 Large-scale Information Network Structures Transfer Framework

3.3.1 Skip-gram in FTLSIN

The FTLSIN, as shown in Figure 3.1, learns random walks by Skip-gram and outputs the network representations from the input of networks in source domain and target domain, respectively. Skip-gram (Mikolov et al., 2013) is a language model exploiting word orders in a sequence and assuming that words closer are statistically more dependent or related. We employ the Skip-gram architecture to FTLSIN which treats the nodes in a sequence and fully uses the structures to make network analysis.

Given a current node u^s in the source network within a certain window, we have a FTLSIN Skip-gram for source networks by maximizing the following log-likelihood function of f^s in observing a neighborhood of $N_S(u^s)$:

$$\max_{f^s} \sum_{u^s \in V^s} \log Pr(N_S(u^s)|f^s(u^s)) \tag{3.1}$$

Given a node u^t in the target network with a certain window, we have a FTLSIN Skip-gram for the target network by maximizing the following log-likelihood function of f^t in observing a neighborhood of $N_S(u^t)$:

$$\max_{f^t} \sum_{u^t \in V^t} \log Pr(N_S(u^t)|f^t(u^t)) \tag{3.2}$$

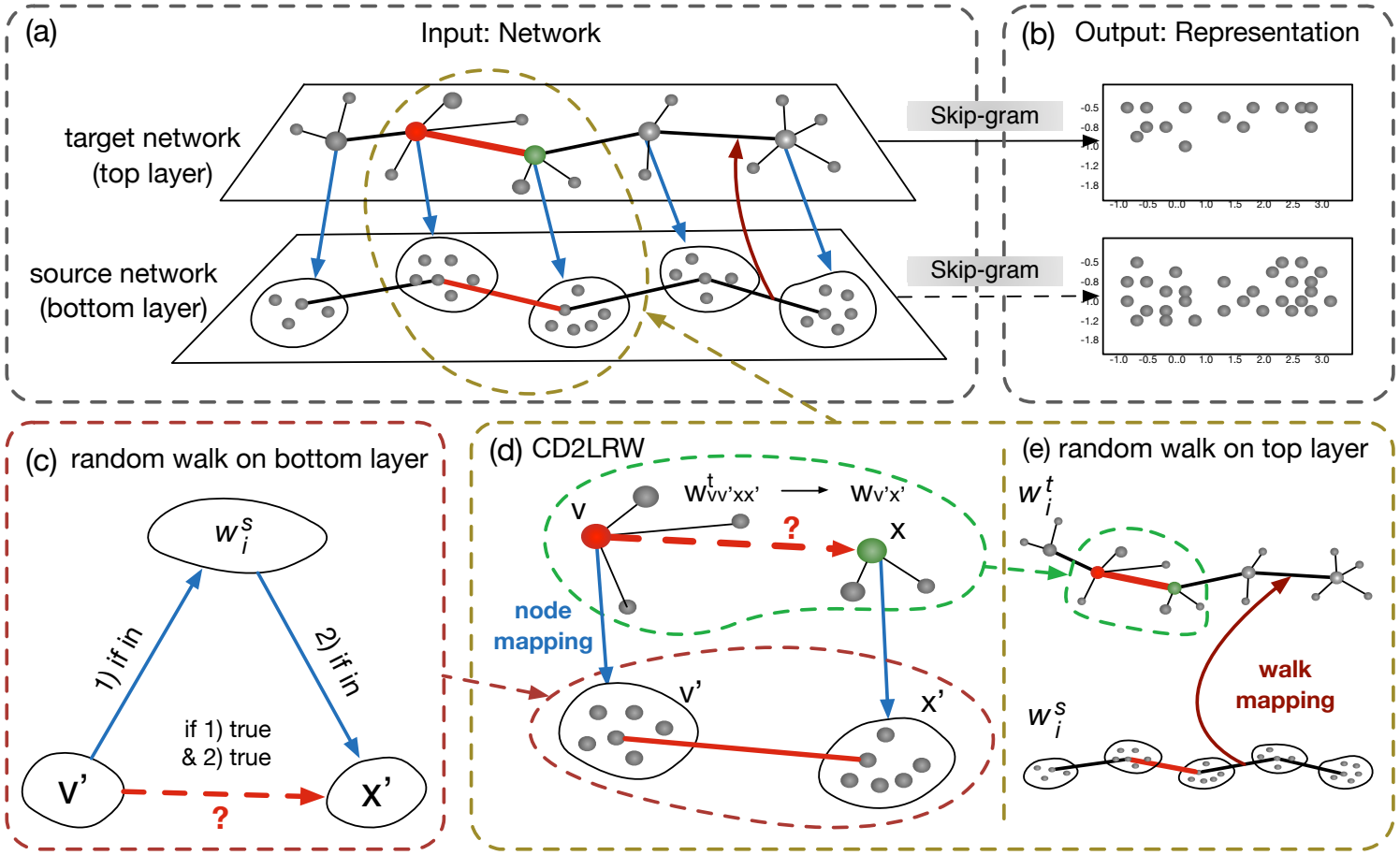


Figure 3.1 An illustration of FTLSIN

Following the standard assumptions of Node2Vec (Grover and Leskovec, 2016), conditional independence and symmetry in feature space are defined in Eqs. (3.3) and (3.4), respectively.

$$Pr(N_S(u)|f(u)) = \prod_{n_i \in N_S(u)} Pr(n_i|f(u)) \tag{3.3}$$

$$Pr(n_i|f(u)) = \frac{\exp\left(f(n_i) \cdot f(u)\right)}{\sum_{v \in V} \exp\left(f(v) \cdot f(u)\right)} \tag{3.4}$$

In our proposed FTLSIN Skip-gram (see in Algorithm 3.2), the network neighborhood strategy applied on the target network can be different from the ones on source networks. Meanwhile, the window length and optimization function f^s and f^t set in FTLSIN Skip-gram also can differ from networks.

3.3.2 Two-layer Random Walk in FTLSIN

The FTLSIN consists of a **C**ross-**D**omain **2**-**L**ayer **R**andom **W**alk (CD2LRW), which includes a bottom-layer random walk and a top-layer random walk. In Figure 3.1, the CD2LRW measures the likelihood between super-nodes $\{v', x'\}$ based on its learning of random walk w_i^s. The top-layer maps two nodes $\{v, x\}$ in the target network to the corresponding super-nodes $\{v', x'\}$ in a source network within a node mapping procedure and a walk mapping procedure. The Algorithm of CD2LRW in FTLSIN is as shown in Algorithm 3.1.

Given a random walk of node in either a source network u^s or a target network u^t, u is in a fixed length of l, i.e., the length of u^s is l^s and the length of u^t is l^t. The CD2LRW allows l^s different from l^t.

Let v_i denote the ith node in the walk, where the start node is $v_0 = u$ and all the nodes in the walk follows the distribution:

$$P(v_i^s = x^s | v_{i-1}^s = v^s) = \begin{cases} \frac{\pi_{v^s x^s}}{Z} & \text{if } (v^s, x^s) \in E^s \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

$$P(v_i^t = x^t | v_{i-1}^t = v^t) = \begin{cases} \frac{\pi_{v^t x^t}}{Z} & \text{if } (v^t, x^t) \in E^t \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

where $\pi_{v^s x^s}$ and $\pi_{v^t x^t}$ are the unnormalized transition probability between nodes v^s and x^s, and between nodes v^t and x^t; and Z is the normalizing constant.

Bottom-layer Random Walk

The design of the bottom-layer random walk is for the network representation both in the target network and in the source network. We employ parameters p and q to guide the walk by considering the network neighborhood. In order to determine which node in the neighborhood have a higher probability to be connected into the random walk, the search bias α is employed into Eqs. (3.7) and (3.9):

$$\pi_{v^s x^s} = \alpha_{pq}(t^s, x^s) \cdot w_{v^s x^s} \tag{3.7}$$

where $w_{v^s x^s}$ is the weight on link (v^s, x^s).

Algorithm 3.1: CD2LRW in FTLSIN

Input:
$G^t = (V^t, E^t)$: a target network;
$G^s = (V^s, E^s)$: a source network.

Output:
W^t: a walk set of target network.

1: $W^s \leftarrow$ Apply bottom-layer random walk to process the source network, Eqs. (3.5)-(3.8).
2: $G^s = (\mathscr{V}^s, \mathscr{E}^s, \mathscr{G}^s, \mathscr{F}^s) \leftarrow$ Samples the source network to a super-graph with super-nodes.
3: **for** w_i^s in W^s **do**
4: $\quad f_{node} \leftarrow$ Node mapping on w_i^s by Eq. (3.11).
5: $\quad w_{v'x'} \leftarrow$ Walk mapping on w_i^s and f_{node} by Eqs. (3.12)-(3.13), where $v', x' \in \mathscr{V}^s$.
6: **end for**
7: $W^t \leftarrow$ Apply bottom-layer random walk to process the target network, Eqs. (3.5)-(3.10), where $w_{v^t x^t} = w_{v'x'}$.
8: **return** W^t

$$\alpha_{pq}(t^s, x^s) = \begin{cases} \frac{1}{p} & \text{if } d_{t^s x^s} = 0 \\ 1 & \text{if } d_{t^s x^s} = 1 \\ \frac{1}{q} & \text{if } d_{t^s x^s} = 2 \end{cases} \tag{3.8}$$

where $d_{t^s x^s}$ is the shortest path between nodes t^s and x^s through node v^s.

$$\pi_{v^t x^t} = \alpha_{pq}(t^t, x^t) \cdot w_{v^t x^t} \tag{3.9}$$

where $w_{v^t x^t}$ is the weight on link (v^t, x^t).

$$\alpha_{pq}(t^t, x^t) = \begin{cases} \frac{1}{p} & \text{if } d_{t^t x^t} = 0 \\ 1 & \text{if } d_{t^t x^t} = 1 \\ \frac{1}{q} & \text{if } d_{t^t x^t} = 2 \end{cases} \tag{3.10}$$

where $d_{t^t x^t}$ is the shortest path between nodes t^t and x^t through node v^t.

Top-layer Random Walk

For the random walk in the top layer, we define a node mapping procedure and a walk mapping procedure. The node mapping procedure starts from one node $v \in V^t$ in the target network to a node set $v' \in V_i^s$ in the source network. The walk mapping procedure learns from a walk $w^s \in \{W_i^s\}$ in the source network to a new walk $w^t \in W^t$ in the target network.

Following the assumption of transfer learning (Lu et al., 2015), the scale of networks in the source domain is much larger than the scale of the network in the target domain, i.e., $|V_i^s| \gg |V^t|$ or $|E_i^s| \gg |E^t|$. The node mapping procedure links a node in target network and a set of nodes in source network. Thus, we employs the definition of super-graph and super-node to process the node mapping procedure. Specifically, the node set v' is denoted as a super-node.

A super-graph (Guo and Zhu, 2014) is represented as $G = (\mathscr{V}, \mathscr{E}, \mathscr{G}, \mathscr{F})$, where $\mathscr{V}$ is a finite set of graph-structure nodes. $\mathscr{E} \subset \mathscr{V} \times \mathscr{V}$ denotes a finite set of links, and $\mathscr{F} : \mathscr{E} \to \mathscr{G}$ is an injective function from $\mathscr{E}$ to $\mathscr{G}$, where $\mathscr{G}$ is the set of single-attribute graphs. A node in the super-graph, represented by a single-attribute graph, is called a super-node.

Algorithm 3.2: Skip-gram in FTLSIN

Input:

W^t: a target random walk set learning from Algorithm 3.1.

Output:

$f^t : V^t \to \mathbb{R}^d$: an optimized mapping function for FTLSIN.

1: $f^{t(0)} \leftarrow$ Initialize the target network mapping function.
2: **for** w_i^t in W^t **do**
3: $\quad f^t \leftarrow$ Apply Eq. (3.2)-(3.4) to optimize f^t.
4: **end for**
5: **return** f^t

As above, our node mapping procedure measures the likelihood of a node in the target network and a super-node in the source network, i.e., $f_{node} : v^t \to V_i^s$.

$$f_{node} = \begin{cases} 1 & \text{if } \deg(v^t) = \deg(V_i^s) \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

In the walk mapping procedure, walk set of target network W^t is jointly determined by the node mapping function f_{node} and weighted random walk kernel (Guo and Zhu, 2014) on super-graph in source network. The walks over the source network $W^s = \{w_i^s\}$ is naturally within a super-graph. The links forming a w_i^s links two super-nodes, as shown in Figure 3.1 (d).

Within the top-layer random walk, an link weight in a target walk ($w_{v'x'}$ in w_i^t) is formed by two terms in Eq. (3.12). The former term is contributed by the virtual weight in the target network $w_{v''x''}$, and the latter term is contributed by the learning weight from a walk mapping $w^t_{vv'xx'}$.

$$w_{v'x'} = \beta \cdot w^t_{vv'xx'} + (1-\beta) \cdot w_{v''x''} \tag{3.12}$$

where $\beta = |V^t|/(|V_i^s| + |V^t|)$.

$$w^t_{vv'xx'} = \max_{f_{walk}} \sum_{W^s_{w_i} \in \mathscr{W}^s} \sum_{\substack{v^s \in v', \\ x^s \in x'}} \log P(x^s|v^s) \tag{3.13}$$

where $P(x^s|v^s) = 1/d_{v^s x^s}$.

3.4 Experiments

3.4.1 Datasets

We select two academic citation networks as the datasets. Both of them are for the multi-class classification problem. Nodes are denoted as papers in these networks.

Table 3.1 FTLSIN dataset statistics

Domain	Network	Num. of Nodes	Num. of Links	Num. of Labels
Source	DBLP	60,744	52,890	4
Target	M10	10,310	77,218	10

- **DBLP**[1] **dataset** (source network), which consists of bibliography data in computer science has been used widely in network and graph analysis (Wu et al., 2014, 2018). Each paper may cite or be cited by other papers, from which naturally forms a citation network. The network in this dataset abstracts a list of conferences from four research areas, i.e., database, data mining, artificial intelligence and computer vision.

[1] http://arnetminer.org/citation (V4 version is used)

- **CiteSeer-M10**[2] **dataset** (target network) is a subset of CiteSeerX data which consists of scientific publications from ten distinct research areas, i.e., agriculture, archeology, biology, computer science, financial economics, industrial engineering, material science, petroleum chemistry, physics and social science.

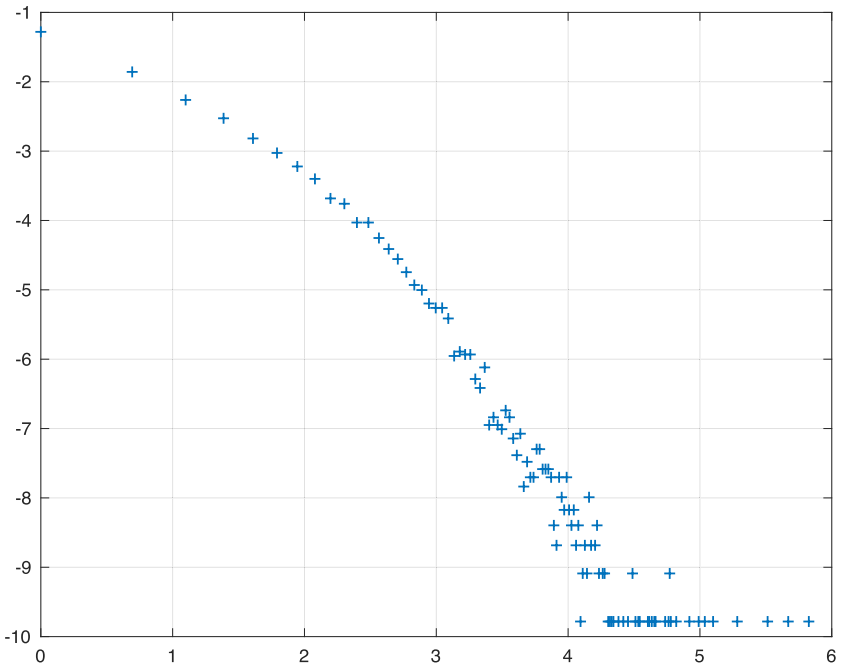
3.4.2 Setups

Our experiment evaluates the latent feature representations on standard supervised learning task: linear SVM classification. We choose the linear classifier instead of non-linear classifier or sophisticated relational classifiers in order to reduce the impact of complicated learning approaches on the classification performance. For evaluations, we randomly partition the dataset in the target domain into two non-overlapping sets for training and testing by nine groups of training percents, $\{0.1, 0.2, \cdots, 0.9\}$. We repeat the above steps for ten times and thus receive ten copies of the training data and the testing data. The reported experiment results are the average of the ten runs and their variance.

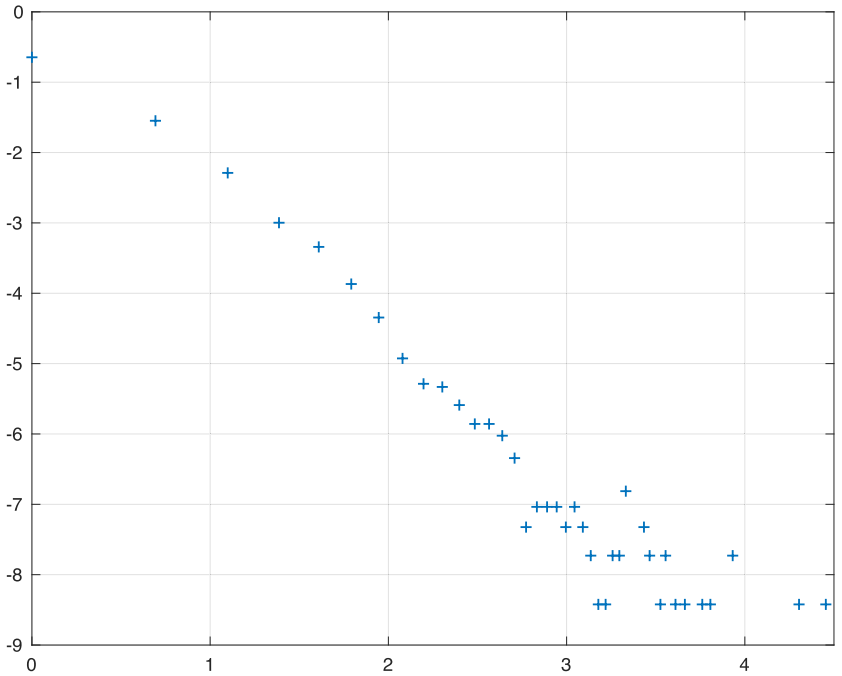
3.4.3 Baselines

Figure 3.2 and Figure 3.3 show power law distributions (Adamic and Huberman, 2000) on the experiment datasets and their random walks, which obey the assumptions of the random walk that if the degree distribution of a connected graph follows a power law distribution, the frequency which the nodes appear in the short random walks will also follow a power law distribution (Perozzi et al., 2014).

[2]http://citeseerx.ist.psu.edu/

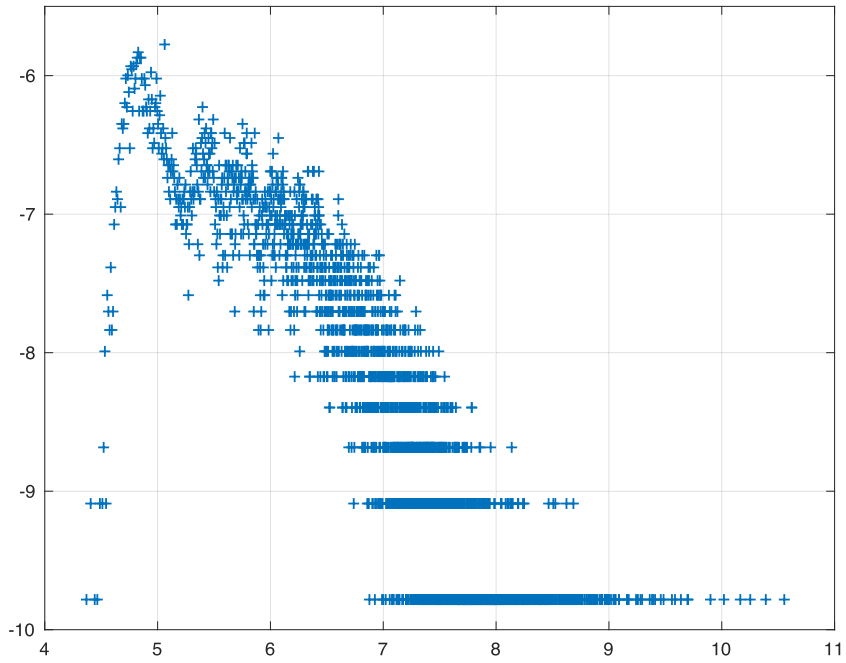


(a) Source Network: DBLP

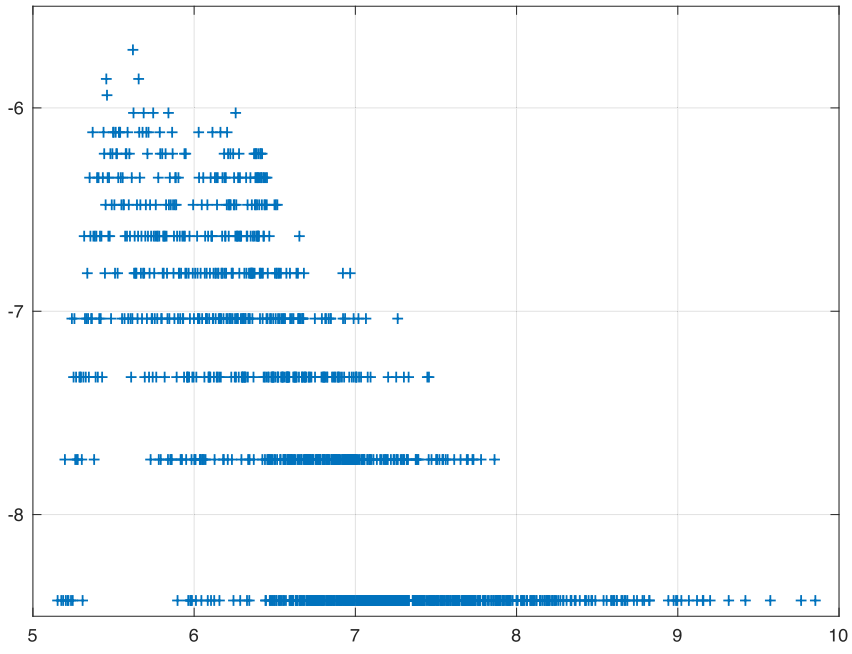


(b) Target Network: M10

Figure 3.2 Power-law distributions of the networks



(a) Source Network: DBLP



(b) Target Network: M10

Figure 3.3 Power-law distributions of the random walks

We implement the following random walk based domain-specific network representation algorithms for comparison. The baselines are applied with the FTLSIN Skip-gram for source networks in Eqs. (3.1), (3.3)-(3.4).

- **DeepWalk** (Perozzi et al., 2014) learns d-dimensional feature representations by simulating uniform random walks. The sampling strategy in DeepWalk can be seen as a special case of FTLSIN with bottom-layer random walk in $p = 1$ and $q = 1$.

- **LINE** (Tang et al., 2015) learns d-dimensional feature representations in two separate phases. In the first phase, it learns $d/2$ dimensions by BFS-style simulations over immediate neighbors of nodes. In the second phase, it learns the next $d/2$ dimensions by sampling nodes strictly at a 2-hop distance from the source nodes.

- **Node2Vec** (Grover and Leskovec, 2016) learns d-dimensional feature representations by BFS-style simulations over immediate neighbors of nodes. The sampling strategy in Node2Vec is also a special case of FTLSIN with bottom-layer random walk in $p = 1$ and $q = 1$.

3.4.4 Parameters Setting

The parameter settings used for FTLSIN are in line with typical values used for DeepWalk, LINE and Node2Vec. Specially for source and target networks, we set the dimensions of feature representation at $d = 128$, set the walk length at $l = 80$, set the number of walks of every source node at $k = 10$, and set the window size at $r = 10$. In this way, the total number of walks over a input network is $w = SampleSize \times k$,

and the shape of walk sets are in $w \times l$. The parameters of search bias α is set at $p = 1$ and $q = 1$.

3.4.5 Result Analysis

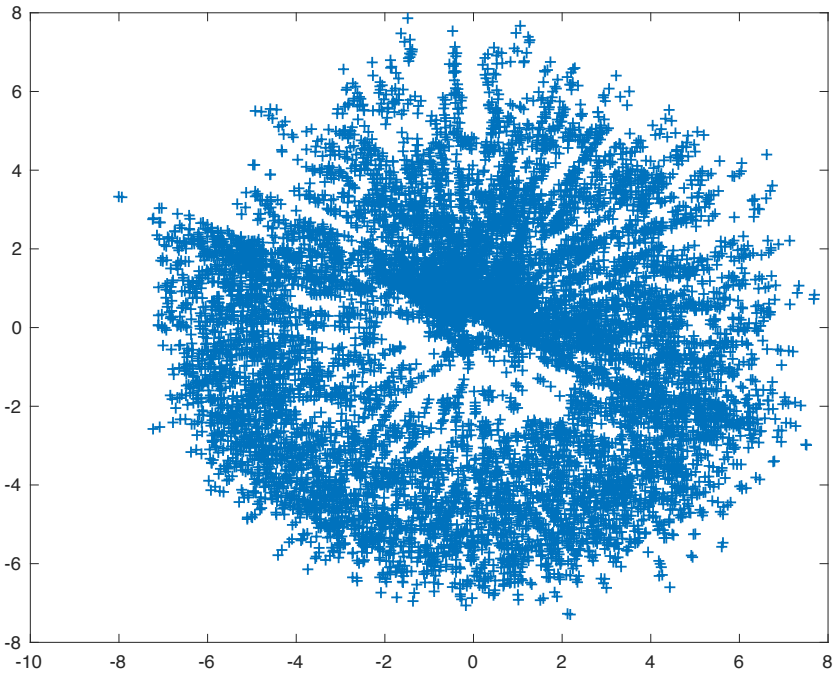
The node feature learning by network representations are input to a one-against-all linear SVM classifier (Hsu and Lin, 2002). We use Macro-F1 and Micro-F1 for comparing performance and the results are shown in Table 3.2. These two measures are popular just like the classification accuracy performance in data mining areas (Wu et al., 2012).

Representation Analysis. Figure 3.4 (a) illustrate the feature spaces of DBLP by FTLSIN bottom-layer random walk, Figure 3.4 (b) illustrate the feature spaces of M10 by FTLSIN two-layer random walk. These two illustrations show almost the same distributions in feature spaces and get good mappings in a low dimension than PCA (Figure 3.5), LLE (Figure 3.6) and Laplacian (Figure 3.7) based network representations.

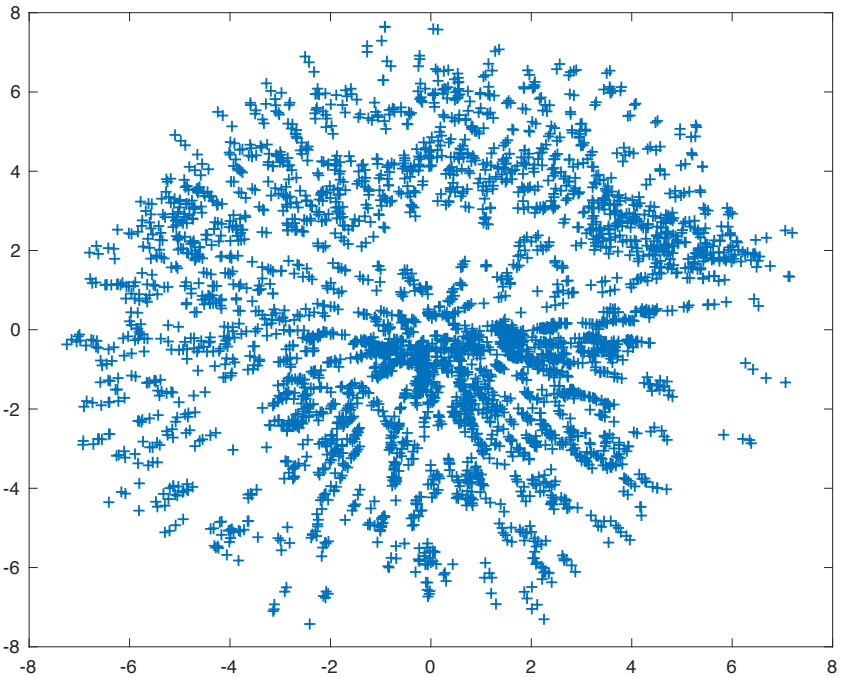
Effectiveness of search priority in random walks. In Table 3.2, DeepWalk and LINE show the worse performance than FTLSIN and Node2Vec, which can be explained by its inability to reuse samples, a feat that can be easily done using the random walk. The outstanding of Node2Vec among benchmark models indicates the exploration strategy is much better than the uniform random walks learned by DeepWalk and LINE. Meanwhile, the poor performance of DeepWalk and LINE is mainly because the network structure is rather sparse, with noises and only contains limited information. FTLSIN and Node2Vec are both good performing on M10

Table 3.2 FTLSIN classification results on target domain network of M10

	Model	Statistic	10%	20%	30%	40%	50%	60%	70%	80%	90%
Micro-F1	DeepWalk	mean	0.1758	0.1833	0.1897	0.2049	0.2051	0.2216	0.2236	0.2420	0.2431
		variance	0.0086	0.0100	0.0122	0.0126	0.0128	0.0111	0.0170	0.0133	0.0220
	LINE	mean	0.2338	0.2362	0.2623	0.2821	0.3269	0.3244	0.3561	0.3508	0.4128
		variance	0.0102	0.0170	0.0110	0.0141	0.0150	0.0087	0.0193	0.0184	0.0486
	Node2Vec	mean	0.3342	0.4166	0.4714	0.5213	0.5550	0.5843	0.6216	0.6353	0.6535
		variance	0.0099	0.0110	0.0153	0.0127	0.0176	0.0092	0.0215	0.0115	0.0324
	FTLSIN	**mean**	**0.3530**	**0.4374**	**0.4980**	**0.5519**	**0.5876**	**0.6179**	**0.6580**	**0.6712**	**0.6967**
		variance	**0.0043**	**0.0049**	**0.0063**	**0.0050**	**0.0065**	**0.0072**	**0.0074**	**0.0078**	**0.0183**
Macro-F1	DeepWalk	mean	0.2523	0.2667	0.2768	0.2945	0.2935	0.3077	0.3101	0.3294	0.3359
		variance	0.0117	0.0051	0.0072	0.0120	0.0081	0.0086	0.0158	0.0123	0.0220
	LINE	mean	0.3160	0.2984	0.3421	0.3596	0.4070	0.4275	0.4498	0.4277	0.4773
		variance	0.0113	0.0127	0.0144	0.0249	0.0382	0.0548	0.0383	0.0302	0.0486
	Node2Vec	mean	0.4326	0.4748	0.5338	0.5900	0.6092	0.6388	0.6866	0.6981	0.6568
		variance	0.0147	0.0156	0.0153	0.0153	0.0290	0.0314	0.0202	0.0572	0.0261
	FTLSIN	**mean**	**0.4662**	**0.5094**	**0.5747**	**0.6354**	**0.6557**	**0.6863**	**0.7377**	**0.7488**	**0.6908**
		variance	**0.0057**	**0.0120**	**0.0121**	**0.0107**	**0.0128**	**0.0147**	**0.0143**	**0.0153**	**0.0200**

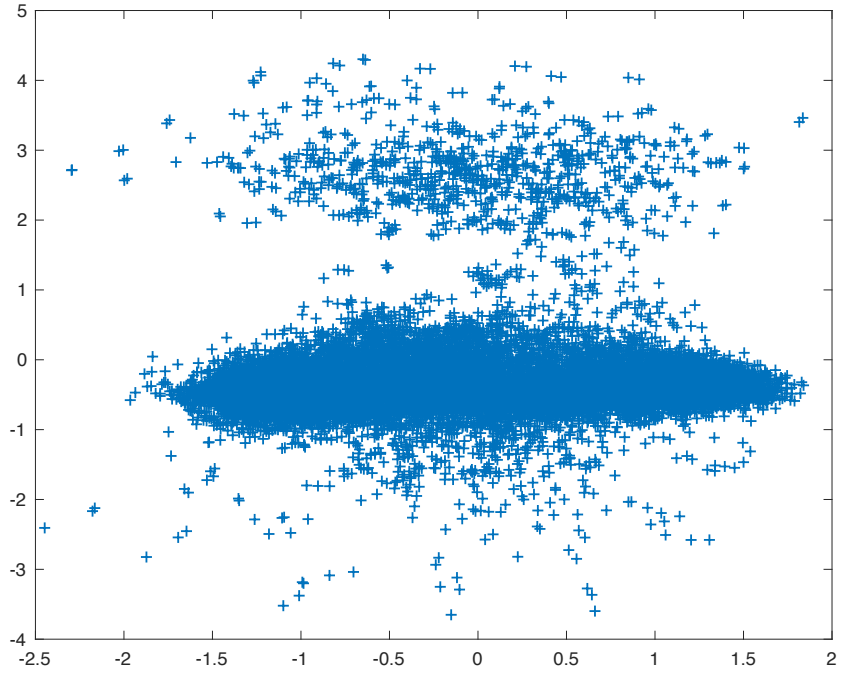


(a) FTLSIN on DBLP

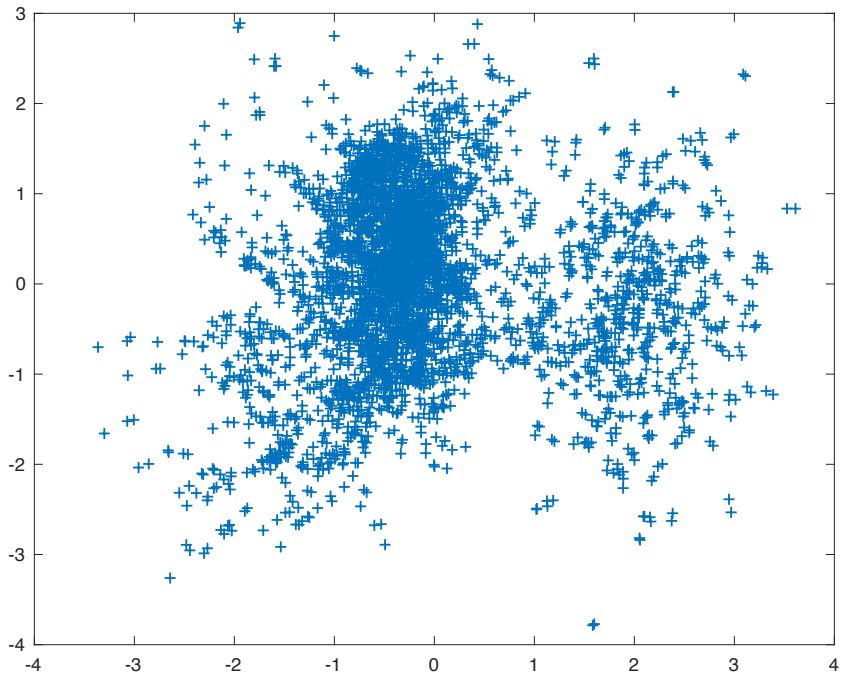


(b) FTLSIN on M10

Figure 3.4 Illustrations of 2-dimensional network representation by FTLSIN

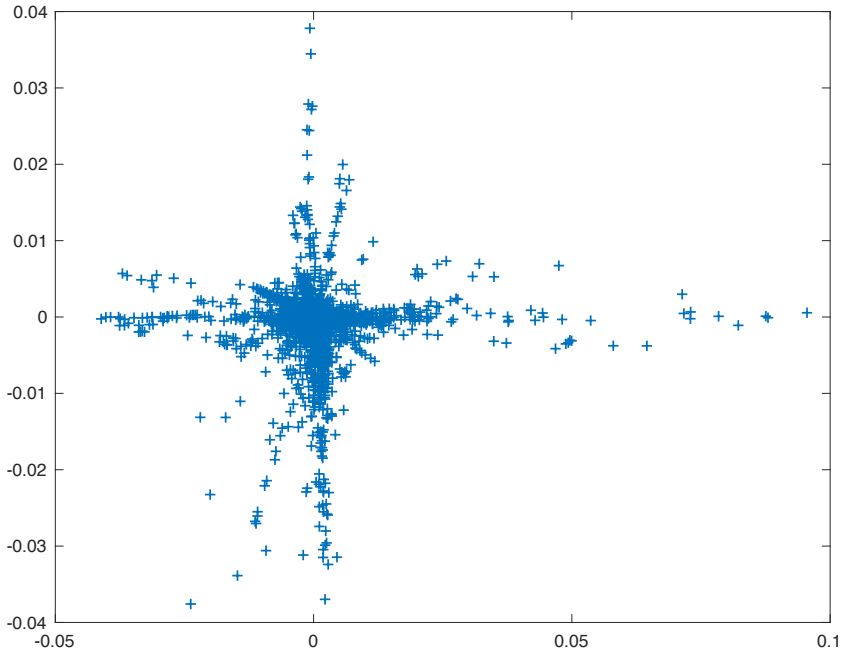


(a) PCA on DBLP

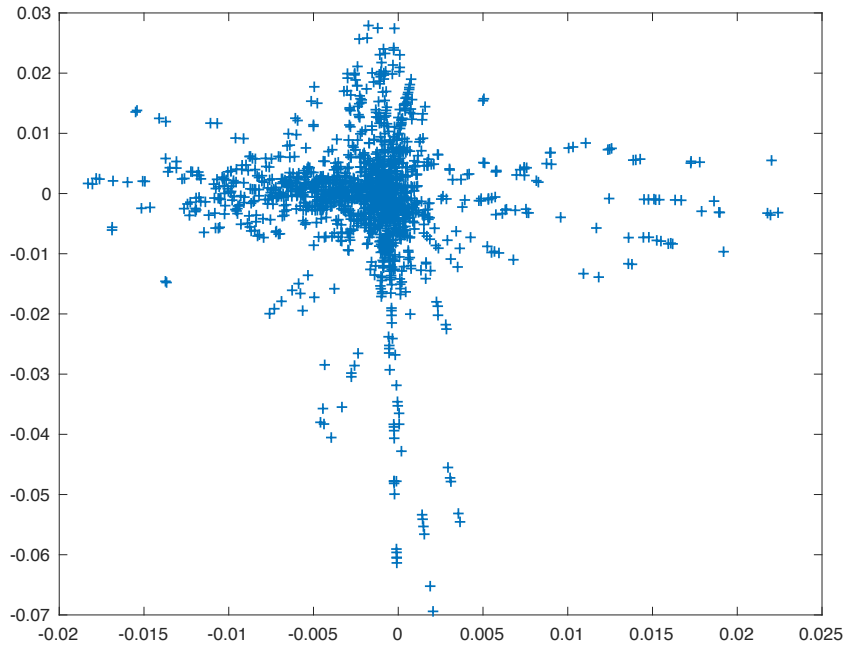


(b) PCA on M10

Figure 3.5 Illustrations of 2-dimensional network representation by PCA

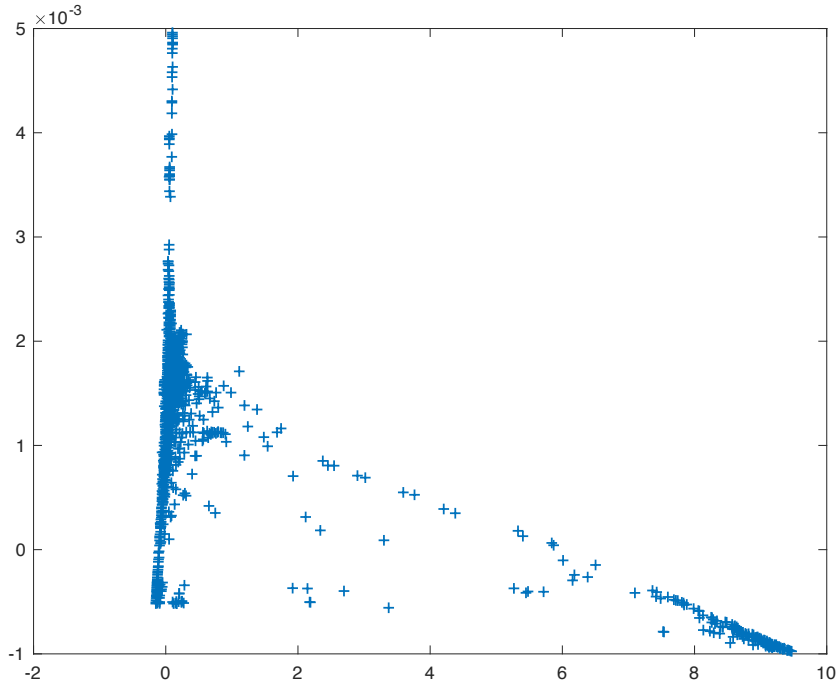


(a) LLE on DBLP

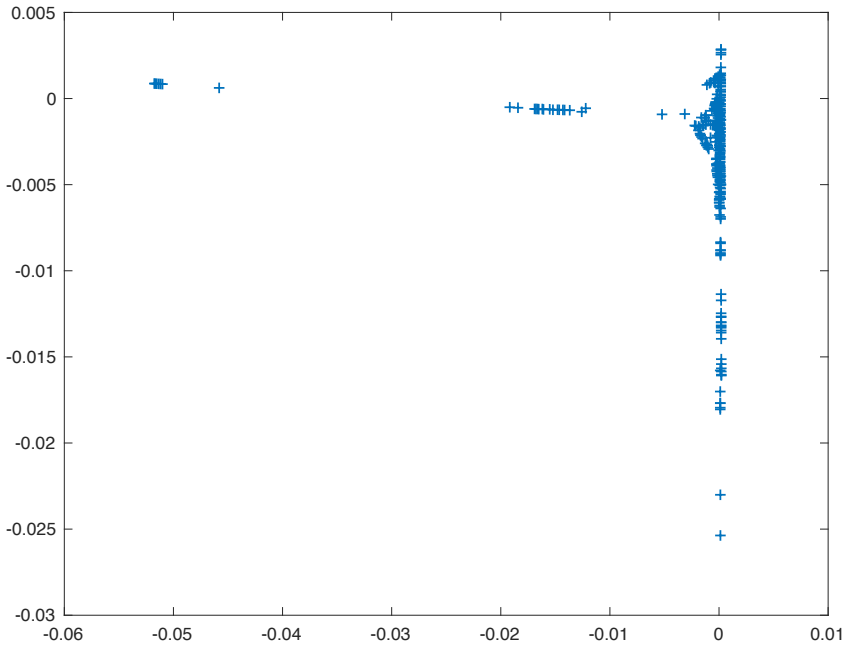


(b) LLE on M10

Figure 3.6 Illustrations of 2-dimensional network representation by LLE



(a) Laplacian on DBLP



(b) Laplacian on M10

Figure 3.7 Illustrations of 2-dimensional network representation by Laplacian

network with above advantages, as parameter of search bias α adds the flexibility in exploring local neighborhoods prior to global network.

Importance of information from source domain. Table 3.2 shows that FTLSIN outperforms the domain-specific benchmark models, which uses topological information from the source domain to learn the network representation in the target domain. When we add a top-layer in FTLSIN, the information in the source network are transferred to the source network by adjusting the weights on the links of the target network.

3.5 Summary

In this chapter, the proposed FTLSIN offers a solution for the scenario in network representation that transferring structures across networks with CD2LRW. The FTLSIN effectively improves the performance of latent feature learning in large-scale citation networks as shown in the experiment. Meanwhile, it reduces learning difficulties of data sparsity and noises. Future works include FTLSIN with multiple labels and deep network representation.

Chapter 4

Cross-domain Network Representations based on Random Walk Transfer

4.1 Introduction

Typical random walk-based network representation algorithms, such as DeepWalk (Perozzi et al., 2014), learn sequences of nodes to model the network structures of deep features. However, these domain-specific network representation algorithms still limit the performance of network representations. Random walk-based network representations are highly dependent on the sliced window for sampling the nodes. When the distance between two nodes that share the same content is larger than the size of window, the random walk fails in this representation round and turns to the next round of sampling. Although the missing representations are covered by a vast amount of sampling in some related works, the process increases computational

complexity (Perozzi et al., 2014). Most previous works are domain-specific random walk-based network representation algorithms that cannot overcome the gap that the network structure is too sparse to obtain a satisfying latent feature space for further machine learning tasks. The sparsity may be caused by the smaller scale of the network which is generating the nodes and links.

In this chapter, we propose a novel CDNR based on FTLSIN proposed in (Xue et al., 2018). CDNR is different from previous deep transfer learning approaches for cross-domain graph-structured data, i.e., context enhanced inductive representation (Hamilton et al., 2017) and intrinsic geometric information transfer (Lee et al., 2017). CDNR has the advantage of using a sampling strategy, as in domain-specific random walk-based network representations, and focuses on large information network representations which show special characteristics that a graph does not. By addressing the challenges of cross-domain relationships and cross-domain random walk constructions, CDNR makes the following contributions:

- **Contribution 1:** The novel algorithm of CDNR is proposed to enable knowledge sharing across related domains by link prediction on the node mapping procedure. By generating new links between two independent networks, CDNR calculates the similarities and determines the relationships across domains.

- **Contribution 2:** The proposed random walk strategy works synchronously for the source and target domains. In CDNR, knowledge is transferred from the source domain to the target domain in a walk mapping procedure. The network in the target domain leverages shared knowledge to make representations.

- **Contribution 3:** An extension of CD2LRW is proposed to complete the contributions of 1 and 2 in an unsupervised feature-learning way. The output of CD2LRW is the set of random walks of the target network, from which CDNR learns the network representation based on the random walks.

4.2 Problem Statement

Let $G = (V, E)$ be a given network, where V represents the members of the network known as nodes and E represents their connections known as links, $E \subseteq (V \times V)$. In the cross-domain problem, we have a $\mathscr{D}^s$ and a $\mathscr{D}^t$. The domain (Pan and Yang, 2010) is denoted as $\mathscr{D} = \{\mathscr{X}, P(X)\}$, where $\mathscr{X}$ is the feature space and $P(X)$ is the marginal probability distribution that $X = \{x_1, \cdots, x_n\} \in \mathscr{X}$. In the domain-specific scenario, $\mathscr{X}^s$ and $\mathscr{X}^t$ are represented by $G^s = (V^s, E^s)$ and $G^t = (V^t, E^t)$ respectively, because $P^s(X^s)$ and $P^t(X^t)$ are different in $P(\cdot)$ or in X. In CDNR, $\mathscr{D}^s = \{\mathscr{X}^s, P^s(X^s)\}$ and $\mathscr{D}^t = \{\mathscr{X}^t, P^t(X^t)\}$. We target the network representation on $\mathscr{X}^t$ which is jointly represented by $G^t = (V^t, E^t)$ and the shared knowledge from $G^s = (V^s, E^s)$. CDNR introduces the structural knowledge from $\mathscr{D}^s$ to $\mathscr{D}^t$ and improves the network representations in $\mathscr{D}^t$ when the per-node scale of G^t is smaller than that of G^s. We first implement a domain-specific latent feature-learning procedure from the topology structures of G^s as a maximum likelihood optimization problem to generate a set of random walks in the bottom layer of CD2LRW. $N_S(u^s) \subset V^s$ is designed as a network neighborhood of node u^s where $u^s \in V^s$ to determine the neighborhood network structures of G^s. A neighborhood sampling strategy (S) with standard biased random walk is generated for $N_S(u^s)$. The top layer of CD2LRW is then designed to make the cross-domain structural

knowledge transfer. Let $f: V^t \rightarrow \mathbb{R}^d$ be the mapping function from nodes to feature representations for the target domain network, where d denotes the lower-dimensions of the latent feature space. The representation outputs of CDNR will be evaluated by a standard classifier in the target domain.

4.3 CDNR with CD2LRW

CDNR extends domain-specific random walks to cross domains and assumes that networks across domains follow the power-law distribution. CD2LRW contributes CDNR under the Skip-gram framework and is optimized by maximum likelihood. As shown in Figure 4.1, CD2LRW first makes a domain-specific random walk in the bottom layer and involves the target network in the top layer. A cross-domain node-mapping procedure then predicts the links from the top layer to the bottom layer and a cross-domain walk-mapping procedure transfers the knowledge from the bottom layer to the top layer. Lastly the random walk learning for the target network is processed in the top layer using the transferred knowledge and its own network structure information. The example of the four steps are as follows.

- **Step 1** generates random walks in the bottom layer, which is the source domain knowledge prepared for cross-domain walk mapping.

- **Step 2** finds the super-nodes V' in the source network using the cross-domain node mapping function Φ_{node}. v^t in the target network and V' in the source network correspond to the close degrees, $Deg(v^t) \sim Deg(V')$. Each pair of (v^t, V') is linked with $e^*_{v^t V'}$.

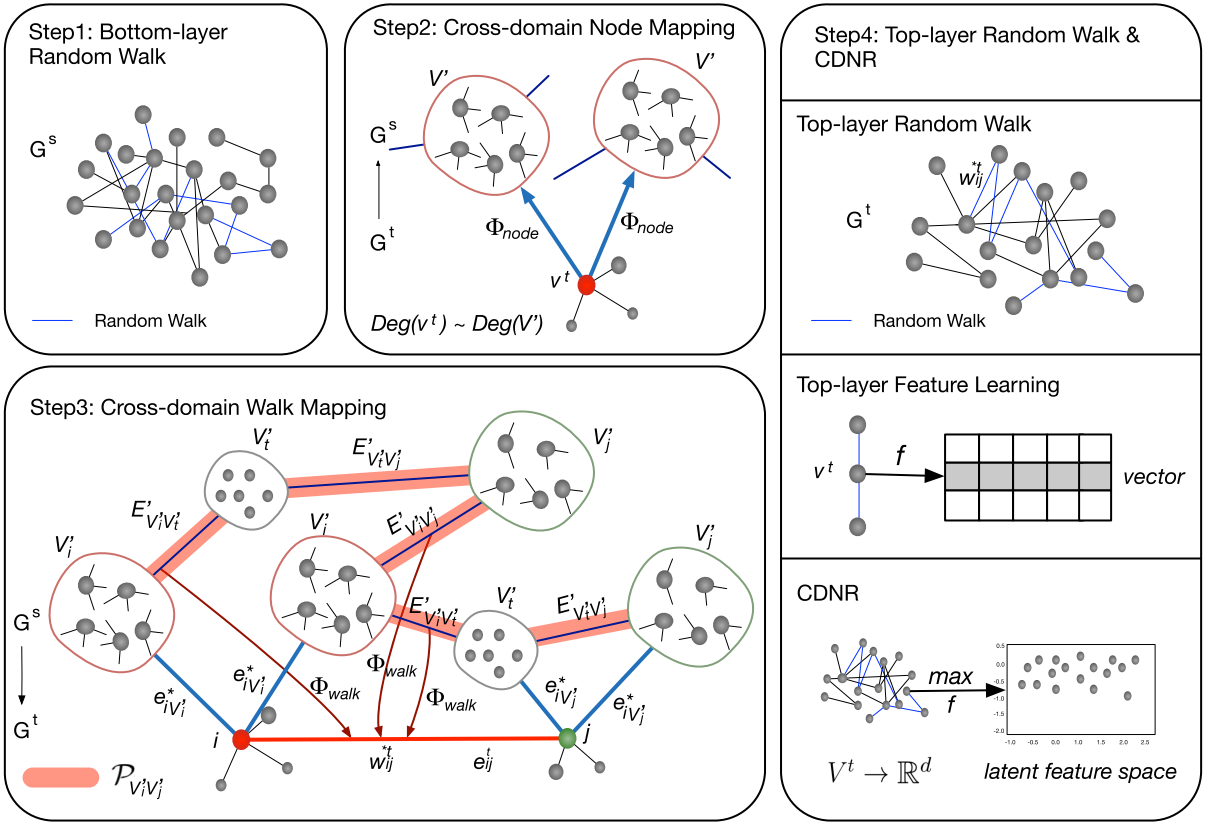


Figure 4.1 An illustration of CDNR in four steps

- **Step 3** transfers the source network random walk knowledge via $e^*_{v'V'}$ and saves the knowledge on the updated weight w^*_{ij}. This procedure Φ_{walk} offers more structural information for the random walk in the target network.

- **Step 4** conducts the CDNR based on the top-layer random walk for the target network. Each node is embedded in a vector format and all nodes are mapped to a latent feature space optimized by the maximized likelihood. The detailed steps in CDNR are explained in this section.

4.3.1 Random Walk Sampling Strategies: Domain-specific

DeepWalk (Perozzi et al., 2014) trains a neural language model on the random walks generated by the network structure. After denoting a random walk that starts from a root node, DeepWalk slides a window and maps the central node to its representation. Hierarchical Softmax factors out the probability distributions corresponding to the random walk and the representation function is updated to maximize the probability. DeepWalk has produced promising results in dealing with sparsity in scalable networks, but has relatively high computational complexity for large-scale information networks. LINE, Node2Vec and Struc2Vec are the other structure-based network representation algorithms that improve the performance of DeepWalk. LINE (Tang et al., 2015) preserves both the local network structure and the global network structure by first-order proximity and second-order proximity respectively and can be applied to large-scale deep network structures that are directed, undirected, weighted and unweighted. Node2Vec (Grover and Leskovec, 2016) explores the diverse neighborhoods of nodes in a biased random walk procedure by employing classic search strategies. Struc2Vec (Ribeiro et al., 2017) encodes structural sim-

ilarities and generates the structural context for nodes using random walks. The above-mentioned works has contributed to network analysis by modeling a stream of short random walks. In this chapter, we propose CDNR which employs biased random walk sampling strategies to learn network structures based on those earlier works.

4.3.2 Power-law Distribution: The Assumption

The power-law distribution exists widely in real-world networks. It is a special degree distribution that follows $P(deg) \sim deg^{-a}$, where deg is a node degree and a is a positive constant (Newman, 2005b). A network that follows the power-law distribution is also regarded as a scale-free network with the scale invariance property (Barabási, 2009). The social networks, biological networks and citation networks being discussed in this chapter are observed to be scale-free in nature (Barabási and Pósfai, 2016). In *log-log* axes, the power-law distribution shows a linear trend on the slope ratio of $-a$ (Figures 4.2-4.7), which reflects that numerous links connect small degree nodes and will not change regardless of network scale (Adamic and Huberman, 2000). It has been observed in (Perozzi et al., 2014) that if a network follows the power-law distribution, the frequency at which a node undertakes in a short random walk will also follow the same distribution. Meanwhile, random walks in power-law distribution networks naturally gravitate towards high degree nodes (Adamic et al., 2001). The link predictions in CD2LRW are therefore leveraged on the power-law distribution as well as the distance calculation between the two independent networks across domains. The network that has small distance to the

target network is regarded as the source domain. The scale invariance property should theoretically ensure that power law-based CDNR is robust.

4.3.3 Bottom-layer Random Walk: Knowledge Preparation

The bottom-layer random walk is designed for random walk sampling in the source domain. The sampled random walks are the structural knowledge waiting to be transferred to the target domain.

The bottom-layer random walk introduces a biased random walk to efficiently explore diverse neighborhoods and sample the nodes along the shortest path[1]. Suppose $\mathscr{W}^s = \{W^1_{v^s}, W^2_{v^s}, \cdots, W^k_{v^s}\}$ is a set of stochastic random walks in the bottom layer that are rooted at node $v^s \in V^s$ where k is the given number of random walks. For each random walk W_{v^s}, we generate a length in l and the nodes in W_{v^s} within the neighborhood are searched along the shortest path. For example, let c denote the ith node and $c-1$ denote the $i-1$th node in W_{v^s} where $v^s = c_0$. The probability that node x is involved in W_{v^s} as $c+1$ is based on the neighborhood $N_S(c)$. If link $e_{xc} \in E^s$, $P(x|c) = \frac{\pi_{xc}}{Z}$, otherwise it is 0. Z is the partition function that ensures a normalized distribution; see definition in (Bengio et al., 2013). π_{xc} is guided by the search bias parameter α and only works in each $N_S(c)$. $\pi_{xc} = \alpha_{pq}(x,c) \cdot w_{xc}$, where w_{xc} refers to the weight on link e_{xc} and $\alpha_{pq}(x,c)$ follows the following search rules: if the shortest path $d_{xc-1} = 0$, $\alpha_{pq}(x,c) = 1/p$; if $d_{xc-1} = 1$, $\alpha_{pq}(x,c) = 1$; and if $d_{xc-1} = 2$, $\alpha_{pq}(x,c) = 1/q$. In this way, p describes the case in which the nodes are revisited by a random walk so that the sampling strategy on random walks is computationally efficient, especially for real-world large-scale networks.

[1]The shortest path is a path between two nodes for which the sum of its links weights is minimized.

4.3.4 Top-layer Random Walk: Knowledge Transfer

CDNR deals with the scenario in which the training sample in the target network is insufficient to make a good network representation. In the procedure described in the bottom-layer random walk, the scale of the sampled random walks is small compared with the scale of the nodes. The top-layer random walk is therefore proposed to counter this obstacle under the common framework of transfer learning (Lu et al., 2015). The cooperation of CD2LRW is achieved by a cross-domain node mapping procedure and a cross-domain walk mapping procedure.

Cross-domain Node Mapping

The source network candidates must first meet the following requirements:

- **Requirement 1** $|V^s| > |V^t|$: the scale of the nodes should be larger than the node scale of the target network; and
- **Requirement 2** $\langle deg^s \rangle > \langle deg^t \rangle$: the average node degree[2] should be larger than the average node degree in the target network.

Candidates are then evaluated in a super-graph structure (Guo and Zhu, 2014) which is formed by super-nodes, as shown in Figure 4.1.

Definition 4.1. Super-node in Source Domain A super-node is a sub-structure of the original source network. Denoting $\mathscr{V} = \{V', E' | V' \subseteq V^s, E' \subseteq E^s\}$, a super-node V' consists of a group of nodes $\{v^s\}$ from the original network that share the same degree and E' are the links appearing in the sub-structure of the original network.

[2]The average node degree is the mean of the degrees of all nodes in the network.

Definition 4.2. Super-graph in Source Domain A super-graph is formed by the super-nodes as $G^s = (\mathscr{V}, \mathscr{E}, \mathscr{G}, \mathscr{F})$, where $\mathscr{E}$ denotes a finite set of links as $\mathscr{F} : \mathscr{G} \rightarrow \mathscr{E}$, and $\mathscr{G} = \{G' = (V', E')\}$.

To map a large-scale source domain network to a small-scale target domain network, a cross-domain node mapping procedure is implemented by predicting links, starting from the nodes in the target domain network to the super-nodes in the source domain network. We attempt to pair each node $v^t \in V^t$ with more than one super-node $V' \in \mathscr{V}$ with an link $e_{v^t V'} \in E^*$ and a weight $w_{v^t V'} \in W^*$, where E^* are the predicted links that arrange the knowledge transfer paths with various weights W^* to control how much knowledge should be transferred from the source domain to the target domain. The node mapping function is as follows:

$$\Phi_{node} : (V^t, \mathscr{V}) \rightarrow E^* \tag{4.1}$$

For each pair of (v^t, V'),

$$e_{v^t V'} = \begin{cases} 1 & \text{if } w_{v^t V'} > 0 \\ 0 & \text{if } w_{v^t V'} = 0 \end{cases} \tag{4.2}$$

$$w_{v^t V'} = \frac{\min(Deg(v^t), Deg(V'))}{\max(Deg(v^t), Deg(V'))} \tag{4.3}$$

where $Deg(v)$ denotes the degree of node v.

Denoting the node degree $Deg(v^t)$ in the target network ranges from 1 to $\max(deg^t)$ where $|deg^t| = n_{deg^t}$ and the node degree $Deg(v^s)$ in the source network ranges from 1 to $\max(deg^s)$ where $|deg^s| = n_{deg^s}$, there are three possible cases in cross-domain node mapping.

- **Case 1:** If $n_{deg^s} = n_{deg^t}$, only one V' maps to v^t. $Deg(V^t)$ and $Deg(\mathscr{V})$ are ranked in decreasing order, respectively. v^t finds the V' that are in the same position as their own degree sequences, denoted as $Deg(v^t) \sim Deg(V')$. In this case, the cross-domain node mapping is completed directly by Eqs. (4.2) and (4.3), $E^* = \{e_{v^t V'}\}$ and $W^* = \{w_{v^t V'}\}$.

- **Case 2:** If $n_{deg^s} > n_{deg^t}$, more than one V' is mapped to v^t.

- **Case 3:** If $n_{deg^s} < n_{deg^t}$, we add $n_{deg^t} - n_{deg^s}$ super-nodes which are empty sets $V'_{null} = \varnothing$ in $\mathscr{V}$, where $Deg(V'_{null}) = 0$.

In Case 2 and Case 3, the cross-domain node mapping $\Phi_{node}\colon (V^t, \mathscr{V}) \to E^*$ is optimized by maximizing the likelihood between V^t and $\mathscr{V}$. Starting from each v^t, we weight w_i to each pair of (v^t, V'_i) according to Eq. (4.3), where $i = 1, \cdots, n_{node}$ and $n_{node} = \max(n_{deg^s}, n_{deg^t})$.

$$\max_{\Phi_{node}} \sum_{v^t \in V^t} \eta \sum_{i=1}^{n_{node}} \left[\log(C) - a_{\oplus} \log(\vec{\delta}_z \vec{w} \vec{w}^\top \vec{\delta}_z^\top) \right]_i \tag{4.4}$$

where $\vec{w} = [w_1, \cdots, w_i, \cdots, w_{n_{node}}]^\top$. $\vec{\delta}_z$ is a vector in size of n_{node} with the value of 0 or 1, which selects out the super-nodes that are $Deg(v^t) \sim Deg(V')$. Let $a_{\oplus} = \min\{a^s, a^t\}$ in which a^s and a^t are the power-law slope ratio of G^s and G^t respectively. $\eta = \frac{1}{n_{deg^t}} e^{\frac{1-n_{deg^s}^2}{n_{deg^s}}} \gamma e^{\lambda}$ controls the range of the likelihood over the global cross-domain node mapping, where γ is the parameter of V^t and λ is the parameter of $\mathscr{V}$. The optimized cross-domain node mapping results are $E^* \propto \vec{\delta}_z$ where $W^* = \vec{\delta}_z \vec{w}$ in Case 2 and Case 3.

Cross-domain Walk Mapping

The cross-domain walk mapping transfers in the random walks the source domain $\mathscr{W}^s$ generated in top-layer random walk via the predicted links E^* and the W^* learned by cross-domain node mapping, and assists the random walk learning in the target domain. The cross-domain walk mapping function is as follows:

$$\Phi_{walk}\colon (G^t, \mathscr{W}^s, E^*, W^*) \to \mathscr{W}^t \tag{4.5}$$

Corresponding to the cross-domain node-mapping procedure which learns from a super-node to a general node, the cross-domain walk mapping procedure learns from the links $\mathscr{E}$ connecting the super-nodes $\mathscr{V}$ along $\mathscr{W}^s$ to the general links E^t in G^t. The learning is therefore designed so that the transferred knowledge works on the weights of E^t.

Suppose that $e^t_{ij} \in E^t$, the weights on e^t_{ij} in the top layer of the CD2LRW are denoted as:

$$w^{*t}_{ij} = w^t_{ij} + \sum_{V'_i}\sum_{V'_j} w^*_{iV'_i} w^*_{jV'_j} \Big[\frac{1}{l_{\mathscr{P}}} \sum_{E'_{V'_t V'_{t+1}} \subseteq \mathscr{P}_{V'_i V'_j}} w^s_{V'_t V'_{t+1}}\Big] \tag{4.6}$$

where w^t_{ij} is the weight of e^t_{ij} in G^t; V'_i is one of the corresponding super-node of node i; V'_j is one of the corresponding super-node of node j; $\mathscr{P}_{V'_i V'_j}$ is the shortest path from V'_i to V'_j; $l_{\mathscr{P}}$ is the length of $\mathscr{P}_{V'_i V'_j}$; and $w^s_{V'_t V'_{t+1}}$ is the weight on $E'_{V'_t V'_{t+1}}$ that $E'_{V'_t V'_{t+1}} \in \mathscr{E}$ forms $\mathscr{P}_{V'_i V'_j}$.

We then learn the random walks on G^t in the cross-domain top layer, using the same procedure as used in the bottom-layer random walk with w^*_{ij} and optimized by the following top-layer feature learning.

4.3.5 Top-layer Feature Learning: Knowledge Representation

CDNR represents the knowledge in the target domain after CD2LRW has completed the knowledge preparation in the source domain and the knowledge transfer to the target domain. CDNR learns the latent feature space by $f\colon V^t \to \mathbb{R}^d$ in the Skip-gram framework.

Given a node u^t in the target domain with the window size r, we obtain a cross-domain Skip-gram for the G^t by maximizing the following log-likelihood function of f in observing a neighborhood of $N_S(u^t)$,

$$\max_{f} \sum_{u^t \in V^t} \log Pr(N_S(u^t)|f(u^t)) \tag{4.7}$$

Algorithm 4.1 of CDNR is formed by a CD2LRW and a Top-layer Feature Learning. The two-layer design completes the knowledge transfer from the source network to the target network. The bottom layer loads the source network and the top layer loads the target network, where the bottom layer pre-learns the useful knowledge of random walks to assist the random walk sampling in the top layer.

In summary, the main advantage of CDNR is that when the network structure lacks the information to generate good network representation, the two layers are designed to share knowledge by overcoming the challenges presented by the unbalanced scale of nodes and random walks. The proposed CD2LRW in CDNR offers efficient cross-domain mapping with a relatively low computational cost of $O(\langle deg^t \rangle |V^t|)$. The computational complexity of Top-layer Feature Learning is in line with Node2Vec of $O(\langle deg^t \rangle^2 |V^t|)$.

Algorithm 4.1: The CDNR algorithm

Input:

$G^t = (V^t, E^t)$ in the target domain and $G^s = (V^s, E^s)$ in the source domain; start node $u^t \in V^t$; and walk length l.

Output:

A latent feature space of G^t in $\mathbb{R}^d$.

CD2LRW

$\mathscr{W}^s \leftarrow$ Apply bottom-layer random walk to process random walks in G^s.
for v^t in V^t **do**
  $\Phi_{node}(v^t, V') \leftarrow$ Cross-domain node mapping by Eqs. (4.1)-(4.4).
end for
$E^* \leftarrow$ Predicted links by cross-domain node mapping.
$W^* \leftarrow$ Weights on E^*.
for e^t_{ij} in E^t **do**
  $\{i, V'_i\} \leftarrow$ Find super-nodes in G^s that $e_{iV'_i} \in E^*$.
  $\{j, V'_j\} \leftarrow$ Find super-nodes in G^s that $e_{jV'_j} \in E^*$.
  $\mathscr{P}_{V'_i V' j} \leftarrow$ Construct shortest paths between V'_i and V'_j.
  $w^s_{v'_t V'_{t+1}} \leftarrow$ Weights on $E'_{v'_t V'_{t+1}} \in \mathscr{E}$.
  $w^{*t}_{ij} \leftarrow$ Update weight on e^t_{ij} by Eq. (4.6).
end for

Top-layer Feature Learning

for u^t in G^t **do**
  $N_S(u^t) \leftarrow$ Search neighborhood of u^t.
  $f \leftarrow$ Apply Skip-gram to optimize.
end for
$\mathscr{W}^t \leftarrow$ Apply CD2LRW to process random walks in G^t by Eqs. (4.5)-(4.7).
return $\mathbb{R}^d \leftarrow$ A latent feature space of G^t by f.

4.4 Experiments

This section evaluates the effectiveness of CDNR compared to the baseline algorithms of network representations in multi-label classifications.

Table 4.1 CDNR dataset statistics

Datasets	Num. of Nodes	Num. of Links	Ave. Degree	Num. of Categories	Labels
Blog3	10,312	333,983	64.776	39	Interests
YouTube	31,703	90,082	8.199	37	Interests
Facebook	4,039	88,234	43.691	10	Groups
PPI	3,890	37,845	19.609	50	States
arXivCit-HepPh	34,546	421,578	24.407	11	Years
arXivCit-HepTh	27,777	352,807	25.409	11	Years

4.4.1 Datasets

We select six real-world large-scale networks of different kinds as the experimental datasets, consisting of three online social networks (Blog3, YouTube, Facebook), two citation networks (arXivCit-HepPh, arXivCit-HepTh) and one biological network (PPI). All of them are for the multi-class multi-label classification problem. In the online social networks, nodes represent users and the users' relationships are denoted as links. In the citation networks, papers are denoted as nodes and links describe the citations in this experiment. In the biological network, genes are denoted as nodes and links represent the relationships between the genes.

- **BlogCatalog3 (Blog3)**[3] **dataset** is a social blog directory which manages bloggers and their blogs. Both the contact network and selected group membership information is included. The network has 10,312 nodes, 333,983 undirected links and 39 different labels. Nodes are classified according to the interests of bloggers.

- **YouTube**[4] **dataset** is a social network dataset generated by users who share videos online. The labels represent groups of website users that enjoy common

[3]http://socialcomputing.asu.edu/datasets/BlogCatalog3
[4]http://leitang.net/code/social-dimension/data/youtube.mat

video genres (e.g. *anime* and *wrestling*). The network has 90,082 undirected links, 31,703 connected and labeled nodes and 37 different labels on connected nodes.

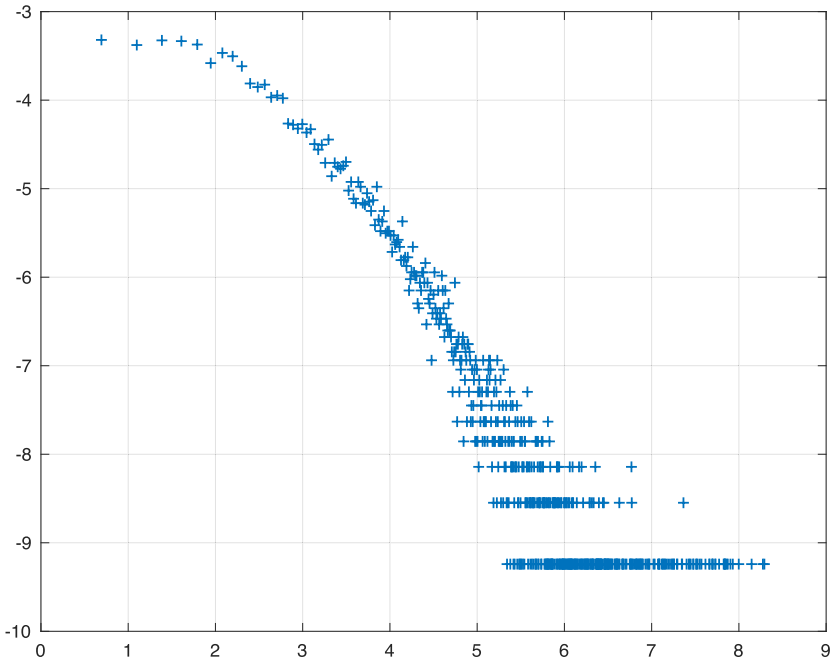
- **Facebook**[5] **dataset** consists of circles (i.e., friends lists) from Facebook. This dataset contains user profiles as node features, and circles as link features and ego networks. The network has 4,039 nodes, 88,234 undirected links and 10 different labels representing groups of users.

- **PPI**[6] **dataset** is a subgraph of the PPI network for Homo Sapiens, which obtains labels from hallmark gene sets and represents biological states. The network has 3,890 nodes, 76,584 undirected links and 50 different labels.

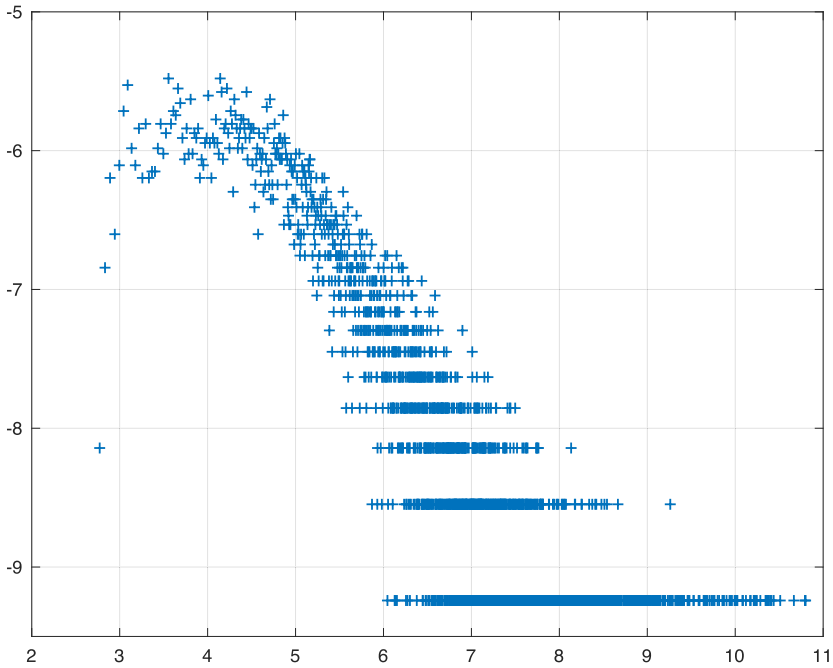
- **arXiv High-energy Physics Citation Network (arXivCit-HepPh)**[7] **dataset and arXiv High-energy Physics Theory Citation Network (arXivCit-HepTh)**[8] **dataset** are abstracted from the e-print arXiv. arXivCit-HepPh covers all the citations within a dataset of 34,546 papers (regarded as nodes) with 421,578 directed links. arXivCit-HepTh covers all the citations within a dataset of 27,777 papers (regarded as nodes) with 352,807 directed links. If a paper i cites paper j, the graph contains a directed link from i to j. The data consist of papers from the period January 1993 to April 2003, categorized by year.

The networks chosen in the experiment follow the power-law distribution (Adamic and Huberman, 2000), as do the random walks on the networks (Perozzi et al., 2014), as shown in Figures 4.2-4.7.

[5]https://snap.stanford.edu/data/egonets-Facebook.html
[6]https://downloads.thebiogrid.org/BioGRID
[7]http://snap.stanford.edu/data/cit-HepPh.html
[8]http://snap.stanford.edu/data/cit-HepTh.html

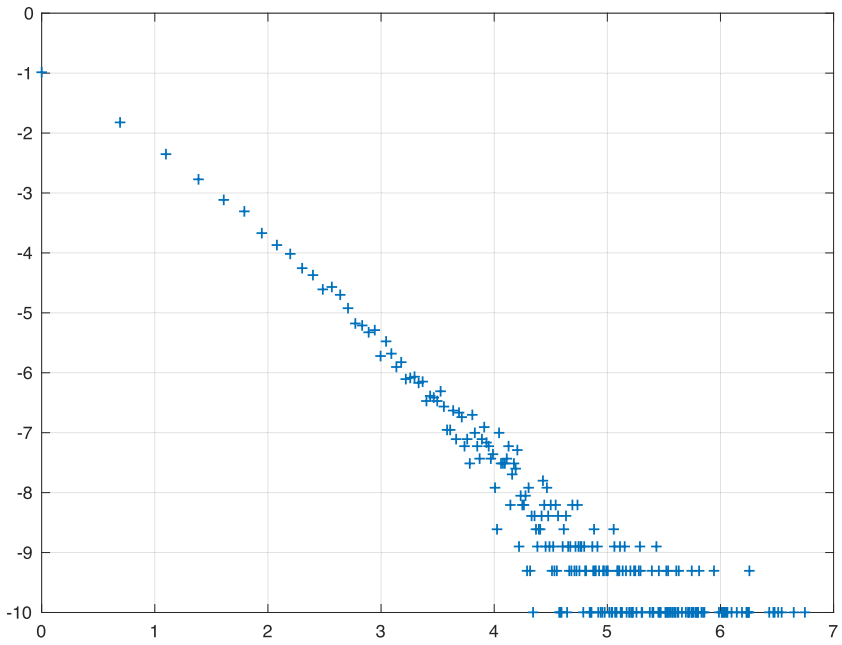


(a) Blog3 Network

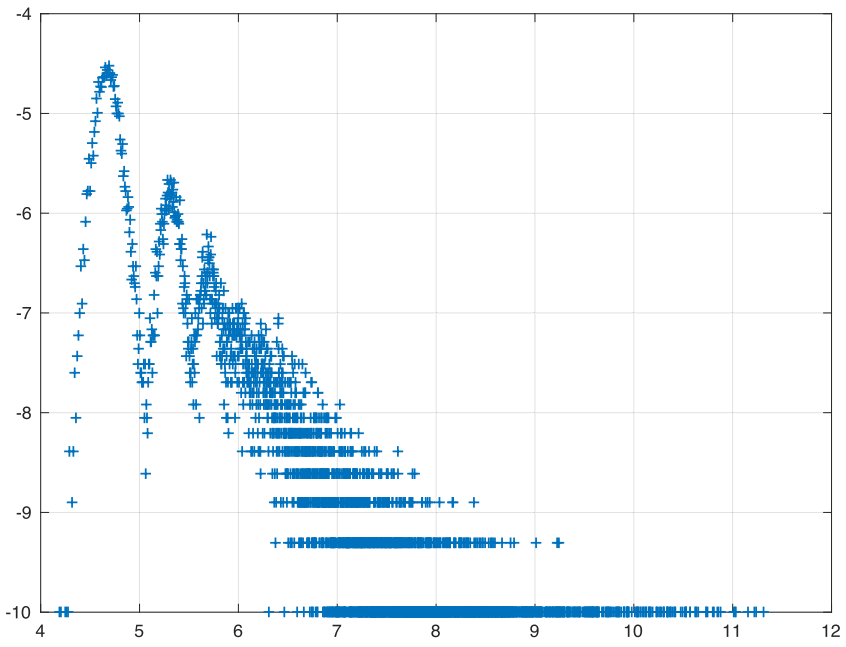


(b) Blog3 Random Walks

Figure 4.2 Power-law distributions of the Blog3 network and its random walks

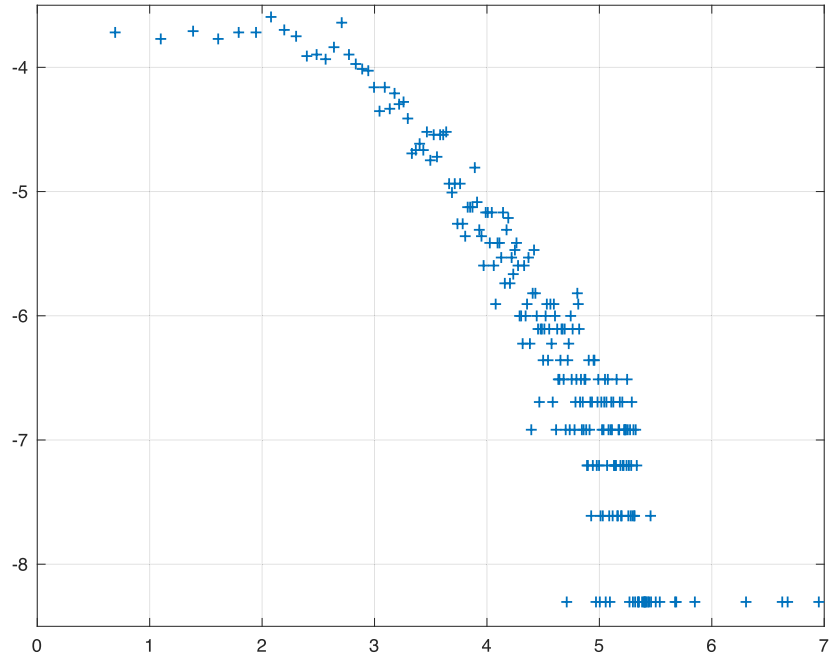


(a) YouTube Network

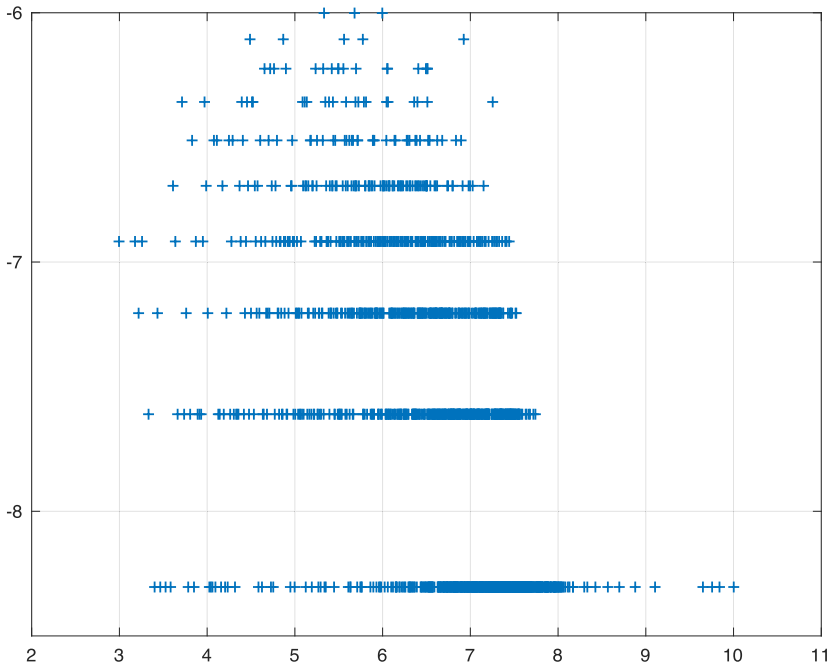


(b) YouTube Random Walks

Figure 4.3 Power-law distributions of the YouTube network and its random walks

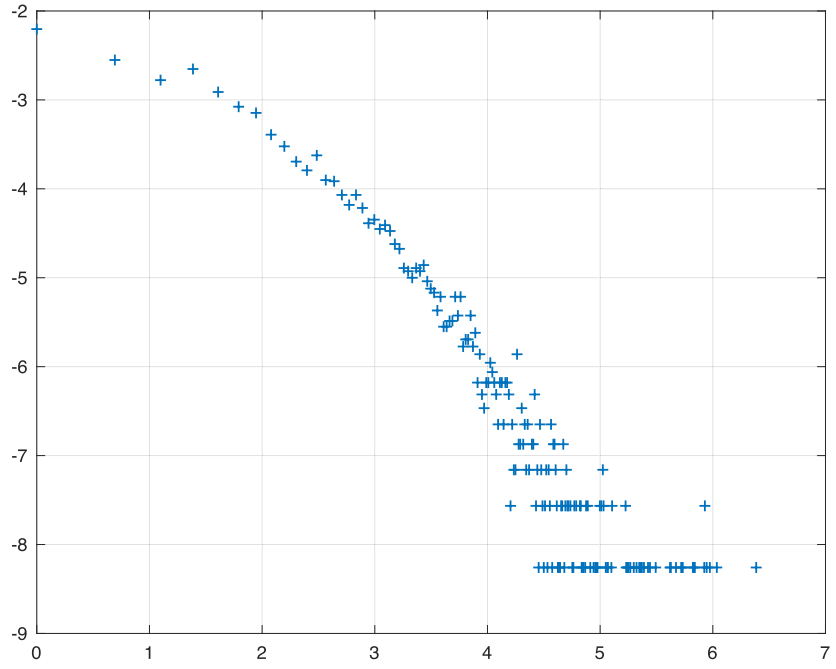


(a) Facebook Network

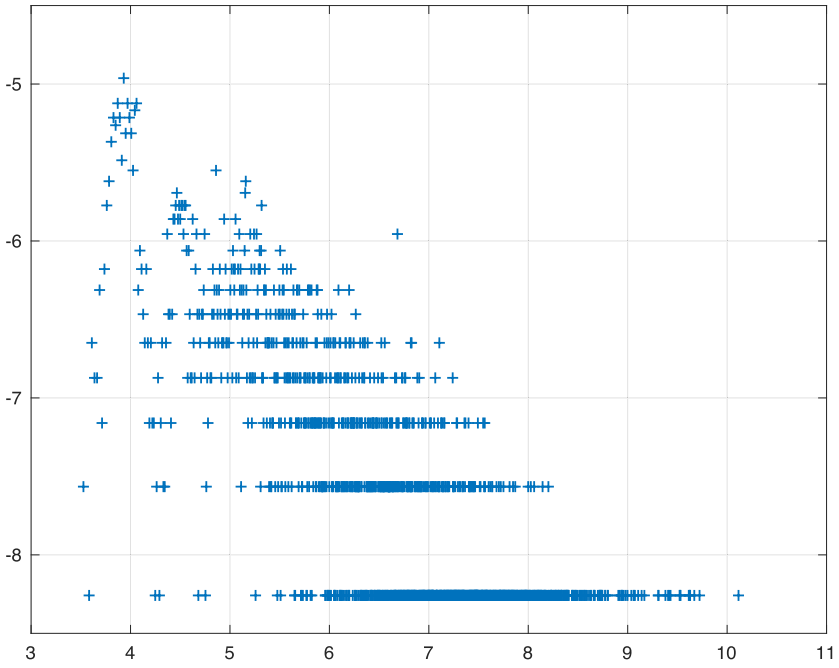


(b) Facebook Random Walks

Figure 4.4 Power-law distributions of the Facebook network and its random walks

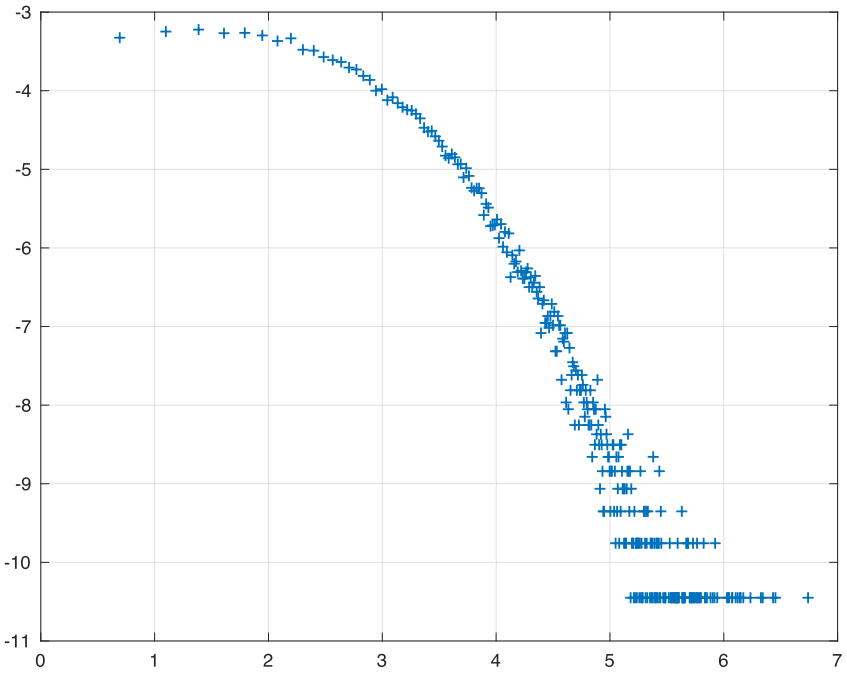


(a) PPI Network

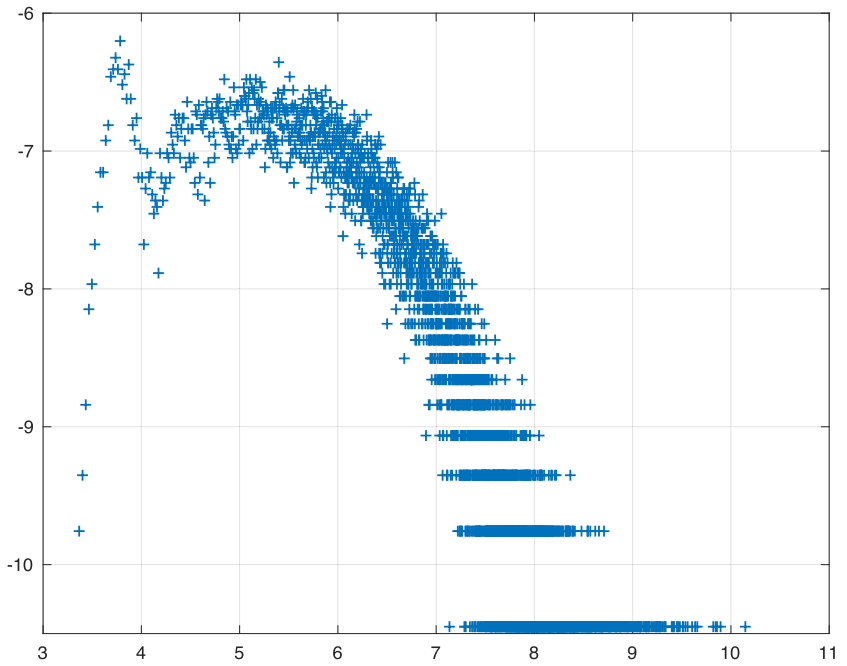


(b) PPI Random Walks

Figure 4.5 Power-law distributions of the PPI network and its random walks

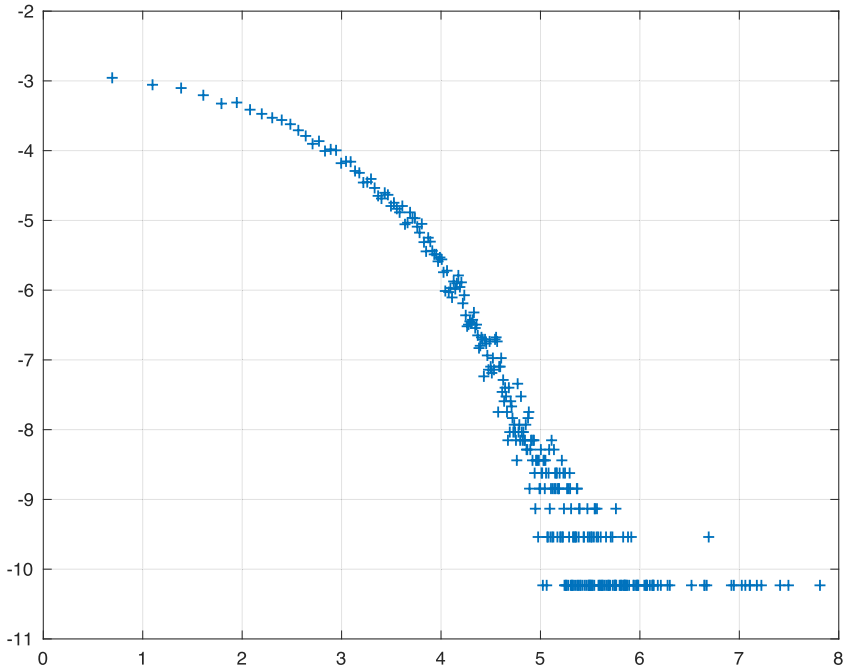


(a) arXivCit-HepPh Network

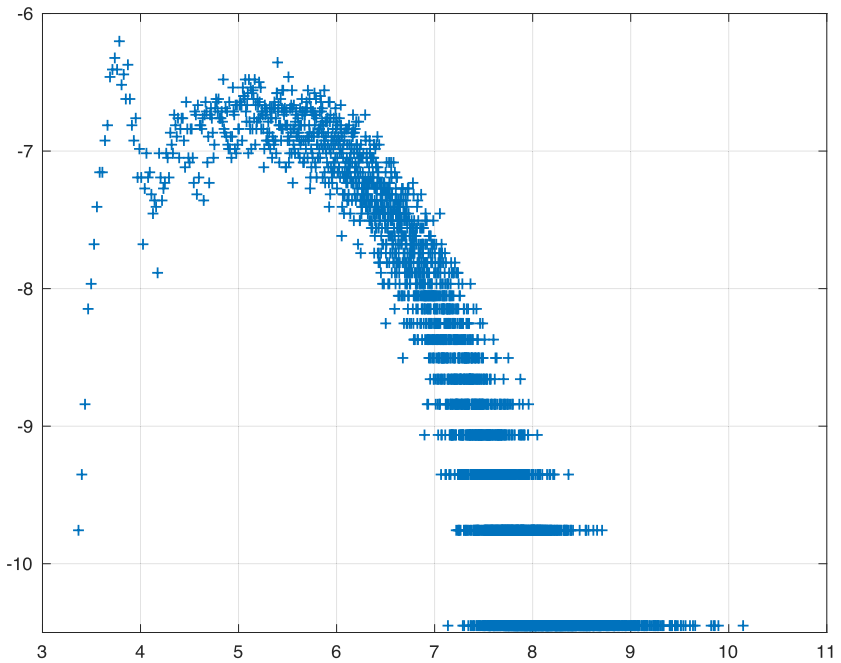


(b) arXivCit-HepPh Random Walks

Figure 4.6 Power-law distributions of the arXivCit-HepPh network and its random walks



(a) arXivCit-HepTh Network



(b) arXivCit-HepTh Random Walks

Figure 4.7 Power-law distributions of the arXivCit-HepTh network and its random walks

4.4.2 Setups

Our experiment evaluates the latent feature representations on a standard supervised learning task: linear SVM classification (Fung and Mangasarian, 2005). A linear classifier is chosen instead of a non-linear classifier or sophisticated relational classifier to reduce the impact of complicated learning approaches on classification performance. For the evaluations, we randomly partition the dataset in the target domain into two non-overlapping sets for training and testing by nine groups of training percentages, $\{0.1, 0.2, \cdots, 0.9\}$. We repeat the above steps ten times and thus obtain ten copies of the training data and testing data. The reported experiment results are the average of the ten runs and their variance.

The node feature learning by network representations are input to a one-against-all linear SVM classifier (Hsu and Lin, 2002). We use Macro-F1 and Micro-F1 (Yang and Liu, 1999) to compare performance and the results are shown in Tables 4.3-4.8. The F1 score is designed to evaluate the effectiveness of category assignments by classifiers.

We use the indicators of true positive (*tp*), false positive (*fp*) and false negative (*fn*) to measure the standard recall (r) and precision (p). For $F_1(r,p) = Micro\_F_1$, let $r = \frac{\sum tp}{\sum tp + \sum fn}$ and $p = \frac{\sum tp}{\sum tp + \sum fp}$. The Micro-F1 score computes the global $n \times m$ binary decisions, where n is the number of total test nodes, and m is the number of categories of binary labels. For $F_1(r,p) = Micro\_F_1$, let $r = \frac{1}{m}\sum \frac{tp}{tp+fn}$ and $p = \frac{1}{m}\sum \frac{tp}{tp+fp}$. The Macro-F1 score computes the binary decisions on individual categories and then averages the categories.

$$F_1(r,p) = \frac{2rp}{r+p} \tag{4.8}$$

This experiment summarizes the network statistics in Table 4.1. Node degree is the number of links reflecting the connection capability of the node. Each network is selected as a source domain or a target domain following $|V^s| > |V^t|$ and $\langle deg^s \rangle > \langle deg^t \rangle$. These selections are shown in Table 4.2.

Table 4.2 CDNR domain selections based on network statistics

Source Domain	Target Domain
Blog3	PPI
arXivCit-HepTh	PPI
arXivCit-HepPh	PPI
Facebook	PPI
Blog3	Facebook
arXivCit-HepPh	YouTube

4.4.3 Baselines

This experiment evaluates the performance of the unsupervised CDNR on the target networks. The representation outputs are applied a standard supervised learning task, i.e., linear SVM classification (Suykens and Vandewalle, 1999), to put less emphasis on the network representation performance by classifiers and reflect the performance of CDNR. The baselines are chosen from the previous domain-specific network representations as follows.

- **DeepWalk** (Perozzi et al., 2014) is the first random walk-based network representation algorithm. By choosing DeepWalks, we exclude the matrix factorization approaches which have already been demonstrated to be inferior to DeepWalk.

- **LINE** (Tang et al., 2015) learns latent feature representations from large-scale information networks by an link-sampling strategy in two separate phases of

first- and second-order proximities. We excluded a recent Graph Factorization algorithm (Ahmed et al., 2013) because LINE demonstrated better performance in the previous experiment.

- **Node2Vec** (Grover and Leskovec, 2016) learns continuous feature representations of nodes using a biased random walk procedure to capture the diversity of connectivity patterns observed in networks with the biased parameter α which is controlled by parameters of p and q.

- **Struc2Vec** (Ribeiro et al., 2017) learns node representations from structural identity by constructing a hierarchical graph to encode structural similarities and generating a structural context for nodes.

4.4.4 Parameters Setting

The parameters of CDNR are set in line with typical values used for DeepWalk, LINE, Node2Vec and Struc2Vec. For networks in both the source domain and the target domain, let the dimensions of feature representation be $d = 128$, the walk length be $l = 80$, the number of walks of every source node be $k = 10$, the window size be $r = 10$, $workers = 8$, and the search bias α be with $p = 1$ and $q = 1$, which comprehensively follows previous settings in DeepWalk (Perozzi et al., 2014), Node2Vec (Grover and Leskovec, 2016) and LINE (Tang et al., 2015). Let the learning rate ρ start from 0.025 as in (Tang et al., 2015) and the convergence track on 0.1 in our experiment. For Struc2Vec as used in (Ribeiro et al., 2017), let OPT1 (reducing the length of degree sequences), OPT2 (reducing the number of pairwise similarity calculations) and OPT3 (reducing the number of layers) all in values of True, and the maximum number of layers be 6. The parameters of the node-mapping

is set as $\gamma = 100$ and $\lambda = 100$. In these settings, the total number of random walks over an input network is $w = SampleSize \times k$ and the size of the random walks is $w \times l$.

4.4.5 Result Analysis

In the multi-label classification setting, every node is assigned one or more labels from a finite set Y. In the training phase of the CDNR node feature representations, we observe a fraction of the nodes and all their labels, and predict the labels for the remaining nodes. This is a challenging task, especially if the $|V|$ and $|E|$ are large. The multi-label classification in our experiment inputs $d = 128$ network representations to a one-against-all linear SVM classifier (Hsu and Lin, 2002). We use the F1 score of Macro-F1 and Micro-F1 to compare performance (Yang and Liu, 1999) in Tables 4.3-4.8.

Experiment results from the algorithmic perspective. A general observation drawn from the results is that the learned feature representations from other networks improve or maintain performance compared to the domain-specific network representation baseline algorithms. CDNR outperforms DeepWalk, LINE and Struc2Vec in all datasets with a gain of 12.95%, 47.66% and 52.21% respectively. CDNR outperforms Node2Vec on the PPI dataset and the YouTube dataset in 72% of the experiment, and outperforms Node2Vec on the Facebook dataset in 94% of the experiment. The losses of CDNR with Node2Vec on all networks average only 0.12%.

Experiment results from the dataset perspective. The general results on the PPI dataset (Tables 4.3 and 4.4) reflect the difficulty of cross-domain learning. Con-

Table 4.3 CDNR classification results of Micro-F1 on the target domain network of PPI

Algorithm	10%	20%	30%	40%	50%	60%	70%	80%	90%
DeepWalk	0.2849 ±0.0181	0.2854 ±0.0116	0.2845 ±0.0193	0.2803 ±0.0170	0.2725 ±0.0168	0.2736 ±0.0200	0.2629 ±0.0241	0.2778 ±0.0215	0.2621 ±0.0344
LINE	0.2900 ±0.0062	0.2772 ±0.0077	0.2807 ±0.0083	0.2715 ±0.0104	0.2702 ±0.0113	0.2649 ±0.0166	0.2710 ±0.0163	0.2494 ±0.0251	0.2398 ±0.0195
Node2Vec	0.3073 ±0.0171	0.2955 ±0.0104	0.3024 ±0.0139	**0.3028 ±0.0120**	0.3028 ±0.0102	0.2995 ±0.0186	**0.3021 ±0.0288**	0.2967 ±0.0197	**0.3005 ±0.0283**
Struc2Vec	0.2693 ±0.0228	0.2713 ±0.0187	0.2696 ±0.0188	0.2515 ±0.0187	0.2603 ±0.0133	0.2499 ±0.0212	0.2493 ±0.0148	0.2419 ±0.0156	0.2338 ±0.0287
CDNR Blog3 2PPI	0.3020 ±0.0109	0.3027 ±0.0120	0.2980 ±0.0120	0.2989 ±0.0064	0.2979 ±0.0077	0.3016 ±0.0113	0.2974 ±0.0142	0.2839 ±0.0191	0.2843 ±0.0375
CDNR arXivCit -HepPh 2PPI	0.2971 ±0.0132	0.3068 ±0.0119	**0.3060 ±0.0164**	0.3001 ±0.0079	**0.3037 ±0.0098**	**0.3042 ±0.0092**	0.2897 ±0.0177	0.2921 ±0.0126	0.2978 ±0.0349
CDNR arXivCit -HepTh 2PPI	**0.3074 ±0.0136**	0.2949 ±0.0140	0.3017 ±0.0125	0.3024 ±0.0097	0.2995 ±0.0136	0.3005 ±0.0105	0.2979 ±0.0263	**0.3008 ±0.0195**	0.2999 ±0.0387
CDNR Facebook 2PPI	0.3064 ±0.0128	**0.3082 ±0.0137**	0.3051 ±0.0135	0.3024 ±0.0092	0.2970 ±0.0135	0.2857 ±0.0163	0.2915 ±0.0192	0.2840 ±0.0142	0.2652 ±0.0249

Table 4.4 CDNR classification results of Macro-F1 on the target domain network of PPI

Algorithm	10%	20%	30%	40%	50%	60%	70%	80%	90%
DeepWalk	0.3416 ±0.0140	0.3378 ±0.0138	0.3364 ±0.0208	0.3406 ±0.0171	0.3306 ±0.0159	0.3336 ±0.0241	0.2949 ±0.0288	0.2825 ±0.0185	0.2041 ±0.0386
LINE	0.3058 ±0.0094	0.3003 ±0.0113	0.3008 ±0.0069	0.2940 ±0.0120	0.2868 ±0.0138	0.2826 ±0.0176	0.2733 ±0.0173	0.2462 ±0.0262	0.1822 ±0.0198
Node2Vec	0.3490 ±0.0193	0.3442 ±0.0141	0.3510 ±0.0205	**0.3500 ±0.0126**	0.3432 ±0.0140	0.3414 ±0.0201	0.3274 ±0.0240	0.3006 ±0.0248	**0.2310 ±0.0385**
Struc2Vec	0.2892 ±0.0197	0.2926 ±0.0232	0.3019 ±0.0227	0.2784 ±0.0267	0.2851 ±0.0152	0.2626 ±0.0202	0.2589 ±0.0177	0.2399 ±0.0287	0.1712 ±0.0262
CDNR Blog3 2PPI	0.3490 ±0.0182	0.3421 ±0.0097	0.3468 ±0.0165	0.3408 ±0.0095	0.3353 ±0.0209	0.3385 ±0.0098	0.3171 ±0.0247	0.2888 ±0.0298	0.2154 ±0.0399
CDNR arXivCit -HepPh 2PPI	0.3494 ±0.0111	0.3501 ±0.0091	**0.3550 ±0.0186**	0.3425 ±0.0132	**0.3484 ±0.0154**	0.3371 ±0.0226	0.3122 ±0.0150	0.2893 ±0.0199	0.2194 ±0.0463
CDNR arXivCit -HepTh 2PPI	**0.3533 ±0.0168**	0.3502 ±0.0159	0.3494 ±0.0173	0.3492 ±0.0119	0.3387 ±0.0154	**0.3444 ±0.0186**	0.3233 ±0.0230	**0.3076 ±0.0229**	0.2301 ±0.0517
CDNR Facebook 2PPI	0.3503 ±0.0115	**0.3519 ±0.0141**	0.3435 ±0.0067	0.3464 ±0.0176	0.3272 ±0.0185	0.3216 ±0.0186	**0.3282 ±0.0172**	0.2789 ±0.0229	0.2060 ±0.0327

Table 4.5 CDNR classification results of Micro-F1 on the target domain network of Facebook

Algorithm	10%	20%	30%	40%	50%	60%	70%	80%	90%
DeepWalk	0.8078 ±0.0449	0.8727 ±0.0177	0.8933 ±0.0062	0.9050 ±0.0059	0.9153 ±0.0060	0.9198 ±0.0061	0.9307 ±0.0039	0.9301 ±0.0103	0.9334 ±0.0175
LINE	0.4627 ±0.0026	0.4654 ±0.0104	0.4719 ±0.0026	0.4739 ±0.0035	0.4765 ±0.0035	0.4761 ±0.0033	0.4760 ±0.0067	0.4787 ±0.0066	0.4755 ±0.0075
Node2Vec	0.9352 ±0.0072	0.9401 ±0.0032	0.9398 ±0.0051	0.9419 ±0.0047	0.9442 ±0.0057	0.9454 ±0.0063	0.9468 ±0.0092	0.9466 ±0.0079	0.9502 ±0.0098
Struc2Vec	0.4152 ±0.0237	0.4521 ±0.0144	0.4716 ±0.0061	0.4994 ±0.0059	0.5161 ±0.0078	0.5381 ±0.0096	0.5461 ±0.0115	0.5639 ±0.0241	0.5530 ±0.0175
CDNR Blog3 2Facebook	**0.9373 ±0.0071**	**0.9420 ±0.0048**	**0.9422 ±0.0050**	**0.9434 ±0.0040**	**0.9461 ±0.0043**	**0.9468 ±0.0064**	**0.9499 ±0.0063**	**0.9529 ±0.0098**	**0.9510 ±0.0105**

Table 4.6 CDNR classification results of Macro-F1 on the target domain network of Facebook

Algorithm	10%	20%	30%	40%	50%	60%	70%	80%	90%
DeepWalk	0.7655 ±0.0185	0.7915 ±0.0242	0.7858 ±0.0331	0.8052 ±0.0308	0.7902 ±0.0306	0.8138 ±0.0327	0.8213 ±0.0504	0.7678 ±0.0317	0.7822 ±0.0378
LINE	0.5063 ±0.0053	0.5040 ±0.0189	0.5083 ±0.0093	0.5129 ±0.0061	0.5091 ±0.0092	0.5040 ±0.0077	0.5020 ±0.0137	0.4981 ±0.0117	0.4961 ±0.0109
Node2Vec	0.8310 ±0.0256	0.8331 ±0.0226	0.8206 ±0.0262	**0.8373 ±0.0359**	0.8343 ±0.0354	0.8214 ±0.0479	0.8192 ±0.0487	0.8018 ±0.0277	0.8104 ±0.0498
Struc2Vec	0.3701 ±0.0156	0.3937 ±0.0157	0.3926 ±0.0174	0.4160 ±0.0155	0.4377 ±0.0235	0.4525 ±0.0131	0.4532 ±0.0144	0.4755 ±0.0260	0.4583 ±0.0347
CDNR Blog3 2Facebook	**0.8329 ±0.0251**	**0.8383 ±0.0320**	**0.8405 ±0.0366**	0.8372 ±0.0338	**0.8355 ±0.0368**	**0.8265 ±0.0272**	**0.8234 ±0.0531**	**0.8163 ±0.0458**	**0.8108 ±0.0508**

Table 4.7 CDNR classification results of Micro-F1 on the target domain network of YouTube

Algorithm	10%	20%	30%	40%	50%	60%	70%	80%	90%
DeepWalk	0.3471 ±0.0068	0.3382 ±0.0064	0.3425 ±0.0144	0.3490 ±0.0071	0.3509 ±0.0037	0.3552 ±0.0112	0.3608 ±0.0067	0.3586 ±0.0124	0.3511 ±0.0176
LINE	0.2785 ±0.0129	0.2732 ±0.0088	0.2726 ±0.0080	0.2703 ±0.0106	0.2703 ±0.0117	0.2633 ±0.0091	0.2648 ±0.0059	0.2654 ±0.0161	0.2558 ±0.0133
Node2Vec	0.4183 ±0.0036	0.4134 ±0.0063	0.4136 ±0.0067	0.4118 ±0.0020	**0.4117 ±0.0031**	0.4118 ±0.0034	0.4106 ±0.0076	**0.4106 ±0.0069**	**0.4119 ±0.0124**
Struc2Vec	0.3232 ±0.0049	0.3268 ±0.0039	0.3260 ±0.0040	0.3261 ±0.0030	0.3276 ±0.0049	0.3289 ±0.0054	0.3276 ±0.0046	0.3285 ±0.0032	0.3327 ±0.0126
CDNR arXivCit -HepPh 2YouTube	**0.4195 ±0.0056**	**0.4152 ±0.0045**	**0.4150 ±0.0037**	**0.4128 ±0.0025**	0.4107 ±0.0047	**0.4124 ±0.0045**	**0.4113 ±0.0073**	0.4094 ±0.0089	0.4041 ±0.0111

Table 4.8 CDNR classification results of Macro-F1 on the target domain network of YouTube

Algorithm	10%	20%	30%	40%	50%	60%	70%	80%	90%
DeepWalk	0.3471 ±0.0097	0.3382 ±0.0117	0.3389 ±0.0127	0.4356 ±0.0109	0.4393 ±0.0078	0.4367 ±0.0136	0.4412 ±0.0100	0.4302 ±0.0135	0.3994 ±0.0190
LINE	0.2592 ±0.0076	0.2610 ±0.0073	0.2618 ±0.0044	0.2629 ±0.0047	0.2611 ±0.0067	0.2640 ±0.0055	0.2587 ±0.0098	0.2501 ±0.0117	0.2410 ±0.0115
Node2Vec	0.4254 ±0.0077	0.4285 ±0.0082	0.4338 ±0.0080	0.4338 ±0.0051	0.4411 ±0.0043	0.4397 ±0.0067	**0.4478 ±0.0110**	0.4403 ±0.0144	**0.4309 ±0.0178**
Struc2Vec	0.2976 ±0.0043	0.2984 ±0.0055	0.3020 ±0.0042	0.3060 ±0.0052	0.3078 ±0.0047	0.3106 ±0.0054	0.3095 ±0.0131	0.3009 ±0.0103	0.2990 ±0.0122
CDNR arXivCit -HepPh 2YouTube	**0.4256 ±0.0074**	**0.4310 ±0.0070**	**0.4358 ±0.0088**	**0.4361 ±0.0072**	**0.4452 ±0.0053**	**0.4438 ±0.0095**	0.4370 ±0.0091	**0.4414 ±0.0074**	0.4073 ±0.0269

sidering the domain similarities, a cross-domain adaption from either the social networks or the citation networks to the biological network as shown in our experiment would not be recommended in transfer learning. However, CDNR is capable of capturing useful structural information from network topologies and removing noise from the source domain networks in an unsupervised feature-learning environment, so CDNR on PPI still shows a slight improvement and almost retains its representation performances. Therefore, cross-domain network knowledge transfer learning works in unsupervised network representations. CDNR is less influenced by domain selections when the transferable knowledge is mainly contributed by network topologies.

The general results on the Facebook dataset (Tables 4.5 and 4.6) show promising improvements by CDNR compared to other baseline algorithms. Examining the results in detail shows that the source domain networks of Blog3, arXivCit-HepPh and arXivCit-HepTh provide a larger volume of information to the Facebook target domain network than other pairs of CDNR experiments, which promote knowledge transfer across domains. The social network of Blog3 transfers 39 categories of Interests with the network average degree of 64.776 to the Facebook social network (10 categories of Groups, network average degree of 43.691). The citation networks of arXivCit-HepPh and arXivCit-HepTh transfer 11 categories of Years to Facebook with a network average degree of 24.407 and 25.409 respectively. The above results show that unsupervised CDNR works especially well in dense networks.

The YouTube dataset in this experiment is an example of a sparse network that lacks connections and contains large numbers of isolated nodes. It can be seen from the general results on the YouTube dataset (Tables 4.7 and 4.8) that CDNR can transfer knowledge from a directed network to an undirected network. Unsupervised

representations of CDNR allow learning from small categories to large categories, and in a heterogeneous label space. In addition, CDNR uses its CD2LRW learning algorithm to capture the useful topologies in a large-scale information network.

To demonstrate that CDNR is indeed statistically superior to the baselines, we summarize our results for all classification evaluation tasks in Table 4.9 by pairwise *t*-test at a confidence level of $\alpha = 0.05$. The statistical significance is validated on every paired CDNR and baseline. On the single-label datasets, for example FTLSIN from the DBLP dataset to the M10 dataset (FTLSINDBLP2M10) is compared with DeepWalk, LINE, Node2Vec and Struc2Vec by pairwise *t*-test. 1.37E-17 in line 3 column 2 of Table 4.9 is a mean significance value averaged from nine significance values on $\{10\%, \cdots, 90\%\}$ training percentages. Each of these significance values is *t*-tested between FTLSINDBLP2M10 and DeepWalk. Since the CDNR multi-label dataset experiment is conducted across six datasets,the statistical significance is validated for each scenario; for example CDNRBlog32PPI is CDNR from Blog3 to PPI, and 3.14E-02 in line 3 column 6 is averaged from the nine significance values by pairwise *t*-testing CDNRBlog32PPI and DeepWalk.

In Table 4.9, each value less than $\alpha = 0.05$ indicates that the difference is statistically significant. The results in Table 4.9 confirm that CDNR statistically outperforms DeepWalk, LINE, Node2Vec and Struc2Vec in all cases.

Table 4.9 Pairwise t-test results of FTLSIN/CDNR versus baselines

	FTLSINDBLP2M10				CDNRBlog32PPI			
	DeepWalk	LINE	Node2Vec	Struc2Vec	DeepWalk	LINE	Node2Vec	Struc2Vec
Micro-F1	1.37E-17	7.23E-14	1.74E-02	1.60E-09	3.14E-02	2.65E-03	4.14E-02	8.60E-04
Macro-F1	1.08E-15	5.89E-10	1.20E-02	8.70E-08	4.41E-02	4.83E-03	4.08E-02	1.65E-03
	CDNRarXivCit-HepPh2PPI				CDNRarXivCitHepTh2PPI			
	DeepWalk	LINE	Node2Vec	Struc2Vec	DeepWalk	LINE	Node2Vec	Struc2Vec
Micro-F1	4.57E-02	2.16E-02	4.99E-02	7.22E-04	3.79E-02	2.84E-03	4.42E-02	9.89E-04
Macro-F1	2.70E-02	4.54E-03	4.77E-02	1.62E-03	4.32E-02	2.03E-03	4.22E-02	7.77E-04
	CDNRFacebook2PPI				CDNRBlog32Facebook			
	DeepWalk	LINE	Node2Vec	Struc2Vec	DeepWalk	LINE	Node2Vec	Struc2Vec
Micro-F1	2.02E-02	8.08E-03	3.91E-02	2.78E-03	4.40E-02	7.46E-28	4.11E-02	6.15E-21
Macro-F1	2.29E-02	9.92E-03	3.69E-02	3.06E-03	1.99E-02	1.22E-13	4.11E-02	1.49E-13
	CDNRarXivCit-HepPh2YouTube							
	DeepWalk	LINE	Node2Vec	Struc2Vec				
Micro-F1	1.95E-09	1.45E-15	4.47E-02	2.09E-11				
Macro-F1	9.52E-03	1.16E-12	3.85E-02	2.25E-10				

4.5 Summary

In this chapter, we propose the CDNR algorithm for FTLSIN. Compared to previous network representation approaches, CDNR enables knowledge transfer from the external domains using a two-layer design. A bottom layer is designed for knowledge preparation and a top layer is designed for CDNR in the target domain. The CD2LRW solves the key problems of unbalanced scales across networks by a cross-domain node mapping procedure that balances the node scales and a cross-domain walk mapping procedure that balances the random walk scales, in which the random walks are the knowledge transferred by CDNR. The experiment results show that CDNR improves the performance of latent feature learning in large-scale information networks.

Chapter 5

Cross-domain Similarity Learning based on Network Patterns

5.1 Introduction

In the last decade, research works on network representation for complex data-based learning tasks have been conducted from a domain-specific point of view (Li et al., 2018a; Ma et al., 2018; Zhou et al., 2018). The main disadvantages of network sparsity and computational complexity have been studied from the aspects of sampling strategy and representation optimization. However, the information scale is still too small to support a satisfying representation that compares with the node scale (Xue et al., 2018). Therefore, it is effective to input external knowledge for the target network representation task.

In this chapter, we use a simple network structure in which the nodes have no additional attributes, and the links are unweighted and undirected. Networks conduct the representation learning task based on two pieces of information, i.e., their own

network structures and the adapted knowledge from the external related networks. To address the similarity determine two independent networks for the CDNR problem, we consider the following challenges: 1) How are domain similarities calculated based on network structure patterns? and 2) Will the domain selection based on domain similarities contribute to CDNR?

To this end, we propose a CDSL algorithm to learn the domain similarity for cross-domain large-scale networks based on the network pattern analysis of node centralities. The main contributions are as follows:

- **Contribution 1:** We design a novel algorithm to learn the cross-domain similarities, starting from the node centralities. We identify the node centralities from two aspects, i.e., *centrality*: the ability to spread information, such as the closeness centrality; and *power*: the ability to control information, such as the betweenness centrality.

- **Contribution 2:** We design a principle that calculates the similarity between cross-domain networks using dual centrality-based biased random walks, and considers the *centrality* and *power* properties. The principle shows the effectiveness by implementing in the CDNR process.

- **Contribution 3:** To evaluate the proposed algorithm, we conduct extensive experiments on similarity-parameter γ training and node classification testing. The experiment results show that our proposed algorithm CDSL significantly outperforms the state-of-the-art algorithms.

5.2 Problem Statement

Definition 5.1. Cross-domain Similarity Learning (CDSL) for Network Representation Given a set of undirected networks $\mathscr{G} = \{G\}$, one of these networks belongs to $\mathscr{D}^t$ and the others are in $\mathscr{D}^s$ where one network will be chosen to provide knowledge to the target domain. The learning process of choosing the most relevant source domain network denotes the cross-domain similarity learning for network representation.

In each network, $v_i \in V$ is the ith node ($i = 1, 2, \cdots, N$) and the link $(i, j) \in E$ connects the nodes v_i and v_j. Each node v_i has a neighborhood $N(v_i)$ which includes the node v_i, the links (i, j) connecting with v_i and the nodes v_j at the other side. The degree k_i of v_i denotes the number of links in $N(v_i)$. The random walk-based network representation aims to maximize the likelihood of $N(v_i)$ when searching over the whole network. The random walk is described as a path shape in G where the path starts from node v_s and ends at node v_t. Path $\mathcal{P}_{st}$ consists of a sequence of nodes. The number of steps along $\mathcal{P}_{st}$ denotes $d(v_s, v_t)$, which is called the distance between v_s and v_t. Of all $\mathcal{P}_{st}$, the shortest path $\mathcal{P}^*_{st}$ is in the smallest $d(v_s, v_t)$. In this paper, $\widetilde{P}_{st}$ denotes the probability of random walks in a fixed length l that start form v_s and end at v_t. $\widetilde{P}^{ref}_{st}$ is the reference path probability. Considering the cost $\widetilde{c}$, let $\widetilde{c}(\mathcal{P}_{st}) = \sum_{(i,j)\in\mathcal{P}_{st}} c_{ij}$ and $c_{ij} = \frac{1}{w_{ij}}$ where w_{ij} is the weight on link (i, j).

The biased random walk strategy introduces a biased parameter α_{pq} to leverage a second-order search with p and q to identify the local connections in $N(v_i)$. When a walk passes v_t and stands on v_i to determine the next step to v_x, the biased random walk sets the walk probability $P(v_x|v_i) = \frac{\pi_{xi}}{Z}$ if $(i,x) \in E$, otherwise 0, where $\pi_{xi} = \alpha_{pq}(v_t, v_x) \cdot w_{ix}$; $\alpha_{pq}(v_t, v_x) = \frac{1}{p}$ if $d(v_t, v_x) = 0$, $\alpha_{pq}(v_t, v_x) = 1$ if $d(v_t, v_x) = 1$, and

$\alpha_{pq}(v_t, v_x) = \frac{1}{q}$ if $d(v_t, v_x) = 2$; and Z is the normalizing constant. In this paper, we only consider the case of $(i,x) \in E$.

5.2.1 Node Centralities

Node centrality C_i reflects the impact of network structures in spreading and controlling information. The bi-directional information flows are evaluated by centrality indicators of closeness and betweenness in this paper. Closeness, as shown in Eq. (5.1), describes a node centrality property of spreading information outward from neighborhood $N(v_i)$.

$$C_i^{col} = \frac{N-1}{\sum_{j=1}^{N} \mathcal{P}_{ij}^*} \tag{5.1}$$

Betweenness, as shown in Eq. (5.2), describes a node centrality property of controlling information inside the neighborhood of $N(v_i)$.

$$C_i^{bet} = \sum_{s,t=1}^{n} \frac{n(i \in \mathcal{P}_{st}^*)}{|\mathcal{P}_{st}^*|} \tag{5.2}$$

where $|\mathcal{P}_{st}^*|$ is the number of shortest paths between node v_s and node v_t, and $n(i \in \mathcal{P}_{st}^*)$ is the number of shortest paths between v_s and v_t that go through node v_i.

5.2.2 Cross-domain Network Representations

FTLSIN (Xue et al., 2018) offers a solution for the CDNR problem by designing a CD2LRW to transfer the useful information along the biased random walks, as shown in Figure 5.1. FTLSIN is completed by a cross-domain node mapping procedure and a cross-domain walk mapping procedure. In this chapter, we keep the procedures

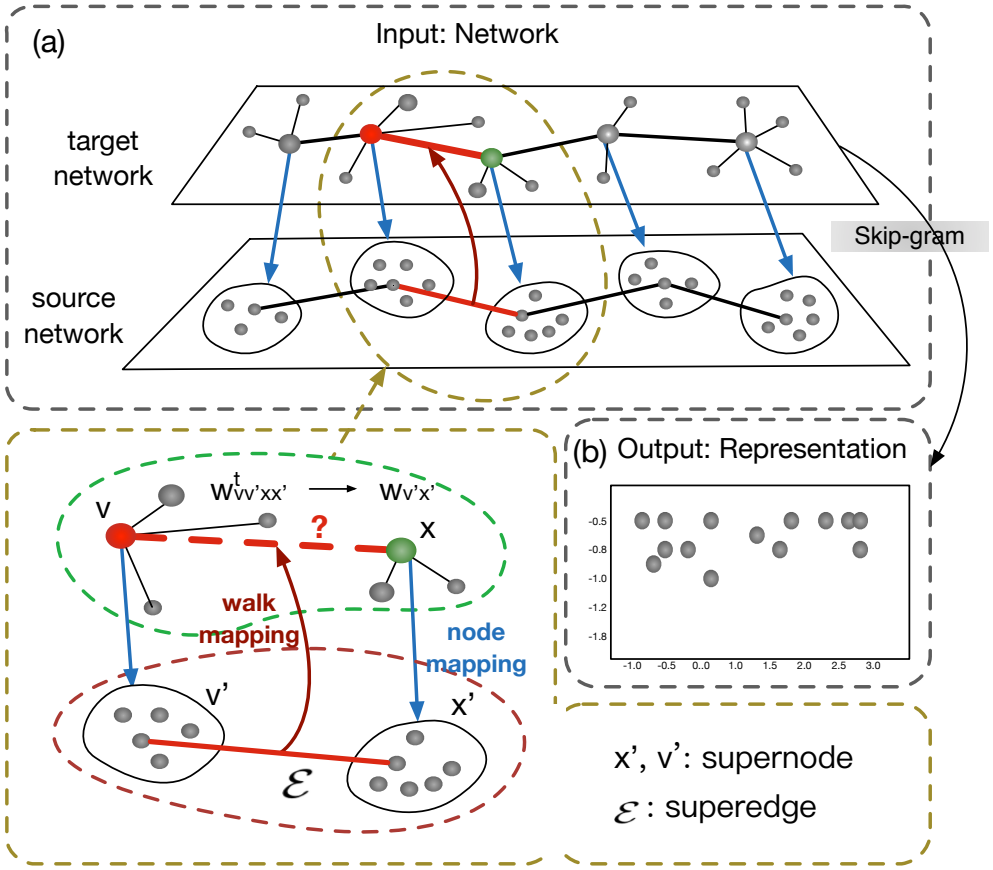


Figure 5.1 An illustration of CDNR in FTLSIN

in FTLSIN and update the mappings with the novel dual centrality indicator. The cross-domain node mapping clusters a group of nodes in the source domain network as the super node V_i^s and maps it to node v^t in the target domain network if their node degree is in the same $deg(v^t) = deg(V_i^s)$. The cross-domain walk mapping maps weights of the G^s biased random walk knowledge on the G^t links so that the biased random walk on the G^t is guided by source domain knowledge.

In this chapter, we propose a novel CDSL algorithm for FTLSIN and CDNR. CDSL selects the most relevant external network in $\mathscr{D}^s$ by learning the parameter γ, which weights dual centrality in $C_i = \gamma C_i^{clo} + (1-\gamma)C_i^{bet}$. The candidatures of the source domain networks $G^s \in \mathscr{G}$ are ranked by the similarity learning with the target domain network G^t in the biased random walk training procedure. When we obtain the optimized γ^*, the corresponding G^s which shows the most similarity pattern of dual centrality in (G^s, G^t) is selected as the source domain $\mathscr{D}^s$. **D**ual **C**entrality based **B**iased **R**andom **W**alk (DCBRW) works with CDSL and is trained in the FTLSIN.

5.3 Cross-domain Similarity Learning based on Network Patterns

In this section, we integrate the two node centrality indicators of closeness centrality and betweenness centrality into a dual centrality in a biased random walk learning. The biased random walk has a randomized shortest path framework and aims to minimize the walk cost along the paths. In finding the likelihood of shortest paths, the goal of our proposed algorithm, i.e., to minimize cost of searching the first- and

second order neighborhood, has the same function as the goal of previous works that to maximize the neighborhood likelihood.

5.3.1 Dual Centrality

The dual centrality consists of two node centrality indicators as shown in Eqs. (5.1) and (5.2). A parameter γ is weighted for these two indications and is trained for biased random walk.

$$\begin{aligned} C_i &= \gamma C_i^{clo} + (1-\gamma) C_i^{bet} \\ &= \sum_{s,t=1,t\neq i}^{n} \left[\frac{\gamma n(n-1)}{\mathcal{P}_{it}^*} + \frac{(1-\gamma)|\mathcal{P}_{it}^*|}{|\mathcal{P}_{st}^*|} \right] \end{aligned} \tag{5.3}$$

where C_i is the dual centrality of node v_i, $\mathcal{P}_{it}^*$ is the shortest path from node v_i to node v_t, $\mathcal{P}_{st}^*$ is the shortest path from node v_s to node v_t, and $|\mathcal{P}_{it}^*|$ and $|\mathcal{P}_{st}^*|$ are the numbers of $\mathcal{P}_{it}^*$ and $\mathcal{P}_{st}^*$ respectively.

Lemma 1. *Suppose* $n(i \in \mathcal{P}_{st}^*) \neq 0$, $n(i \in \mathcal{P}_{st}^*) = |\mathcal{P}_{it}^*|$.

Proof. Let $|\mathcal{P}_{it}^*| = n(\min \sum_{(u,v)\in\mathcal{P}_{it}} w_{uv})$, and

$n(i \in \mathcal{P}_{st}^*) = n(\min \sum_{(u,v)\in\mathcal{P}_{si}} w_{uv} \oplus \min \sum_{(u,v)\in\mathcal{P}_{it}} w_{uv})$.

$\oplus$ denotes that $n(\min \sum_{(u,v)\in\mathcal{P}_{si}} w_{uv} \oplus \min \sum_{(u,v)\in\mathcal{P}_{it}} w_{uv}) = 0$, if either $\mathcal{P}_{si}$ or $\mathcal{P}_{it}$ is $\varnothing$; and otherwise $n(\min \sum_{(u,v)\in\mathcal{P}_{si}} w_{uv} \oplus \min \sum_{(u,v)\in\mathcal{P}_{it}} w_{uv}) = n(\min \sum_{(u,v)\in\mathcal{P}_{si}} w_{uv}) = n(\min \sum_{(u,v)\in\mathcal{P}_{it}} w_{uv})$.

Because $n(i \in \mathcal{P}_{st}^*) \neq 0$, $\mathcal{P}_{si} \neq \varnothing$ ($n(\min \sum_{(u,v)\in\mathcal{P}_{si}} w_{uv}) \neq 0$). Therefore, $n(i \in \mathcal{P}_{st}^*) = n(\min \sum_{(u,v)\in\mathcal{P}_{it}} w_{uv}) = |\mathcal{P}_{it}^*|$. □

Theorem 5.1. *Let* $0 < C_i < 1$ *in Eq. (5.3), we have* $|\gamma| < \frac{min(C_i^{bet}, 1-C_i^{bet})}{|C_i^{clo} - C_i^{bet}|}$.

Proof. In Eq. (5.1), $0 < |\mathcal{P}_{it}^*| \leq n-1$ for each j, so $0 < C_i^{clo} < 1$. Similarly, $0 < C_i^{bet} < 1$ in Eq. (5.1) by Lemma 1 that $n(i \in \mathcal{P}_{st}^*) < |\mathcal{P}_{st}^*|$.

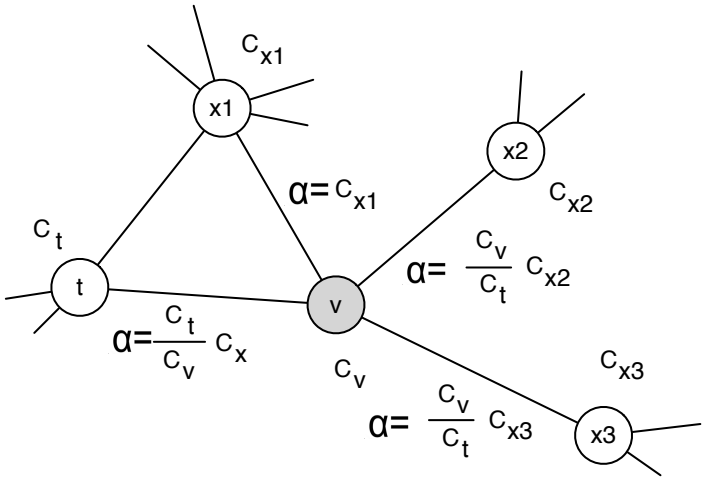


Figure 5.2 An example of DCBRW

Then, we let $0 < \gamma C_i^{clo} + (1-\gamma)C_i^{bet} < 1$, $|r| < \frac{1-C_i^{bet})}{|C_i^{clo}-C_i^{bet}|}$, if $C_i^{bet} > \frac{1}{2}$; and $|r| < \frac{C_i^{bet})}{|C_i^{clo}-C_i^{bet}|}$, if $C_i^{bet} < \frac{1}{2}$.

Therefore, $|\gamma| < \frac{min(C_i^{bet},1-C_i^{bet})}{|C_i^{clo}-C_i^{bet}|}$ under $0 < C_i < 1$. □

5.3.2 Dual Centrality based Biased Random Walk

Parameters p and q for biased random walk parameter α_{pq} as shown in Eq. (5.4) and α_{pq} is discussed in the following cases.

$$P(v_i = x | v_{i-1} = v) = \frac{\alpha_{pq}(v_t, v_x) \cdot w_{vx}}{Z} \tag{5.4}$$

where we only consider $(v,x) \in E$.

- If $d(v_t, v_x) = 0$, p is set at a high value to ensure that it does not link back to node v_t, which reflects a high value of C_x^{clo}.

- If $d(v_t, v_x) = 0$, p is set at a low value that tends to revisit node v_t, which reflects a high value of C_t^{bet}.
- If $d(v_t, v_x) = 2$, $q > 1$ is set to stand for the random walk that is biased towards nodes close to v_t, which reflects a high value of C_v^{bet}.
- If $d(v_t, v_x) = 2$, $q < 1$ is set to stand for the random walk that is biased towards nodes far away from v_t, which reflects a high value of C_v^{clo}.

In DCBRW, we summarize the above rules with C_i^{clo} and C_i^{bet} and also analyze the cases for $d(v_t, v_x)$.

- **Case 1:** If C_i^{clo} and C_i^{bet} are both in high values, Ci is in a high value which means that there is a high probability that node v_i will be connected.
- **Case 2:** If C_i^{clo} is in a high value and C_i^{bet} is in a low value, node v_i is spreading information outwards from $N(v_i)$.
- **Case 3:** If C_i^{clo} is in a low value and C_i^{bet} is in a high value, node v_i is controlling information inside $N(v_i)$.
- **Case 4:** If C_i^{clo} and C_i^{bet} are both in low values, Ci is in a low value which means that there is a small probability that node v_i will be connected.

For the above cases, DCBRW first describes Case 1 and Case 4 that $P(v_i = x|v_{i-1} = v)$ is in a high value if C_x is high. To this end, biased parameter α multiples C_x for all cases in $d(v_t, v_x)$. For Case 2 and Case 3, it is more likely that v_t will be revisited when $C_t > C_v$; and it is in a high probability to go far away for v_x when $C_v > C_t$. In summary, the probability on α with C_i in DCBRW as shown in Figure 5.2 is as follows,

$$P(v_i = x | v_{i-1} = v) = \frac{\alpha(v_t, v_x) \cdot w_{vx}}{Z} = \begin{cases} \frac{1}{Z} \cdot \frac{C_t}{C_v} \cdot C_x \cdot w_{vx} & \text{if } d(v_t, v_x) = 0 \\ \frac{1}{Z} \cdot C_x \cdot w_{vx} & \text{if } d(v_t, v_x) = 1 \\ \frac{1}{Z} \cdot \frac{C_v}{C_t} \cdot C_x \cdot w_{vx} & \text{if } d(v_t, v_x) = 2 \end{cases} \tag{5.5}$$

5.3.3 Dual Centrality based Randomized Shortest Path

The randomized shortest path is based on the probability distribution over the set $\mathcal{P}_{st}$ of absorbing *s-t* random walks. It expects the minimal cost of the walks to be as follows,

$$\min_{\widetilde{P}_{st}} \sum_{\mathcal{P}_{st}} \widetilde{P}_{st}(\mathcal{P}_{st}) \cdot \widetilde{c}(\mathcal{P}_{st}) \quad \text{subject to} \quad \begin{cases} J(\widetilde{P}_{st} || \widetilde{P}_{st}^{ref}) = J_0 \\ \widetilde{P}_{st} = 1 \end{cases} \tag{5.6}$$

where $J(\widetilde{P}_{st} || \widetilde{P}_{st}^{ref})$ is KL divergence and J_0 is a desired level.

The solution of Eq. (5.6) with the dual centrality C_i is,

$$\widetilde{P}_{st}(\mathcal{P}_{st}) = \frac{\widetilde{P}_{st}^{ref} \exp(-\sum_{(i,j) \in \mathcal{P}_{st}} p_{ij}^{ref} c_{ij})}{\sum_{\mathcal{P}_{st}} \widetilde{P}_{st}^{ref} \exp(-\sum_{(i,j) \in \mathcal{P}_{st}} p_{ij}^{ref} c_{ij})} \tag{5.7}$$

where $p_{ij}^{ref} = P(v_j | = v_i)$.

$\widetilde{P}_{st}(\mathcal{P}_{st})$ is normalized by the sum of likelihood of all paths in minimizing costs, and optimized by $J(\widetilde{P}_{st} || \widetilde{P}_{st}^{ref}) \to J_0$.

Lemma 2. *The desired level J_0 is constrained by fixed repeated times of each rooted node in the random walks, denoted as r, in* $-(\ln r+2) < J_0 < 0$.

Proof. $J(\widetilde{P}_{st}||\widetilde{P}^{ref}_{st}) = \sum_{\mathcal{P}_{st}} \widetilde{P}_{st} \ln \frac{\widetilde{P}_{st}}{\widetilde{P}^{ref}_{st}}$ by KL divergence. Let $\widetilde{P}^{ref}_{st} \exp(-\sum_{(i,j)\in\mathcal{P}_{st}} p^{ref}_{ij} c_{ij}) = A$, $J(\widetilde{P}_{st}||\widetilde{P}^{ref}_{st}) = -\frac{\sum_{\mathcal{P}_{st}} \left[A \sum_{(i,j)\in\mathcal{P}_{st}} p^{ref}_{ij} c_{ij}\right]}{\sum_{\mathcal{P}_{st}} A} - \frac{\sum_{\mathcal{P}_{st}} \left[A \ln \sum_{\mathcal{P}_{st}} A\right]}{\sum_{\mathcal{P}_{st}} A}$.

Because $0 < p^{ref}_{ij} < 1$ and $0 < c_{ij} < 1$, $0 < p^{ref}_{ij} c_{ij} < 1$ leads to $0 < \sum_{(i,j)\in\mathcal{P}_{st}} p^{ref}_{ij} c_{ij} < 1$. Because $A > 0$, $-A < -\sum_{(i,j)\in\mathcal{P}_{st}} p^{ref}_{ij} c_{ij} < 0$ so that $-1 < -\frac{\sum_{\mathcal{P}_{st}} \left[A \sum_{(i,j)\in\mathcal{P}_{st}} p^{ref}_{ij} c_{ij}\right]}{\sum_{\mathcal{P}_{st}} A} < 0$. Furthermore, $0 < \exp[\sum_{(i,j)\in\mathcal{P}_{st}} p^{ref}_{ij} c_{ij}] < e$ and $0 < A < e\widetilde{P}^{ref}_{st} < e$ which leads to $0 < \ln \sum_{\mathcal{P}_{st}} A < \ln re = \ln r + 1$, so $-(\ln r+1) - \frac{\sum_{\mathcal{P}_{st}} \left[A \ln \sum_{\mathcal{P}_{st}} A\right]}{\sum_{\mathcal{P}_{st}} A} < 0$.

This completes the proof that $-(\ln r+2) < J_0 < 0$ when $J(\widetilde{P}_{st}||\widetilde{P}^{ref}_{st}) \to J_0$. □

5.3.4 Algorithm of DCBRW-based CDSL

Algorithm 5.1 of DCBRW-based CDSL is formed by a DCBRW Algorithm and a CDSL Algorithm.

In this algorithm, we first focus on the dual centrality parameter γ learning for all the networks, which includes the network in the target domain and the networks in the source domain. Dual centrality C_i reflects the network pattern in information distributions in two aspects, i.e., the control of information and the spread of information on nodes. It is influenced by the local structures of $N(v_i)$ and is also determined by the global structure of node importance on centralities. To this end, $\gamma^{(t)}$ is updated with biased random walks in the randomized shortest paths. When the KL divergence is close to the desired level, J_0, γ^* is optimized with DCBRW. CDSL then calculates the dense similarity for each pair of optimized

$(\gamma^{*t}, \gamma^{*s})$. The network in the source domain that has the smallest dense similarity is chosen as the source network to perform the CDNR.

In summary, the main advantage of the proposed DCBRW-based CDSL is that 1) when the network structure lacks the information to generate a good network representation, the external network can offer knowledge of random walks; and 2) the external network is chosen by domain similarity learning instead of trials or expertise experience previous works, which may lead to negative transfer in domain adaptations.

The computational complexity of DCBRW-based CDSL is in line with closeness centrality and betweenness centrality in $O(N^3)$.

5.4 Experiments

This section evaluates the effectiveness of DCBRW-based CDSL compared to the baselines of domain-specific network representation and CDNR. The experiment in this paper consists of two parts: the DCBRW parameter γ-learning and the node classification testing.

5.4.1 Datasets

The datasets chosen for the source domain are four real-world large-scale networks (Blog3, Facebook, PPI and wiki), and the dataset of the target domain network is a classic dataset in the graph research domain, LesM. The statistical details of the datasets are as shown in Table 5.1. The datasets in the source domain are in the multi-label classification setting, in which every node is assigned one or more labels

Algorithm 5.1: DCBRW-based CDSL

Input:

$G^t = (V^t, E^t)$ in the target domain and a group of networks $G^s = (V^s, E^s) \in \mathscr{G}^s$ in the source domain; and random walk repeated times r.

Output:

$\gamma^{*t} \rightarrow$ Optimized dual centrality parameter of target domain network; and $\{\gamma^{*s}\} \rightarrow$ a set of optimized dual centrality parameters of source domain networks.

DCBRW

for $\mathscr{G}^s$ and G^t **do**
  $C_i^{clo} \rightarrow$ Node closeness centrality by Eq. (5.1).
  $C_i^{bet} \rightarrow$ Node betweenness centrality by Eq. (5.2).
  $\gamma^{(0)} \rightarrow$ Initial γ by Lemma 1.
  $C_i \rightarrow$ Dual centrality by Eq. (5.3).
  $\mathbf{P}^{ref} = [p_{ij}^{ref}] \rightarrow$ Reference transition probability matrix on biased random walks by Eq. (5.5).
  $J_0 \rightarrow$ Set desired level by Lemma 2.
  while $J(\widetilde{P}_{st} || \widetilde{P}_{st}^{ref}) \rightarrow J_0$ **do**
    $\widetilde{P}_{st}(\mathcal{P}_{st}) \leftarrow$ DCBRW probability in minimizing costs by Eqs. (5.6)-(5.7).
  end while
  return $\gamma^* \rightarrow$ Optimized DCBRW parameter.
end for

CDSL

for $(\gamma^{*t}, \gamma^{*s})$ **do**
  $|dense(\gamma^{*t}) - dense(\gamma^{*s})| \leftarrow$ Dense similarity.
end for
$R \leftarrow$ Rank dense similarities.
return $G^{s*} \leftarrow$ The most relevant source domain network from $\mathscr{G}$ who is with $\min(R)$.

from a finite set. The dataset in the target domain has only one label for each node. All the datasets have multi-class labels.

Diameter is the shortest distance between the two most distant nodes in the network. When the shortest path length from every node to all other nodes has been calculated, the diameter is the longest path length of all the calculated path lengths.

Table 5.1 CDSL dataset statistics

Networks		Num. of Nodes	Num. of Links	Label Categories	Labels	Network Diameter	Ave. Degree
Blog	Citation Network	10,312	333,983	39	Interests	5	64.776
Facebook	Social Network	3,959	84,243	10	Groups	17	42.558
PPI	Biological Network	3,860	37,845	50	States	8	19.609
wiki	WWW	4,733	32,026	40	POS Tags	6	13.533
LesM	Novel Character Network	77	254	8	Appearance	5	6.597

- **Blog3**[1] **dataset** (Source domain network 1) is a social blog directory which manages bloggers and their blogs. Information about both the contact network and selected group membership is included. The network has 10,312 nodes, 333,983 undirected links, and 39 different labels, which excludes the isolated nodes and unlabeled nodes.

- **Facebook**[2] **dataset** (Source domain network 2) consists of circles (or friends lists) from Facebook. This dataset includes node features (profiles), circles and ego networks. The network has 3,959 nodes, 84,243 undirected links, and 10 different labels, which excludes the isolated nodes and unlabeled node.

- **PPI**[3] **dataset** (Source domain network 3) in this experiment is a subgraph of the PPI network for Homo Sapiens. The subgraph corresponds to the graph induced by nodes for which we could obtain labels from the hallmark gene sets and represent biological states. The network has 3,860 nodes, 37,845 undirected links, and 50 different labels, which excludes the isolated nodes and unlabeled node.

- **Wikipedia (wiki)**[4] **dataset** (Source domain network 4) is a co-occurrence network of words appearing in the first million bytes of the Wikipedia dump. The labels represent the POS tags. The network has 4,733 nodes, 32,026 undirected links, and 40 different labels, which excludes the isolated nodes and unlabeled node.

[1] http://socialcomputing.asu.edu/datasets/BlogCatalog3
[2] https://snap.stanford.edu/data/egonets-Facebook.html
[3] https://downloads.thebiogrid.org/BioGRID
[4] http://www.mattmahoney.net/dc/text.html

- **Les Misérables (LesM)**[5] **dataset** (Target domain network) is a network containing the co-occurances of characters in Victor Hugo's novel 'Les Misérables'. A node represents a character and a link between two nodes shows that these two characters appear in the same chapter of the book. The labels represent the appearance of the character. The network has 77 nodes, 254 links, and 8 different labels, which excludes the isolated nodes and unlabeled node.

5.4.2 Baselines

We implement the following algorithms for comparison. There are three domain-specific network representation algorithms and a CDNR algorithm.

- **DeepWalk** (Perozzi et al., 2014) is the first random walk-based network representation algorithm. By choosing DeepWalk, we exclude the matrix factorization approaches which have already been demonstrated to be inferior to DeepWalk.

- **Node2Vec** (Grover and Leskovec, 2016) learns continuous feature representations of nodes using a biased random walk procedure to capture the diversity of connectivity patterns observed in networks with the biased parameter α, which is controlled by the parameters of p and q.

- **Struc2Vec** (Ribeiro et al., 2017) learns node representations from structural identity by constructing a hierarchical graph to encode structural similarities and generating a structural context for nodes.

[5]https://github.com/gephi/gephi/wiki/Datasets

- **CDNR** learns node embeddings using the external random walk knowledge from the manually selected source domain and extends the biased random walk with parameter α_{pq}.

5.4.3 Setups

Our experiment evaluates the latent feature representations on a standard supervised learning task, i.e., linear SVM classification (Fung and Mangasarian, 2005). We choose the linear classifier instead of a non-linear classifier or sophisticated relational classifiers to reduce the impact of complicated learning approaches on classification performance. For the evaluations, we randomly partition the dataset in the target domain into two non-overlapping sets for training and testing by five groups of training percentages, $\{0.3, 0.4, \cdots, 0.7\}$. We repeat the above steps ten times and thus obtain ten copies of the training data and the testing data. The reported experiment results are the average of the ten runs. The random walk parameters are set as follows. Let the dimensions of feature representation be $d = 128$, the walk length be $l = 80$, the number of walks of every source node be $r = 10$, the window size be $k = 10$, $workers = 8$, and the search bias α_{pq} be with $p = 1$ and $q = 1$, which comprehensively follows previous settings in DeepWalk (Perozzi et al., 2014) and Node2Vec (Grover and Leskovec, 2016). For Struc2Vec as used in (Ribeiro et al., 2017), let OPT1 (reducing the length of degree sequences), OPT2 (reducing the number of pairwise similarity calculations) and OPT3 (reducing the number of layers) all be True values, and the maximum number of layers be six.

5.4.4 DCBRW Parameter γ-Learning

The γ-Learning results of the source network candidature networks and the target network are shown in Figures 5.3-5.7. The learning rate of γ is calculated based on $\gamma^{(0)} = 20$ which is large enough for learning convergence. The learning results are evaluated by $J(\widetilde{P}_{st}||\widetilde{P}_{st}^{ref}) - J_0$ where J_0 is set as $r = 10$. The convergence point γ^* is calculated by dense function $dense(\gamma)$. The rank results for the source domain network candidates on $|dense(\gamma^{*t}) - dense(\gamma^{*s})|$ are $R = \{$PPI, Facebook, wiki, Blog3$\}$ in ascending order. The PPI dataset is selected as the source domain network according to the DCBRW γ-learning result, which is a little different from the usual practice, in which PPI and LesM are in quite different domains, such that PPI will not be chosen as the source domain network. However, γ learns their similarity on the information distributions, which reflects that the network structure pattern is difficult to acquire in the usual way. The performance of the CDSL (source domain selection) is tested in the next section.

5.4.5 Result Analysis

The F1 score is designed to evaluate the effectiveness of category assignments by classifiers. We use Macro-F1 and Micro-F1 (Yang and Liu, 1999) to compare performance and the results are shown in Table 5.2. We make the following observations:

- The algorithms of CDNR and CDSL on FTLSIN outperform the domain-specific network representations of DeepWalk, Struc2Vec and Node2Vec. DeepWalk cannot output a latent feature space that the linear SVM classifier can work with.

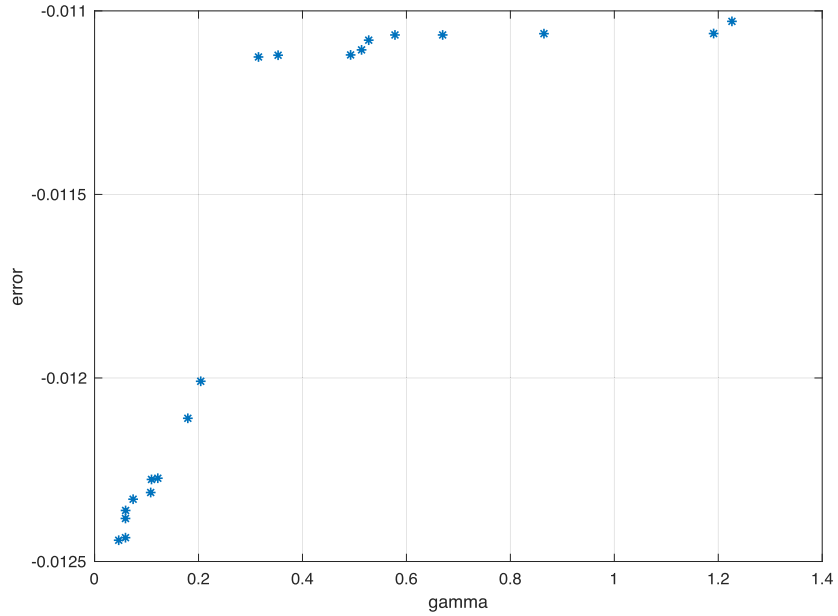


Figure 5.3 LesM γ-learning results: $\gamma^* = 2.0423E-01$

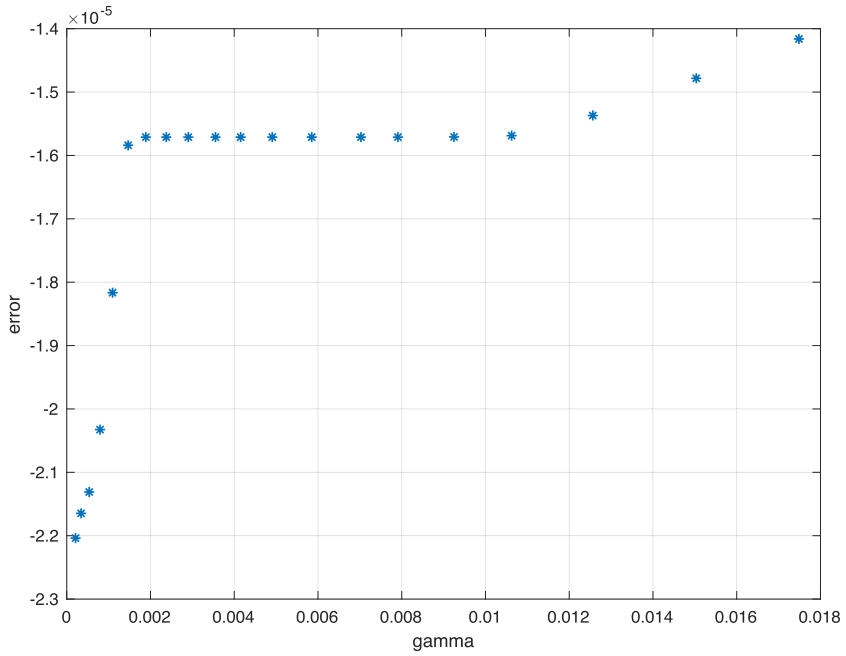


Figure 5.4 Facebook γ-learning results: $\gamma^* = 5.1496E-04$

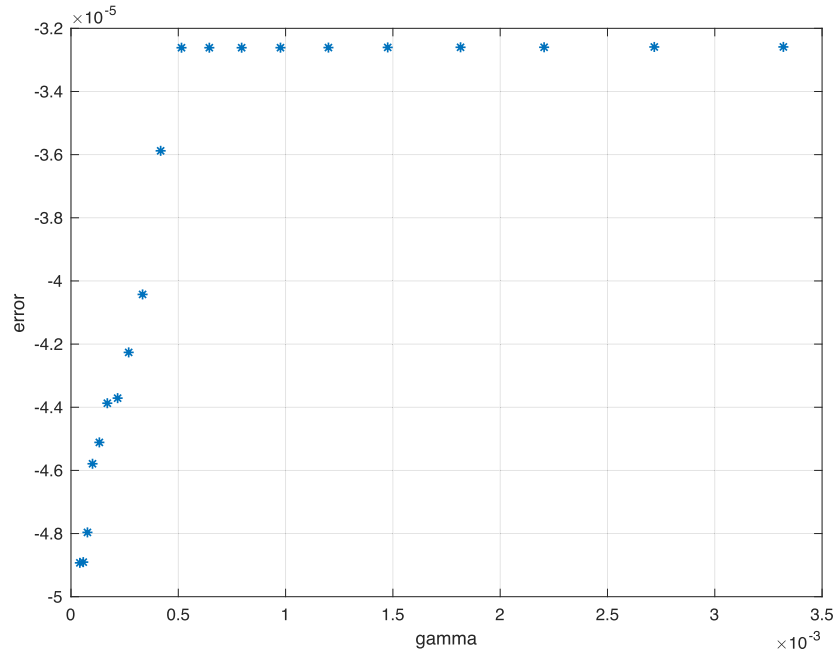


Figure 5.5 Blog3 γ-learning results: $\gamma^* = 1.8890E-03$

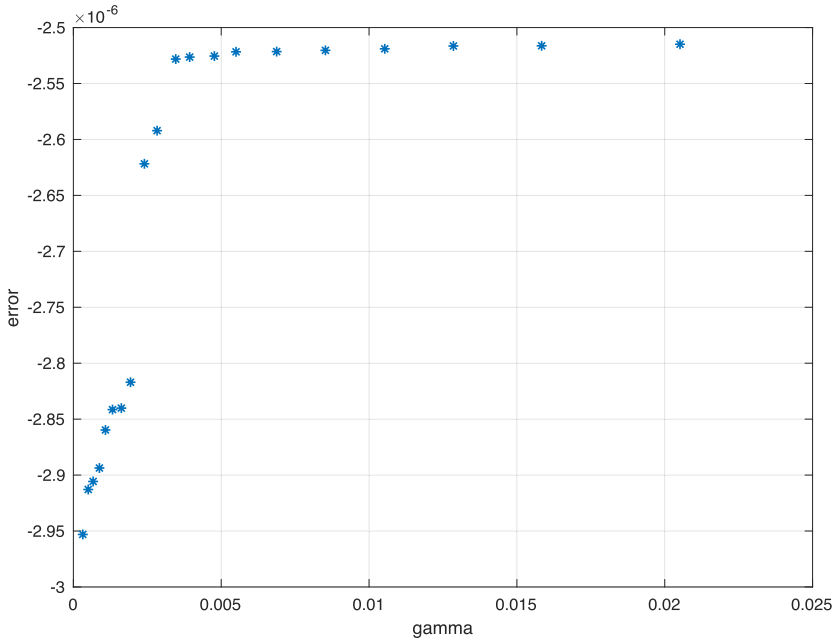


Figure 5.6 PPI γ-learning results: $\gamma^* = 3.4623E-03$

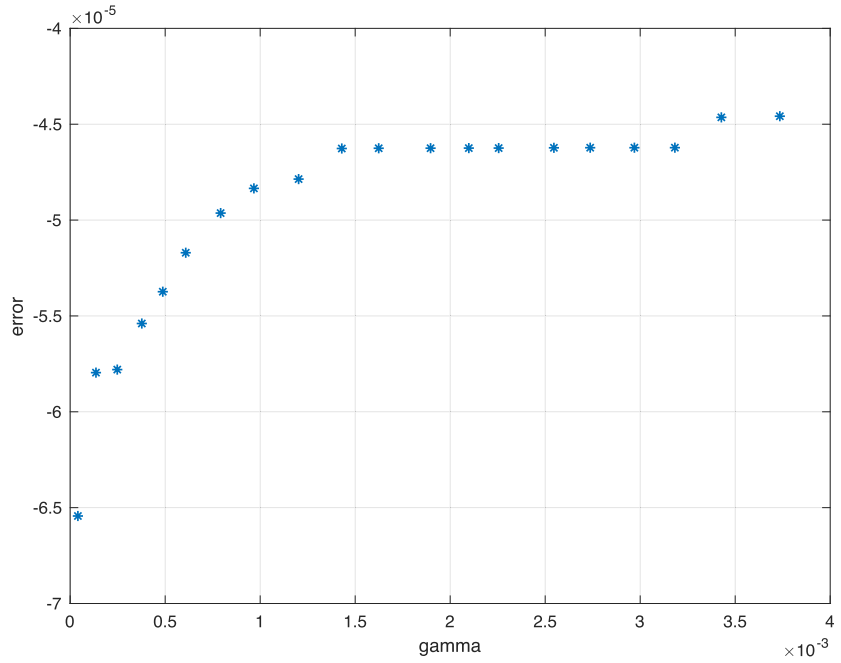


Figure 5.7 wiki γ-learning results: $\gamma^* = 1.4295E-03$

- CDNR achieves good performance on average, and CDSL has the best performance on the selected source domain network. Therefore, FTLSIN is important for adapting external knowledge such as learned biased random walks. The performance of CDNR shows that α_{pq} searches network local structure $N(v_i)$ properly but pre-training is required to obtain the setup values for p and q. The performance of CDSL proves that the individual similarity learning works for each network candidature in the source domain.

- The DCBRW-based CDSL on parameter γ chooses the correct source domain network for the LesM dataset. The PPI dataset with close dense distance to LesM achieves the best performance on the node classification testing. The rank of CDSL R is in line with the rank of the performance on the node classifications.

Table 5.2 CDSL classification results on the target domain network of LesM

Macro-F1	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**
Struc2Vec	0.345	0.384	0.389	0.357	0.582
Node2Vec	0.403	0.653	0.664	0.692	0.768
CDNR-Facebook	0.442	0.822	0.837	0.893	0.914
CDNR-wiki	0.369	0.693	0.749	0.855	0.882
CDNR-Blog3	0.389	0.725	0.739	0.823	0.835
CDNR-PPI	0.432	0.730	0.778	0.843	0.845
CDSL-Facebook	0.339	0.844	0.857	0.915	0.939
CDSL-wiki	0.357	0.452	0.532	0.780	0.790
CDSL-Blog3	0.311	0.517	0.675	0.726	0.757
CDSL-PPI	**0.483**	**0.923**	**0.937**	**0.948**	**0.962**
Micro-F1	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**
Struc2Vec	0.345	0.385	0.389	0.357	0.496
Node2Vec	0.604	0.653	0.664	0.692	0.768
CDNR-Facebook	0.427	0.822	0.837	0.893	0.914
CDNR-wiki	0.470	0.693	0.749	0.855	0.882
CDNR-Blog3	0.419	0.725	0.739	0.823	0.835
CDNR-PPI	0.487	0.730	0.778	0.843	0.845
CDSL-Facebook	0.362	0.844	0.857	0.915	0.939
CDSL-wiki	0.363	0.521	0.617	0.780	0.790
CDSL-Blog3	0.246	0.417	0.675	0.726	0.757
CDSL-PPI	**0.457**	**0.923**	**0.937**	**0.948**	**0.962**

- The node classification results on CDSL and CDNR, which adapt knowledge from the wiki dataset and Blog3 dataset, are lower than the results gained by Node2Vec in some training percentages. According to the usual practice, these two datasets may have some similarities with the LesM dataset because they are all in the publication domain. However, CDNR leads to negative transfers based on these experiment results.

5.5 Summary

In this chapter, we propose a solution of DCBRW-based CDSL. In contrast to previous network representation approaches, CDSL enables knowledge transfer from the external domains in the learned source network. A dual centrality is proposed to train the parameter γ and the domain similarity is learned based on γ. The experiment results show that DCBRW-based CDSL improves the performance of latent feature learning in real-world information networks compared to the state-of-the-art baselines, and the performance rank is in line with the similarity rank.

Chapter 6

Conclusion and Future Research

This chapter concludes the thesis and provides further research directions for this topic.

6.1 Conclusions

Random walk-based network representation has attracted much attention in the areas of data mining and machine learning due to its powerful knowledge representation and its flexible latent feature learning, especially for large-scale real-world information networks with structural sparsity. Transfer learning contributes to knowledge transfer across different datasets which differs in data distribution or feature spaces. It is powerful in learning efficiency compared with traditional machine learning approaches and requires less training data for the target domain hence avoiding the need to rebuild models for each learning task. The random walk-based network representations with knowledge adaptations is a newly emerging area, which we study in this thesis. The findings of this study are summarized as follows:

1. *The development of a novel framework of transferring structures across large-scale information networks (FTLSIN).*

FTLSIN is proposed for a new scenario in network representation of the target domain sparse network. It is designed on a two-layer random walk which includes a bottom layer for the network in the source domain and a top layer for the network in the target domain. The links are predicted between these two layers which construct the paths so that knowledge of the source domain network can be transferred to the target domain network. Knowledge adaptation across domains reduces learning difficulties caused by data sparsity and noise; and effectively improves the performance of latent feature learning in large-scale information networks.

2. *The development of a novel cross-domain network representation algorithm (CDNR) consists of a cross-domain node mapping procedure and a cross-domain walk mapping procedure.*

CDNR extends FTLSIN, especially in relation to the procedures developed for cross-domain node mapping and cross-domain walk mapping. The main advantage of CDNR is that when the network structure lacks information to generate good network representation, the two layers share knowledge by overcoming the challenges presented by the unbalanced scale of nodes and random walks. The cross-domain mappings incur a relatively low computational cost due to the node clustering process. The computational complexity of top-layer feature learning is in line with the domain-specific baselines which is challenging work for cross-domain learning tasks. The experiment results on the multi-label multi-class real-world datasets show that CDNR improves the performance of latent feature learning in large-scale information networks.

3. *The development of a novel cross-domain similarity learning algorithm (CDSL) is based on network patterns of node importance.*

Compared to previous network representation approaches, the proposed CDSL algorithm increases the flexibility of CDNR due to the intelligent calculation of the similarity for each pair of source domain network and target domain network. The proposed dual centrality indicator plays an important role in CDSL and network pattern mining. It is formed by a closeness centrality and a betweenness centrality which describes node importance in spreading and controlling the information flows over the global and local networks. By embedding dual centrality evaluation in biased random walk learning and optimizing the randomized shortest path search costs, the patterns of the networks in both domains are acquired so that the most relevant source domain network is selected for CDNR. The experiment results show that CDSL improves the performance of latent feature learning in real-world information networks compared to the state-of-the-art baselines, and the performance rank is in line with the performance rank of CDSL.

6.2 Future Study

This thesis identifies the following directions as future work:

- The current research focus on the transfer learning problem which is largely dependent on the exact network structure. If the network is sparse, large-scale, dynamically changing, hierarchical and even heterogeneous, then it might be challenging to apply traditional graph theory and machine learning algorithms, and some further research required in this aspect. Therefore, network pattern

analysis is required in many applications, e.g. node classification, recommender system and data visualization. A dynamical model with time-varying parameters can be considered and tested by using the developed algorithms in order to show the generality.

- The network patterns based on the node importance mining method, especially in transmission mechanisms, should fully consider the negative spread of information in cross-domain network representations. However, node power importance is rarely discussed. Power analysis of network patterns helps measure the negative transfer in cross-domain learning for network representations and controls the cross-domain knowledge adaption process. In future applications, power implemented algorithms will also be able to rank negative node importance, such as rumor spreading in social networks.

- Computational complexity is a crucial issue in applications that are related to cross-domain learning for network representations. The bad data problems, such as outliers, noisy, high-dimensional, stochastic, nonlinear, sparse and missing, exist universally in the real-world. It's highly desired to further explore the network modeling issues from bad or incomplete data structure. When applying multiple measurements to achieve cross-domain knowledge adaptation, computational complexity dramatically increases, which directly influences the method-application transform. Future work should address the algorithm optimizations to reduce costs in cross-domain network sampling and learning.

In summary, handling cross-domain learning for network representation is an urgent and important issue. It is a key technique in accomplishing artificial intelligence

learning tasks. Future research on transferable network representation techniques, especially for large-scale systems, has great prospects. Real-time experiments are going to be examined along with the simulation examples.

Bibliography

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

Adamic, L. A., Lukose, R. M., Puniyani, A. R., and Huberman, B. A. (2001). Search in power-law networks. *Physical Review E*, 64(4):046135.

Ahmed, A., Shervashidze, N., Narayanamurthy, S., Josifovski, V., and Smola, A. J. (2013). Distributed large-scale natural graph factorization. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 37–48. ACM.

Alderson, A. S. and Beckfield, J. (2004). Power and position in the world city system. *American Journal of Sociology*, 109(4):811–851.

Allen, J. (2009). Three spaces of power: Territory, networks, plus a topological twist in the tale of domination and authority. *Journal of Power*, 2(2):197–212.

Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., and Spyropoulos, C. D. (2000). An experimental comparison of naïve Bayesian and keyword-based anti-spam

filtering with personal e-mail messages. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 160–167. ACM.

Arnold, A., Nallapati, R., and Cohen, W. W. (2007). A comparative study of methods for transductive transfer learning. In *Proceedings of the 7th IEEE International Conference on Data Mining Workshops*, pages 77–82. IEEE.

Arribas-Bel, D., Kourtit, K., and Nijkamp, P. (2013). Benchmarking of world cities through self-organizing maps. *Cities*, 31:248–257.

Baktashmotlagh, M., Harandi, M. T., Lovell, B. C., and Salzmann, M. (2014). Domain adaptation on the statistical manifold. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2481–2488.

Bandyopadhyay, A. and Gamarnik, D. (2008). Counting without sampling: Asymptotics of the log-partition function for certain statistical physics models. *Random Structures & Algorithms*, 33(4):452–479.

Barabási, A.-L. (2009). Scale-free networks: A decade and beyond. *Science*, 325(5939):412–413.

Barabási, A.-L. and Pósfai, M. (2016). *Network Science*. Cambridge University Press.

Baralis, E., Chiusano, S., and Garza, P. (2008). A lazy approach to associative classification. *IEEE Transactions on Knowledge and Data Engineering*, 20(2):156–171.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. (2010). A theory of learning from different domains. *Machine Learning*, 79(1):151–175.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828.

Bhagat, S., Cormode, G., and Muthukrishnan, S. (2011). Node classification in social networks. In *Social Network Data Analytics*, pages 115–148. Springer.

Bonacich, P. (1987). Power and centrality: A family of measures. *American Journal of Sociology*, 92(5):1170–1182.

Borgatti, S. P. and Everett, M. G. (2006). A graph-theoretic perspective on centrality. *Social Networks*, 28(4):466–484.

Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., and Erhan, D. (2016). Domain separation networks. In *Proceedings of Advances in Neural Information Processing Systems 29*, pages 343–351.

Boyd, J. P., Mahutga, M. C., and Smith, D. A. (2013). Measuring centrality and power recursively in the world city network: A reply to Neal. *Urban Studies*, 50(8):1641–1647.

Breitkreutz, B.-J., Stark, C., Reguly, T., Boucher, L., Breitkreutz, A., Livstone, M., Oughtred, R., Lackner, D. H., Báhler, J., Wood, V., et al. (2008). The bioGRID interaction database. *Nucleic Acids Research*, 36(1):D637–D640.

Cao, Y., Long, M., and Wang, J. (2018). Unsupervised domain adaptation with distribution matching machines. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 2795–2802.

Chaker, R., Al Aghbari, Z., and Junejo, I. N. (2017). Social network model for crowd anomaly detection and localization. *Pattern Recognition*, 61:266–281.

Chen, B., Lam, W., Tsang, I., and Wong, T.-L. (2009). Extracting discriminative concepts for domain adaptation in text mining. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 179–188. ACM.

Chen, C., Tong, H., Prakash, B., Tsourakakis, C., Eliassi-Rad, T., Faloutsos, C., and Chau, D. (2016). Node immunization on large graphs: Theory and algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 28(1):113–126.

Choi, K., Toh, K.-A., and Byun, H. (2012). Incremental face recognition for large-scale social network services. *Pattern Recognition*, 45(8):2868–2883.

Ciaramita, M. and Chapelle, O. (2010). Adaptive parameters for entity recognition with perceptron HMMs. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 1–7. Association for Computational Linguistics.

Cook, K. S., Emerson, R. M., Gillmore, M. R., and Yamagishi, T. (1983). The distribution of power in exchange networks: Theory and experimental results. *American Journal of Sociology*, 89(2):275–305.

Corley, H. and David, Y. (1982). Most vital links and nodes in weighted networks. *Operations Research Letters*, 1(4):157–160.

Dai, W., Xue, G.-r., Yang, Q., and Yu, Y. (2007). Transferring naïve Bayes classifiers for text classification. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, pages 540–545. Citeseer.

De Domenico, M., Solé-Ribalta, A., Omodei, E., Gómez, S., and Arenas, A. (2015). Ranking in interconnected multilayer networks reveals versatile nodes. *Nature Communications*, 6:6868.

Derudder, B., Vereecken, L., and Witlox, F. (2004). An appraisal of the use of airline data in assessments of the world city network. *Urban Studies*, 42(13):2371.

Ding, S., Lin, L., Wang, G., and Chao, H. (2015). Deep feature learning with relative distance comparison for person re-identification. *Pattern Recognition*, 48(10):2993–3003.

Du, Y., Gao, C., Hu, Y., Mahadevan, S., and Deng, Y. (2014). A new method of identifying influential nodes in complex networks based on TOPSIS. *Physica A: Statistical Mechanics and its Applications*, 399:57–69.

Duan, Y. and Lu, F. (2013). Structural robustness of city road networks based on community. *Computers, Environment and Urban Systems*, 41:75–87.

Feng, R., Yang, Y., Hu, W., Wu, F., and Zhuang, Y. (2018). Representation learning for scale-free networks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 282–289.

Freeman, L. C. (1977). A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41.

Freeman, L. C. (1978). Centrality in social networks conceptual clarification. *Social Networks*, 1(3):215–239.

Frideman, J. (1986). The world city hypothesis: Development & change. *Urban Studies*, 23(2):59–137.

Fung, G. M. and Mangasarian, O. L. (2005). Multicategory proximal support vector machine classifiers. *Machine Learning*, 59(1-2):77–97.

Fung, G. P. C., Yu, J. X., Lu, H., and Yu, P. S. (2006). Text classification without negative examples revisit. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):6–20.

Gao, J., Fan, W., Jiang, J., and Han, J. (2008). Knowledge transfer via multiple model local structure mapping. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 283–291. ACM.

Goh, K.-I., Oh, E., Kahng, B., and Kim, D. (2003). Betweenness centrality correlation in social networks. *Physical Review E*, 67(1):017101.

Grover, A. and Leskovec, J. (2016). 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.

Guo, T. and Zhu, X. (2014). Super-graph classification. In *Proceedings of the 18th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 323–336. Springer.

Hadlak, S., Schumann, H., Cap, C. H., and Wollenberg, T. (2013). Supporting the visual analysis of dynamic networks by clustering associated temporal attributes. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2267–2276.

Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. In *Proceedings of the 31st Advances in Neural Information Processing Systems*, pages 1025–1035.

Heckman, J. et al. (2013). Sample selection bias as a specification error. *Applied Econometrics*, 31(3):129–137.

Hennemann, S. and Derudder, B. (2014). An alternative approach to the calculation and analysis of connectivity in the world city network. *Environment and Planning B: Planning and Design*, 41(3):392–412.

Hsu, C.-W. and Lin, C.-J. (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425.

Hu, J., Du, Y., Mo, H., Wei, D., and Deng, Y. (2016). A modified weighted TOPSIS to identify influential nodes in complex networks. *Physica A: Statistical Mechanics and its Applications*, 444:73–85.

Hu, P., Fan, W., and Mei, S. (2015). Identifying node importance in complex networks. *Physica A: Statistical Mechanics and its Applications*, 429:169–176.

Huang, J., Gretton, A., Borgwardt, K. M., Schölkopf, B., and Smola, A. J. (2007). Correcting sample selection bias by unlabeled data. In *Proceedings of Advances in Neural Information Processing Systems 20*, pages 601–608.

Huang, X., Li, J., and Hu, X. (2017). Label informed attributed network embedding. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 731–739. ACM.

Jancura, P. and Marchiori, E. (2010). Dividing protein interaction networks for modular network comparative analysis. *Pattern Recognition Letters*, 31(14):2083–2096.

Jeong, M., Lin, C.-Y., and Lee, G. G. (2009). Semi-supervised speech act recognition in emails and forums. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, volume 3, pages 1250–1259. Association for Computational Linguistics.

Jia, Z., Li, Y., Jingru, L., and Peng, Z. (2015). Identifying influential nodes based on graph signal processing in complex networks. *Chinese Physics B*, 24(5):058904.

Jiang, J. and Zhai, C. (2007a). Instance weighting for domain adaptation in NLP. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, volume 7, pages 264–271.

Jiang, J. and Zhai, C. (2007b). A two-stage approach to domain adaptation for statistical classifiers. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management*, pages 401–410. ACM.

Kermarrec, A.-M., Le Merrer, E., Sericola, B., and Trédan, G. (2011). Second order centrality: Distributed assessment of nodes criticity in complex networks. *Computer Communications*, 34(5):619–628.

Kononenko, I. (1993). Inductive and Bayesian learning in medical diagnosis. *Applied Artificial Intelligence an International Journal*, 7(4):317–337.

Konstantakis, K. N., Michaelides, P. G., Tsionas, E. G., and Minou, C. (2015). System estimation of GVAR with two dominants and network theory: Evidence for BRICs. *Economic Modelling*, 51:604–616.

Kuncheva, L. I. and Rodriguez, J. J. (2007). Classifier ensembles with a random linear oracle. *IEEE Transactions on Knowledge and Data Engineering*, 19(4):500–508.

Kurant, M., Markopoulou, A., and Thiran, P. (2011). Towards unbiased BFS sampling. *IEEE Journal on Selected Areas in Communications*, 29(9):1799–1809.

Lai, D., Lu, H., and Nardini, C. (2010). Enhanced modularity-based community detection by random walk network preprocessing. *Physical Review E*, 81(6):066118.

Latour, B. (2005). *Reassembling the Social: An Introduction to Actor-Network-Theory*. Oxford University Press.

Lee, J., Kim, H., Lee, J., and Yoon, S. (2017). Transfer learning for deep learning on graph-structured data. In *Proceedings of the 31st Association for the Advancement of Artificial Intelligence*, pages 2154–2160.

Lelis, L. H., Otten, L., and Dechter, R. (2013). Predicting the size of depth-first branch and bound search trees. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 594–600.

Leskovec, J., Kleinberg, J., and Faloutsos, C. (2005). Graphs over time: Densification laws, shrinking diameters and possible explanations. In *Proceedings of the 11st ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pages 177–187. ACM.

Leskovec, J. and Mcauley, J. J. (2012). Learning to discover social circles in ego networks. In *Proceedings of the 26th Advances in Neural Information Processing Systems*, pages 539–547.

Lewis, D. D. (1992). *Representation and Learning in Information Retrieval*. PhD thesis, University of Massachusetts at Amherst.

Li, W., Duan, L., Xu, D., and Tsang, I. W. (2014). Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1134–1148.

Li, Y., Sha, C., Huang, X., and Zhang, Y. (2018a). Community detection in attributed graphs: An embedding approach. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 338–345.

Li, Z., Wei, Y., Zhang, Y., and Yang, Q. (2018b). Hierarchical attention transfer network for cross-domain sentiment classification. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5852–5859.

Liu, J.-G., Ren, Z.-M., Guo, Q., and Wang, B.-H. (2013). Node importance ranking of complex networks. *Acta Physica Sinica*, 62(17):178901–178901.

Liu, W. and Lü, L. (2010). Link prediction based on local random walk. *Europhysics Letters*, 89(5):58007.

Liu, X. and Derudder, B. (2012). Two-mode networks and the interlocking world city network model: A reply to Neal. *Geographical Analysis*, 44(2):171–173.

Liu, X., Derudder, B., and Taylor, P. (2014a). Mapping the evolution of hierarchical and regional tendencies in the world city network, 2000–2010. *Computers, Environment and Urban Systems*, 43:51–66.

Liu, X., Derudder, B., Witlox, F., and Hoyler, M. (2014b). Cities as networks within networks of cities: The evolution of the city/firm-duality in the world city network, 2000–2010. *Tijdschrift Voor Economische en Sociale Geografie*, 105(4):465–482.

Lu, J., Behbood, V., Hao, P., Zuo, H., Xue, S., and Zhang, G. (2015). Transfer learning using computational intelligence: A survey. *Knowledge-based Systems*, 80:14–23.

Lu, J., Xuan, J., Zhang, G., and Luo, X. (2018). Structural property-aware multilayer network embedding for latent factor analysis. *Pattern Recognition*, 76:228–241.

Ma, J., Cui, P., and Zhu, W. (2018). DepthLGP: Learning embeddings of out-of-sample nodes in dynamic networks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 370–377.

Ma, Y., Luo, G., Zeng, X., and Chen, A. (2012). Transfer learning for cross-company software defect prediction. *Information and Software Technology*, 54(3):248–256.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Moon, S. and Carbonell, J. G. (2017). Completely heterogeneous transfer learning with attention - what and what not to transfer. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 2508–2514.

Neal, Z. (2011). Differentiating centrality and power in the world city network. *Urban Studies*, 48(13):2733–2748.

Neal, Z. (2012). Structural determinism in the interlocking world city network. *Geographical Analysis*, 44(2):162–170.

Neal, Z. (2013a). Brute force and sorting processes: Two perspectives on world city network formation. *Urban Studies*, 50(6):1277–1291.

Neal, Z. (2013b). Does world city network research need eigenvectors? *Urban Studies*, 50(8):1648–1659.

Neal, Z. (2014). Validity in world city network measurements. *Tijdschrift Voor Economische en Sociale Geografie*, 105(4):427–443.

Newman, M. (2010). *Networks: An introduction*. Oxford University Press.

Newman, M. E. (2001). Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality. *Physical Review E*, 64(1):016132.

Newman, M. E. (2005a). A measure of betweenness centrality based on random walks. *Social Networks*, 27(1):39–54.

Newman, M. E. (2005b). Power laws, Pareto distributions and Zipf's law. *Contemporary Physics*, 46(5):323–351.

Niu, Q., Zeng, A., Fan, Y., and Di, Z. (2015). Robustness of centrality measures against network manipulation. *Physica A: Statistical Mechanics and its Applications*, 438:124–131.

Noh, J. D. and Rieger, H. (2004). Random walks on complex networks. *Physical Review Letters*, 92(11):118701.

Paalanen, P., Kamarainen, J.-K., Ilonen, J., and Kälviäinen, H. (2006). Feature representation and discrimination based on Gaussian mixture model probability densities—practices and algorithms. *Pattern Recognition*, 39(7):1346–1358.

Pan, S., Wu, J., Zhu, X., Zhang, C., and Wang, Y. (2016). Tri-party deep network representation. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 1895–1901.

Pan, S. J., Ni, X., Sun, J.-T., Yang, Q., and Chen, Z. (2010). Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th International Conference on World Wide Web*, pages 751–760. ACM.

Pan, S. J., Tsang, I. W., Kwok, J. T., and Yang, Q. (2011). Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210.

Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Pei, S. and Makse, H. A. (2013). Spreading dynamics in complex networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2013(12):P12002.

Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). 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.

Poulin, R., Boily, M.-C., and Mâsse, B. R. (2000). Dynamical systems to define centrality in social networks. *Social Networks*, 22(3):187–220.

Ribeiro, L. F., Saverese, P. H., and Figueiredo, D. R. (2017). Struc2Vec: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 385–394. ACM.

Roy, D. M. and Kaelbling, L. P. (2007). Efficient Bayesian task-level transfer learning. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, volume 7, pages 2599–2604.

Sagae, K. (2010). Self-training without reranking for parser domain adaptation and its impact on semantic role labeling. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 37–44. Association for Computational Linguistics.

Saito, K., Kimura, M., Ohara, K., and Motoda, H. (2012). Efficient discovery of influential nodes for SIS models in social networks. *Knowledge and Information Systems*, 20(3):613–635.

Sanderson, M. R., Derudder, B., Timberlake, M., and Witlox, F. (2015). Are world cities also world immigrant cities? An international, cross-city analysis of global centrality and immigration. *International Journal of Comparative Sociology*, 56(3–4):173–197.

Sandu, O., Carenini, G., Murray, G., and Ng, R. (2010). Domain adaptation to summarize human conversations. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 16–22. Association for Computational Linguistics.

Sassen, S. (2010). Global inter-city networks and commodity chains: Any intersections? *Global Networks*, 10(1):150–163.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47.

Shi, L., Mihalcea, R., and Tian, M. (2010). Cross language text classification by model translation and semi-supervised learning. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1057–1067. Association for Computational Linguistics.

Sigler, T. J. and Martinus, K. (2017). Extending beyond 'world cities' in World City Network (WCN) research: Urban positionality and economic linkages through the

Australia-based corporate network. *Environment and Planning A*, 49(12):2916–2937.

Solé-Ribalta, A., De Domenico, M., Gómez, S., and Arenas, A. (2016). Random walk centrality in interconnected multilayer networks. *Physica D: Nonlinear Phenomena*, 323:73–79.

Sugiyama, M., Nakajima, S., Kashima, H., Buenau, P. V., and Kawanabe, M. (2008). Direct importance estimation with model selection and its application to covariate shift adaptation. In *Proceedings of Advances in Neural Information Processing Systems 21*, pages 1433–1440.

Sun, Q., Qiao, Y., Wang, J., and Shen, S. (2016a). Node importance evaluation method in wireless sensor network based on energy field model. *EURASIP Journal on Wireless Communications and Networking*, 2016(1):199.

Sun, Q., Qiao, Y., Wang, J., and Shen, S. (2016b). Node importance evaluation method in wireless sensor network based on energy field model. *EURASIP Journal on Wireless Communications and Networking*, page 199.

Suykens, J. A. and Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300.

Tan, S., Cheng, X., Wang, Y., and Xu, H. (2009). Adapting naïve Bayes to domain adaptation for sentiment analysis. *Advances in Information Retrieval*, pages 337–349.

Tan, S., Wang, Y., Wu, G., and Cheng, X. (2008). Using unlabeled data to handle domain-transfer problem of semantic detection. In *Proceedings of the 2008 ACM Symposium on Applied Computing*, pages 896–903. ACM.

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., and Mei, Q. (2015). LINE: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077. ACM.

Tang, L. and Liu, H. (2009). 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.

Tsuboi, Y., Kashima, H., Hido, S., Bickel, S., and Sugiyama, M. (2009). Direct density ratio estimation for large-scale covariate shift adaptation. *Journal of Information Processing*, 17:138–155.

Tu, C., Zhang, W., Liu, Z., and Sun, M. (2016). Max-margin DeepWalk: Discriminative learning of network representation. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pages 3889–3895.

Wang, D., Cui, P., and Wenwu, Z. (2018a). Deep asymmetric transfer network for unbalanced domain adaptation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 443–450.

Wang, H., Wang, J., Wang, J., Zhao, M., Zhang, W., Zhang, F., Xie, X., and Guo, M. (2018b). GraphGAN: Graph representation learning with generative adversarial nets. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 2508–2515.

Wang, S. (2014). Research on the methods of identifying key node based on classical networks. *Applied Mechanics & Materials*, 530–531:489–495.

Wang, X., Cui, P., Wang, J., Pei, J., Zhu, W., and Yang, S. (2017). Community preserving network embedding. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 203–209.

Wang, Z., Ye, X., Wang, C., Wu, Y., Wang, C., and Liang, K. (2018c). RSDNE: Exploring relaxed similarity and dissimilarity from completely-imbalanced labels for network embedding. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 475–482.

Wu, J., Cai, Z., and Ao, S. (2012). Hybrid dynamic k–nearest–neighbour and distance and attribute weighted method for classification. *International Journal of Computer Applications in Technology*, 43(4):378–384.

Wu, J., Hong, Z., Pan, S., Zhu, X., Cai, Z., and Zhang, C. (2014). Multi-graph-view learning for graph classification. In *Proceedings of the 2014 IEEE International Conference on Data Mining*, pages 590–599. IEEE.

Wu, J., Pan, S., Zhu, X., Zhang, C., and Philip, S. Y. (2018). Multiple structure-view learning for graph classification. *IEEE Transactions on Neural Networks and Learning Systems*, 29(7):3236–3251.

Xie, F., Lin, J., and Cui, W. (2015). Exploring express delivery networks in China based on complex network theory. *Complexity*, 21(2):166–179.

Xue, G.-R., Dai, W., Yang, Q., and Yu, Y. (2008). Topic-bridged PLSA for cross-domain text classification. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 627–634. ACM.

Xue, S., Lu, J., Zhang, G., and Xiong, L. (2018). A framework of transferring structures across large-scale information networks. In *Proceedings of 2018 International Joint Conference on Neural Networks*, pages 1137–1142. IEEE.

Xue, S., Xiong, L., Lu, Z., and Wu, J. (2017). Graph-theoretic node importance mining in world city networks: Methods and applications. *Information Discovery and Delivery*, 45(2):57–65.

Yang, C., Liu, Z., Zhao, D., Sun, M., and Chang, E. Y. (2015). Network representation learning with rich text information. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 2111–2117.

Yang, J. and Leskovec, J. (2015). Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1):181–213.

Yang, Y. and Liu, X. (1999). A re-examination of text categorization methods. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 42–49. ACM.

Yin, X., Han, J., Yang, J., and Yu, P. S. (2006). Efficient classification across multiple database relations: A crossmine approach. *IEEE Transactions on Knowledge and Data Engineering*, 18(6):770–783.

Zhang, Z., Wang, H., Wang, C., and Fang, H. (2015). Modeling epidemics spreading on social contact networks. *IEEE Transactions on Emerging Topics in Computing*, 3(3):410–419.

Zhou, L., Yang, Y., Ren, X., Wu, F., and Zhuang, Y. (2018). Dynamic network embedding by modeling triadic closure process. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 571–578.

Zhu, P., Zuo, W., Zhang, L., Hu, Q., and Shiu, S. C. (2015). Unsupervised feature selection by regularized self-representation. *Pattern Recognition*, 48(2):438–446.

Abbreviations

arXivCit-HepPh	arXiv High-energy Physics Citation Network
arXivCit-HepTh	arXiv High-energy Physics Theory Citation Network
Blog3	BlogCatalog3
CD2LRW	**C**ross-**D**omain **2**-**L**ayer **R**andom **W**alk
CDNR	**C**ross-**D**omain **N**etwork **R**epresentation
CDSL	**C**ross-**D**omain **S**imilarity **L**earning
DCBRW	**D**ual **C**entrality based **B**iased **R**andom **W**alk
EM	**E**xpectation **M**aximization
FTLSIN	**F**ramework of **T**ransferring Structures across **L**arge-**S**cale **I**nformation **N**etwork
KL	**K**ullback-**L**eibler
KLIEP	**K**ullback-**L**eibler **I**mportance **E**stimation **P**rocedure
KMM	**K**ernel **M**ean **M**atching
LesM	**Les** **M**isérables
MMD	**M**aximum **M**ean **D**iscrepancy
NBTL	**N**aïve **B**ayes **T**ransfer **L**earning
NLP	**N**atural **L**anguage **P**rocessing
non-i.i.d.	**non**-**i**ndependent and **i**dentically **d**istributed

POS	**P**art-**o**f-**S**peech
PPI	**P**rotein-**P**rotein **I**nteractions
SI	**S**usceptible-**I**nfected
SIR	**S**usceptible-**I**nfected-**R**ecovery
SIS	**S**usceptible-**I**nfected-**S**usceptible
$\mathscr{D}^s$	source domain
SVM	**S**upport **V**ector **M**achine
$\mathscr{D}^t$	target domain
TNB	**T**ransfer **N**aïve **B**ayes
TOPSIS	**T**echnique for **O**rder of **P**reference by **S**imilarity to **I**deal **S**olution
WCN	**W**orld **C**ity **N**etwork
wiki	Wikipedia