Context-Based Social Trust Relationship Prediction

By

Rongwei Xu

A thesis submitted to Macquarie University
for the degree of Master of Research
School of Computing
September 2022

Examiner's Copy

Typeset in $\LaTeX\,2_\varepsilon$.

Statement of Originality

This work has not previously been submitted for a degree or diploma in any university. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

(Signed)________________________ Date: 05/09/2022________________________

Rongwei Xu

Acknowledgements

I would like to express my gratitude to my supervisor, Dr. Guanfeng Liu, for his invaluable advice, continuous guidance, and patience during my research. I would also like to thank my family and friends for their encouragement.

Abstract

Social trust relationship prediction targets using attributes to quantify the interrelationships in trust between users, and this trust can be applied in both decision-making and product recommendation. Most of the existing algorithms do not consider the heterogeneity and the semantics of information included in online social networks, leading to low adaptability in capturing user preferences. What's more, they only focus on directly connected nodes and treat all the information propagation paths equally, leading to the lack of structured context information. Given the incomplete graph structure on online social networks constructed by existing algorithms, they can hardly have good performance in the trust prediction.

To solve the above-mentioned problems, we propose a novel Context-based Social Trust Relation Prediction (CSTRP) model, which can capture different features on both nodes and paths adaptively based on the complex contexts and take multi-hop neighbours into account. Specifically, in our model, we construct a heterogeneous graph of three kinds of nodes: User, Interest, and Relationship, as well as two different meta-paths: User-Interest-User, and User-Relative-User. Then, we adopt a two-level attention mechanism to obtain the attention value on both the node-level and path-level. To incorporate the multi-hop neighbours' information, we develop a 2-hop attention diffusion to aggregate the information from the indirectly connected nodes. The experimental results on real-world datasets have demonstrated that CSTRP outperforms the state-of-the-art methods in terms of the accuracy of social trust prediction.

Contents

List of Figures

List of Tables

1 Introduction

An online social network can be modeled as a network system shaped by social relationships between users of the network. The interrelationships between users can include relatives and friends, actions, and sending and receiving information. With the development of new media and the advent of the 5G era, online social networks have become popular and influential. Users have been accustomed to using online social networks to exchange ideas, share information, and transfer knowledge in work and life. This flood of context information on the Internet changes and updates the ideas of people, which ultimately affects users' choices of recommended products. In this way, users could influence each other in the online social network.

Trust relationship reflects the credibility and reliability between users, and users could interplay through their trust relationship in online social networks. Trust can have an influence on a user's decision-making, including product purchasing and recommendations, etc. For example, people tend to accept suggestions or recommendations from whom they trust. However, most users in online social networks are unknown to each other. Thus, we could not find an explicit trust relationship between them. Even though users in the online social network do not directly connect with each other, they can still establish a trust relationship with each other based on trust inference properties [1]. As shown in Figure 1.1, if user A trusts user D and they are both sports lovers, A tends to buy the sports brands like Nike and Adidas recommended by D. If A can establish a trust relationship with E based on the trust inference, then the mobile phone brand recommended by E is more likely to be successfully accepted by A. *Thus, it is essential to find an effective way to perform the accurate trust prediction within users.* In addition, trust evaluation is not only based on the users' similarity measurement, but also affected by the relationship between users and the preferences of users, etc. Therefore, trust evaluation is a challenging task in making decisions.

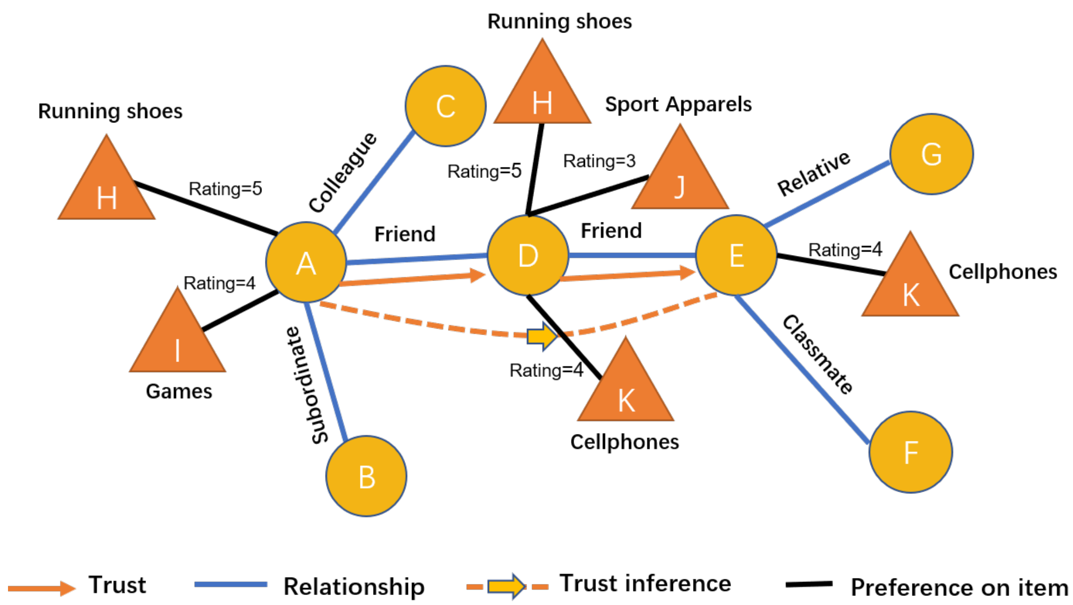

Figure 1.1: Example of users interplay through their trust relationships in online social networks.

1.1 Motivation and Challenge

Most of the existing studies in the literature for the trust prediction in online social networks suffer from two drawbacks causing the low prediction accuracy, including 1) neglecting the heterogeneous and the semantically latent relationships and 2) ignoring the multi-hop neighbourhood context on the whole graph structure. For the first drawback, the existing studies do not take the heterogeneous and the semantic relationships between users as well as users and items into account, which ignores the different weights between attributes and the structure of nodes. On the one hand, an online social network can contain different kinds of relationships, such as friends, colleagues, relatives, etc., and these relationships result in different degrees of trust. For example, as shown in Figure 1.1, user D has more influence on user A compared with user C, since users A and D are close friends while users A and C are only colleagues. On the other hand, users inherently have different levels of preference corresponding to different items. If these different levels of users' preferences on items cannot be adaptively captured, it can hardly deliver an accurate trust prediction. For example, as shown in Figure 1.1, user A is a sports lover and buys lots of sports products. Although he/she also buys electronic products once or twice, he/she has no preference for electronic products. Therefore, compared with electronics enthusiasts, sports lovers tend to trust user A's recommendation. These different relationships between users and users' preferences on items should also be considered in trust prediction. Besides, these semantic social relationships between users and their preferences on items belong to heterogeneous information. At first, users could purchase items and then forms a graph with the purchasing relations. In addition, the interactions between users can form another graph with the interaction relations. These two types of graphs have different types of nodes, attributes and links, together forming a heterogeneous graph, where the heterogeneity of information included in the online social networks should be also considered in trust prediction.

For the second drawback, most of the existing studies ignore the information transferred from directly connected nodes to indirectly connected nodes. In real scenarios, the social context information, including social relationships and preferences, can be transferred and propagated between users. This transfer of information exists between the indirectly connected users, and users can interplay through this multi-hop neighbouring context. For example, a friend of a user's friend can also be a friend of the user. Thus, the user can also establish a trust relationship with his/her friend's friend. As shown in Figure 1.1, if the propagated information is only limited to the directly connected users (e.g., user A and user D, user D and user E), it will lead to an incomplete graph structure due to the lack of the information propagated from indirectly connected users A and E.

Based on the above-mentioned problems in the existing studies, we target to tackle the missing information problems on the heterogeneous and semantically latent relationships (inadaptive weights on paths and nodes) and the multi-hop neighbourhood context on the graph structure (inefficient attention hops on propagation). In this paper, we propose a novel Context-based Social Trust Relationship Prediction (CSTRP) model. This method can adaptively capture the structure information as well as semantic information between directly and non-directly connected nodes. Incorporating this substantial contextual information to add more features to users will strengthen the dynamic of the real-time characteristic of the model on trust prediction.

The process flowchart with all the methodological components of CSTRP is shown in Figure 1.2. Our model first adopts PMF and Doc2vec to obtain the latent vectors from users' ratings and reviews. Then, our model leverages the attention mechanism and multi-hop attention diffusion mechanism to build the heterogeneous knowledge graph with triples, i.e., user-rating-item. Furthermore, our model employs a pairwise neural network to project the features of users into lower dimension space. Finally, our model computes the cosine similarity based on these users' latent representations and predicts their trust relationships.

1.2 Contributions

Our contributions can be summed as follows.

- Our method conducts the trust relationship prediction with a combination of heterogeneous attention mechanism and multi-hop diffusion mechanism based on the pairwise neural network.

- Our model is a context-dependent model that preserves the abundant information of the network structure and semantics simultaneously, including the information on homophily and diffusion.

- For the user modeling, we take users and their interests as well as relationships as different types of nodes. Then we use an attention-based method to assign the multi-attention weights between nodes and edges, allowing different weights to be allocated to different paths based on their influence on trust prediction.

- We develop a CSTRP algorithm that could capture the directly connected relationship and indirectly connected relationship through the diffusion layer. By introducing the 2-hop attention diffusion mechanism, we can aggregate more information from non-neighbouring nodes, optimizing the network structure representation.

- We have conducted experiments on the real-world dataset Epinions and experimental results have illustrated that our model can outperform all the state-of-the-art methods.

1.3 Thesis Organisation and Overview

The rest structure of the thesis is as follows.

- Chapter 2 provides the related work and the analysis on the embedding and trust prediction of social networks.

- Chapter 3 explains the key concept and terms of the heterogeneous graph in this paper, and details the framework of our research method, which incorporates an attention mechanism and a 2-hop diffusion mechanism as well as a pairwise neural network.

- Chapter 4 illustrates the experiment and analysis. The chapter starts with four questions to investigate the different aspects of the performance of our model. Next, it introduces the dataset, the experiment setting, baselines and the implementation. Finally, it focuses on the results comparison and the analysis.

- Chapter 5 discusses our conclusion and feature research direction. This chapter summarizes the work of the whole thesis and elaborates on future directions.

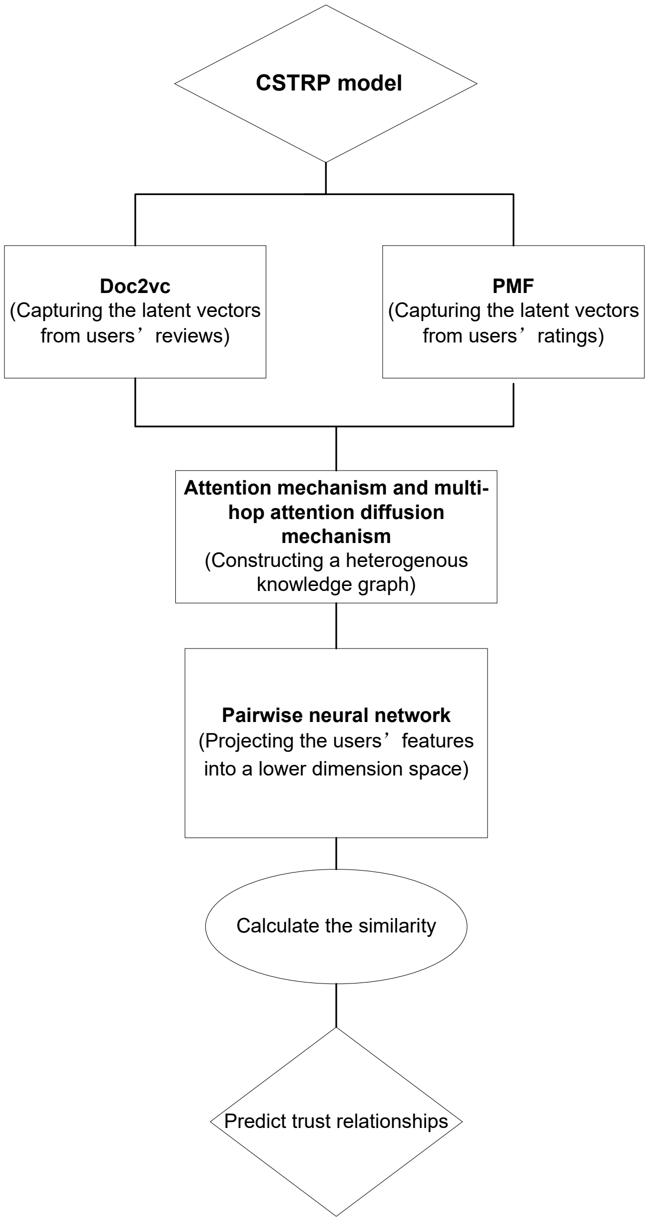

Figure 1.2: The flowchart of CSTRP model.

2

Literature Review

This chapter mainly discusses the literature review from the following aspects. In section 2.1, we first discuss the network embedding methods in the literature. Network embedding methods can extract the user's features into the vector space, and extracting the users' features is a crucial part to predict the trust relationships on the social networks. In section 2.2, we analyze the latest trust prediction methods, including static trust prediction methods and dynamic trust prediction methods. In section 2.3, we summarize the main drawbacks of these existing methods. The details are as follows.

2.1 Network Embedding

Before conducting the trust prediction on social networks, we need to capture the information on the structure of social networks and properties of users, and embed them into the vector space, such as the clustering of users and the preferences of users, etc. As an imperative part of trust prediction, the social network embedding can project these social actors' information into a low-dimensional vector space (also known as an embedding space). Currently, the method of social network embedding can be divided into the following four categories: unsupervised network embedding, supervised network embedding, semi-supervised network embedding, and heterogeneous network embedding.

2.1.1 Unsupervised Network Embedding

Unsupervised learning can be used to analyze and cluster unlabelled datasets. The main unsupervised network embeddings include PCA [2], MDS [3], DeepWalk[4], LINE [5], GraRep [6], Node2vec [7], GAT [8], Q-Learning [9], etc.

In the conventional sense, network embedding is regarded as a process of dimensional reduction [10]. Principal Component Analysis (PCA) [2] and Multi-Dimensional Scaling (MDS) [3] are two of the main methods. All these methods can project the raw data into a lower dimensional space. Other methods, such as IsoMap [11] and LLE [12], are developed to maintain the overall structure of the nonlinear manifold. Generally speaking, these methods perform well on small-scale networks. However, the time complexity of these methods makes them difficult to be applied to large-scale networks.

DeepWalk [4], as a network embedding method, can be used to learn the latent representation of nodes in the network. DeepWalk treats nodes as words and generates short random walks as sentences, bridging the gap between the embedding of the network and the word. The application of the original DeepWalk to a weighted graph was not fully discussed in this work [4]. Then, the work in [13] proposes a modified version of DeepWalk for the attributed network and specifies the applications of the modified model on the weighted graphs. This work converts an attributed graph into a weight matrix graph via the similarity of attributes and structure between the connected nodes.

Different from DeepWalk, LINE [5] considers the weight of edges and clearly defines the first-order similarity and the second-order similarity. The first-order similarity is the self-similarity between two vertices (regardless of other vertices), and the second-order similarity is the similarity between the adjacent network structures of two nodes. That is, the vertices sharing similar neighbours tend to be similar to each other. By using a Breadth-First Search (BFS) strategy, LINE can be applied to various large-scale graphs, including directed graphs and undirected graphs.

After that, GraRep [6] considers the third-order, fourth-order and other higher-order similarities. GraRep extracts global information with a node feature vector learning algorithm.

Similar to the use of node co-occurrence information of different scales, GraRep obtains the low-dimensional node representation with a singular value decomposition functioning to the adjacency matrix exponentiation.

The aforementioned methods depend on the assumption that similar neighbours are close to each other. However, Node2vec [7] believes that two nodes that are not close neighbours may also have high similarities, and the spatial structure similarity of the nodes should also be considered. In this sense, Node2vec can be regarded as an extension of DeepWalk, which combines the BFS and DFS neighbourhood searching methods in exploration with biased random walk progress.

Later, Graph Attention [8] proposes an attention model that can learn multi-aspect representations and predict the links in the original graph. Graph attention automatically learns attention to the power set of the graph transformation matrix, rather than predetermining hyper-parameters to control the distribution of context nodes. However, this graph attention only cares about the information of one layer connected to the directed nodes.

Reinforcement learning is also an unsupervised network embedding method. Reinforcement learning uses the reward and punishment function to converge to the correct answer. Many studies have adopted the reward and punishment policy to make the decision of node mapping in the network embedding. For instance, Yao et al., [14] propose a novel virtual network embedding based on the reinforcement learning method. They use historical data to train the policy network according to the virtual network requests. This policy network is an artificial neural network, which can give the result of node mapping while observing the status of substrate networks. They also employ an exploration strategy and a greedy strategy in the training and evaluating process to optimize the model, respectively. Yuan et al., [9] propose another typical reinforcement learning method for virtual network embedding, Q-learning. Their method updates the Q-matrix by a reward function with respect to the effect of virtual link embedding. These methods are relatively easy to generate the evaluation strategy when the amount of data used is relatively small. However, convergence is not guaranteed when nonlinear approximations such as neural networks are used, especially for a large number of state-action pairs.

2.1.2 Supervised Network Embedding

Supervised learning is a process that functions on labeled data. After we use these labelled data to train a model, we will use this trained model to predict new samples. Through this kind of study, when a similar task appears in the next time, we can finish it based on our existing experience before. KNN [15], SVM [16], DT [17] and ANN [18] are classical machine learning algorithms. Based on these popular ML algorithms, researchers have proposed some supervised network embedding methods [19–25]. In addition, some other supervised learning methods such as Trans-Net [26] and STNE [27] can also be used in network embedding. These can be presented in more detail as follows.

The basic idea of the KNN [15] method can be described as (1) inputting the testing data; (2) comparing its features with the training set whose data and label are known; (3) finding the top K similar data in the training that corresponds to the test data. Then the most frequent occurrence of the category in the test data is the category that belongs to the test data. The simple and intuitive process of KNN makes it need no estimate parameters. In addition, KNN can handle classification problems, particularly suitable for multi-classification issues, and performs better than SVM [16]. Islam et al. [19] leverage the KNN technique to extract paraphrases from user comments to detective the depression emotion among users of Facebook. Based on KNN, Boahen et al. [20] propose an enhanced KNN method, WE-KNN, to extract features and select behaviors of online social users. This method improves the performance of classifying users in online social networks. However, the heavy calculation in these methods can be a drawback, especially for a large number of features.

Support Vector Machine (SVM) [16] shows a good performance in statistical classification and regression analysis. SVM maps the vector into a higher-dimensional space and adopts the hyperplane to differentiate the data from different classes. To find the right hyperplane, we need to find the maximum distance between the hyperplane and the nearest class data point. This distance is called the margin. SVM has a good learning ability in linear and nonlinear classification as well as regression since it has a low generalization error rate. Besides, it can solve both low-dimensional and high-dimensional problems, avoiding neural network structure selection as well as local minima problems. Based on SVM, Bin et al. [21] introduce an embedded space ranking SVM method to capture the preferences of users by separating venues into different characteristics. The embedded space ranking SVM method can reduce the time complexity of training the personalized recommendation model for users. Xin et al. [22] put forward an MF-SVM friend recommendation model, which uses SVM method to train people's features and conducts the link prediction. The classical SVM algorithm only focuses on the binary classification, which makes it difficult to solve the multi-classification problem. Some SVM-based methods consider the combination of multiple two-class SVM methods, multiple classifiers and some other algorithms to address this problem. However, these SVM-based methods are sensitive to the noises in datasets. Namely, the missing data or different choices of parameters can lead to inaccurate classification results in these methods.

Decision Tree (DT) [17] is a tree structure whose internal nodes, branches and leaf nodes stand for a judgment on an attribute, an output of a judgment result and a classification result, respectively. Supervised learning in this model gives the known information of attributes and classification results. Then a decision tree can give the correct classification of the new data by training these samples. DT is very intuitive and easy to be visualized and explained. Based on DT, Wu et al. [23] come up with a DTCA model which uses the DT technique to select credible comments as evidence for the explainable claim verification. In this method, DT allows the model to have optimal decision thresholds in the network embedding and makes the evidence discovery process transparent. Mateusz et al. [24] combine DT and random forest techniques to classify members of organizations in social networks. They construct a multitude of decision trees and employ the individual trees as the mode of output. However, DT-based methods are prone to overfitting, when the training data has many features.

Artificial Neural Network (ANN) [18] simulates a biological neural network with a computer network system. Each node on the network can be regarded as a neuron with the property of memorizing, processing, and working in parallel with other nodes. The process is to input information to some nodes of ANN through nodes processing and output to other nodes, until the training process of the entire neural network is finished, and the final result is delivered. In the artificial neural network, the degree of node participation in the work is controlled by the weighted value. The positive weight is equivalent to the stimulation of the neuron synapse and the excitement. The negative weight is equivalent to being inhibited and paralyzing the neuron until it does not work. ANN can learn all kinds of nonlinear functions. Therefore, ANN is commonly referred to as a universal function approximator. ANN can learn the weights that map all kinds of inputs to outputs due to its activation function. The activation function introduces nonlinear characteristics into the network. This helps the network learn complex relationships between inputs and outputs. Aakash et al. [25] put forward an ANN-based HGS model, which can identify the crucial characteristics from the social network data and conduct the prediction of hotel customer satisfaction based on these characteristics. However, a large number of parameters are required in these neural networks, such as network topology, thresholds as well as initial values of weights, and these parameters are sensitive to noise and outliers. What's more, it is hard to interpret the output results due to the unobserved learning process.

Trans-Net [26] introduces the idea of machine translation to the intermediate process. It encodes the labels on the edge (a vector) through an autoencode, maps the nodes and edges to the same space for addition and subtraction, and uses the decoder part to restore the element-binary label set to get the prediction results. This model considers both rich semantic information over edges and modeling relations between vertices with the translation mechanism when learning vertex representation. Thus, Trans-Net achieves promising results on social relation extraction compared to Deepwalk, LINE and Node2vec.

Recently, STNE [27] has been proposed as a social network embedding, which preserves both the trust transfer pattern as well as the relations simultaneously based on the structural balance theory. Even though this social trust network embedding succeeds in constructing the structure, it tends to fail to explain the performance of the null relation.

2.1.3 Semi-supervised Network Embedding

Semi-supervised learning works on the training data in which only a small part is labelled while most of it is unlabelled. Therefore, the semi-supervised learning methods do not require all the training data to be labelled. The main semi-supervised network embedding can be MMDW [28], CANE [29], SSC-GCN [30], etc.

Maxmargin DeepWalk (MMDW) [28] is a semi-supervised method that learns the representation of nodes in a partially labeled network. MMDW is composed of two parts: the first part is to use a matrix factorization to get the node embedding model; the second part is to train the maximum marginal SVM classifier on the labeled node using the learned representation as a feature. By introducing the bias gradient, the parameters in the two parts can be updated jointly. Since this MMDW is a semi-supervised method, and it works on the dataset with a small part of labeled data and a large part of unlabeled data, it is adaptive to real applications. However, the noise-free data used by MMDW is difficult to be obtained in real applications, and the basic assumptions of MMDW do not take the uncertainty and the complexity of the data distribution into account.

CANE [29], as a CNN-based semi-supervised learning method, has been used in network embedding. Its local connection and weight sharing allow a smaller number of parameters to be set in the model, to some degree, reducing the risk of over-fitting. In the CANE framework, the pooling layer is introduced into the mutual attention mechanism to obtain context-aware text embedding. In addition, through a mutual attention mechanism, CANE learns context-aware embedding for vertices, which could improve the precision of modeling the semantic relationships between vertices. Except for that, CANE has a good capability for automatic feature extraction. Besides, its shared convolution kernel makes it under no pressure to process high-dimensional data. However, the pooling layer in this model ignores the correlation between the part and the whole.

Semi-supervised learning based on graph convolutional neural network (GCN) provides a new idea for the processing of graph structure data, applying deep learning neural network to graph data. It can perform end-to-end learning of node feature information and structural information at the same time with a faster process. It is currently a good choice for graph data learning tasks. Semi-Supervised Classification with Graph Convolutional Networks (SSC-GCN) [30] is a semi-supervised learning method using a variant of convolutional neural networks to operate on graph-structured data. SSC-GCN studies hidden layer representations which encode local graph structure as well as features of nodes through a spectral convolution operation. However, the convolution in this model cannot support for the mini-batch.

2.1.4 Heterogeneous Network Embedding

The above-mentioned works are for homogeneous networks, but the real-world network is undoubtedly heterogeneous. Therefore those works cannot model a real network. To adapt to the real network, some researchers use different types of nodes and edges to build the heterogeneous graph, such as PTE [31] and HINES [32]. Their results illustrate the effectiveness of the heterogeneous network embedding.

PTE [31] defines three types of networks, word-word, word-document, and word-label. They are all made into a similar bipartite graph, and their respective loss functions are summarized together. PTE is relatively simple and intuitive since it does not directly nest a complex prediction model like the CNN/RNN model to extract predictive information in the final embedding.

HINES [32] embeds multiple heterogeneous networks with different types of nodes and different types of edges in the graph. This method introduces the concept of meta path, that is, the connecting edges between different points are connected according to certain meta information, such as A1(Author)-P1(Paper)-A2(Author). This concept can be well extended to many real scenarios. However, this model only considers the directly connected nodes and doesn't take the indirectly connected nodes into account, lacking a transmit diffusion.

2.2 Review for Trust Prediction

The trust level is used to measure the degree of reliance between two online users. The pair of users with high trust values tend to accept information recommended or shared by each other. Since we apply the Internet for nearly every aspect of our daily life, including entertainment, sports, shopping, news reading, etc., our daily life is overwhelmed with tremendous comments, ratings and discussions, etc., relevant or irrelevant, reliable or unreliable. Conducting trust prediction between users from these large amounts of information can reduce the cost of searching for information and precisely find the target customers under product recommendations. In recent years, trust prediction has attracted many researchers' attention, and they have already proposed several models. They can be divided into two categories: *static trust prediction* and *dynamic trust prediction.*

2.2.1 Static Trust Prediction

Static trust prediction method refers to those trust methods that do not take the previous experience and time factor into account. This static trust prediction method considers users' static context behaviors, irrelevant to their historical behaviors. The extracted preferences from these behaviors will not change over time. Based on the main techniques, we can categorize the static prediction method into three parts: structure-based trust method, graph-based trust method and low-rank approximation-based method.

Structure-based Trust Prediction Method

Neighbourhood structure-based and transitivity-based methods can be two classical sub-classes of structure-based trust methods. Neighbourhood structure-based methods have been mentioned in [33, 34]. These methods believe that trust value between users is aggregated from the information of their neighbours. However, these methods do not consider the different weights when aggregating under different semantic contexts. Transitivity property of trust has been studied in [33, 35, 36]. These methods mainly focus on the trust transitivity path between the target user and the source user. Walter et al. [37] take the multiplication of trust values in the transitivity path between intermediate users as the predicted trust values. Lei et al. [38] and Gary et al. [39] use the aggregated trust value as the trustworthiness to predict the trust between unknown users. However, these methods equally treat all the propagation paths from source users to target users. In reality, trust prediction can be complex and varied through different propagation paths. To address this issue, Liu et al. [40, 41] and Zhao et al. [42] add the social background context into the trust path. Their results show that these context-aware methods benefit representation learning. However, these methods have the imbalance attribute problem and ignore the influence of those indirectly connected neighbours.

Graph-based Trust Prediction Method

Apart from these structure-based methods, graph-based trust models [43] also have been applied in the trust prediction, building the block on the concept of web-of-trust or Friend-of-a-Friend. In these approaches, users can be considered as nodes, and their relationships (value of their trust relations) can be regarded as edges [44]. For instance, based on this Friend-of-a-friend (FOAF) concept, Golbeck et al. [45, 46] propose two methods. The first method [45] is to build a web of trust based on different types of topics to conduct the trust prediction between users. The other one [46] is to compute the trust value based on the option of people with the combination of the binary trust ratings and continuous ratings on the networks.

Similar to the usage of the trust network graph in [45, 46], Ziegler et al. [47] propose a semantic-web-based method, Appleseed, to compute the trust value with a local group trust metrics. Subsequently, to obtain a better representation of users' graphs and conduct the trust prediction, Zhang et al. [48] introduce a semantic-based method to do the trust reasoning. Compared with the required tremendous human effort in traditional pairwise trust prediction, they introduce a role-based and behavior-based reasoning mechanism under the domain ontology, improving trust prediction's efficiency.

In addition, Hang et al. [49] leverage a graph similarity design to tackle the recommendation problem based on the trust networks of users, while Zuo et al. [50] use a trust certificate graph to compute the trust value following a trust chain. To further study the different interactions between users, Caverlee et al. [51] take the different weights of users' feedback into account to research the social relationships on the trust graph.

In recent years, Graph Neural Network (GNN) has been regarded as a powerful tool in graph data learning. Based on GNN, Jing et al. [52] maximize a signed graph of mutual information and consider the positive point-wise mutual information into the model, conducting accurate trust predictions. To consider the diffusion from a single-hop trust in a trust social network, Nguyen et al., [53] propose a heuristic algorithm. They take each node's propagation probability and the multi-hop neighbours' influence into account. Liu et al., [54] put forward an iterative algorithm to contain the information about multi-hop neighbours. They use neural networks to conduct the multi-hop trust assessment. Then, based on the single-hop trust computation rules obtained from the neural networks, they employ a breadth-first search strategy to assess the multi-hop trust relationship over the trust social network. However, these methods neglect the heterogeneous information in their models.

To consider different types associated with nodes and edges in the social network, these studies [55–58] adopt the heterogeneous information network to compute the items' and users' similarity through the meta-path, and they [55, 56] consider the information propagating between indirectly connected nodes. However, their transmit diffusion only focuses on the node representation's attention score and does not consider the diffusion on the edge's attention score.

Low-rank Approximation-based Trust Prediction Method

Low-rank approximation-based methods have been widely applied to deal with the data sparsity problem in trust prediction. These methods contain collaborative filtering [59, 60], document clustering [61, 62] and matrix factorization (MF) [61]. They employ low-rank latent representation to represent the sparse and low-rank trust relation matrix, which has a relatively good performance on trust prediction of social networks[63].

Traditional algorithms have drawbacks in dealing with data sparsity and scalability on the model, which leads to the low prediction accuracy. Therefore, Ma et al. [64] propose the probabilistic matrix factorization method. This method extracts information from the user rating records and social networks with a regularization term of social ties in the optimization. The result illustrates that this method can handle large and sparse datasets with a high quality of trust prediction.

Similar to using regularization terms in optimization in [64], Jamali et al. [65] propose another model-based matrix factorization approach that takes the trust propagation mechanism into account, reducing the cold start users' problem. Later, based on [64, 65], Yang et al. [66] propose an extension CircleCon model that considers category-specific social trust circles factors under the assumption that users have different attitudes (trust or distrust) toward each other on various categories.

Paul et al.[67] put forward a constrained Non-Negative Matrix Factorization-based method (CNMF). This method uses the regularization of some additional constraints to obtain the identification and classification of representation. However, this CNMF-based method ignores the situation of local or pairwise and only considers from the global scope. To address this problem, Cai et al. [68] raise a Graph Regularized Nonnegative Matrix Factorization (GNMF) algorithm, which uses an affinity graph as an encoder to account for the geometrical information with a matrix factorization.

Jiang et al. [69] bring up a trust prediction model named Relative Pairwise Relationships Non-negative Matrix Factorization. It imposes a penalty on the pairwise relationship and treats them as triples. This method obtains close approximation and performs better than traditional low-rank approximation methods.

Wang et al. [70] also leverage the pairwise neural network to address the sparsity problem of trust prediction. Differently, they use matrix factorization (MF) and Doc2Vec to obtain the information of users' features distilled from users' review behaviors. However, they ignore the structure factors when conducting the trust prediction.

2.2.2 Dynamic Trust Prediction

Dynamic trust prediction refers to the trust prediction that considers the dynamic factors of time, previous action and experience, etc. Since every element is correlated and nothing is unchanged in the real world, the level of trust can increase or decrease according to the previous interactions between users. The dynamic trust prediction methods can be categorized into the time window-based method and the HMM-based method.

Time Window-based Trust Prediction

To adapt to the actual dynamic social networks, some researchers propose a trust time window to study the trust prediction. Shi et al. [71] develop a dynamic feedback peer-to-peer trust model which could set a historical time to cope with the dynamic trust prediction. In addition, a decay factor [72] is combined with the trust time window method to weaken the impact of past information while updating new information. Furthermore, under the same task of measuring the trustworthiness of peers in [71], Jigyasu et al. [73] raise a Bayesian network-based trust model. It uses a time window according to the previous transactions of task processor peers, which can identify the malicious peers in an early stage. Fang et al. [74] establish a Time-window-based Resilient Trust Management Scheme (TRTMS) method, which considers the behaviors of compromised nodes over a period of time and introduces a control factor with a time window to defend against the time-varying attack of reputation in the network. Shilpa et al. [75] put forward EBTEM, an evidence-based dynamic trust prediction that uses a time window to treat various attributes as evidence factors at different times, adaptive to the changes in the service behavior. Ye et al. [76] propose a Dynamic Trust Evaluation Model (DTEM), which uses a sliding window to adjust different weights on implicit and explicit trust, enhancing the flexibility of the trust evaluation. Zhang et al. [77] use a forgetting rate to aggregate the failure and the success of transactions with a time window to adjust the trust prediction. These methods may improve the accuracy of trust prediction in some cases, but it is not suitable for the environment where data is changed in high frequency.

HMM-based Trust Prediction

HMM (Hidden Markov Model) as a probabilistic model can describe the process of randomly generating an unobservable random sequence (state sequence) from a Hidden Markov Chain, and then the process of generating an observation sequence (observation sequence) from each state. Zheng et al. [78] take contextual transaction information as well as the outcome as the observation sequences and the states of HMM to conduct the dynamic trust prediction. This method addresses the problem of the lack of contextual information about each transaction in the HMM-based trust prediction model [72, 79]. However, it is difficult in line with the real situation that a seller cannot transact a series of times with the same customer. What's more, since these methods take the past transaction outcomes as observation sequences, they cannot identify the infrequent dishonest behavior. To address these problems, Liu et al. [80] use an HMM-based trust model to capture the dynamic behavior based on the interaction contextual information. Fulvia et al. [81] apply hidden Markov theory to longitudinal sampling weights to deal with repeated and missing nodes from the data to investigate the public trust. This method considers time-fixed individual covariates along with observed time-varying factors, which effectively figure out the problem of identifying dishonest behaviors.

2.3 Summary

In summary, we reviewed four kinds of important network embedding methods: unsupervised network embedding, supervised network embedding, semi-supervised network embedding, and heterogeneous network embedding. As we discussed above, each type of embedding method has its own limitations. For example, the high time complexity of some unsupervised learning methods makes them difficult to be applied to large-scale networks; in addition, overfitting is prone to occur in some supervised learning methods when the training data has many features.

Then we reviewed the trust prediction methods in both static and dynamic categories. The main disadvantages of all the above trust prediction methods are mainly as follows. 1) Trust matrix sparsity problem. Most of the methods still suffer from the data sparsity problem. These methods depend highly on the trust relation matrix, and these trust relationships in the matrix are sparse and difficult to observe in the actual social networks. Therefore, there are insufficient trust relations to be factorized in these methods. 2) Hardness to preserve semantics and structure simultaneously. Social networks contain both semantic information, i.e., social relationships, status, preferences of users, and structure information, i.e., relationships between attributes. However, these methods only consider one of them and seldom consider both simultaneously, which cannot provide enough attribute information when computing the similarity between users. 3) Improperly treating all the propagation paths equally. Since the trust between users can be propagated and transferred, and these trust values are different on the trust chain, the propagation paths cannot be set as equal. However, these methods ignore these differences between the propagation paths. 4) Only considering on one-hop attention score on the direct connections. Most methods only care about the directly connected nodes and ignore the indirectly connected nodes, which cannot address the null relation problem. All these drawbacks lead to the low accuracy of trust prediction on social networks.

To address these problems, in this thesis, we develop a new CSTRP model, which combines a heterogeneous graph attention mechanism with a pairwise deep neural network and a 2-hop attention diffusion layer. We will introduce our model in detail in Chapters 3.

3

Methodology

This chapter mainly introduces the preliminary and the method that we use to conduct social trust relationships. In section 3.1, we explain the notations and concepts adopted in our theory. In section 3.2, we introduce our specific CSTRP model.

3.1 Preliminary

This section explains all the related concepts and terms about the attention mechanism and multi-hop diffusion mechanism on the social network. It starts with the explanation of the heterogeneous attention graph [82] setting and ends with the summary of the notations in this thesis in Table 3.1. All this information is adopted in the methodology of the proposed CSTRP model.

- **Heterogeneous graph [82].** The heterogeneous graph is defined as $G = (U, I, R, P)$. U, I, R, P stand for users, interests, relationships and their meta-paths, respectively. This heterogeneous graph contains different types of nodes and edges, as can be shown in

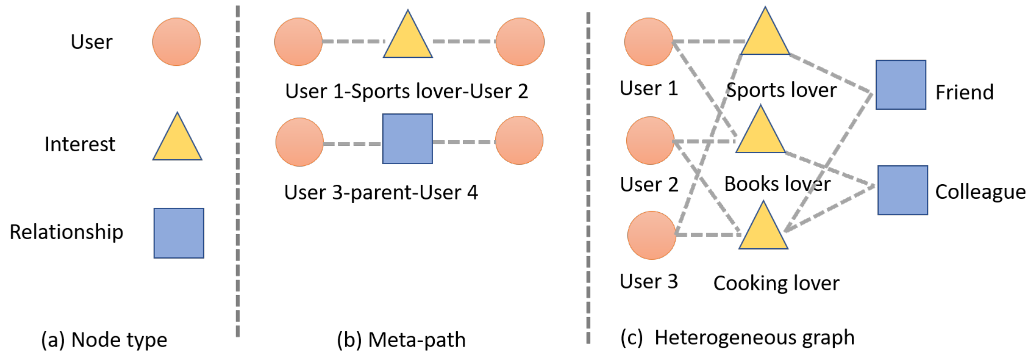

Figure 3.1: An example of our social network with different types of nodes and paths.

Figure 3.1 (d), three types of nodes including users, interests and relationships as well as two types of paths on relations, including User1-Interest-User2 and User3-Relative-User4.

- **Meta-path [70].** Meta-path is adopted to explain the composite relations between different nodes. Let the total number of nodes equals to n, from node i to j, $1 < i < j \leq n$, we define the meta-path P in the form of $U_i \xrightarrow{R} U_j$ and $U_i \xrightarrow{I} U_j$. R and I refer to composite relations $R = R_1 \circ R_2 \circ ... \circ R_l$ and a composite interests $I = I_1 \circ I_2 \circ ... \circ I_k$ between user U_i and U_j, respectively. "$\circ$" refers to the composition operator on relations and interests. As shown in Figure 3.1 (b), Different meta-paths contain different semantics. For instance, User 1-SportsLover-User 2 means that they have the same interest in sports, while User-parents-User means that they have the same relative like parents.

- **Meta-path based Neighbours [82].** Given a node U_i, its meta-path-based neighbours denote as $U_j \in N^P_{U_i}$. It refers to the sets of nodes directly connected with node U_i through meta-path P in this heterogeneous graph, where N denotes the neighbours of node U_i, P denotes their meta-path. Note that the node itself is also included in its neighbour sets. Take an example in Figure 3.1, given the meta-path User 1-SportsLover-User 2, the meta-path-based neighbours of User 1 are User 1 and User 2. In this way, different semantics and structure information can be considered via these meta-path-based neighbours which can be learned from the multiplication of adjacency matrices' sequences.

- **Multi-hop Neighbours.** Multi-hop neighbours refer to the pair of nodes without a direct connection through an edge, which means that each of these nodes cannot directly compute the representation of its next layer by participating in the immediate neighbours. Take an instance, users A and D, users D and E are two pairs of directly connected nodes in Figure 1.1. User A and E are two-hop neighbours without an edge connection and so on.

Table 3.1: NOTATION AND EXPLANATION

Notation	Explanation
P	Meta-path
Tr_P	The specific type of transformation matrix
f_u	Original feature of node u
f'_u	Projected feature of node u
$Im^P_{U_i,U_j}$	Importance of U_j on U_i based on P
att_{node}	Node level attention
$N^P_{U_i}$	Neighbour of U_i on the meta-path P
σ	Activation function
α_P	Vector of node-level attention on the meta-path P
$a^P_{U_i,U_j}$	Weight coefficient of P based on U_j and U_i
$e^P_{U_i}$	Embedding of node U_i based on P
$\|\|$	Concatenation
att_{path}	Path level attention
W	Weight matrix
b	Semantic level attention vector
β_{P_i}	Importance of each meta-path
c	Bias
a_{P_i}	Weight coefficient of meta-path
X	Final embedding of two-level attention mechanism
$sc^n_{i,k,j}$	Attention score of the nodes' edge on the nth layer
$A^n_{i,j}$	Attention value in the nth layer gathering information from the node U_i to U_j
$d \cdot (1-d)$	Decay factor of multi-hop diffusion
L^n	Input user embedding of the n-hop
x	Original user embedding from the first hop
T_U	The output of the pair-wise multi-layer projection

3.2 CSTRP Model

In this section, we describe our Context-based Social Trust Relationship Prediction Model (CSTRP). After we construct a heterogeneous network graph, we first introduce our two-level attention mechanism motivated by HAN model [82], including the attention value on the node-level and path-level as shown in Figure 3.1 to clarify all the potential relationships and semantic information. Then, we add a 2-hop attention diffusion layer into our model to aggregate the information from the indirectly connected nodes. Furthermore, we use the pair-wise deep neural network to further learn the user's features. After we get the nodes embedding through the pair-wise deep neural network, the degree of their trust prediction is the similarity between their nodes embedding. The nodes in the neighbourhood with similar node sets have high values of cosine similarity [83]. In contrast, the low values of cosine similarity means the different potential representations between users. This conclusion is based on the homophily theory, a famous psychological theory that owes people's trust prediction to their similar experiences and background [84], [85].The framework of CSTRP is shown in Figure 3.2.

3.2.1 Attention Mechanism

Graph neural network has been proved as a powerful deep representation learning for social networks, and attention mechanism as a recent research trend has been applied in graph neural network. Existing studies have shown the superior representation learning of attention mechanism on different sizes of datasets graph, since it can focus on the most salient parts based on the datasets and embed accurate information. Graph Attention Network [8] introduces an attention mechanism with only one type of node and edge in the homogeneous graph, which cannot be applied to the social networks that contain different types of nodes and edges [86]. It is because a single type of node and edge cannot model the rich semantics and comprehensive graph structure. In this case, we introduce the attention mechanism based on a heterogeneous graph to capture the information on multi-types of nodes and edges. To clarify the semantics in both nodes and edges, we leverage a two-level attention mechanism, i.e., node-level attention and path-level attention.

Node-level Attention

In a heterogeneous graph, to simulate the real social networks, nodes can be connected by different kinds of relationships. These connections under different relationships can be described

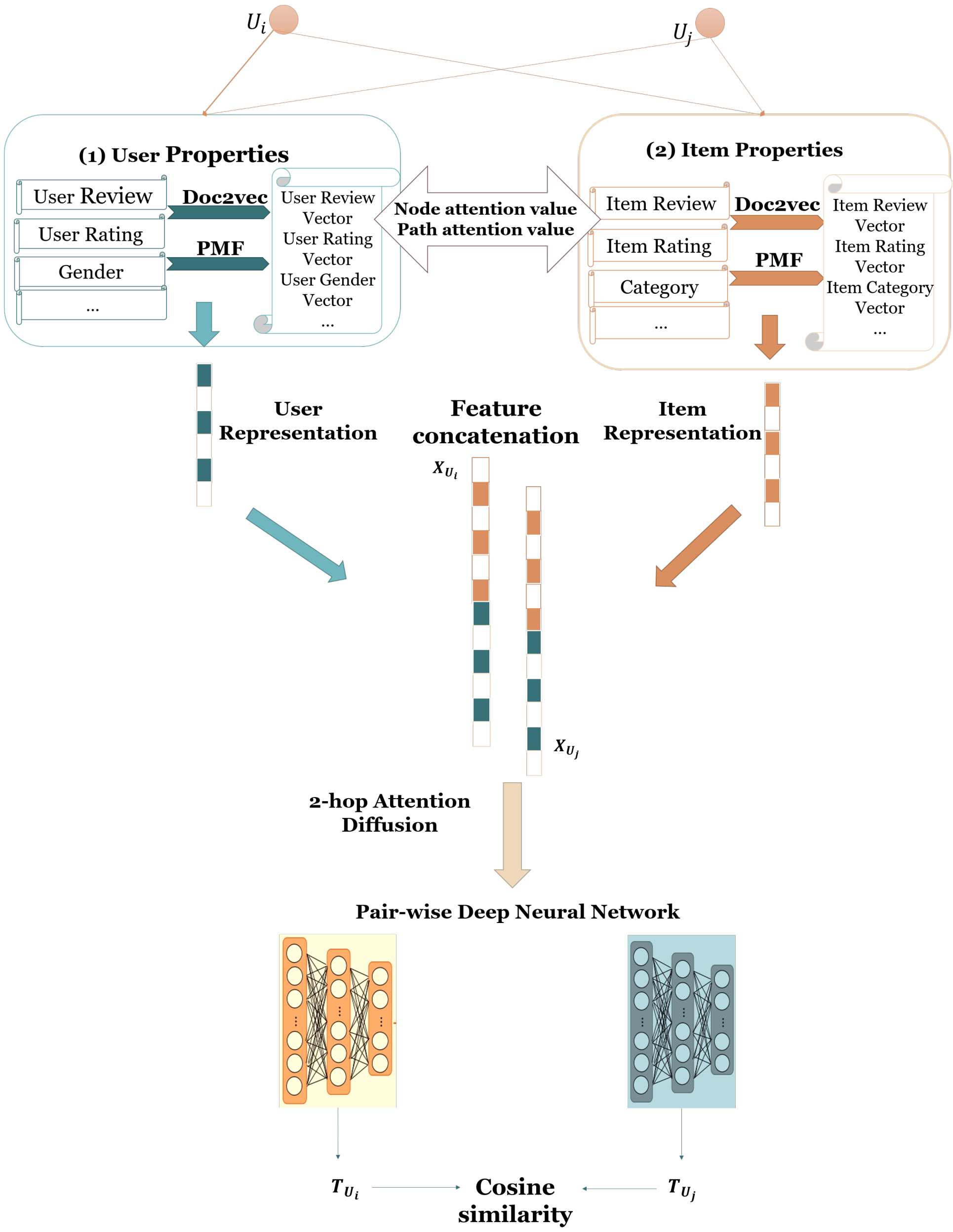

Figure 3.2: An overview of the proposed CSTRP model, constituted of a heterogeneous and comprehensive user modeling, a 2-hop attention diffusion, a pair-wise multi-layer projection and the cosine similarity trust measurement.

through meta-path. Since each node owns many meta-path based neighbours, it is required to distinguish the difference between the node's meta-path based neighbours and highlight the neighbours that are important to the node. This different information of meta-based neighbours is finally aggregated to embed the node's representation. In this consideration, we use node-level attention to distinguish the different influences of these neighbours on this node.

For each user, the target of node-level attention is to define the different importance of the directly linked neighbours based on the meta-path P between the user and each of the neighbours and assign different attention values to them. Since user, interest and relationship are in different types of feature spaces due to the heterogeneity of nodes, a specific type of transformation matrix Tr_P is needed to project them into the same space as the input. Here we define f_u and f'_u as the original feature and the projected feature of node u, respectively. The projected feature of node u is calculated by Equation 3.1:

$$f'_u = Tr_P * f_u \tag{3.1}$$

To figure out the association of each node, we use self-attention to learn the weight of each node. $Im^P_{U_i,U_j}$ represents the importance of U_j on U_i on the meta-path P. Given pairs of nodes (U_i, U_j) connected by a meta-path P, the importance of U_j on U_i can be learned through the node level attention and can be calculated by Equation 3.2:

$$Im^P_{U_i,U_j} = att_{node}(f'_{U_i}, f'_{U_j}, P) \tag{3.2}$$

where att_{node} denotes the node-level attention in the deep neural network. Since the similarity of connection patterns exists under one path, the same att_{node} value is shared between the nodes on the same P. It is noted that $Im^P_{U_i,U_j}$ is asymmetric. This is because trust is asymmetric in real scenarios.

Then, we use the softmax function to normalize $Im^P_{U_i,U_j}$ to get the weight coefficient, which can be calculated by equation 3.3:

$$a^P_{U_i,U_j} = softmax(Im^P_{U_i,U_j}) = \frac{\exp\left(\sigma(a^T_P \cdot [f'_{U_i}||f'_{U_j}])\right)}{\sum_{U_k \in N^P_{U_i}} \exp\left(\sigma(a^T_P \cdot [f'_{U_i}||f'_{U_k}])\right)} \tag{3.3}$$

where U_j is the neighbour of U_i. Let $U_j \in N^P_{U_i}$ denote the neighbour of U_i on the meta-path P. σ is the activation function, and k denotes all the neighbours of U_i on the meta-path P. α_P denotes the vector of node-level attention on the meta-path P. From the equation of (3.3), we can see the reason why the attention value $a^P_{U_i,U_j}$ is asymmetric. Expect for the difference of concatenate orders in the numerator, different neighbours of each other can also lead to the asymmetric weight coefficients.

In addition, by multiplying these weight coefficients with projected features of the neighbour nodes, an aggregate function needs to be activated to get the embedding of node U_i based on the meta-path. Namely, we sum the hidden features of adjacent nodes on the meta-path of P according to different weights and then add the activation function to get the updated $e^P_{U_i}$, the embedding of node U_i based on P, which can be calculated by Equation 3.4:

$$e^P_{U_i} = \sigma\left(\sum_{U_j \in N^P_{U_i}} a^P_{U_i,U_j} \cdot f'_{U_j}\right) \tag{3.4}$$

To reduce the data variance, we also add a multi-head attention mechanism to stabilize the training process. Node-level attention is repeatedly calculated M times and then concatenated as a semantic-specific embedding. The M here refers to the number of the node's meta-path.

$$e^P_{U_i} = \sigma \mathop{\Big\|}_{m=1}^{M} \left(\sum_{U_j \in N^P_{U_i}} a^P_{U_i,U_j} \cdot f'_{U_j}\right) \tag{3.5}$$

Through our process, we can get the specific semantic node embeddings, expressed as $e^P_{U_i}$ when given q different paths (P_i, $1 \leq i \leq q$)

Path-level Attention

In real online social networks, people have different relationships and preferences toward different people and different items. Correspondingly, each node has several meta-paths, reflecting different semantic information in the heterogeneous graph. We need to clarify and embed the specific semantics contained in each meta-path to learn the representation on the graph, and it is imperative to distinguish the most meaningful meta-path. Thus, we define the path-level attention to study the importance of each meta-path and assign appropriate attention values to them. The weight of different meta-path P can be defined in equation 3.6:

$$(\omega_{P_1}, \omega_{P_2}, \ldots, \omega_{P_q}) = att_{path}(e^P_{U_1}, e^P_{U_2}, \ldots, e^P_{U_q}) \tag{3.6}$$

where att_{path} calculates the path-level attention. To obtain the attention scores of each meta-path, the non-linear transformation process of semantic-specific embedding is needed. Then, we use the cosine similarity to calculate the transformed embedding and the path-level attention vector b in Equation 3.7. The similarity result is averaged and regarded as the different importance of these meta-paths. β_{P_i} denotes the importance of each meta-path and can be calculated by Equation 3.7:

$$\beta_{P_i} = \frac{1}{|U + I + R|} \sum b^T \cdot Tanh(W \cdot e^P_{U_i} + c) \tag{3.7}$$

where W denotes the weight matrix, c denotes the bias, and b denotes the semantic level attention vector.

Similar to the node-level attention, we also use the softmax function to normalize the importance of each meta-path.

$$\omega_{P_i} = \frac{\exp(\beta_{P_i})}{\sum_{i=1}^{q} \exp(\beta_{P_i})} \tag{3.8}$$

where ω_{P_i} denotes the weight of meta-path through the softmax function. A larger value of ω_{P_i} means this kind of meta-path has more contributions to the trust prediction. Last but not least, to get the final embedding X, these semantic-specific embeddings can be added in Equation 3.9:

$$X = \sum_{i=1}^{q} \omega_{P_i} \cdot e_{U_i}^{P} \tag{3.9}$$

The whole process of this two-level attention mechanism is shown in algorithm 1.

3.2.2 2-hop Attention Diffusion

As we mentioned in section 1.1, most of the existing methods only focus on the information of directly connected nodes and do not consider the information of indirectly connected nodes. Therefore, it can lead to inaccurate trust prediction results. It is because the message-passing layer in the real world is multiple, and users can interplay through their multi-hop neighbouring context. Therefore, the existing methods are inevitably bombarded with the noise produced by the incomplete representation learning on the graph of social networks, resulting in low prediction accuracy. To tackle the above issue and obtain the information from indirectly connected users, we introduce a 2-hop attention diffusion, which operates attention scores at each layer. After we compute the attention scores on all edges, the attention values between indirectly connected pairs of nodes are computed at the attention diffusion layer. This process will be introduced intuitively below.

Firstly, we take a triple set $(X_i,\ E_k,\ X_j)$ as an input to our 2-hop attention diffusion operator, where X_i and X_j represent the embedding of U_i and U_j under the semantics information and we could calculate them by Equation (3.9). E_k denotes the trainable edge embedding.

Secondly, we use LeakyRelu [87] as our activate function to calculate the attention score of the direct connections relations between users on the first layer. LeakyRelu function is to adjust the vanishing gradient problem for negative values. $sc_{i,k,j}^{1}$ denotes the attention score of the direct connection relations between users on the first layer and can be calculated by Equation

Algorithm 1: Two-level Attention of CSTRP

Input: Graph $G = (U, I, R, P)$
The node feature $f_u, \forall_u \in \{U, I, R\}$
The meta-path set $\{P_1, P_2, \ldots, P_q\}$
The times for Node-level attention repeating M

Output: The node-level attention coefficient $a^P_{U_i,U_j}$
The path-level attention coefficient a_{P_i}
The final embedding X

for $P_i \in \{P_1, P_2, \ldots, P_q\}$ **do**
  for $m = 1 \ldots M$ **do**
    The specific type of transformation $f'_u \leftarrow Tr_P * f_u$;
    for $u \in \{U, I, R\}$ **do**
      Select $N^P_{U_i}$;
      ($N^P_{U_i}$ is the meta-path based neighbours of node U_i);
      for $U_j \in N^P_{U_i}$ **do**
        Compute the weight coefficient $a^P_{U_i,U_j}$;
        $a^P_{U_i,U_j} = softmax(Im^P_{U_i,U_j}) = \frac{\exp\left(\sigma(a^T_P \cdot [f'_{U_i} || f'_{U_j}])\right)}{\sum_{U_k \in N^P_{U_i}} \exp\left(\sigma(a^T_P \cdot [f'_{U_i} || f'_{U_k}])\right)}$;
      end
      Compute the path-specific node embedding;
      $e^P_{U_i} \leftarrow \sigma(\sum_{U_j \in N^P_{U_i}} a^P_{U_i,U_j} \cdot f'_{U_j})$;
    end
    Concatenate the whole learned embeddings for M times;
    $$e^P_{U_i} \leftarrow \sigma \mathop{||}_{m=1}^{M} (\textstyle\sum_{U_j \in N^P_{U_i}} a^P_{U_i,U_j} \cdot f'_{U_j})$$
  end
  Compute the attention coefficient of meta-path ω_{P_i};
  Fuse the semantic-specific embedding;
  $X \leftarrow \sum_{i=1}^{q} \omega_{P_i} \cdot e^P_{U_i}$
end

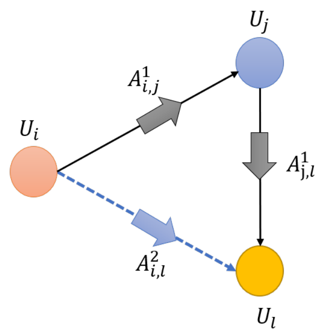

Figure 3.3: An specific example to compute the attention score between node U_i and its indirectly connected node U_l.

3.10:

$$sc^1_{i,k,j} = LeakyRelu(a_1 tanh(W_a \cdot X_i || W_b \cdot X_j || W_c \cdot E_k)) \tag{3.10}$$

where W_a, W_b, W_c and a_1 are the trainable parameter weights shared by the first layer. "||" denotes the concatenation. After applying Equation (3.10) to each edge in the graph, the attention score matrix $sc^1_{i,j}$ on the first layer is obtained as follows:

$$sc^1_{i,j} = \begin{cases} sc^1_{i,k,j}, & \text{if } U_i, E_k, U_j \text{ appear in the } G \\ -\infty, & \text{otherwise} \end{cases} \tag{3.11}$$

Thirdly, taking $sc^1_{i,j}$ as an input through a softmax function, the attention matrix can be obtained as follows, where $A^1_{i,j}$ represents the attention value in the first layer when information is gathered from the node U_i to U_j.

$$A^1_{i,j} = softmax(sc^1_{i,j}) \tag{3.12}$$

In the following stage, we aim to figure out the attention score between nodes that are indirectly connected in the network. Since, in the real world, the trust is decay in propagation, we add an attention decay factor into our diffusion layer to calculate the attention score.

$$A' = d \cdot (1 - d) \cdot A^1 \tag{3.13}$$

$$L^2 = A' \cdot x \tag{3.14}$$

$$A^2 = AttDiffusion\ (G,\ L^2) = A'L^2 \tag{3.15}$$

where $d \in (0, 1]$ is the trainable parameter and $d \cdot (1 - d)$ is the decay factor for the second layer. A^1 is the power of the attention matrix of the first hop.

Finally, after getting the second hop attention score, we then propose the aggregation of graph attention diffusion-based characteristics as Equation (3.13), where L^2 denotes the input user embedding of the second hop which transfers from the first hop, x denotes the original user embedding from the first hop. Take an instance, the specific process to figure out the importance between indirectly connected nodes like U_i and U_j in Figure 3.3 can be shown as follows. From Equation (3.12), we can get the first-hop attention $A^1_{i,j}$ and $A^1_{j,l}$ from U_i to U_j and U_j to U_l. Then from equations (3.13) (3.14) (3.15), we can get the second hop attention $A^2_{i,l} = d \cdot (1 - d) \cdot A^1_{i,j} \cdot A^1_{j,l}$. As shown in Figure 3.3, $A^1_{i,j}$ and $A^1_{j,l}$ refer to the attention score from U_i to U_j and U_j to U_l based on their direct connection, from their first hop. $A^2_{j,l}$ refers to the attention score from U_i to U_l based on its indirect connection, from the second hop. This 2-hop attention diffusion makes it accessible to preserve a relatively complete structure between nodes, since it aggregates more information from indirectly connected nodes via adaptive different weight coefficients through different hops and decay factors. Finally, we feed these users' features into a pair-wise multi-layer projection process to further capture their characteristics.

3.2.3 Pair-wise Multi-layer Projection

After we integrate a two-hop diffusion mechanism into our model, we take the advantage of the pair-wise deep neural network to further capture the latent features of users. This pair-wise multi-layer projection consists of a multi-layer perception unit as well as a similarity calculation unit. Suppose our input user feature as X', output latent representation as Z and we have H_i intermediate hidden layers, our deep neural network equation can be shown below,

$$H_1 = w_1 X' + \epsilon_1 \tag{3.16}$$

$$H_i = f(W_i H_{i-1} + \epsilon_i), i = 2, 3, 4, ..., N - 1 \tag{3.17}$$

$$Z = f(W_N H_{N-1} + \epsilon_N) \tag{3.18}$$

where w refers to the vector of the weight matrix, W_i refers to the weight matrix, and ϵ_i refers to the bias term on the i^{th} layer. In order to increase the nonlinearity of the neural network model, we define RELU as our activation function f during our pair-wise multi-layer projection, where $f(X') = max(0, X')$. Through this pair-wise multi-layer projection, the features of user pairs (U_i, U_j) could be projected into a low dimension space and this process can be written as follows:

$$T_{U_i} = f(...f(W_{a2}f(X'_{U_i})W_{a1})...) \tag{3.19}$$

$$T_{U_j} = f(...f(W_{b2}f(X'_{U_j})W_{b1})...) \tag{3.20}$$

where (T_{U_i},T_{U_j}) is the output of the pair-wise multi-layer projection, the N^{st} layer weighting matrices for the user pair (U_i, U_j) are W_{aN} and W_{bN}. After we conduct the multi-layer perception unit, we will feed the output into the similarity calculation unit. This unit uses the cosine similarity to evaluate the trust prediction between the users based on their latent representations, which can be written as:

$$CosineSimilarity(T_{U_i}, T_{U_j}) = \frac{T_{U_i}^T T_{U_j}}{\|T_{U_i}\|\|T_{U_j}\|} \tag{3.21}$$

For our CSTRP model, we employ the squared loss function to optimize our model, which can be written as:

$$L = \sum_{T \in T^+ \cup T^-} \theta_{ij} (T^+_{(u_i,u_j)} - T'_{(u_i,u_j)})^2 \tag{3.22}$$

where θ_{ij} refers to the threshold parameter of training loss. T is our training set. T^+ and T^- stand for the explicit trust relations and the unobserved trust relations, respectively. $T'_{(u_i,u_j)}$ refers to the ground-truth label of user pairs. Note that the output of CSTRP is the value of the trust at the range of [0, 1], which is different from the binary classification of either 1 or 0. Here, 1 refers to the trusted pairs, and 0 refers to the opposite. We leverage back-propagation to train and update the parameters of the weight matrix with batches on every layer.

Finally, we rank the similarity scores of the testing pairs based on the descending order of the scores. The testing pairs whose similarity scores are larger than 0.5 are regarded to have a trust relationship.

4

Experiment and Analysis

In chapter 4, we propose a framework to conduct social networks trust prediction, which consists of a two-level attention mechanism, 2-hop diffusion mechanism and a pair-wise neural network based on a heterogeneous graph. We explain the principles and the equations of these three components and use the square loss function to optimize our model. In this chapter, we conduct a set of experiments on the real-world dataset Epinions to evaluate and analyze our CSTRP model. Our experiment mainly focuses on the following questions to validate our model.

- Q1: Whether our proposed CSTRP method has a better performance compared with both classical and state-of-the-art methods?
- Q2: Whether our CSTRP is stable in the large-scale social networks with semantics information?
- Q3: How do the two-level attention mechanism and the 2-hop diffusion mechanism contribute to the trust prediction in the learning representation, respectively?
- Q4: What's the effect of the parameters taking in our trust prediction performance?

4.1 Datasets and Settings

4.1.1 Datasets

Table 4.1: STATISTICS OF EPINIONS DATASET

Dataset	Number of Users	Number of Items	Number of Ratings/Reviews	Number of Trust Relationships
E100	2,138	100	20,542	31,560
E200	3,163	200	34,119	45,319
E300	4,162	300	48,202	63,510
E400	4,998	400	60,743	76,902
E500	5,553	500	70,814	86,405
Epinions	8,356	34,483	364,114	153,486

Table 4.2: STATISTICS OF TRIPLES

Dataset	Number of Entities	Number of Relationships	Number of Triples	Training	Test
E100	2,238	5	20,542	94,680	18,936
E200	3,363	5	34,119	135,957	27,192
E300	4,462	5	48,202	190,530	38,106
E400	5,398	5	60,747	230,706	46,143
E500	6,053	5	70,814	259,215	51,843
Epinions	42,839	5	364,114	460,455	92,091

We aim to predict the trust relationship between users who do not have an explicit trust relationship. Here, we use our CSTRP model to compute the similarity based on users' latent representations (see equation 4.2.1). If the similarity score of the pairwise users is larger than 0.5, then this user pair is predicted to have a trust relationship. However, in the literature, there is no existing dataset that contains all the attributes, such as social relationships and preferences, etc. To evaluate our model, we use a real-world public dataset, Epinions [88]. This is a consumer website that contains consumers' ratings and reviews, as well as the trust relationships between online users. After purchasing an item, a user can give a rating and write a review on this item.

This review is visible to other users. Other users can give another score (helpfulness) based on the quality of the review to form a trust relationship between them. This trust relationship is publicly assessable and the information of dataset can reveal users' preferences, the items' attributes and the interaction between users-and-items and user-and-user. This dataset has been widely used in the study of trust prediction research [70].

To further explore the effectiveness and stability of CSTRP on the different sizes of datasets, we extract five sub-datasets from Epinions based on the frequency of items clicked by users. They are named as E100, E200, E300, E400 and E500 where the frequency of items clicked by users is on top of 100, 200, 300, 400 and 500, respectively, among all items. For these datasets, each user has at least 5 interactions with items, including ratings and reviews and each item has at least 15 interactions with users. We use 80% of the datasets for training and 20% of them for testing. The details of these datasets are shown in Table 4.1 and Table 4.2

4.1.2 Parameter Setting

For the proposed CSTRP model, we set the number of the negative sample as 2, the learning rate as 0.0001, the dimension of latent features from PMF and Doc2vec both as 32, the hidden layer of the deep neural network as 5, the window size as 15, the dimension of the two-level attention vector as 64, the epoch as 50, and the training batch as 64.

4.2 Baselines

To answer Q1, We compare our model with the classical and the state-of-the-art methods to verify the effectiveness of our proposed CSTRP model. In addition, two variants of CSTRP are tested to investigate the effectiveness of our heterogeneous attention mechanism and 2-hop diffusion mechanism, respectively. In our experiment, we take the accuracy of trust prediction as our evaluation metrics, which can be written as numbers of predicted trust relationships/numbers of labeled trust relationships $\times$100 %. Although some general GNN models like GAT [8] can also be used to address the prediction problem, they do not consider the attention score on edges in the diffusion process. Therefore, their performance is not good as the methods that adopt the attention score on edges in the diffusion process, which has already been studied and indicated by Wang et al. [87]. In contrast, our model considers the edges' attention score in the diffusion mechanism. Therefore, it will have better performance than the general GNN model like GAT.

So, we compare our method with the following baselines.

- **Trust propagation** [35]: It is a classical structure-based network trust prediction method that experiments on the path between users to evaluate their trust relationships.
- **Matrix Factorization** [61]: It is a typical low-rank approximation-based trust prediction method, which obtains the representation of trust relationships by using the matrix factorization on the matrix of document-term and linkage adjacency.
- **hTrust** [89]: It is a state-of-art model for trust network prediction, which contains the low-rank approximation method as well as additional knowledge information.
- **DeepTrust** [70]: As a state-of-the-art model for trust prediction, it is insensitive to data sparsity, and it carries out trust relationship prediction, which combines item properties and user behaviors through a pairwise neural network.
- **CSTRP**$_{\text{att}}$: It is a variant of CSTRP, which removes the heterogeneous attention mechanism and treats all the propagation paths equally by assigning the same weight on each path.
- **CSTRP**$_{\text{2hop}}$: It is a variant of CSTRP, which removes the 2-hop diffusion mechanism and only focuses on the directly connected neighbours. Therefore, it is the 1-hop variant of CSTRP.
- **CSRTP**: The proposed heterogeneous graph attention neural network employs a heterogeneous attention mechanism and a 2-hop diffusion mechanism simultaneously.

4.3 Implementation

4.3.1 Extracting Features from Rating and Reviews

To conduct our trust prediction experiment, we first use PMF (Probability Mass Function) and Doc2vec to capture the latent features of both users and items from ratings and reviews. PMF is a probabilistic matrix factorization algorithm. It can decompose the user rating matrix R_{i*j} into the product of a user feature matrix U_{i*k} and an item feature matrix V_{k*j} with the same dimension. This matrix decomposition shares the vector and feature space, aiming at discovering their potential features, including the latent vector of users and items. As an extension model of

Word2Vec, Doc2vec can learn fixed-length feature representations from sentences, paragraphs, and documents. Doc2vec overcomes the shortcomings of the bag-of-words model since it can work on unfixed sentence length and accept sentences of different lengths as training samples. Here we leverage the Doc2vec algorithm to obtain the vector representation of users and items from different review documents.

4.3.2 Index Matching

While we use PMF for data processing, we also generate the index of users and items to bridge a connection between the representation and the entities (users and items). Although we generate the index number to label the user and item in the above data processing, the index numbers may be different in different tasks in this model. Therefore, we carry out a code mapping operation, aligning the indices on both sides, and taking the serial number, so that the subsequent construction of the heterogeneous graph can correspond to the previous index.

4.3.3 Constructing a Heterogeneous Network Graph

A. Construction of a Bipartite Graph

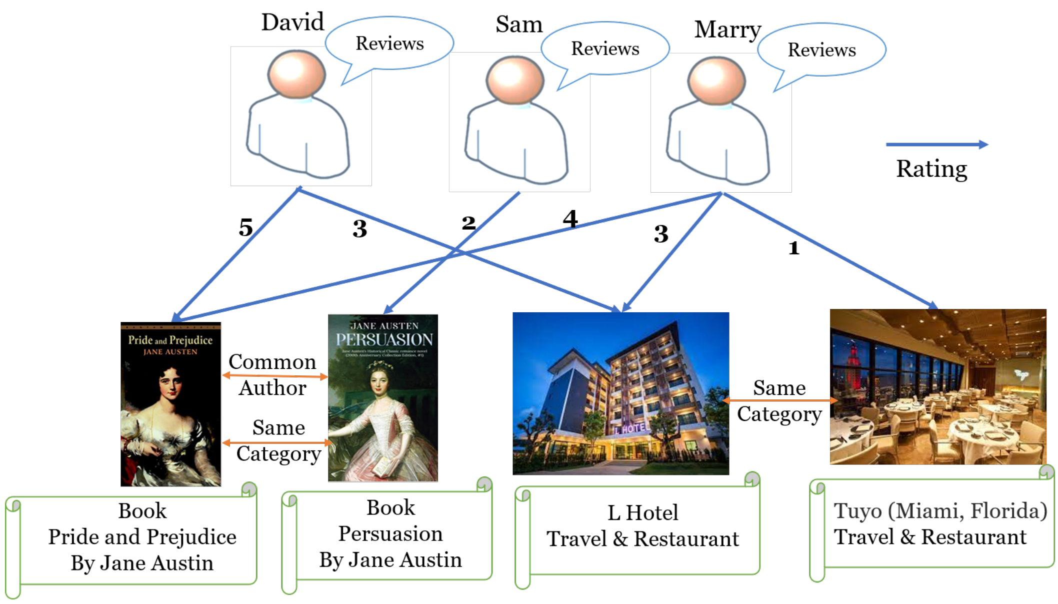

Figure 4.1: An example of A superimposed bipartite graph and triples

To construct our heterogeneous social network graph, we need a bipartite graph between users and items as an auxiliary role in the following knowledge graph construction. Figure 4.1

shows that David-Pride and Prejudice, David-L Hotel, Sam-Persuasion, Pride and Prejudice-written by-Jane Austen, etc., are examples of bipartite graphs. Through this bipartite graph, we label whether a user clicks on an item or not. Then, we sort the items that users clicked on by time to generate the sample and its required representation sequences. Depending on the above bipartite graph (user-item), we then superpose a knowledge graph of triples (e.g., David-5 (ranking) - Pride and Prejudice) to better obtain the representation of users. Our entire framework is based on a superimposed bipartite graph with triples, which is shown in Figure 4.1.

B. Construction of a Knowledge Graph

Before we enter the heterogeneous attention mechanism part, we need to learn the representation of distributed vectors between entities and relations. For this purpose, we construct a knowledge graph as the input of our model. As a special graph data, a knowledge graph (KG) can be regarded as a knowledge base on the semantic web. The purpose of a knowledge graph is to represent the relationship between entities that refer to things in the real world, such as people, items, places, etc. In essence, relationships are used to express some kind of connection between different entities, such as Dan and David are “colleagues”, Lily and Alex are ”friends ”, etc. Building such a knowledge graph gives us a structured knowledge base, and when we perform a search, we can directly get the final answer by keyword extraction (”Lily”, ”Alex”, ”friend”) and matching on the knowledge base.

In our social network knowledge graph, the nodes inside are the entities; the edges symbolize such a triple (h, l, t), (head, label, tail); and each edge represents the labeled relationship between the entity head and tail. We currently treat users, items and the attributes of items as entities and the KG can be “David-R5-Pride and Prejudice”, “Prejudice-written by-Jane Austen”, etc. Here, ranking “R5” and “written by” are this KG’s relationships. In the dataset Epinions, we construct a heterogeneous graph that comprises 8,356 users, 34,483 items and 5 kinds of ratings. In this experiment, we take ranking scores as our main relationship elements, since the ranking score from 1 to 5 shows the level of approval, from disapproval to highly approval. The meta-path set user-rating-item is employed to perform the experiment.

4.3.4 Adding Heterogeneous Attention Mechanism and 2-hop Attention Mechanism

The heterogeneous attention mechanism and the 2-hop diffusion mechanism are added to the model to investigate whether the user has clicked on an item or not. With this operation, we can construct a training set and a test set and obtain a better representation of users and items as the input of the CSTRP model. In the training of the representation vectors, we first map the entities into the relational space and then construct translation relations between the two projected entities. The specific training steps are shown as follows. Firstly, we employ the encoder-decoder framework: the encoder uses our heterogeneous attention mechanism to compute the embedding of entities and relations on the social network graph, and then the decoder is assigned to the relationship prediction task. Instead of only aggregating information passed from directly connected nodes, our CSRTP model collects information from the multi-hop neighbourhood. By using the attention diffusion mechanism, we can aggregate more information on the context neighbourhood compared with existing methods that only focus on the one-hop neighbourhood.

After adding this hierarchical attention mechanism and the 2-hop diffusion mechanism into our graph triples, we feed them into our pair-wise neural network to further capture the characteristics of users'. Since the representation training part and the trust relationship prediction part are two separate parts of our model, we do not need to rerun the whole process to perform the prediction for each pair of nodes. We can reuse the trained representations of users during the process of inference and only rerun the trust prediction part to obtain the result of each user pair.

4.4 Result and Analysis

To answer the questions from Q1 to Q4, we conduct two comparison experiments and use a case study to verify the effectiveness and the stability of the CSTRP model in the social network trust prediction. In addition, we also analyze the impact of parameters on our embedding dimension, the negative instances, and the depth of layers in the neural network on the trust accuracy.

4.4.1 Performance Comparison

Experiment 1: In this experiment, we compare the effectiveness of our CSTRP with baselines in the trust prediction on the largest Epinions dataset. Figure 4.2 shows the experiment results.

Based on the results, our observations that can answer Q1 are as follows:

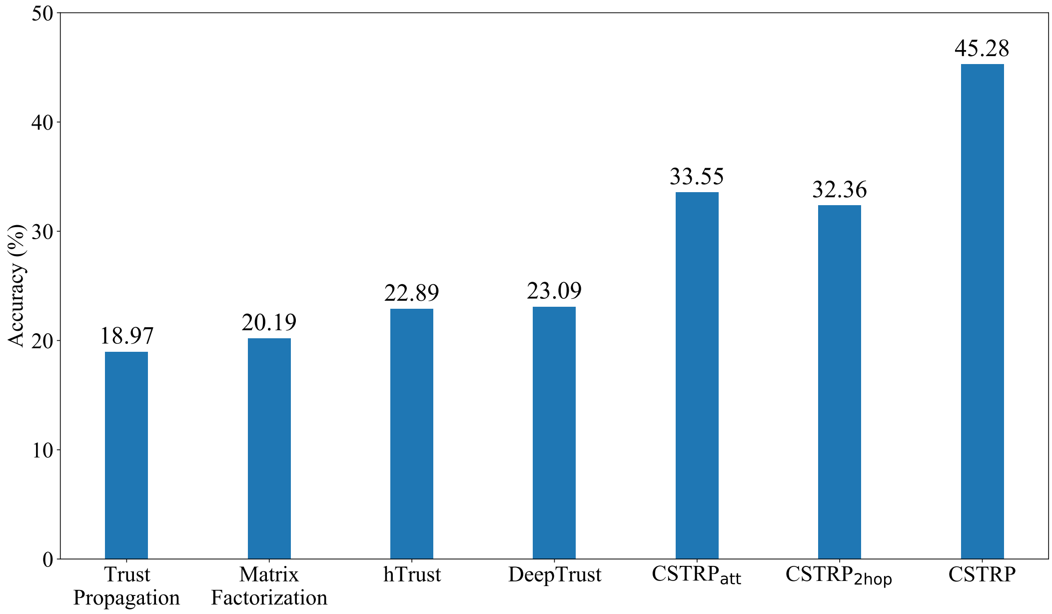

Figure 4.2: The Comparisons of Trust Prediction on Epinions

- **Observation 1:** Our proposed CSTRP model outperforms all the baseline methods, which supports the theoretical analysis of our CSTRP model. As shown in Figure 4.2, the accuracy of trust prediction of our CSTRP is 45.28%, which is much better than the other methods. Compared to the two classical methods, i.e., Trust Propagation and Matrix Factorization, that only consider the low approximation or the network structure, our CSTRP takes both the semantic and the structure of social networks into account. Compared to hTrust which uses matrix factorization with the regularization of rating similarity to capture features, our CSTRP applies the neural network, which better mines features from the latent space. Compared to DeepTrust method which also employs the pair-wise deep neural network, our CSTRP incorporates a heterogeneous graph attention mechanism and a multi-hop diffusion mechanism, which can comprehensively obtain the users' representation.

- **Observation 2:** Both pair-wise neural network models (CSTRP and DeepTrust) outperform the other baseline methods that are based on the low approximation and the network structure. It demonstrates that adding this pair-wise neural network into the prediction task can better capture users' characteristics.

- **Observation 3:** hTrust outperforms MF, pointing out the importance of adding additional knowledge such as aggregating the information from other group reputations in the trust prediction task. From this aspect, it verifies the imperative role in collecting the neighbours' information on the node representation.

- **Observation 4:** Trust Propagation [35] has the worst performance over all the methods. It means that using the low-rank approximation-based algorithm (CSTRP, DeepTrust, hTrust, MF) can improve the accuracy of trust prediction compared to the Trust Propagation method which only focuses on a simple network structure.

- **Observation 5:** The results of CSRTP, $\text{CSRTP}_{\text{att}}$ and $\text{CSRTP}_{\text{2hop}}$ in Figure 4.2 can show the vital influence of the hierarchical attention mechanism and multi-hop diffusion mechanism intuitively. Both $\text{CSRTP}_{\text{att}}$ and $\text{CSRTP}_{\text{2hop}}$ outperform other baselines and have similar accuracy values. It demonstrates that the heterogeneous attention mechanism is as important as the 2-hop attention diffusion mechanism in trust prediction and makes similar contributions to the accuracy. Besides, CSTRP has a better performance than $\text{CSTRP}_{\text{att}}$. As we mentioned in the baselines, $\text{CSTRP}_{\text{att}}$ is a variant of CSTRP that only discards the heterogeneous attention mechanism and keeps the 2-hop diffusion mechanism at the same time. Therefore, the out-performance of CSTRP indicates that the 2-hop diffusion mechanism based on the heterogeneous attention graph helps boost the performance in the trust prediction task.

Discussion 1: Based on the above observations in Experiment 1, we have analyzed the effectiveness of CSTRP from the numerical result perspective. The superior performance of CSTRP exists in Experiment 1 due to the effectiveness of the heterogeneous attention mechanism and the 2-hop diffusion mechanism on the trust prediction. To make clear the intrinsic influence mechanism of effectiveness on the trust prediction, we will provide a further discussion about the reasons for the effectiveness of these two components.

- **Effectiveness of hierarchical attention mechanism.** The two-level attention mechanism adopted in CSTRP assigns different weights to both nodes and their belonged meta-paths. On the one hand, such different node-level attention weights strengthen the identification on the preference of the node's neighbours. On the other hand, these different attention weights reflect the different contributions that neighbours make inherently. Similarly, the different attention weights on path-level are also highly correlated with the user's preference identification on each meta-path. So, CSTRP has the capability of semantic adaptability when capturing the characteristics of users. Therefore, CSTRP can get more accurate information from the semantic context in the online social network with this two-level attention mechanism, thereby improving the accuracy.

- **Effectiveness of 2-hop diffusion mechanism.** As the observations from Figure 4.2 and Figure 4.3, the performance of CSTRP is better than $\text{CSTRP}_{\text{2hop}}$, which illustrates the good performance of introducing the 2-hop diffusion mechanism. 2-hop diffusion mechanism makes it accessible for the CSTRP to aggregate information from indirectly connected nodes and weights the propagated information according to the decay factor, enlarging the receptive field and capturing more details of relationships between users, thereby improving the accuracy of trust prediction.

- **Summary.** Thus, a combination of the two-level attention mechanism and the 2-hop attention diffusion mechanism benefits our CSTRP model. With these two components, CSTRP outperforms both classical and state-of-the-art models in Figure 4.2. From this perspective, we can also get the answer for Q3.

Experiment 2: In this experiment, we compare CSTRP with DeepTrust (the state-of-the-

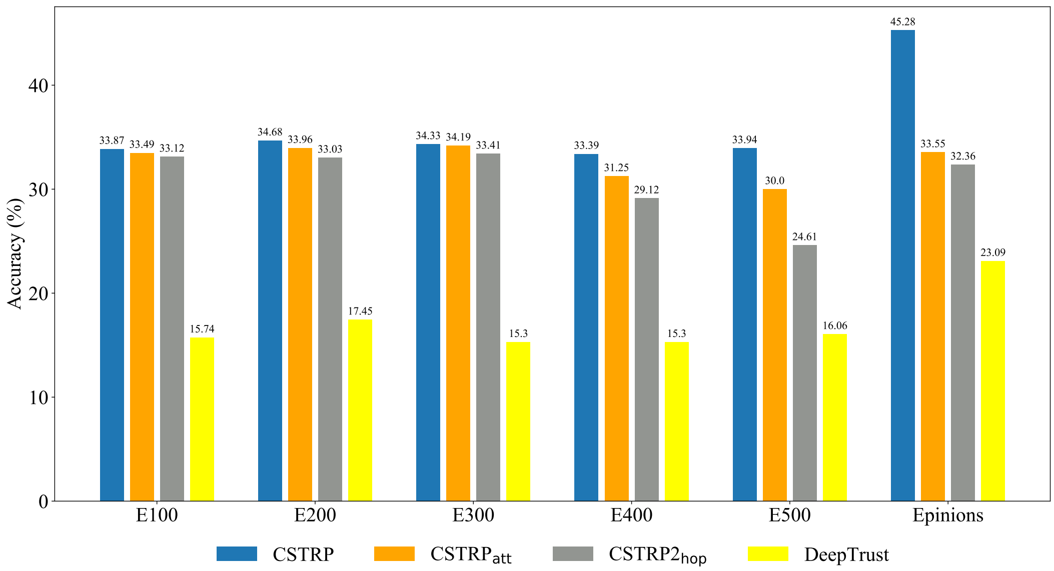

Figure 4.3: Performance on Different Size Datasets

art) and two variants of CSTRP on the different sizes datasets of Epinions to test its stability. The experimental results in Figure 4.3 can answer Q2, and our observations are summarised as follows.

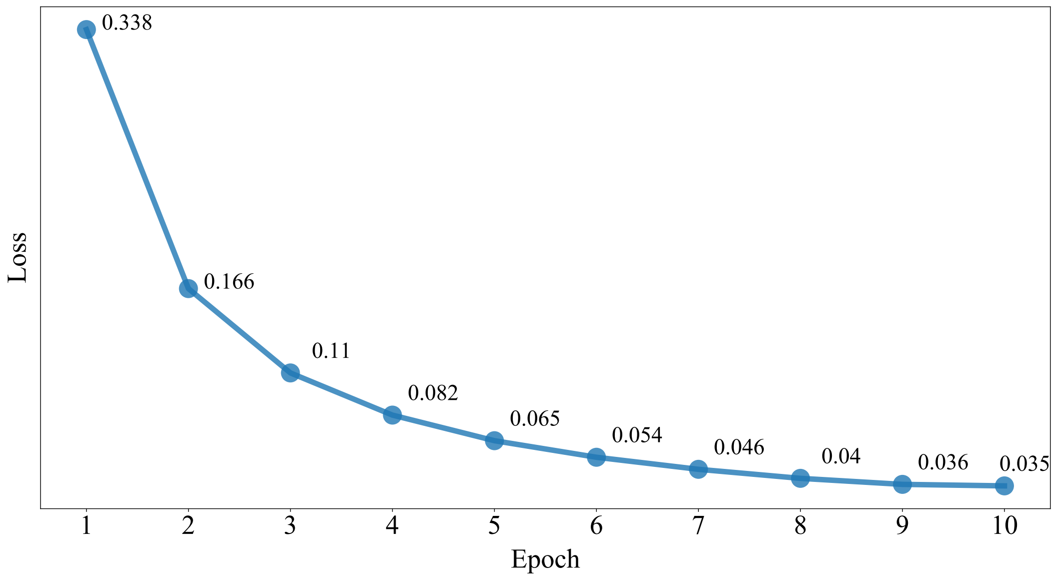

Figure 4.4: Loss of CSTRP on Each Epoch

- **Observation 1:** Our CSTRP achieves the best performance compared with the state-of-the-art DeepTrust method and the two variants of CSTRP over the six datasets on the trust prediction, including the smallest dataset E100 and the largest dataset Epinions. The trust prediction accuracy value of CSTRP shows a steady trend from E100 to Epinions. It illustrates that the CSTRP behaves steadily as the dataset information increases.

- **Observation 2:** $\text{CSTRP}_{\text{2hop}}$ and $\text{CSTRP}_{\text{att}}$ also outperform DeepTrust over all the datasets, which also demonstrates the stability of heterogeneous attention mechanism and 2-hop diffusion mechanism on the online social networks with the semantics context.

- **Observation 3:** To investigate the stability of CSTRP on the Epinions dataset, we also count the loss of CSTRP at each epoch. Although we set 50 epochs when setting the parameters, as we use an early stop to train the model to avoid overfitting, it ends at the 10^{th} epoch. Figure 4.4 shows that our model is convergent and has certain generalization performance and stability.

Discussion 2: As we mentioned above in observations, Experiment 2 has illustrated the stability of the CSTRP model. To analyze the reasons for the stability of CSTRP and how these influence the trust prediction, we will give a further explanation and discussion.

- **Noise immunity.** The noise immunity is one of the reasons that support the stability of CSTRP. Specifically speaking, the two-level attention mechanism adopted in CSTRP assigns different weights on both the structure and the semantics, which discards the redundant and irrelevant information while focusing on the highly relevant information. Therefore, CSTRP is capable of some noise immunity based on the online social networks with the semantics context.

- **Comprehensive representation learning.** The comprehensive representation learning can be another reason for the stability of CSTRP. More specifically, three elements contribute to this comprehensive representation of learning and can be summed as follows. (1) The two-level attention mechanism makes it accessible for CSTRP to have a more complete representation learning capability than baselines, since it shapes a more accurate heterogeneous relation graph. (2) Furthermore, the 2-hop diffusion mechanism can aggregate information between indirectly connected users, which completes the graph structure and extracts more information from the original online social network. (3) In addition, the pair-wise deep neural network of CSTRP can better capture and represent users' characteristics from the latent space, contributing to the representation learning and stabilizing the trust prediction results. In this regard, CSTRP can mine a comprehensive representation learning in online social networks and thus improve the trust prediction accuracy.

- **Summary.** Under the combined effect of the three elements: two-level attention mechanism, multi-hop diffusion mechanism and pair-wise neural networks, our CSTRP model can equip with the capabilities of reducing noise and forming a comprehensive and accurate social network graph structure. These two capabilities can adapt to the semantically real-world social networks and finally stabilize the CSTRP model.

4.4.2 Case Study

In this section, we use a case study to further explain the effectiveness of the two-level attention mechanism of our CSTRP. To illustrate **the effectiveness of node-level attention**, we take the user *beaner* in the dataset Epinions as an illustrative example. Given a meta-path User-Rating-Item which describes the level of user's approval on the item, we calculate the neighbours' importance of *beaner* based on the meta-path. The following Figure 4.6 shows their attention scores. As we can see from Figure 4.5, *beaner* connects to *Mountain*, *Las Vegas* and *Mammoth Mountain*, which all belong to the category of Hotel and Travel. In addition, it connects to *Bust-A-Move 99 for PlayStation 1* and *Outburst for Windows* as well, while both *Bust-A-Move 99 for PlayStation 1* and *Outburst for Windows* belong to the category of Game. Furthermore, it

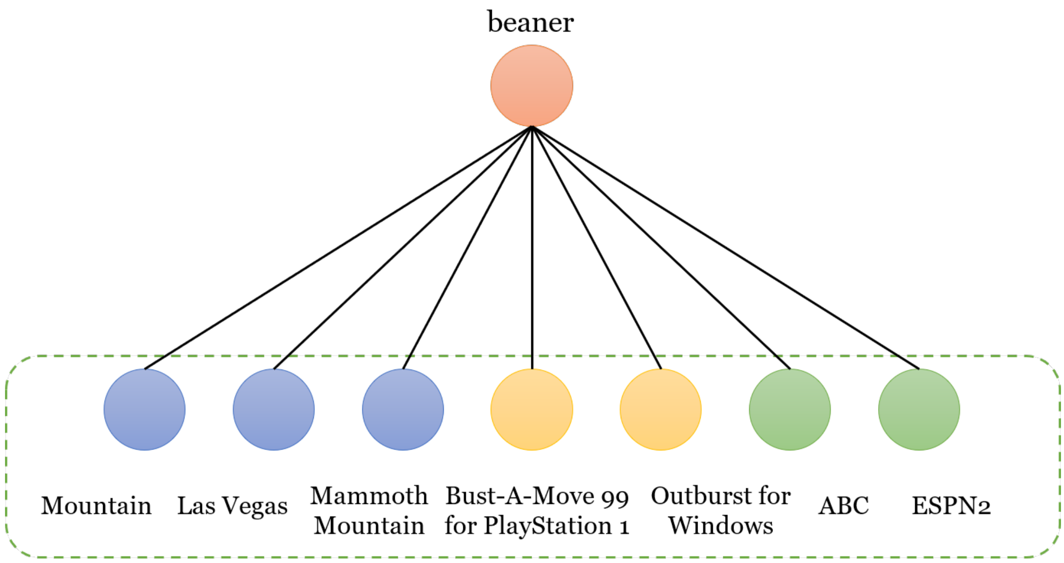

Figure 4.5: Neighbours of *beaner* based on meta-path

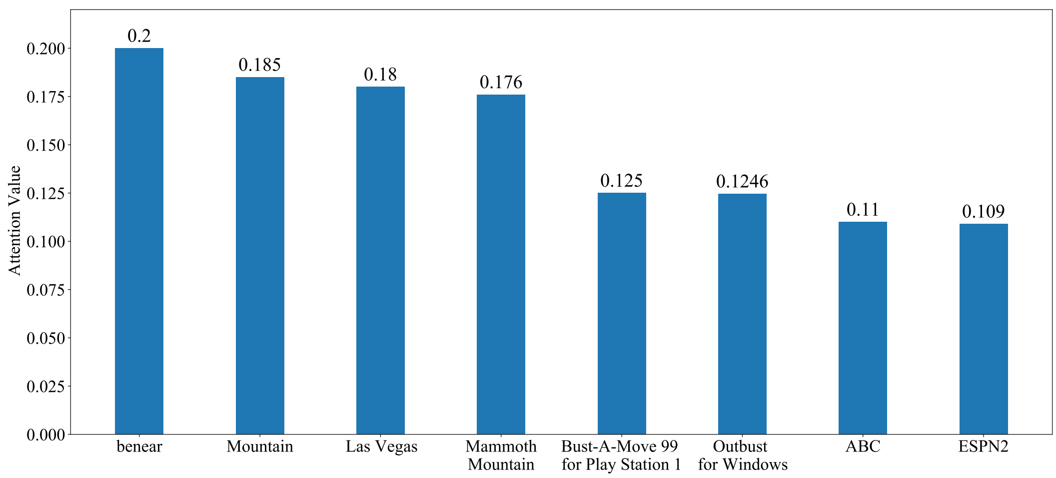

Figure 4.6: Attention values of the *beaner* neighbours

also connects to *ABC* and *ESPN2*, while both *ABC* and *ESPN2* belong to the category of Media. As we can see from the result, the highest node-level attention value of *beaner* illustrated in Figure 4.6 is *beaner* itself, which indicates the most crucial part is the node itself when learning its representation. Since neighbours' information is used to support the representation of node learning, this result can be considered reasonable. Then, the second-highest attention value is *Mountain* and the third-highest is *Las Vegas*, which indicates that the category of Hotel and Travel plays an essential role in the information aggregation of *beaner*. Since the rest neighbours do not belong to the category of Hotel and Travel, they have lower attention scores in the ranking. It also means that they are not so crucial as items in the category of Hotel and Travel when aggregating the information in *beaner* representation learning. By highlighting the

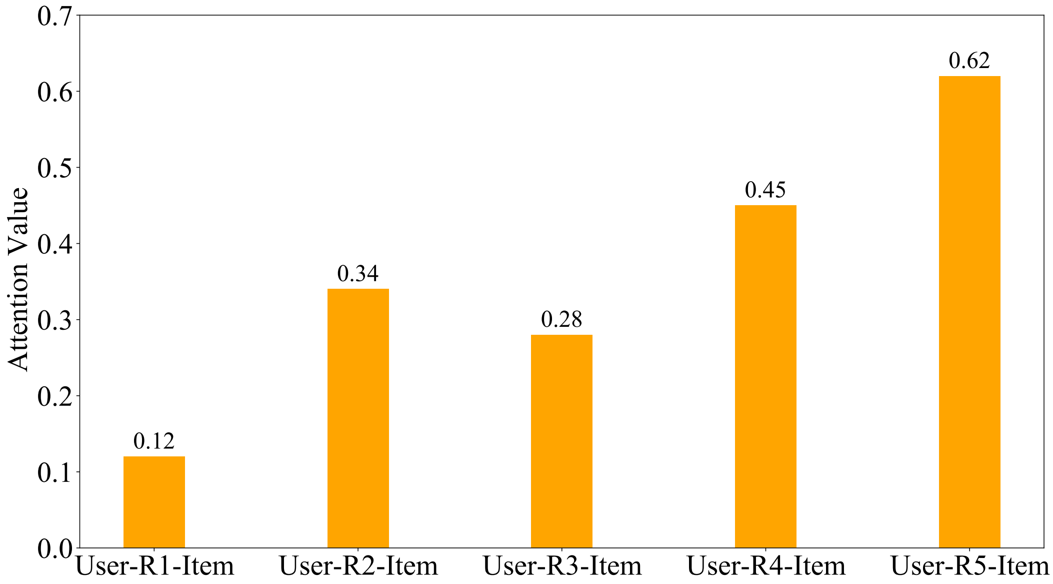

Figure 4.7: Attention value on single meta-path

top contributors with high attention values, the node-level attention mechanism can distinguish contributions among different neighbours, thereby improving the accuracy of trust prediction.

To analyze **the effectiveness of meta-path level attention** under the heterogeneous attention mechanism, we report the attention value of the single meta-path in Figure 4.7. In our dataset, we have five kinds of relationships between users and items, named from R1 to R5, to indicate the relationships from disapproval to highly approval. In our experiment, CSTRP assigns the highest attention scores on User-R5-Item and lowest attention scores on User-R1-Item. It means that User-R5-Item is considered as the most crucial meta-path while User-R1-Item as the least important in CSTRP when considering the users' preferences. It is reasonable for the reason that the user's preference and the ratings are highly correlated. For instance, a sports lover may give high marks for sports products, while a user who does not like sports may give low marks. On the contrary, if we assign the equal attention value to all these meta-paths, the user's preferences can not be clarified well under different relationships between users and items. Thus, it will lead to the low accuracy of trust prediction. In this sense, the meta-path level attention allows different semantic weights on these meta-paths, thereby equipping CSTRP with the capability of semantic adaptability.

For these reasons, our method consider both node-level and path-level attention mechanisms in this trust prediction task.

4.4.3 Influence of Parameters

To answer Q4, we conduct several experiments on different parameters to explore their influence on the trust prediction.

Dimension of Embedding

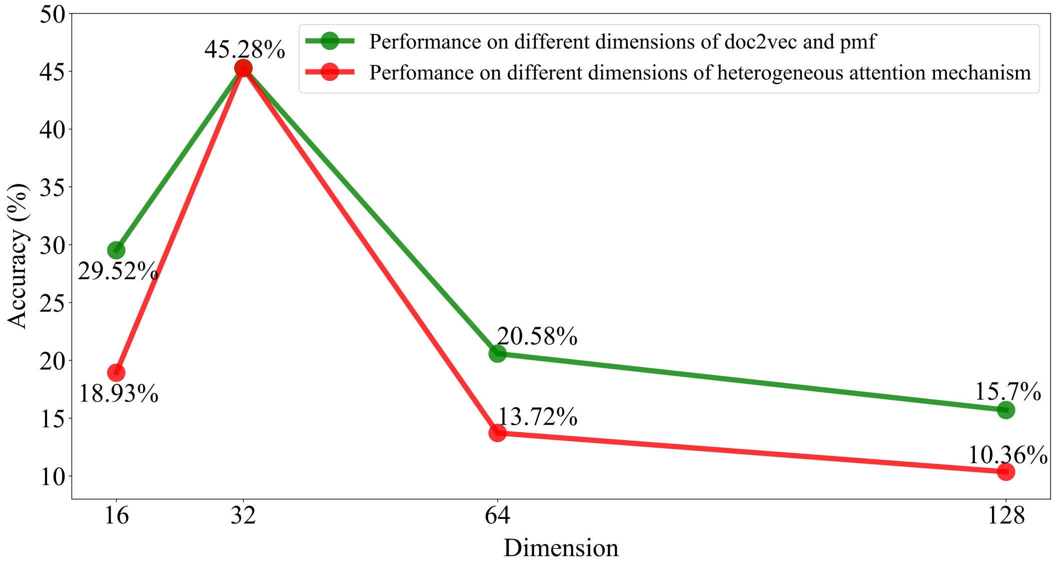

Figure 4.8: Performance of CSTRP on different dimensions

- **Dimension of doc2vec and pmf.** Since both the review and rating vectors should be concatenated together as the representation of users' attributes, these two dimensions need to be consistent with each other. To explore the impact of these two dimensions working on the performance of trust prediction, we change the dimension of doc2vec and pmf in the range of [16, 32, 64, 128] simultaneously with a fixed model setting of 50 epochs, 64 batches and 0.0001 learning rate. As shown in Figure 4.8, we observe that the accuracy of CSTRP grows with the dimension of the review and the rating vectors. The accuracy arrives at the highest point when the dimensions of these two vectors are set as 32. After increasing the dimension by 32, the performance of CSTRP starts to get worse due to overfitting.

- **Dimension of heterogeneous attention mechanism.** To explore the influence that the heterogeneous attention embedding dimension functions on the trust prediction performance, we vary the dimension under node-level and path-level attention mechanisms in the range of [16, 32, 64, 128]. The result in Figure 4.8 shows that the accuracy of the trust prediction increases and reaches the best performance when the dimension is 32. It is because that CSTRP requires a suited dimension for the sake of encoding the semantics information, and excessive dimension may interrupt the performance by producing additional noise.

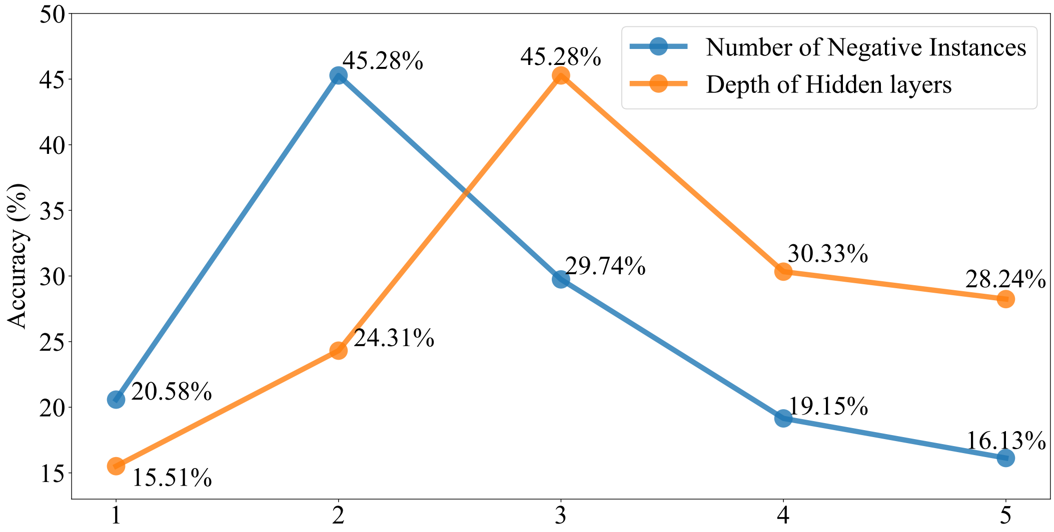

Figure 4.9: The Performance on Datasets with Different Sizes.

Number of Negative Instances

The process of our pairwise neural network training is actually the process of continuously adjusting the network weights in line with the ground truth trust relationships. To handle a large neural network, negative sampling [70] is used to modify a small amount of the weights, rather than the whole training sample. In our experiment, we take user pairs without trust relationships as negative instances during our neural network training process. We vary the number of negative instances from 1 to 5 to discover the variance of performance. As shown in Figure 4.9, the accuracy of trust prediction achieves the best when the number of negative instances is 2. The accuracy of CSTRP begins to decline after the negative instance number increases to 2, due to the noises produced by too many unobserved negative instances. In this case, the optimal number of our negative instances is 2.

Depth of Layers in Deep Neural Network

To investigate the effect of hidden layers on the CSTRP's performance, we conduct experiments to vary the number of hidden layers from 1 to 5. From the result in Figure 4.9, we can observe that the performance of CSTRP increases when the hidden layer increases from 1 to 3. It expresses that the number of hidden layers has a positive effect on the latent feature learning. However, the performance becomes worse in the trust prediction when the layers are larger than 3. From the result, we set the depth of layers as 3. The results also point out that different depths of layers affect the performance of CSTRP model with its impact on the final user representation. It is due to deeper hidden layers owning the better capability to capture complex user features from latent space.

5

Conclusions and Future Research

In this chapter, we summarize our work and conceive our future research direction extended on this work.

To conduct social networks trust prediction on the semantically social network, it is imperative to consider both structure and semantics based on a heterogeneous graph of social networks at the cost of the least noise. To this account, we study the social network embedding problem and propose a model CSTRP. We design a heterogeneous graph including these three types of nodes: User, Relationship and Interest. Then according to these three types of nodes, we introduce a stratification of attention mechanism to clarify the importance of all these relationships and semantic information by assigning different attention values to them. Besides, we also develop a 2-hop attention diffusion to enlarge the receptive field based on the edge attention scores between users. After doing that, we leverage a pairwise multi-layer projection to further extract the users' characteristics. For the optimization, we employ the squared loss function to supervise the learning process. Extensive experiments have demonstrated that the CSTRP model can better capture users' preferences from online social networks with semantics context and achieve superior performance on the trust prediction compared with other state-of-the-art methods.

This work has brought up a method to carry out the social network trust prediction by leveraging both structural and semantics information. However, social networks contain rich information, including the static and dynamic, and we only focus on a small comer of the static perspective to compute the user similarity based on the context. To further improve our model, we will study the future work in the following direction. First, we will consolidate our CSTRP framework by embedding data from more meta-paths such as the category, place and register time, etc. Second, we will take dynamic elements into account to adapt to the real dynamic world. Third, we will modify our model and make it into an end-to-end model to improve the effectiveness of the parameter adjustment and deployment.

References

[1] A. Jøsang and S. Pope. *Semantic constraints for trust transitivity*. In *APCCM*, vol. 43, pp. 59–68 (2005).

[2] I. T. Jolliffe. *Principal components in regression analysis*. Principal Component Analysis pp. 129–155 (1986).

[3] T. Cox and M. Cox. *Multidimensional scaling*. Chapman and Hall/CRC (1994).

[4] B. Perozzi, R. Al Rfou, and S. Skiena. *Deepwalk: Online learning of social representations*. In *SIGKDD*, pp. 701–710 (2014).

[5] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. *Line: Large-scale information network embedding*. world wide web pp. 1067–1077 (2015).

[6] S. Cao, W. Lu, and Q. Xu. *Grarep: Learning graph representations with global structural information*. In *CIKM*, pp. 891–900 (2015).

[7] A. Grover and J. Leskovec. *node2vec: Scalable feature learning for networks*. In *SIGKDD*, pp. 855–864 (2016).

[8] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. *Graph attention networks*. In *ICLR* (2018).

[9] Y. Yuan, Z. Tian, C. Wang, F. Zheng, and Y. Lv. *A q-learning-based approach for virtual network embedding in data center*. Neural Computing and Applications **32**(7), 1995 (2020).

[10] C. Williams. *On a connection between kernel pca and metric multidimensional scaling*. Advances in Neural Information Processing Systems **46**(1), 11 (2002).

[11] J. B. Tenenbaum, V. De Silva, and J. C. Langford. *A global geometric framework for nonlinear dimensionality reduction*. Science **290**(5500), 2319 (2000).

[12] S. T. Roweis and L. K. Saul. *Nonlinear dimensionality reduction by locally linear embedding*. Science **290**(5500), 2323 (2000).

[13] K. Berahmand, E. Nasiri, M. Rostami, and S. Forouzandeh. *A modified deepwalk method for link prediction in attributed social network*. Computing **103**(10), 2227 (2021).

[14] H. Yao, X. Chen, M. Li, P. Zhang, and L. Wang. *A novel reinforcement learning algorithm for virtual network embedding*. Neurocomputing **284**, 1 (2018).

[15] N. S. Altman. *An introduction to 0kernel and nearest-neighbor nonparametric regression*. The American Statistician **46**(3), 175 (1992).

[16] C. Cortes and V. Vapnik. *Support-vector networks*. Machine Learning **20**(3), 273 (1995).

[17] J. R. Quinlan. *Simplifying decision trees*. International Journal of Man-machine Studies **27**(3), 221 (1987).

[18] S. E. Dreyfus. *Artificial neural networks, back propagation, and the kelley-bryson gradient procedure*. Journal of Guidance, Control, and Dynamics **13**(5), 926 (1990).

[19] M. R. Islam, A. R. M. Kamal, N. Sultana, R. Islam, M. A. Moni, *et al. Detecting depression using k-nearest neighbors (knn) classification technique*. In *IC4ME2*, pp. 1–4 (2018).

[20] E. K. Boahen, W. Changda, and B.-M. Brunel Elvire. *Detection of compromised online social network account with an enhanced knn*. Applied Artificial Intelligence **34**(11), 777 (2020).

[21] B. Xia, Z. Ni, T. Li, Q. Li, and Q. Zhou. *Vrer: context-based venue recommendation using embedded space ranking svm in location-based social network*. Expert Systems with Applications **83**, 18 (2017).

[22] M. Xin and L. Wu. *Using multi-features to partition users for friends recommendation in location based social network*. Information Processing & Management **57**(1), 102125 (2020).

[23] L. Wu, Y. Rao, Y. Zhao, H. Liang, and A. Nazir. *Dtca: Decision tree-based co-attention networks for explainable claim verification*. In *ACL* (2020).

[24] M. Nurek and R. Michalski. *Combining machine learning and social network analysis to reveal the organizational structures*. Applied Sciences **10**(5), 1699 (2020).

[25] A. Aakash, A. Tandon, and A. Gupta Aggarwal. *How features embedded in ewom predict hotel guest satisfaction: An application of artificial neural networks*. Journal of Hospitality Marketing & Management **30**(4), 486 (2021).

[26] C. Tu, Z. Zhang, Z. Liu, and M. Sun. *Transnet: Translation-based network representation learning for social relation extraction.* In *IJCAI*, pp. 2864–2870 (2017).

[27] P. Xu, W. Hu, J. Wu, W. Liu, B. Du, and J. Yang. *Social trust network embedding*. In *ICDM*, pp. 678–687 (2019).

[28] C. Tu, W. Zhang, Z. Liu, and M. Sun. *Max-margin deepwalk: Discriminative learning of network representation.* In *IJCAI*, vol. 2016, pp. 3889–3895 (2016).

[29] C. Tu, H. Liu, Z. Liu, and M. Sun. *Cane: Context-aware network embedding for relation modeling*. In *ACL*, pp. 1722–1731 (2017).

[30] T. N. Kipf and M. Welling. *Semi-supervised classification with graph convolutional networks*. In ICLR (2017).

[31] J. Tang, M. Qu, and Q. Mei. *Pte: Predictive text embedding through large-scale heterogeneous text networks*. In *ACM SIGKDD*, pp. 1165–1174 (2015).

[32] Z. Huang and N. Mamoulis. *Heterogeneous information network embedding for meta path based proximity*. IEEE Transactions on Knowledge and Data Engineering (2017).

[33] J. Golbeck and J. Hendler. *Filmtrust: Movie recommendations using trust in web-based social networks*. In *IEEE CCNC*, vol. 96, pp. 282–286 (2006).

[34] P. Massa and P. Avesani. *Controversial users demand local trust metrics: An experimental study on epinions. com community*. In *AAAI*, vol. 5, pp. 121–126 (2005).

[35] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. *Propagation of trust and distrust*. World Wide Web pp. 403–412 (2004).

[36] J. Golbeck and J. Hendler. *Inferring binary trust relationships in web-based social networks*. ACM Transactions on Internet Technology **6**(4), 497 (2006).

[37] L. Xiong and L. Liu. *Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities*. IEEE Transactions on Knowledge and Data Engineering **16**(7), 843 (2004).

[38] L. Li, Y. Wang, and E.-P. Lim. *Trust-oriented composite service selection and discovery*. In *SOC*, pp. 50–67 (2009).

[39] E. Gray, J.-M. Seigneur, Y. Chen, and C. Jensen. *Trust propagation in small worlds*. In *IFIPTM*, pp. 239–254 (2003).

[40] G. Liu, Y. Wang, M. A. Orgun, and E.-P. Lim. *Finding the optimal social trust path for the selection of trustworthy service providers in complex social networks*. IEEE Transactions on Services Computing **6**(2), 152 (2011).

[41] G. Liu, F. Zhu, K. Zheng, A. Liu, Z. Li, L. Zhao, and X. Zhou. *Tosi: A trust-oriented social influence evaluation method in contextual social networks*. Neurocomputing **210**, 130 (2016).

[42] Y. Zhao, G. Liu, K. Zheng, A. Liu, Z. Li, and X. Zhou. *A context-aware approach for trustworthy worker selection in social crowd*. World Wide Web **20**(6), 1211 (2017).

[43] S. M. Ghafari, A. Beheshti, A. Joshi, C. Paris, A. Mahmood, S. Yakhchi, and M. A. Orgun. *A survey on trust prediction in online social networks*. IEEE Access **8**, 144292 (2020).

[44] W. Sherchan, S. Nepal, and C. Paris. *A survey of trust in social networks*. ACM Computing Surveys (CSUR) **45**(4), 1 (2013).

[45] J. Golbeck, B. Parsia, and J. Hendler. *Trust networks on the semantic web*. In *CIA*, pp. 238–249 (2003).

[46] J. A. Golbeck. *Computing and applying trust in web-based social networks*. University of Maryland, College Park (2005).

[47] C. Ziegler and G. Lausen. *Spreading activation models for trust propagation*. In *EEE*, pp. 83–97 (2004).

[48] Y. Zhang and T. Yu. *Mining trust relationships from online social networks*. Journal of Computer Science and Technology **27**(3), 492 (2012).

[49] C. Hang and M. P. Singh. *Trust-based recommendation based on graph similarity*. In *AAMAS TRUST*, pp. 71–81 (2010).

[50] Y. Zuo, W. Hu, and O. Timothy. *Trust computing for social networking*. In *ITNG*, pp. 1534–1539 (2009).

[51] J. Caverlee, L. Liu, and S. Webb. *Socialtrust: tamper-resilient trust establishment in online communities*. In *JCDL*, pp. 104–114 (2008).

[52] Y. Jing, H. Wang, K. Shao, and X. Huo. *Relation representation learning via signed graph mutual information maximization for trust prediction*. Symmetry **13**(1), 115 (2021).

[53] D. L. Nguyen, T. H. Nguyen, T. H. Do, and M. Yoo. *Probability-based multi-hop diffusion method for influence maximization in social networks*. Wireless Personal Communications **93**(4), 903 (2017).

[54] G. Liu, C. Li, and Q. Yang. *Neuralwalk: Trust assessment in online social networks with neural networks*. In *IEEE INFOCOM*, pp. 1999–2007 (IEEE, 2019).

[55] J. Yu, M. Gao, J. Li, H. Yin, and H. Liu. *Adaptive implicit friends identification over heterogeneous network for social recommendation*. In *CIKM*, pp. 357–366 (2018).

[56] H. Zhao, Y. Zhou, Y. Song, and D. L. Lee. *Motif enhanced recommendation over heterogeneous information network*. In *CIKM*, pp. 2189–2192 (2019).

[57] J. Shokeen and C. Rana. *A study on features of social recommender systems*. Artificial Intelligence Review **53**(2), 965 (2020).

[58] L. Hou, W. Kong, Y. Gao, Y. Chen, and X. Li. *Pa-gan: Graph attention network for preference-aware social recommendation*. In *Journal of Physics: Conference Series*, vol. 1848, p. 012141 (IOP Publishing, 2021).

[59] Y. Koren. *Factorization meets the neighborhood: a multifaceted collaborative filtering model*. In *SIGKDD*, pp. 426–434 (2008).

[60] J. Tang, H. Gao, and H. Liu. *mtrust: Discerning multi-faceted trust in a connected world*. In *WSDM*, pp. 93–102 (2012).

[61] S. Zhu, K. Yu, Y. Chi, and Y. Gong. *Combining content and link for classification using matrix factorization*. In *SIGIR*, pp. 487–494 (2007).

[62] C. Ding, T. Li, and M. I. Jordan. *Nonnegative matrix factorization for combinatorial optimization: Spectral clustering, graph matching, and clique finding*. In *ICDM*, pp. 183–192 (2008).

[63] J. Tang and H. Liu. *Trust in social media*. Synthesis Lectures on Information Security, Privacy, & Trust **10**(1), 1 (2015).

[64] H. Ma, H. Yang, M. R. Lyu, and I. King. *Sorec: social recommendation using probabilistic matrix factorization*. In *CIKM*, pp. 931–940 (2008).

[65] M. Jamali and M. Ester. *A matrix factorization technique with trust propagation for recommendation in social networks*. In *RecSys*, pp. 135–142 (2010).

[66] X. Yang, H. Steck, and Y. Liu. *Circle-based recommendation in online social networks*. In *SIGKDD*, pp. 1267–1275 (2012).

[67] V. P. Pauca, J. Piper, and R. J. Plemmons. *Nonnegative matrix factorization for spectral data analysis*. Linear Algebra and Its Applications **416**(1), 29 (2006).

[68] D. Cai, X. He, J. Han, and T. S. Huang. *Graph regularized nonnegative matrix factorization for data representation*. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(8), 1548 (2010).

[69] S. Jiang, K. Li, and R. Y. Da Xu. *Relative pairwise relationship constrained non-negative matrix factorisation*. IEEE Transactions on Knowledge and Data Engineering **31**(8), 1595 (2018).

[70] Q. Wang, W. Zhao, J. Yang, J. Wu, W. Hu, and Q. Xing. *Deeptrust: A deep user model of homophily effect for trust prediction*. In *ICDM*, pp. 618–627 (2019).

[71] Z.-g. Shi, J.-w. Liu, and Z.-l. Wang. *Dynamic p2p trust model based on time-window feedback mechanism*. Journal on Communications **31**(2), 120 (2010).

[72] E. ElSalamouny, V. Sassone, and M. Nielsen. *Hmm-based trust model*. In *FAST*, pp. 21–35 (2009).

[73] J. Dubey and V. Tokekar. *Bayesian network based trust model with time window for pure p2p computing systems*. In *GCWCN*, pp. 219–223 (2014).

[74] W. Fang, W. Zhang, Y. Yang, Y. Liu, and W. Chen. *A resilient trust management scheme for defending against reputation time-varying attacks based on beta distribution*. Science China Information Sciences **60**(4), 1 (2017).

[75] S. Deshpande and R. Ingle. *Evidence based trust estimation model for cloud computing services*. International Journal of Network Security **20**(2), 291 (2018).

[76] Z. Ye, T. Wen, Z. Liu, X. Song, and C. Fu. *An efficient dynamic trust evaluation model for wireless sensor networks*. Journal of Sensors (2017).

[77] J. Zhang and R. Cohen. *Evaluating the trustworthiness of advice about seller agents in e-marketplaces: A personalized approach*. Electronic Commerce Research and Applications **7**(3), 330 (2008).

[78] X. Zheng, Y. Wang, and M. A. Orgun. *Modeling the dynamic trust of online service providers using hmm*. In *ICWS*, pp. 459–466 (2013).

[79] M. E. Moe, B. E. Helvik, and S. J. Knapskog. *Tsr: Trust-based secure manet routing using hmms*. In *Q2SWinet*, pp. 83–90 (2008).

[80] X. Liu and A. Datta. *Modeling context aware dynamic trust using hidden markov model*. In *AAAI* (2012).

[81] F. Pennoni and E. Genge. *Analysing the course of public trust via hidden markov models: a focus on the polish society*. Statistical Methods & Applications **29**(2), 399 (2020).

[82] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu. *Heterogeneous graph attention network*. World Wide Web pp. 2022–2032 (2019).

[83] L. F. Ribeiro, P. H. Saverese, and D. R. Figueiredo. *struc2vec: Learning node representations from structural identity*. In *SIGKDD*, pp. 385–394 (2017).

[84] M. McPherson, L. Smith Lovin, and J. M. Cook. *Birds of a feather: Homophily in social networks*. Annual review of sociology **27**(1), 415 (2001).

[85] H. Liu, E. P. Lim, H. W. Lauw, M.-T. Le, A. Sun, J. Srivastava, and Y. A. Kim. *Predicting trusts among users of online communities: an epinions case study*. In *EC*, pp. 310–319 (2008).

[86] C. Shi, Y. Li, J. Zhang, Y. Sun, and S. Y. Philip. *A survey of heterogeneous information network analysis*. IEEE Transactions on Knowledge and Data Engineering **29**(1), 17 (2016).

[87] G. Wang, R. Ying, J. Huang, and J. Leskovec. *Direct multi-hop attention based graph neural network*. International Joint Conference on Artificial Intelligence (2021).

[88] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu. *Pathsim: Meta path-based top-k similarity search in heterogeneous information networks*. Proceedings of the VLDB Endowment **4**(11), 992 (2011).

[89] J. Tang, H. Gao, X. Hu, and H. Liu. *Exploiting homophily effect for trust prediction*. In *WSDM*, pp. 53–62 (2013).