

Movie Recommendation Using Link Prediction

by

Md Alif-uz-zaman Dhrubo
16101173

Tasfia Chowdhury Supty
21101001

Md Rakib Hossin
17101543

SM Ashfaque Rahman
16101052

Ananya Das
17101382

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
September 2021


© 2021. Brac University
All rights reserved.

Declaration

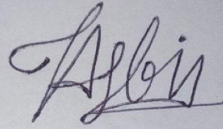
It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

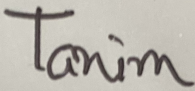
Student's Full Name & Signature:



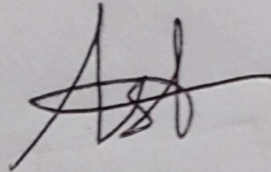
Md Alif-uz-zaman Dhrubo
16101173



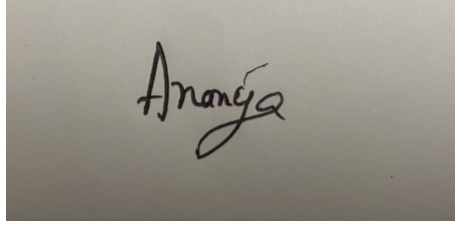
Tasfia Chowdhury Supty
21101001



Md Rakib Hossin
17101543



SM Ashfaqr Rahman
16101052



Ananya Das
17101382

Approval

The thesis/project titled “Movie Recommendation Using Link Prediction” submitted by

1. Md Alif-uz-zaman Dhrubo(16101173)
2. Tasfia Chowdhury Supty(21101001)
3. Md Rakib Hossin(17101543)
4. SM Ashfaqur Rahman(16101052)
5. Ananya Das(17101382)

Of Summer, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on 26 September,2021

Examining Committee:

Supervisor:
(Member)

Dr. Mohammad Zavid Parvez
Assistant Professor
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)



Md Anwarul Kaium Patwary, PhD
Lecturer
School of Physics, Mathematics and Computing
The University of Western Australia

Head of Department:
(Chair)

Sadia Hamid Kazir
Chairperson and Associate Professor , Computer Science and Engineering
Department of Computer Science and Engineering
Brac University

Ethics Statement

We, hereby declare that this thesis is based on the results we obtained from our work. Due acknowledgement has been made in the text to all other material used. This thesis, neither in whole nor in part, has been previously submitted by anyone to any other university or institute for the award of any degree.

Abstract

Link prediction is an important task for analyzing movie recommendation which also has applications in other domain like, information retrieval and bioinformatics. Proximity measure quantify the closeness or similarity between nodes in movie recommendation and form the basis of a range of applications in social sciences different quality based movie, information about user's choice, networking and connecting . Recommendation can be effective of link prediction sub-process, with unique nodes (users and items) and connections (similar user/item relationships and user/item interections). Through specific methods and techniques, the recommending systems try to identify the most appropriate items, such as types of information and good and propose the closest to the user's tastes. One of the easiest and most understandable and authorisation for locating people with the same preferences in the recommendation systems is mutual filtering that provides active performance data based on the ranking of a segment of people. In this model, the process is subject to scalability, with a growing number of users and movies. Across the other hand, when there is little information available on the ratings, it is essential to promote the system's performance. This study proposes an efficient dynamic graph prediction using link algorithm to predict the user's choice and recommended the movie based on that link prediction. Temporal information offers link occurrence behavior in the dynamic network, while community clustering shows how strong the connection between two individual nodes is, based on whether they share the same community. These model and methods have achieved higher prediction of recommending. We got better prediction by implementing Jaccard coefficient into methods. Furthermore, in the future, we will use more algorithms to improve the recommending based on the rating of the movies by sorting them for the users.

Keywords: Link prediction, Recommendation system, Graph algorithm, Jaccard coefficient, Network analyzing, Sparse network, Potential connection, The Naive Bayes.

Dedication

Firstly, this thesis is dedicated to our parents for their love, effort, endless support and encouragement which they always provided.

We are also thankful to our supervisor Dr. Mohammad Zavid Parvez. Without his guidance, support and motivation it would have been impossible for us to conduct the research.

Last but not the least, we want to share our gratitude towards the friends we found along the way, who shined the brightest in our darkest of days.

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.

Secondly, We would like to thank our co-supervisor Md Anwarul Kaium Patwary, PhD. He helped us through our difficult times and guided us to go forward

And finally to our parents without their throughout sup-port it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

Declaration	i
Approval	iii
Ethics Statement	iv
Abstract	v
Dedication	vi
Acknowledgment	vii
Table of Contents	viii
List of Figures	x
List of Tables	xi
Nomenclature	xi
1 Introduction	1
1.1 Recommendation System and Link Prediction	1
1.2 Problem Statement	3
1.3 Research Objective	4
2 Literature Review	5
2.1 Review	5
2.2 Similarity Based Methods	7
2.3 Local Approaches	7
2.4 Global Approaches	8
2.5 Quasi-Local Approaches	8
3 Background Studies	9
3.1 Link Prediction	9
3.2 Jaccard Coefficient	10
3.3 Sparse Network	12
3.4 Naive Byes	13

4	Methodology	15
4.1	Methodology	15
4.2	Algorithm	16
4.3	Dataset Analysis	17
5	Experimental Results and Analysis	19
5.1	Cs and Sm	19
5.2	Graph	19
5.3	SR	20
6	Conclusions	22
6.1	conclusion	22
6.2	future work	22
	Bibliography	25

List of Figures

2.1	Taxonomy for link prediction techniques	7
3.1	Finding missing link from link-prediction	9
3.2	Recommending movie by link-prediction	10
3.3	Pragmatic of Jaccard coefficient	11
3.4	Understand Jaccard Index	11
3.5	Network analyzing by Jaccard Coefficient	11
3.6	Finding potential connection by Sparse Graph	12
3.7	Finding connection capability	13
3.8	Naïve Bayes algorithm	14
4.1	Flowchart	15
4.2	A sample movie network	17
5.1	Cs and Smn	19
5.2	sm vs rating	20
5.3	sm vs cscore	20
5.4	graph1	21
5.5	graph2	21
5.6	graph3	21

List of Tables

4.1	dataset	18
4.2	Dataset After Pre-processing	18

Chapter 1

Introduction

1.1 Recommendation System and Link Prediction

A recommendation system is nothing but a means that helps us exert information that may be relevant to a specific user from a large pool of information and it is considered to be one of the most powerful tools in the digital world today. The main idea behind it is to build a system that can help a user to make calculated, smart decisions by using the information that has been gathering for ages [1]. The most common examples of recommendation systems would be the “Recommended for you” tab in Netflix or the “Friends you may know” section on your homepage on Facebook.

The idea of helping a user find items that they might like, based on the information we already had, has been around since the dawn of computing. The idea was first sprung in early 1979 by a system named Grundy [2]. Grundy was designed to be one of the first computer-based librarians, which could provide suggestions to what to read next based on the user’s history of the books that he or she has already checked out. A simple idea sprout out possibilities of playing around with the technology and people began to research the potential this system might have. Hence in early 1990, the first commercial recommending system was launched named the Tapestry. This inspired individual entities to make their recommending systems like the GroupLens Recommender System launched by a research lab at the University of Minnesota, USA which could help people find the desired article. This followed more study and development in the late 1990s and the milestone of that research was the Amazon Collaborative Filtering which is one of the most famous recommendations technology in the world today. Amazon Collaborative Filtering has completed changed the recommendation system outlook which has led to further development in the technology. New applications to the science are being recognized every day, usually called hybrid approaches because it integrates numerous approaches. Since then, the study and growth of Recommendation Systems have become widespread and play a huge role in online systems. Today, Netflix holds the title of being the most competitive in the recommendation system industry as they launch the competition Netflix Prize in 2006 announcing 1 million US dollars will be the prize money for the team that provides the best recommender system, and they finally got a winner in 2009. There are four types of Recommendation systems in the market, which are Collaborative Recommendation system, Content-based recommendation system, Popularity-based recommendation system, and Classification model[3].

Among all the four types, Content-based Recommender system and Collaborative Recommender system are the ones that are more popular and generally more used. The Classification model is fairly simple. It uses the features of the products and takes into account the preference of the users and determines if the user should be recommended a particular item or not. The main problem with the Classification model was that it was near impossible to collect so much information about the products and different users, and as the number of users and items increase in the dataset, the task becomes even more arduous. Moreover, even if the dataset is complete, the products and users could be so diverse that it is rigorous to categorize the nodes and determine an accurate classification. On the other hand, Popularity-based recommendation systems are more flexible. This method allows the ongoing popular trend in the market, for example, if an e-commerce business concludes that one of their product has heavy sales, it will be recommended to every customer. The biggest merit of this method is that there is no need for historical data because we are following the present trends. However, one of the main ideas behind recommendation systems is to provide a personalized recommendation to every individual, which this method fails to do.

Another type of Recommendation system is the Content-based Recommendation system which looks up the similar content of a specific item the user is looking for [4]. For instance, if a user is watching a movie, the system will offer recommendations of other movies of the same genre, actors, etc. In other words, similar products are recommended to the users. The items are classified based on their defining features, so this limits the different types of products that can be used to compare. The similarity between products is calculated using Euclidean Distance, Cosine Similarity, and Jaccard Similarity. This process does not require us to have any information about the users, just the features of the item, so using this recommendation system can be used to recommend an item for a new user. Nevertheless, this system requires extensive data regarding that particular item, which includes features so we can similarities between two different products.

Lastly, the most famous recommendation system method is Collaborative filtering . Collaborative filtering takes into account what content-based recommendation system was lacking. Collaborative filtering combines rating of objects to identify resemblances between people based on their ratings and tries to produce new recommendations for the users using these inter-personal comparisons. The reason why collaborative filtering is superior to all the other systems is that it doesn't require any machine-readable depiction of the substances and can be used on all products despite their features. The basic principle behind Collaborative filtering is the assumption that if people agreed on one topic in the past, there is a high possibility that they might agree on it in the future. Nonetheless, the system can get complicated if there are many users with conflicting interests, which can result in the data overlap and create a sparsity problem.

Recommendation systems have revolutionized many industries like, retail, media, banking, telecom, etc. but above all, it has made a huge impact on the e-commerce businesses. Remunerations of having a recommendation system are two folds for both the service providers and the users [5]. These recommendations have proven to help decision-making easily and also confirmed quality control. They cut transaction costs and boost profits as using the recommendation systems has made selling more effective. These systems also increased the Average Order Value (AOV) which is sig-

nificant to e-commerce businesses. There is nothing better than relevant and above all customized suggestions when you are shopping, and these systems bring that to you. When you order milk for an online grocery store, if the eggs are not there right underneath for you just to press add, imagine how time-consuming the shopping would be. These suggestions help increase user satisfaction and make sure that the user comes back to the store to shop again. Examples of e-commerce corporate which has amazing recommendation systems would be Amazon and Alibaba.

Video and audio streaming platforms are other areas where recommendation systems have made another huge impact as well. The algorithms used by Netflix, YouTube, Spotify, and many more are sophisticated and they are trying to keep up with the market trends and update their recommender engines accordingly. Recommendation systems are a technology that is being adopted by more and more businesses as the days go by. Even in research labs and universities, recommendation systems are used to move outside the catalog searches, however, the algorithm and data used for these recommendations will differ from a video streaming platform or an e-commerce business. Hence, the organization needs to understand which recommendation system to use to get more accurate and efficient results.

1.2 Problem Statement

Nowadays, movie recommendations are an easier way to classify the dynamic link prediction between the people in a community. These recommendations can be represented as graphs where the peak of the graph denotes an individual in any specific group and the trough indicate any kind of connotation amongst the people in that group. Implementation of link prediction in complex recommendations is one of the popular research topics and the main focus of all this research is to solve real-world problems, as it reflects the information about network topology. Network analysis has already been popular long before websites like Netflix, Amazon, Vudu. But there is some lack in predicting the dataset of it.

Application data storage methods have greatly increased our capacity to retain vast nodes over the years. Watts et al., famous for their report on the small-world effect and the implementation of Barabasi and Albert's scale-free network model a year later, started a new wave of study on this matter. For example, in the work of Jin et al., Barabasi et al., and Davidsen et al. or the survey of Newman, the work done on recommendation system has been thoroughly evaluated by questioning whether they can replicate some of the definite global structural features that is witnessed in real-life networks [6]. As the accessibility of database systems and networking is growing, more and more real-world network datasets are available and also getting complicated

As a result, in the real-world application, it's getting difficult to predict the dataset in the recommended system. The recommendation problem of movies has been studied before, they use machine learning to solve it but here we are using a graph algorithm of link prediction to solve it. To recognize the dynamics that manage the growth of connection is a complicated problem because the system and the datasets contain a giant number of capricious parameters. But approximately, a simpler problem is to work with is the relationship between two specific nodes.

1.3 Research Objective

This dissertation is based on developing a recommendation system for a movie using a graph algorithm of link prediction. Link prediction finds application in any context in which the network is only partially observable and we want to guess the unobserved part. Typically, link creation can be explained by one of two main reasons: interest identity or personal social relations. In this study, we consider users' profiles as a new dimension of an analysis classically composed of three-dimension users, rating and categories of movies and we build these connections using recommendations by the algorithm of link prediction [7].

One of the most basic problems that is faced when working with these data is that the link information between two nodes in the graph may be of uncertain nature, for example, there are exist an incorrectly stated link between two unrelated nodes or a simple link between two related nodes may be overlooked. So our goal of social link prediction is to solve these two problems at hand and try to predict the links between two nodes as accurately as possible.

We attempt to introduce a graph-based recommendation algorithm using link prediction which will incorporate the topological property in the recommendation system, trying to explain the links between users and movies. The choice of link prediction algorithm makes an implicit assumption of how the graph; the mechanism for how the graph grows. We require user interests and user details to get movie recommendations. The scheme will filter the data and send suggestions accordingly. Sequential information gives us a pattern of behavior and, on the other hand, community clustering tells us how strongly the nodes interact and affiliate with one another depending on the fact if they share the same community.

Chapter 2

Literature Review

2.1 Review

Assuming a sample of an unconstrained network in time, where each node represents an entity and each link indicates a relation between the pair of entities connected together. Link prediction problem refers to finding links that are not in the network but have not been detected in the current t -image or will be constructed during $t + \Delta t$ [8]. A complex network, consisting of areas representing individuals or organizations and links that define relationships or connections between areas, is considered an effective new way to represent and study many complex systems [9][10]. An effective recommendation system will help users easily find what they are looking for and thus save time, improve customer purchases and improve sales [11][12].

Recommendation algorithms generally take user characteristics and objects, and user object behavior as the basis of a recommendation program (such as open reviews and explicit searches, ordering or tapping tasks) as a response to user preferences [13]. One of the most effective and promising collaborative filtering (CF) offers suggestions that use user-only combination that can be defined as user-related or object-based depending on whether the collection of recommendations is based on finding similar users based on their interaction or similar items based on common users who have expressed interest in them [14][15]. CF also suffers from sparsity of data, without its success, when a separate combination of user object contributes to improper collection of users or objects. To alleviate this problem, a number of different suggestions have been made.

Recommendation (RS) systems, introduced by the Tapestry project in 1992, are one of the most effective information management systems [15]. Advanced helpful apps help users filter useless information to deal with overload information and provide personalized suggestions. There has been a lot of success in e-commerce to get the customer access to the products they like, and to improve the profitability of the business. In addition, to improve the ability to customize, the recommendation system is widely used on many multimedia websites to target media products to specific customers. Nowadays, co-filtering (CF) is the most effective method used by movie recommendation programs, based on a process close to the neighbors.

Jussi Karlgren was the first to propose the concept of a recommending system. He defined the notion of a recommending system as a "digital bookshelf" in his 1990 Columbia University technical study, "The Systems Development and Artificial Intelligence Laboratory, An Algebra for Recommendations" 1990. Karlgren proposes

that a researcher may stumble across a fascinating title when browsing for a paper in a bookshelf and take it out before returning to their main quest, regardless of their core objective. The result of this is that records of the same value are collected in small batches. As a result, when browsing certain pages, people who use the bookshelf quickly come across interesting texts.

A combination of all hybrid content-based recommendation programs and collaborative strategies in the provision of advice. The combination of these two approaches often deals with the use of content-based or collaborative-only tools [16] But the latest trends are suffering from a decline in demand for information. Jan and Gregory's algorithm uses simple, unambiguous versions of similarity measurements but the bonds between areas of similar communities have been shown to be more likely and more rated than interdependence between multiple cultures [17].

The most common form of suggestion is collaborative filtering. Collaborative filtering approaches are classified into two types: built-in and model-based. Neighborhood-based techniques are the most commonly used prediction algorithms in co-operative filtering systems [18][19]. Local-based interactive filtering methods are divided into two types: user-used and object-based approaches. Object-based techniques anticipate active user ratings based on computer information for comparable objects picked by the active user, whereas user-based methods predict active user ratings based on the same measures. As comparable calculation techniques, user-based and user-friendly methods generally employ the PCC (Pearson Correlation Coefficient) algorithm [19][20] and the VSS (Vector Space Similarity) algorithm. Classification models are utilized in model-based methods to train the previously mentioned model [21]. Kohrs and Merialdo proposed a collaboration sorting algorithm based on the similarity of locations, with the goal of balancing the accuracy of forecasts, particularly when only a few features are known.

Hofmann[22]proposed an algorithm based on a broad range of semantic analysis that may be present in the dynamic response. These methods all focus on capturing the user's object measurement matrix using low-level predictions, and use it to make other predictions. The basic premise of a low-level feature is that there are only a few numbers of features that affect preferences, and that the vector of user preferences is determined by how each feature works for that user. Niu et al [23].It is recommended that people watch online videos. This framework integrates videos with the same response to user viewing behavior by using video viewing logs and video information. Watch logs and video metadata, on the other hand, may not be available for actual use, particularly on cold video sites. Furthermore, they disregard user input, which can potentially be exploited to install comparable movies.

Luis M Capos et al.[24]looked at content-based filtering and interactive filtering as two classic recommendation systems. He proposed a new method that combines the Besesia network with collaborative filtering because they both have issues. The suggested approach is tailored to a specific situation and offers a variety of ways to create meaningful indications. Harpreet Kaur et al. created the hybrid system [25]. Using the technology, integrate the interaction algorithm and content screening. Utkarsh Gupta et al utilize a caterpillar to produce particular person data or specific item details in order to construct a collection. This is a useful method that is based on the hierarchical compilation of a recommendation program. Urszula Kulewska and colleagues [26]integration is proposed as a method to handle recommendation programs. Two computer group representation approaches were in-

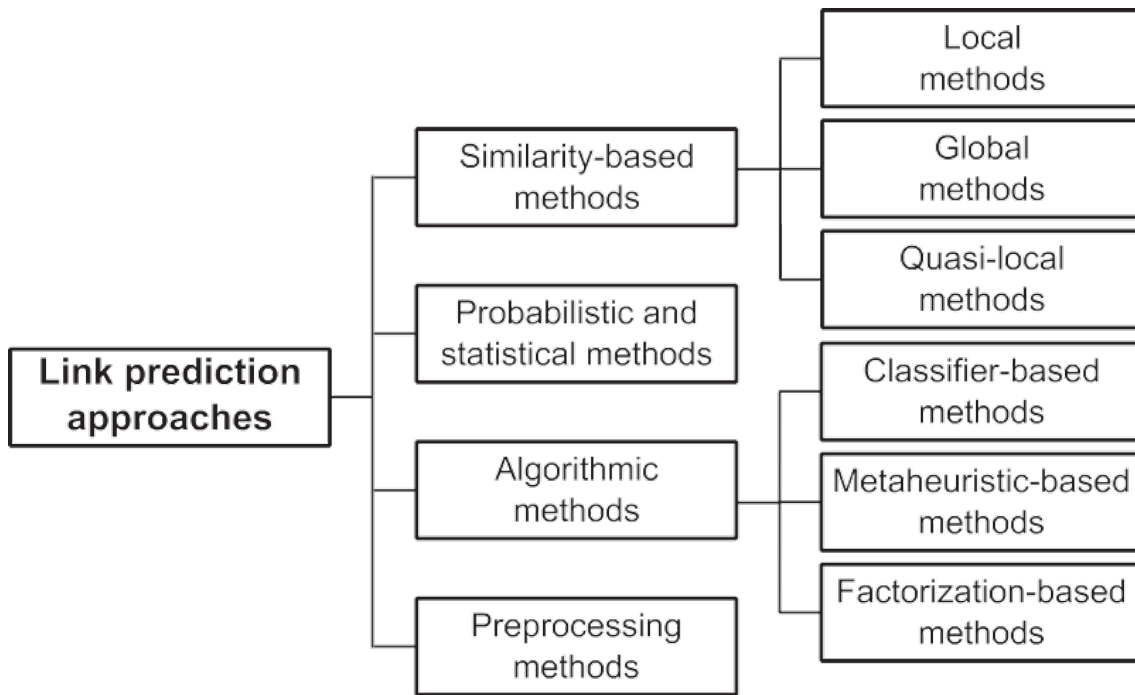


Figure 2.1: Taxonomy for link prediction techniques

roduced and evaluated. The performance of these two recommended approaches was compared using a centroid-based solution and memory-based filtering methods. As a consequence, the accuracy of the suggestions generated has increased significantly when compared to the centroid-based technique. Costin-Gabriel Chiru and colleagues [38] Suggested Movie Recommender is a software that provides movie suggestions based on user-generated data. This software attempts to address the issue of various suggestions to the user caused by disregarding certain data.

2.2 Similarity Based Methods

It is expected that if two nodes are similar in some aspects, they will have a better likelihood of creating a link. If two nodes are connected to one of their common nodes, for example, the first two nodes will have a better probability of forming a link. This method specifies a function $f(x, y)$ that returns parallel points between nodes x and y . This rating is determined for each node in the network that is intriguing, particularly those that do not have a visible link between them.

2.3 Local Approaches

This methodology uses neighborhood node configuration data to compute the similarity of one location to other nodes in a network [8]. In the case of vast networks, techniques focused on local comparisons are quicker than other methods available. These techniques perform admirably in locating connections to a wide range of social networks. In social networks, individuals form connections with one another; in this situation, people frequently establish contact with someone with whom they have mutual relationships. On Facebook, for example, there is a feature named by

both friends. In this feature, it is possible to discover how many common friends both have. When a person has a large number of regular friends, there is a good probability that you will submit a friend request to that person.

2.4 Global Approaches

These procedures link all nodes in the network and then detach them. These approaches award points to all linkages between places based on the specifics of each node. These approaches, unlike local methods, are not confined to assessing similarities. However, the complexity of these algorithms is quite high because they must be considered across each node's network. The complexity of these approaches is relatively significant, especially in dispersed networks [8].

2.5 Quasi-Local Approaches

These links connect the local and international routes. These approaches consider the average neighborhood as well as a fraction of the whole network. As a result, the complexity of these techniques is smaller than that of global methods, and since it causes a specific portion of the network and the local network, it performs better than local comparable methods [8]. Some Quasi-Local techniques have full network access. Therefore, the complexity of these approaches remains far lower than that of the rest of the globe.

Chapter 3

Background Studies

3.1 Link Prediction

Link prediction is the issue of forecasting the presence of a link between two elements in a network in network theory. It seeks to assess the likelihood of each non-existing connection in the network's existence (or creation) in order to discover a collection of missing between the users. These techniques make advantage of several of the network properties described earlier in this chapter, such as degree, clustering, and path lengths. This chapter discussed some of the essential techniques for both goals. As mentioned above, many more sophisticated computational approaches for both link prediction and entity resolution exist, and they will make useful further reading for computer scientists interested in this field.

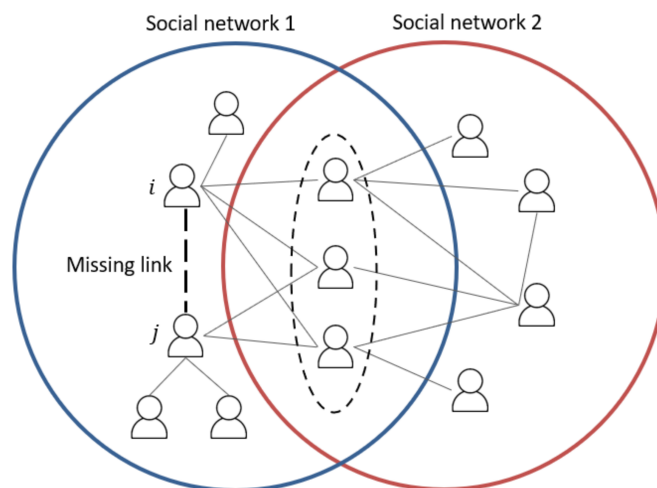


Figure 3.1: Finding missing link from link-prediction

Consider the network $G=(V,E)$, where V represents the network's entity nodes and $EE \subseteq |V| \times |V|$ represents the network's set of "true" links. We are provided the set of entities V as well as a subset of genuine links known as observed linkages. The objective of link prediction is to find undiscovered genuine connections.

The observed links correspond to genuine links at time t in the temporal formulation of link prediction, and the aim is to infer the set of true links at time $t+1$. In most cases, we are also provided a subset of unobserved links known as prospective links E' , and we must find real linkages among these potential links. The assessment of

the ranking of the predicted associations among the candidate nodes/links may be used to do link prediction. For example, depending on the author’s recent publishing history and the trend of research on related themes, we can forecast which papers that author will produce, read, or quote. Such investigations frequently need examining the vicinity of network nodes/links as well as the trends and connections of their similar neighbors. Link prediction is also referred to as link mining.

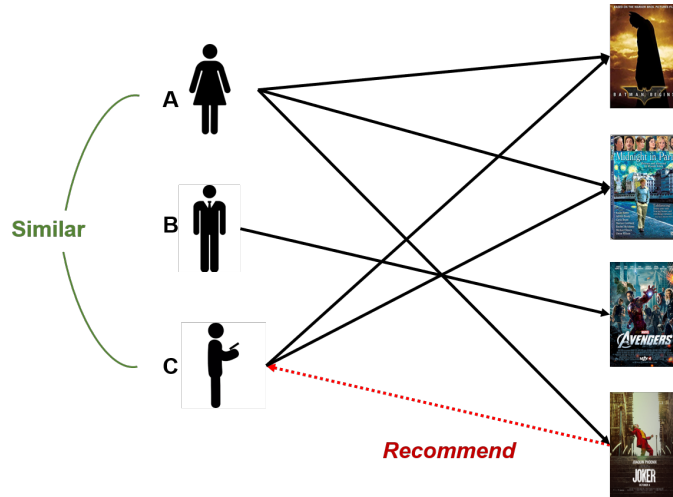


Figure 3.2: Recommending movie by link-prediction

The item descriptions are frequently technical data found in papers, online pages, and news bulletins. These item descriptions are weighted vectors, and user profiles are modeled as weighted vectors. The capacity to promote freshly launched goods to consumers is a benefit of this technique [27]. Complete item descriptions and thorough user profiles are required for content-based recommendation systems. This is the most significant drawback of such systems. Another drawback of content-based systems is privacy concerns, such as users’ reluctance to share their preferences with others.

3.2 Jaccard Coefficient

The Jaccard index, often known as the Jaccard similarity coefficient, is a statistic used to determine the similarity and diversity between sample sets. The Jaccard similarity coefficient analyzes members of two sets in order to determine which members are similar and which are different. It’s a percentage that ranges from 0 percent to 100 percent. The more similar two populations are, the larger the percentage. While analyzing text similarity, Jaccard similarity is useful when duplication is not an issue, but cosine similarity is useful when duplication is an issue. It is preferable to utilize Jaccard similarity for two product descriptions since repetition of a term does not diminish their similarity. It is calculated by dividing the intersection size by the combined size of the two sets. Duplicate elements are weighted the same way in multisets.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

This percentage indicates the degree to which the two sets are comparable. 1. Two

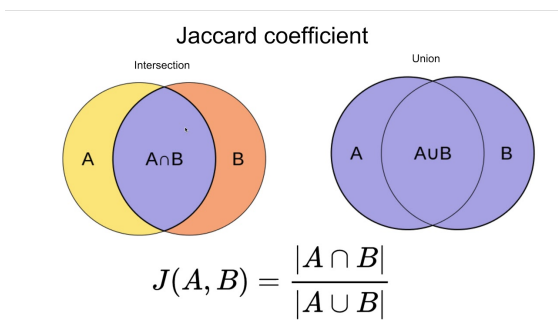


Figure 3.3: Pragmatic of Jaccard coefficient

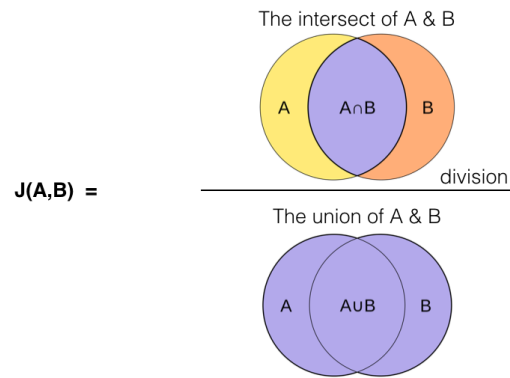


Figure 3.4: Understand Jaccard Index

sets that share all members are identical in every way. The greater the resemblance, the closer to 100 percent (e.g. 90 percent is more similar than 89 percent). 2. If they have no members in common, they are 0 percent similar. 3. The halfway point — 50 percent — denotes that the two sets share half of the members.

Based on close proximity of the two data sets, this approach may be used effectively without the need of data redundancy [28]. When significant document names were examined, the represented documents were shown appropriately, according to the findings. As a result, the system has a better precision and a smaller database than a conventional search page with other providers. According to [29] the search procedure starts with importing users' queries and comparing them to the database. If an input keyword matches the database's index of words, those words can be used to account for the primary keywords presented throughout the search process. The

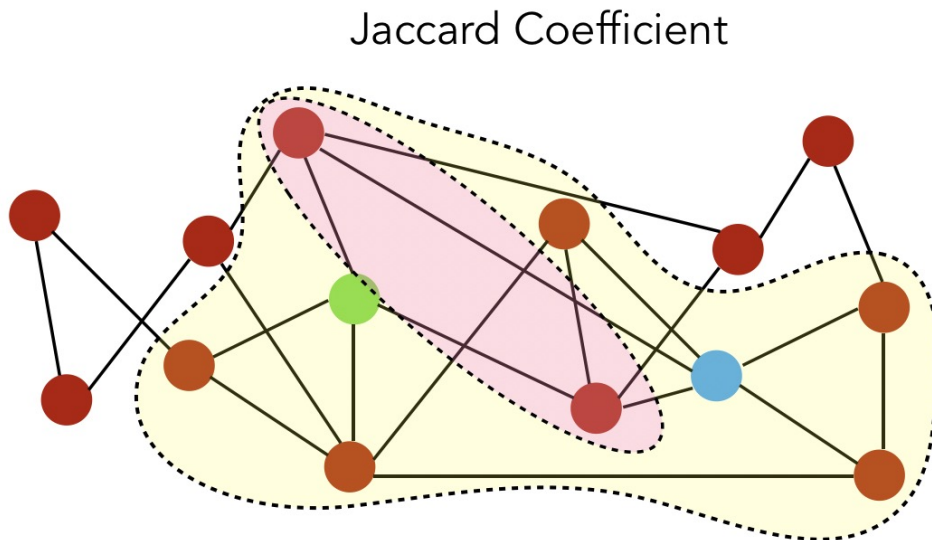


Figure 3.5: Network analyzing by Jaccard Coefficient

Jaccard coefficient is widely used in several fields. When utilized as a bag of words, the coefficient may be used to estimate how similar two pieces of text are . It might

also be used in social network analysis [30] to compare keywords in articles to see how similar writers are in their primary study areas. In cellular manufacturing, the metric was also used to categorize machines based on their individual components. Another interesting use of the Jaccard coefficient was to compare the lesion detection performance of a computer-aided diagnostic system to that of radiologists' manual detections. Lu et al. [31] have enhanced the scalability of a news recommendation system by including the coefficient into kmeans clustering while evaluating user similarity (distance).

3.3 Sparse Network

A sparse neural network is one in which just a subset of the potential connections exists. Consider a completely linked layer with some connections missing. It has a type of neural network that has a small number of connections. Because the high number of neurons and synapses impedes effective NN processing, researchers suggested a variety of training strategies to prune redundant synapses and neurons without sacrificing accuracy, including Sparse Coding, Auto Encoder/Decoder, and Deep Belief Network (DBN) [32]. After all, the idea of many fewer links is still a slang

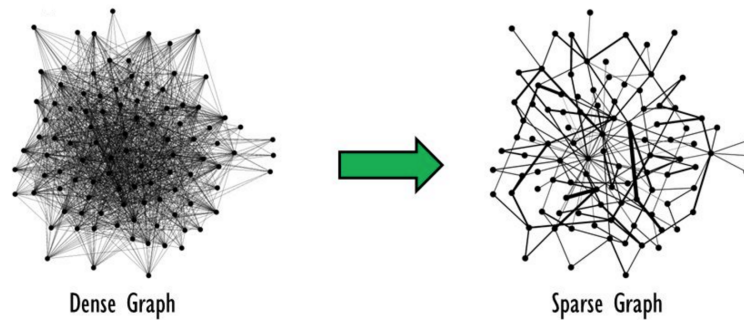


Figure 3.6: Finding potential connection by Sparse Graph

term. While a threshold may exist for a certain network, there is no universal criterion that defines what significantly fewer actually entails. As a result, there is no formal definition of sparsity for any finite network, despite widespread agreement that most empirical networks are sparse. However, there is a formal definition of sparsity in infinite network models, which is given by the behaviour of the number of edges (M) and/or the mean degree ($\langle k \rangle$) as the number of nodes (N) approaches infinity [33].

The network is considered to be sparse if the number of links M in a basic unweighted network of size N is significantly fewer than the maximum possible number of links M_{max} .

$$M \ll M_{max} = \binom{N}{2}$$

However, when dealing with a synthetic graph sequence G_N , or a well-defined network model for networks G_N of any size $N = 1, 2, \dots, \infty$, the displaystyle $\| \cdot \|$ takes on its

normal formal meaning:

$$M \ll M_{max} \iff M = o(M_{max}) \iff \lim_{N \rightarrow \infty} \frac{M}{M_{max}} = 0$$

The structural components of a network can be represented using a mapping function if the nodes in the network are not weighted. A sparse matrix is one in which the majority of the entries in the matrix are zero. However, if the majority of the elements are nonzero, the matrix is dense. The sparsity or density of the matrix is determined by the percentage of the zero element to the total number of elements in the matrix. As a result, in graph theory, if the number of connections is close to its maximum, the graph is said to as dense. Sparse graphs have fewer linkages than the maximum number of possible linkages. The sparsely linked network distribution

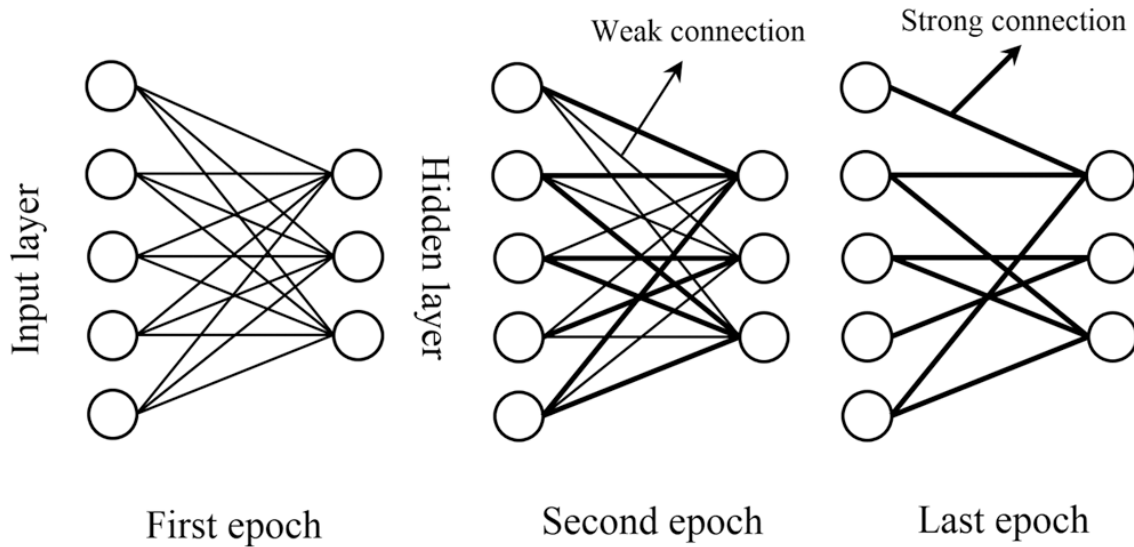


Figure 3.7: Finding connection capability

is scale free and power law. As the number of network connections grows, so does the network's departure from power law. The node similarity is one of the most important variables determining network connection. People in social networks, for example, are more likely to be linked if they have similar social backgrounds, interests, likes, and perspectives. If the complex surfaces of proteins or other molecules are identical or complementary, they are linked in biological networks [34].

3.4 Naive Byes

The naive Bayes model is a significantly simplified Bayesian probability model. In this model, consider the likelihood of a final conclusion given many connected evidence factors. Naive Bayes uses a similar method to estimate the likelihood of different classes based on different characteristics. This technique is commonly used in text classification and with issues that include numerous classes, such as in our movie recommendation, where we classify our movies for the user. It is a fast algorithm, and its classification approaches are based on the base theorem with the assumption of predictor independence. It is divided into two sections: Naive and Byes. A Naive-based model is relatively simple and straightforward to construct,

especially for a large number of datasets. The Bayes theorem, on the other hand, informs us the likelihood of a scenario based on the prior fact of the condition that may be connected to that situation [35]. It is highly efficient when working with a lot of network. The following is the connection between Hypothesis H and Evidence E

$$P(H | E) = \frac{P(E | H) P(H)}{P(E)}$$

The Naive Bayes presupposes that a feature’s influence on a class is independent of other features. In terms of prediction, the Naive Byes algorithm is highly popular. The naïve Bayes classification classifier has the advantage of using just a little amount of training data to estimate the classification parameters (variable means and variances). Since control variables are assumed, just the values of the data for each category must be computed, rather than the entire covariance matrix. Depending on the input, the Bayesian Classifier may calculate the most likely output. It is possible to input additional raw data and improve the probabilistic classifier during runtime. It also produces a high-quality and efficient result. In our work, we utilized the following algorithm. We evaluate a hypothesis given various evidence in

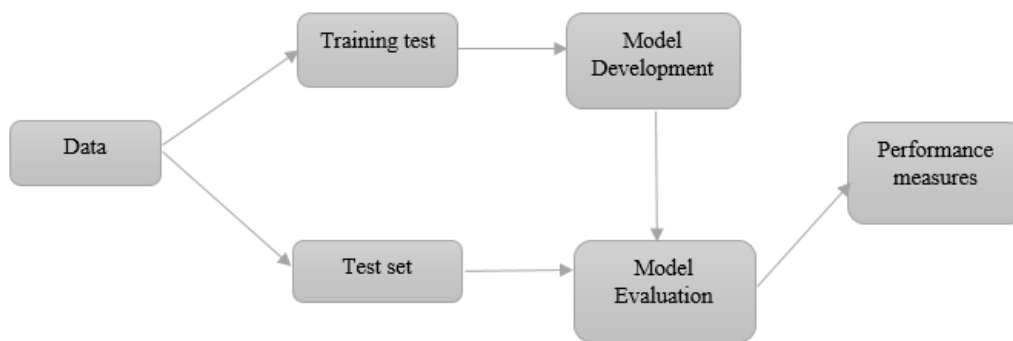


Figure 3.8: Naïve Bayes algorithm

actual datasets. As a result, computations become more difficult . It doesn’t need as much training data as other methods, yet it can handle both continuous and discrete data. We’re talking about learning classifiers here, and there are two types of learning classifiers: supervised and unsupervised learning classifiers. In a supervised learning context, naive Bayes classifiers can be trained very efficiently depending on the particular form of the probability model [36]. It frequently performs far better than one may expect in a variety of complicated real-world scenarios. Independent variables are taken into account for the aim of predicting or predicting the occurrence of an event. Despite its basic design and simplistic assumptions, naive Bayes classifiers typically outperform expectations in a wide range of complicated real-world scenarios.

Chapter 4

Methodology

4.1 Methodology

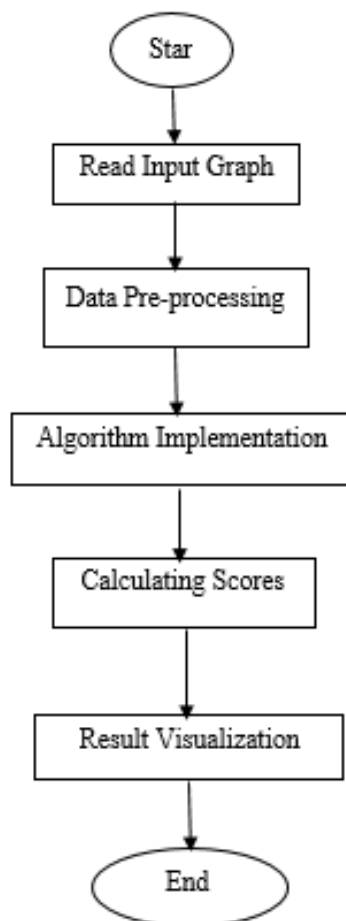


Figure 4.1: Flowchart

To recommend a movie to a user in according to the preferences of the user we need to find out the movies that the user is most likely to watch. To predict the movies that are likely to watch by the user, we need to take consideration of some factors that decides the probability of watching a movie. In our algorithm, we have taken

the user's previous watch history, ratings of the movies and favorite category of the user as our measurement factors.

In our algorithm, we have taken the user's previous watch history, ratings of the movies and favorite category of the user as our measurement factors. To achieve our desired goal, we need to take a heterogeneous graph as our input data where there will be four types of nodes, they are user, movies, ratings and category of a movie. So, we need to find a dataset including these nodes information to implement our algorithm. The flow of the processes to implement our algorithm is shown in fig-1. In this study, we have used similarity-based methods to find out the missing links in a movie network. In particular, we have built an algorithm which will generate a score measuring various similarities between movie to user and user to movie. In this case, we are observing the ratings of the movie and the categories as key factors. To get the links among movie to user and movie to movie we have taken a data set which includes all the four kinds of nodes that is required to execute our algorithm. However, we have used data splitting and cleansing to perform our algorithm smoothly.

4.2 Algorithm

Input Graph $G = [V, E]$

Assume user node U

Take preferred category list as input

Assume the list is C_p

Store all types of categories in a list C

Take a movie from the graph and its category set C_m

Find category score, $C_s = \frac{P(C_p) \cap P(C_m)}{P(C_p) \cup P(C_m)}$

Visit all movie node and find probability score for each movie, $S_m = (0.5 * \text{rating}) + (C_s * 0.75)$

Sort all the movies in descending order according to S_m .

Take one movie, M_1 and visit all movies for M_1 to generate a weight W_m , for the links with all other movies,

Assume category set for M_1 is C_{m1} and for another movie M_i , C_{mi}

Calculate similarity score $S_r = \frac{P(C_{m1}) \cap P(C_{mi})}{P(C_{m1}) \cup P(C_{mi})}$

Assign S_r for the particular edge as weight.

First of all, we take an input graph and a list of a user's preferred categories of movies. The graph has four types of nodes labeled as User, category, movies and ratings. After that we walk through each movie nodes and find out the Jaccard coefficient between the user's preferred categories and each movie's categories. For each movie in the network say for R_1 , we take the set of adjacent category nodes C . Then we take the set of preferred categories by the user. Finally, we find the Jaccard coefficient of these two sets and assign them to the particular movie in a new node adjacent to the movie node as category score. Then we take the rating of the movie and the category score which was generated by Jaccard coefficient and put them in the formula that we have formed. This will calculate the probability score for the movie to the user. Then we sort the movie nodes according to their overall probability score to be watched by the user. This concludes the first part of our algorithm. After that we move on to find movie to movie link prediction. To

find that we will firstly take one movie from the graph and then we will calculate the Jaccard coefficient for all the other movies in the network by the categories they belong to. We will assign the score as a weight to the edge of the two movies. This process will be done for all the movies in the network. Hence, all the movies will have an edge to any other movies in the network and the edge will carry the similarity score based on Jaccard coefficient.

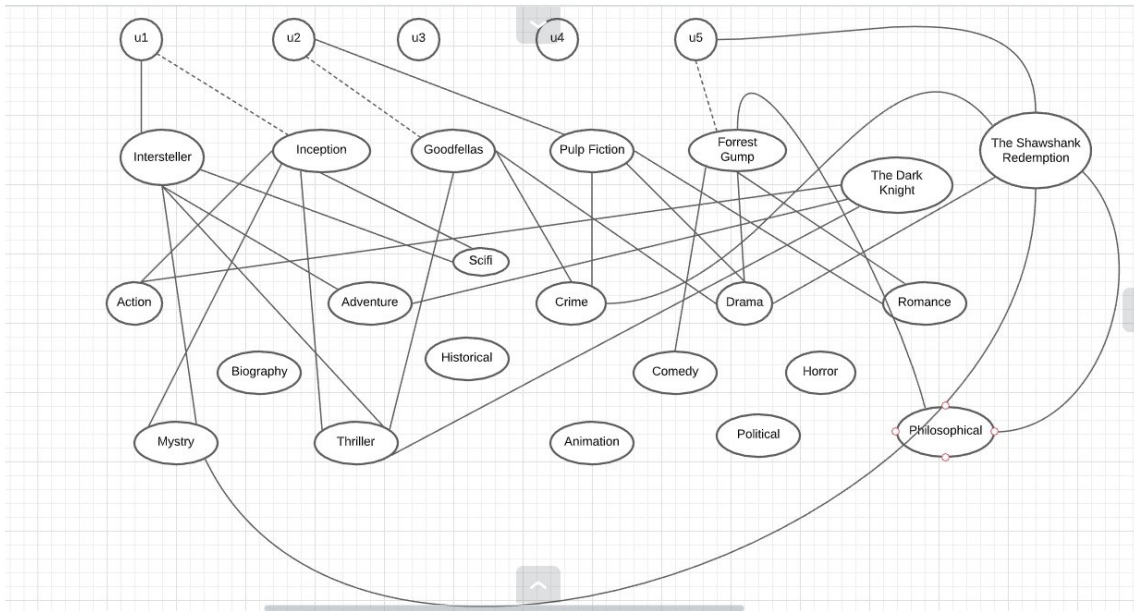


Figure 4.2: A sample movie network

For example, if we implement our algorithm for graph G in figure -2, for a user u1 who have previously watched the movie Intersteller which belongs to the scifi genre, is more likely to watch Inception as we implement our algorithm. The movie Intersteller and Inception has high value as per the Jaccard and the rating based probability score from our algorithm. Hence, the probability of having a link with the user and Interception movie is higher.

Secondly, in figure-2 we can see that each movie has an edge to all the other movie in the graph. By implementing our algorithm we will assign a weight for each of the link available in the network. This weight will determine the similarity between the movies and will help determining future possible links with the user.

4.3 Dataset Analysis

We obtained our datasets of movies from <http://networkrepository.com> . As we are using link prediction and dynamic graph to predict it is, it is very hard to get data in terms of our requirement.

This website they got the data from Amazon prime and they have built a multilevel interactive graph analytics engine that allows users to visualize the structure of the network data as well as macro-level graph data statistics as well as important micro-level network properties of the nodes and edges. Figure-3 here represents the initial condition of our dataset. In this dataset the first column refers to movie name, the second column and the fourth column are not necessary in our case and the

AH3QC2PC1VTGP	0000143561	2.0	1216252800
A3R5OBKS7OM2IR	0000143502	5.0	1358380800
A3LKP6WPMP9UKX	0000143588	5.0	1236902400
AVIY68KEPQ5ZD	0000143588	5.0	1232236800
A1CV1WROP5KTTW	0000589012	5.0	1309651200
AP57WZ2X4G0AA	0000589012	3.0	1366675200

Table 4.1: dataset

third column means the rating of the particular movie. As we do not need column two and four, we will drop these two columns from the data set in the data pre-processing stage. However, we will still need to do some pre-processing as we need a list of categories for each movie node. Hence, we will manually add a column in the dataset which contain the category list for the particular movie. The categories assigned for the movie will be done randomly from all the list of categories. We will assign at least one category for each movie randomly. After performing data cleansing and preprocessing, the newly formed dataset will be used for implementing our algorithm. However, we will keep manipulating the data in the process until we reach our goal. After performing the pre processing the data set will look like the

Movie Name	Category List	Rating
AH3QC2PC1VTGP	[Thriller, Action, Crime]1	2.0
A3R5OBKS7OM2IR	[Sci-fi, Action]	5.0
A3LKP6WPMP9UKX	[Romance, Drama]	5.0
AVIY68KEPQ5ZD	[Animation]	5.0
A1CV1WROP5KTTW	[Psychological Thriller, Crime]	5.0
AP57WZ2X4G0AA	[War, Crime]	3.0

Table 4.2: Dataset After Pre-processing

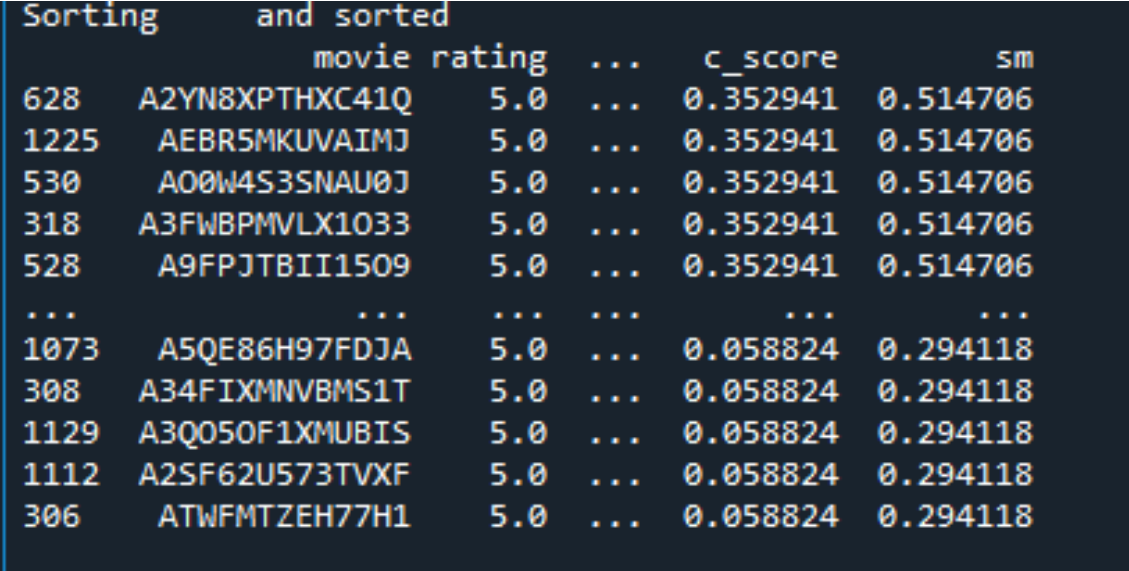
table in fig-4. However, as our dataset is too large and because of there being lack of resources, for simplicity purposes we have excluded some data which ensures the algorithm to run efficiently.

Chapter 5

Experimental Results and Analysis

5.1 Cs and Sm

After plotting data on the existing model We found our category score or cs by



```
Sorting and sorted
      movie rating ... c_score      sm
628   A2YN8XPTHXC41Q  5.0 ... 0.352941 0.514706
1225  AEBR5MKUVAIMJ  5.0 ... 0.352941 0.514706
530   A00W4S3SNAU0J  5.0 ... 0.352941 0.514706
318   A3FWBPMVLX1033  5.0 ... 0.352941 0.514706
528   A9FPJTBII1509  5.0 ... 0.352941 0.514706
...   ...           ... ...   ...   ...
1073  A5QE86H97FDJA  5.0 ... 0.058824 0.294118
308   A34FIXMNVBMS1T  5.0 ... 0.058824 0.294118
1129  A3Q050F1XMUBIS  5.0 ... 0.058824 0.294118
1112  A2SF62U573TVXF  5.0 ... 0.058824 0.294118
306   ATWFMTZEH77H1  5.0 ... 0.058824 0.294118
```

Figure 5.1: Cs and Smn

applying JACCARD COEFFICIENT on the dataset.

$$Cs = \frac{P(Cp) \cap P(Cm)}{P(Cp) \cup P(Cm)}$$

By finding the cs score we then easily found the Sm by applying formula $Sm = (0.5 * rating) + (Cs * 0.75)$. After that we sorted all the movies in descending order according to Sm.

5.2 Graph

In this graph we can see that the rating of the movies are going lower when the sm is going lower. It means when the rating of a movie is lower the sm gets lower. So the movie won't be recommended that much.

In this graph we can see that the sm is in increasing with the category score. The higher the category score gets, the higher sm will get. So if the cs is higher it will be recommended more by the model.

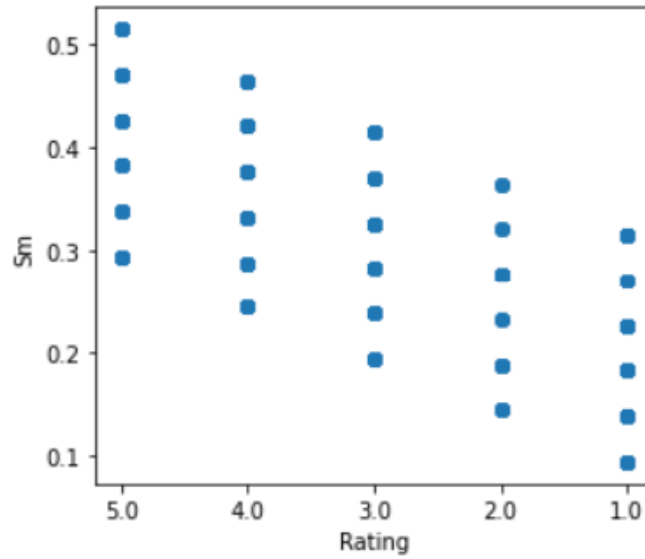


Figure 5.2: sm vs rating

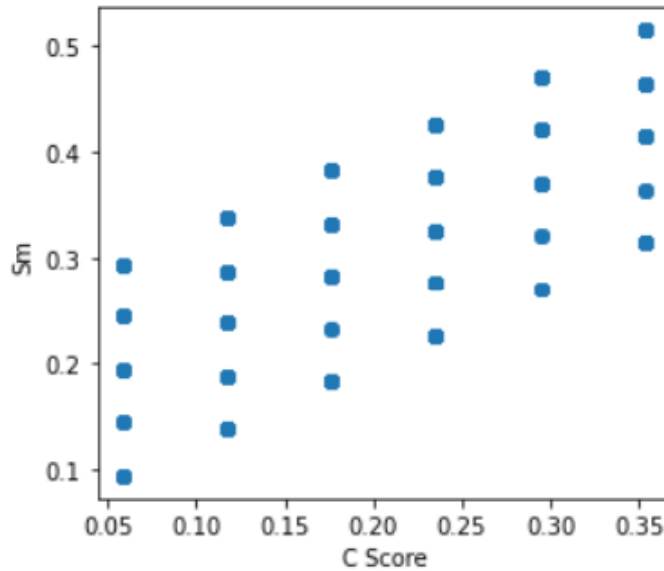


Figure 5.3: sm vs cscore

5.3 SR

In this part of our algorithm we have connected all the movie nodes among themselves with an edge between every two movie nodes. This will be a weighted graph as we will assign a weight to the edge. The weight will be derived from the formula of finding jaccard coefficient. In the figure we can see all the movie nodes are connected to each other. We have calculated the weight of all the edges. The higher the weight of the edges the more similar the movies are to each other. Because of the simplicity of representation we have not shown the weight of the edges in the graph..

We took only 10 data , but if we take 100 the connections are overlapping . And if we plot 2000 it gets overcrowded. For massive data it becomes dense.

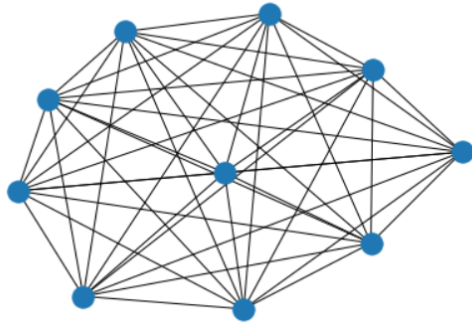


Figure 5.4: graph1

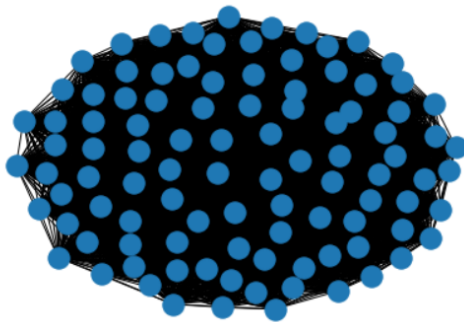


Figure 5.5: graph2

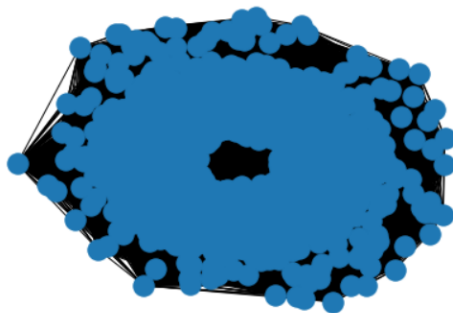


Figure 5.6: graph3

Chapter 6

Conclusions

6.1 conclusion

This dissertation is centered on the development of a movie recommendation system utilizing a graph algorithm of link prediction, which we have successfully completed using our method. In this dissertation, we include user profiles as a new dimension of an analysis that is traditionally made of three dimensions: users, ratings, and movie genres, and we create these relationships using a recommendation by link prediction method. Recommendation is only a sub-problem of link prediction algorithms. This approach is a crucial component of link prediction and is a response to numerous problems where we can frequently be confused by the enormous number of data. We implement our algorithm and got the result. At first, we got recommended movie sorting by rating for the user like which movie have high rating, then based on the category that user usually do watch; our recommending system will suggest that movies to the users. As there is no recommending system by using link prediction yet; so our methods and algorithm will help to recommend movies or network. As an example, in Netflix they also recommended movies but they don't use same method like us. So, if any OTT platform want to recommended the movies and series for their users they can easily use our method. In future, we will develop our system so that for massive network it can recommended the movies by taking in short time.

6.2 future work

From our above analysis, we got a great recommending from our model and our future goal is to extend the research and use it for a wide range of network. Furthermore, this type of research and analysis can be used for others social network to finding the recommended. Although the task of link prediction is crucial since it creates the complete network and depicts the interaction between individuals, it is also a very difficult one. As a result, we will build more linkages that will be predicted on social networks. The research on this can be useful in assessing the interaction between persons and in reflecting on the social conduct of individuals.

Bibliography

- [1] L. Liling, “Summary of recommendation system development,” in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1187, 2019, p. 052044.
- [2] N. N. Qomariyah and A. N. Fajar, “Learning pairwise preferences from movie ratings,” in *2020 International Conference on ICT for Smart Society (ICISS)*, IEEE, 2020, pp. 1–6.
- [3] S. S. Khanal, P. Prasad, A. Alsadoon, and A. Maag, “A systematic review: Machine learning based recommendation systems for e-learning,” *Education and Information Technologies*, vol. 25, no. 4, pp. 2635–2664, 2020.
- [4] J.-H. Passoth, “Music, recommender systems and the techno-politics of platforms, data, and algorithms,” in *TechnoScienceSociety*, Springer, 2020, pp. 157–174.
- [5] F. O. Isinkaye, Y. Folajimi, and B. A. Ojokoh, “Recommendation systems: Principles, methods and evaluation,” *Egyptian informatics journal*, vol. 16, no. 3, pp. 261–273, 2015.
- [6] R. Albert and A.-L. Barabási, “Statistical mechanics of complex networks,” *Reviews of modern physics*, vol. 74, no. 1, p. 47, 2002.
- [7] F. Abel, Q. Gao, G.-J. Houben, and K. Tao, “Analyzing temporal dynamics in twitter profiles for personalized recommendations in the social web,” in *Proceedings of the 3rd international web science conference*, 2011, pp. 1–8.
- [8] V. Martínez, F. Berzal, and J.-C. Cubero, “A survey of link prediction in complex networks,” *ACM computing surveys (CSUR)*, vol. 49, no. 4, pp. 1–33, 2016.
- [9] A. Barabasi, “L., albert, r,” *Emergence of scaling in random networks*, pp. 509–512, 2002.
- [10] G. Palla, A.-L. Barabási, and T. Vicsek, “Quantifying social group evolution,” *Nature*, vol. 446, no. 7136, pp. 664–667, 2007.
- [11] Y. Jiang, J. Shang, and Y. Liu, “Maximizing customer satisfaction through an online recommendation system: A novel associative classification model,” *Decision Support Systems*, vol. 48, no. 3, pp. 470–479, 2010.
- [12] K.-W. Cheung, J. T. Kwok, M. H. Law, and K.-C. Tsui, “Mining customer product ratings for personalized marketing,” *Decision Support Systems*, vol. 35, no. 2, pp. 231–243, 2003.
- [13] M. J. Pazzani, “A framework for collaborative, content-based and demographic filtering,” *Artificial intelligence review*, vol. 13, no. 5, pp. 393–408, 1999.

- [14] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, “Collaborative filtering recommender systems,” in *The adaptive web*, Springer, 2007, pp. 291–324.
- [15] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, “GroupLens: An open architecture for collaborative filtering of netnews,” in *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, 1994, pp. 175–186.
- [16] C. A. Kumar and S. Srinivas, “Concept lattice reduction using fuzzy k-means clustering,” *Expert systems with applications*, vol. 37, no. 3, pp. 2696–2704, 2010.
- [17] A. Clauset, M. E. Newman, and C. Moore, “Finding community structure in very large networks,” *Physical review E*, vol. 70, no. 6, p. 066 111, 2004.
- [18] D. Nichols, B. Oki, and D. Terry, “Using collaborative filtering to weave an information tapestry, comm,” *Assoc. Comput. Mach.*, vol. 35, pp. 51–60, 1992.
- [19] G. Linden and B. Smith, “York and j., et al (2003).“,” *Amazon. com recommendations: Item-to-item collaborative filtering*, *Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003.
- [20] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, “GroupLens: An open architecture for collaborative filtering of netnews,” in *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, 1994, pp. 175–186.
- [21] H. Li, J. Cui, B. Shen, and J. Ma, “An intelligent movie recommendation system through group-level sentiment analysis in microblogs,” *Neurocomputing*, vol. 210, pp. 164–173, 2016.
- [22] T. Hofmann, “Collaborative filtering via gaussian probabilistic latent semantic analysis,” in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 2003, pp. 259–266.
- [23] J. Niu, X. Zhao, L. Zhu, and H. Li, “Affivir: An affect-based internet video recommendation system,” *Neurocomputing*, vol. 120, pp. 422–433, 2013.
- [24] T. Saito and Y. Okada, “Bicluster-network method and its application to movie recommendation,” in *Knowledge and Systems Engineering*, Springer, 2014, pp. 147–153.
- [25] H. K. Virk, E. M. Singh, and A. Singh, “Analysis and design of hybrid online movie recommender system,” *International Journal of Innovations in Engineering and Technology (IJJET) Volume*, vol. 5, 2015.
- [26] U. Gupta and N. Patil, “Recommender system based on hierarchical clustering algorithm chameleon,” in *2015 IEEE International advance computing conference (IACC)*, IEEE, 2015, pp. 1006–1010.
- [27] S. Guha, R. Rastogi, and K. Shim, “Rock: A robust clustering algorithm for categorical attributes,” *Information systems*, vol. 25, no. 5, pp. 345–366, 2000.
- [28] S. Nivattanakul, “Access to knowledge based-on an ontology model,” Ph.D. dissertation, Ph. D. thesis. University of La Rochelle, 2008.

- [29] F. Öztemiz and A. KARCI, “Akademik yazarların yayınları arasındaki ilişkinin sosyal ağ benzerlik yöntemleri ile tespit edilmesi,” *Uludağ University Journal of The Faculty of Engineering*, vol. 25, no. 1, pp. 591–608,
- [30] M. Lu, Z. Qin, Y. Cao, Z. Liu, and M. Wang, “Scalable news recommendation using multi-dimensional similarity and jaccard–kmeans clustering,” *Journal of Systems and Software*, vol. 95, pp. 242–251, 2014.
- [31] C. Ekanadham and H. Lee, “Sparse deep belief net models for visual area v2,” *Advances in Neural Information Processing Systems*, vol. 20, 2007.
- [32] M. Ontivero-Ortega, A. Lage-Castellanos, G. Valente, R. Goebel, and M. Valdes-Sosa, “Fast gaussian naïve bayes for searchlight classification analysis,” *Neuroimage*, vol. 163, pp. 471–479, 2017.
- [33] H. Kamel, D. Abdulah, and J. M. Al-Tuwaijari, “Cancer classification using gaussian naive bayes algorithm,” in *2019 International Engineering Conference (IEC)*, IEEE, 2019, pp. 165–170.
- [34] —, “Cancer classification using gaussian naive bayes algorithm,” in *2019 International Engineering Conference (IEC)*, IEEE, 2019, pp. 165–170.
- [35] E. L. Newman and E. Witsell, *The Food Network Recipe: Essays on Cooking, Celebrity and Competition*. McFarland, 2021.
- [36] M. Scholz, “Node similarity as a basic principle behind connectivity in complex networks,” *arXiv preprint arXiv:1010.0803*, 2010.