U.PORTO

FEP FACULDADE DE ECONOMIA UNIVERSIDADE DO PORTO

SPROUT – a Supervised recommender system for link PRedictiOn in bipartite mUltilayer neTworks

Victor Fernandes Malheiro - 201406480

Dissertation Plan

Master in Modeling, Data Analysis and Decision Support Systems

Supervised by

Prof. Dr. Pedro José Ramos Moreira de Campos

Prof. Dr. Hélder Fernando Cerqueira Alves

2023

Abstract

To achieve the goal of predicting links within bipartite networks with multiple layers, each with a varying number of nodes, we introduce a novel link prediction algorithm for multilayer networks named SPROUT (Supervised link PRedictiOn in mUltilayer bipartite neTworks). This method is designed to address the link prediction problem by utilizing similarity-based measures to determine whether a link should be established between a specific pair of nodes within a particular layer of the network. Our new methodology not only relies on information provided by the layer under consideration for link prediction but also incorporates information from other layers. This integration is achieved through a synthesizer that leverages both intralayer and interlayer information to generate meaningful insights to decide which links should be formed within a given layer of the network.

The interlayer information required for the synthesizer is derived from similarity measures between pair of nodes in different layers. It combines an intralayer similarity measure, which assesses the significance of a pair of nodes within its own layer, with an interlayer similarity measure, which evaluates whether a pair of nodes importance and relevance remain consistent across multiple layers. This computation is performed for all possible pairs of layers, including the layer where the model is being applied and all other layers in the network. The results are then aggregated and normalized to determine the interlayer probability of link formation for all node pairs within a given layer.

To optimize the balance between intralayer and interlayer information, we introduce a control parameter that allows for the weighting of the intralayer measure relative to the interlayer probability. This ensures that the synthesizer utilizes the most pertinent information within the network effectively. Through exploratory data analysis, it becomes possible to identify the most crucial layers in the network that provide the most valuable information for the model implementation.

We put this proposed methodology to the test by applying it to a multilayer bipartite network created from the MovieLens Dataset, which contains movie ratings from various users. This dataset serves as a valuable resource for building a recommender system for each user within the network. SPROUT, as outlined in our approach, can easily be adapted for use with other types of datasets, providing a versatile framework for developing recommender systems.

Keywords: Link Prediction, Multilayer Bipartite Networks, Recommender Systems, Similarity Measures

Resumo

Para atingir o objetivo de prever links dentro de redes bipartidas com múltiplas layers, cada uma com um número variável de nós, introduzimos um novo algoritmo de previsão de links para redes multilayer denominado SPROUT (Supervised link PRedictiOn in mUltilayer bipartite neTworks). Este método foi projetado para abordar a previsão de links, utilizando medidas baseadas em similaridade para determinar se um link deve ser estabelecido entre um par específico de nós, dentro de uma layer específica da rede. Esta nova metodologia não depende apenas da informação fornecida pela layer em consideração para previsão dos links, mas também incorpora informação de outras layers. Essa integração é alcançada através de um sintetizador que aproveita a informação intralayer e interlayer para gerar resultados significativos para decidir quais são os links que devem ser formados dentro de uma determinada layer da rede.

A informação interlayer necessária para o sintetizador é derivada de medidas de similaridade entre pares de nós em diferentes layers. Ele combina uma medida de similaridade intralayer, que avalia a importância de um par de nós dentro da sua própria layer, com uma medida de similaridade interlayer, que avalia se a importância e a relevância de um par de nós permanecem consistentes em múltiplas layer. Este cálculo é realizado para todos os pares possíveis de layers, incluindo a layer onde o modelo está a ser aplicado e todas as outras layers da rede onde o par de nós está presente. Os resultados são então agregados e normalizados para determinar a probabilidade interlayer de formação de conexões para todos os pares de nós.

Para otimizar o equilíbrio entre a informação intralayer e interlayer, introduzimos um parâmetro de controlo que permite a ponderação da medida intralayer em relação à probabilidade interlayer. Isso garante que o sintetizador utiliza efetivamente a informação mais pertinente dentro da rede. Através da análise exploratória dos dados, torna-se possível identificar as layers mais cruciais da rede que fornecem as informações mais valiosas para a implementação do modelo.

Colocamos esta metodologia proposta à prova, aplicando-a a uma rede bipartida multilayer criada a partir de um dataset da MovieLens, que contém classificações de filmes de vários utilizadores. Este conjunto de dados serve como um recurso valioso para a construção de um sistema de recomendação para cada utilizador da rede. O SPROUT, conforme descrito na nossa proposta, pode ser facilmente adaptado para uso com outros

tipos de conjuntos de dados, fornecendo uma estrutura versátil para o desenvolvimento de sistemas de recomendação.

Palavras-chave: Previsão de conexões, Redes bipartidas multilayer, Sistemas de recomendação, Medidas de similaridade.

Contents

List of tables

List of figures

1. Introduction

With technological advancements, the abundance of digital information has presented a challenge in the form of data overload that both companies and consumers must deal with. Data overload occurs when individuals and organizations are confronted with an amount of information that exceeds their capacity to process effectively. Information is generated and disseminated at a pace far swifter than our ability to digest it. Without mechanisms for creating filters tailored to their specific needs, individuals and businesses risk losing control over accessing the information that is most relevant to them (Edmunds & Morris, 2000).

This information overload challenge demanded the development of Recommender Systems (RS), that help users find the information they need and when they need it. With the personalized recommendations created, a recommender system guides the users, using different techniques, with several sources of data, to achieve useful results. Using a group of items or users with known preferences, over which the recommendations are made, the recommendation algorithm can predict the user's preferences for a specific item (Burke, 2002).

It is of great help to have an advisor that helps us to choose the best option when we make a decision, instead of wasting many hours searching the internet analyzing reviews and suggestions. Recommender systems can help companies attract users that will remain on their site for a longer time, resulting in higher revenues and purchases. Data Mining (DM) and Artificial Intelligence (AI) advances have facilitated innovations and boosted the customer experience and that is why businesses across the globe have invested in custom recommender systems. Because producing more accurate predictions with an affordable computational power is one of the biggest challenges, a big effort has been done by the researchers to develop new algorithms and methodologies (Singh, Pramanik, Dey & Choudhury, 2021).

Recommender systems can be naturally represented by a bipartite network that has two types of nodes and one type of edge between those two nodes, with one node being the user and the other the item and the edge being the relation between them. Therefore, the recommender systems can be modelled as a Link Prediction (LP) problem with the task of predicting future links in a network. This network could be of various types, e.g. an e-commerce site relating the buyers with the items they buy, could be a video streaming platform, linking the viewers to the movies or TV shows they watch. Several other cases can be modelled as a link prediction problem on a graph-based RS. This LP problem is commonly a classification task that gives the top k-items recommended for a user. With the massification

of social networks and online shopping, LP gained a special importance with the possibility of representing those networks not only with different types of nodes, but also in different layers, giving the opportunity to realistically represent the different interactions the users have on those networks (Lakshmi & Bhavani, 2021).

Bipartite networks are composed by two different types of nodes (Barber, Faria, Streit & Strogan, 2008) and recommender systems are natural examples of bipartite networks that contain two types of nodes and a single type of edges linking those nodes. The main task of a recommender system is to predict the unrated entries in a rating matrix.

Real-world systems involve multiple types of relations among their components. One of the most important challenges within the complex systems we see in the real world is to inform which entities are related to which other and what are the types of the relationships between them. When several relationships exist within a network, we can define different layers in a multilayer network, where nodes exist in separate layers, representing different forms of interactions. We can have a bipartite network on a network with different layers, for example, in e-commerce the buying and selling of goods over the internet, there are multiple types of interactions (e.g., click, rating and buy) and nodes (e.g. costumers and items) (Najari, Salehi, Ranjbar & Jalili, 2019).

Multilayer link prediction is the problem of finding missing links between nodes based on information of a certain layer and also from other layers. Structured data can be represented by multilayered networks, in which each layer represents a different type of interaction, for instance, trips that can be made by different means of transportation, or social interactions that can happen through various communication channels, e.g. face-to-face, e-mail or social media (Li & Wang, 2022).

In this work we propose a methodology called SPROUT – a Supervised link PRediction On mUltlayer bipartite neTworks. SPROUT is a model developed to predict links between users and items on a bipartite network with multiple layers, using the information of both intralayer and interlayer connections based on similarity measures, even if some of the nodes are not matching between the layers. Results of the implementation of the model will produce meaningful information that can be used in a recommender system.

SPROUT creates the interlayer probability of link existence accounting the information of the other layers of the network where a certain pair of nodes exist. For that, a matrix with the differences of all the matching nodes Betweenness similarity measure between two layers is computed, considering that the lowest that difference is, the bigger its

predicting power. The intralayer similarity measure for the other layer, the one where the links are not being predicted, is also computed, to use the internal information of a certain pair of nodes in the other layers together with the interlayer similarity measure. SPROUT is able to work with layers that can have different sizes, with different number of nodes, using the information of the whole layer. After the interlayer probabilities are computed and normalized, the intralayer probability of the layer being studied is computed using the Jaccard similarity measure, a synthesizer is used to combine both intralayer and interlayer probabilities using a control parameter a, that can have a value between zero and one, with 0.1 intervals, to use the value of the total probability that maximizes the area under the curve computed from the receiver operating characteristic curve, that is used to evaluate the results of the model.

To evaluate the performance of the model, the receiver operating characteristic curve (ROC curve) is created comparing the adjacency matrix created from the graph with the original links, and the one with the adjacency matrix created after 30% of the links were removed from the original graph's adjacency matrix and the new links predicted. A minimum threshold β for the total probability is also used, to evaluate which links should be considered to be created, this parameter also uses values between zero and one, in 0.1 intervals. It is used the combination of a and β that maximizes the area under the curve (AUC) of the model. Another two evaluation measures are also used to evaluate the output of the model, those are the accuracy and the F1 score. The good results of these three evaluation measures lets us conclude that SPROUT has a good predicting power and can be adapted with good results for different datasets.

SPROUT consistently achieved good performance, exhibiting high accuracy, F1 scores, and robust AUC values, both in smaller and larger networks. This adaptability enables its utilization across various bipartite multilayer networks. The ability to fine-tune a and β is crucial, as it empowers us to harness more relevant information within the network, whether it pertains to intralayer or interlayer dynamics.

The application of link prediction tasks on bipartite multilayer networks has not been done extensively by researchers, because most of the similarity measures were created for one-mode networks and the models are developed for multiplex networks, that use the same nodes on all the layers. In this way, this work aims to contribute for the exploration of link prediction on bipartite multilayer networks, generalizing the work outside of multiplex networks.

This dissertation is organized in six chapters. In the first chapter we present the introduction of the topic and the statement of the problem, as well the motivation for it.

In Chapter 2 we present the literature review, where we begin with the introduction of basic concepts about e-commerce, followed by a detailed literature review of recommender systems and graph theory, with a focus on bipartite networks. We also have a section about multilayer networks being followed by a section of a literature review about link prediction, with a focus on the different link prediction approaches. Finally, we present the literature review about the performance metrics used to evaluate the link prediction models.

Chapter 3 is focused on the methodology and data. Research questions are introduced, and we discuss different methodologies that also explored link prediction tasks, followed by the detailed presentation of the methodology that will be implemented. We start by illustrating how the model works with a small example and after that, the MovieLens dataset is presented, describing all the features and data used to implement the model. Finally, the model is briefly presented for a case of a node that is introduced on a layer and how the interlayer information can be used to create links for that node.

In Chapter 4 the methodology is applied to the MovieLens dataset and SPROUT is used to compute the probabilities for the link prediction task.

In Chapter 5 we show how the final output of the model can be used to create recommendations for the users, providing meaningful information for a company and the respective costumers.

Finally, in Chapter 6 conclusions and the future work are presented with a discussion about the strengths, limitations and challenges.

2. Literature review

2.1 E-commerce

Electronic commerce, or e-commerce, is defined as a commercial transaction that implies the transfer of information through Internet (Hendricks & Mwapwele, 2023). It plays an important role in the communication between different economies being the responsible for essential improvements in the way people communicate through the internet, improving the logistics of companies and the deliveries from suppliers and also to costumers, reducing production and delivery times.

The boom of e-commerce happened in the 1990s with ebay[1] and amazon[2] and that opened the doors to many organizations to trade at a global scale. E-commerce allowed organizations, regardless of their size or location, to participate in the global trade. The internet's worldwide availability enabled businesses to reach customers everywhere, breaking down geographical barriers. It also gave companies the advantage of reducing costs, increase their revenues and profits, facilitating marketing campaigns that can reach a wider consumer range and also making the communication between businesses and consumers quicker and easier (Khoo, Ahmi & Saad, 2018).

Several business models exist, including businesses to consumers (B2C), where online retailers sell products and services to consumers through the internet. There is also business to business (B2B), where electronic commerce is used by companies to purchase what is needed from their suppliers or to sell to costumers that are other businesses. E-commerce is essential for the daily operations of all companies that need to trade in a globalized world. Other types of e-commerce are customer to customer (C2C), where consumers trade directly with each other, being the most famous example ebay. There are also cases of consumer to business, e.g. e-commerce websites that reward consumers for reviewing their products (Pandey & Agarwal, 2014).

E-commerce has several benefits to customers, being very convenient because it allows them to buy 24/7 and saves them time because customers do not need to travel to a store to create the order. Another benefit, is the information that is accessible to everyone that facilitates comparison of prices and easily shows the specifications of the items, helping customers to more effectively use the large amount of information, using search tools or

[1] https://e-bay.com/

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

recommender systems that personalize the purchases giving the possibility to customers to buy what they need without spending too much time searching for the right product with the desired price (Taher, 2021).

Companies benefit from e-commerce with the possibility of trading almost without geographical limitations, needing only to worry with the delivery of what is ordered, providing huge cost savings, and increasing the efficiency of their operations because they can create one online shop for the entire world. Companies are also able to target whom they want in a very large group of potential customers and with this, they can get a higher return on their investments (Parikshith & Natesan, 2023).

With these different approaches, companies that use e-commerce could also have to deal with some disadvantages and those disadvantages can be technical or non-technical. The inability of not physically test the items before creating the order is perhaps the biggest disadvantage because as much professional the online presentation of an item is, it could lead to misunderstandings. The damaging of the packages during transportation and the delay on the deliveries are a common disadvantage that takes away the advantage of the time savings on the order creation and with these issues, if the customer services are not competent, it could lead to discontent on the consumer side. Another disadvantage it is the inability that some people still have to connect to the internet, discriminating them on the access to e-commerce (Taher, 2021).

One of the oldest concerns is the possibility of the pour security on the platform where the order is being created, with the possible unauthorized access to personal data, but also to payment frauds that could have a big impact on both customers and companies. Companies must implement and update their security measures without harming the business, but always providing their customers the trust they need to purchase with them (Saeed, 2023).

According to Fuller, Harding, Luna & Summers (2022), the timing of adoption of e-commerce by an organization will have an important impact on their online performance, with some organizations benefitting more if they adopt these capabilities earlier and others having more benefits if they do it latter.

Although the resources are usually limited, companies must create a plan to adopt e-commerce, giving companies different capabilities that are associated with an increased business value. The first one is information, it is essential to have a good and clear communication with customers to have a good interaction. Another is the transaction created, there

should always exist a trusted process to proceed with the transaction, providing clarity, transparency, and security to everyone involved. The last one is customization, with the 4th industrial revolution it is a very important feature for certain companies that need to offer customized products or services to customers and with that, gain competitive advantages with the possible retaining of those customers. The diffusion of the company's innovation on the e-commerce platform and allowing customers that want to purchase a certain item to be able to customize it, is a very important capability that companies must develop (Kabrilyantsa, Obeidata, Alshuridehc & Masa'deh, 2021).

All companies must develop various e-commerce facilitators to be capable of competing with the best competitors. Internet is the most important one because other way, customers will not be able to access the online shop. A secure and easy way to use payment gateways is another important facilitator, the payment transactions are crucial to make the deal, so companies must provide a secure way to make the payment transactions. Social media is having an increasing relevance for e-commerce because it allows e-commerce companies to effectively place their marketing campaigns, targeting the desired group of people. Analytics also play a decisive role, with companies being able to transform data into decision making information because businesses must research the behaviors of clients to target them more effectively with the products they desire and the ones the businesses want to sell to them (Jain, Malviya & Arya, 2021).

According to Rita & Ramos (2022), consciousness regarding sustainability in consuming, is a key topic guiding the future of e-commerce. This can be done looking at how companies are looking at packaging materials and its waste, to cyber security procedures, social topics and specially the defense of the human rights on the sites of goods production, and because of the global trade, on how to do it all cross borders. The relationship between consumer behavior and e-commerce is of the most importance to mold the way companies respect the sustainability topics either it is socially, environmentally, or economically and it is essential on the development of e-commerce from a one channel online store to a multichannel e-commerce environment.

2.2 Recommender systems

To help customers to more easily find the content that is more relevant for them, companies with online presence invest and develop recommender systems, that allow them to suggest to customers, or potential customers, attractive products, or services, saving time

and increasing sales (Deepjyoti & Mala, 2022).

Because RS apply information filtering techniques, they are a lot more efficient than search engines, informing customers of content that they are not aware of when they only use a simple search. RS usually use two types of computational techniques: heuristics algorithms techniques, where algorithms are designed based on measures computed, for example, to find the most bought item; and model based techniques, that build models based on existing data instances, e.g. when graph related features are used in the learning of the model, using measures that are defined to extract information on connected nodes in a bipartite graph (Isinkaye, Folajimi & Ojokoh, 2015).

Two different types of features are generally used on the construction of RS. On one hand we have local features, that capture the collective characteristics of a certain user or item and those characteristics can be the item details, the context of the transaction, the temporal information about the usage of the online platform and the content that more interested the customers. With this information, the recommender systems can show directly the differences between customers and the products bought, and this allows the RS designers to use similarity measures to cross-recommend similar items to similar users. On the other hand, there are the graph related features, consisting of interactions between users not directly related or items, that can be identified and modeled in a graph structure. The designed bipartite graph with the user-item relations can also be projected to a unipartite (or one mode graph), that simplifies the graph structure and the analysis of it (Li & Chen, 2011).

The biggest strength of RS is not only to help companies to decide which products or services to present to customers, but they also increase the cross-selling by offering additional content to customers allowing companies to sell something that is not searched primarily by the customers (Daher, Brun & Boyer, 2017).

Despite the advantages, there are also some challenges that RS designers need to overcome:

- Data sparsity – Because the number of available items or services and costumers are sometimes very large, often with millions of items, the relationship between two users is small and even when a user or item has a lot of evaluations, that distribution is very unbalanced because most of the items or users only received a small amount of evaluations (Choi, Lee D., Jang, Park & Lee S., 2023).

- Scalability – The data related to users and items can encompass millions of entries, resulting in rising computational costs. Therefore, as the data expands, designers must develop strategies for running algorithms that enable the company to harness all relevant data for optimal outcomes (Xin, 2015).
- The cold start problem - is when there is not enough information about a new user or item in the system, is one of the most critical challenges when designing a RS. The solution is usually based on hybrid techniques (Han, Castells, Gupta, Xu & Salaka, 2022).
- Diversity vs accuracy – recommending a popular item to a user does not have a lot of value to the company because that item can be easily found by the user without recommendations, so the list of items to be recommended should also have less obvious items that in other way, the user did not got knowledge of it (Lü, Medo, Yeung, Zhang, Zhang & Zhou, 2012).
- Vulnerability to attacks – recommender systems can be the target of attacks to promote or hide content to the users and to exploit cybersecurity issues (Ferreira, Silva & Itzazelaia, 2023).
- The value of time – The interests of users can fluctuate over time due to factors as the seasonal preferences. Users may exhibit intense interest in a particular product or topic for a brief period of time, but they may have more long-term interests. Recommender systems must account for this temporal variability, accommodating both short-lived and sustained user interests effectively (Lü, Medo, Yeung, Zhang, Zhang & Zhou, 2012).

According to Farashah, Etebarian, Azmi and Dastjerdi (2021) researchers have at their disposal a multitude of practical examples since recommender systems play a pivotal role in helping customers and companies navigate the challenges posed by information overload. RS can effectively guide both customers and companies toward their respective objectives. For customers, the primary goal is to locate items or services accurately and promptly. For companies, the aim is to sustain the customer engagement and encourage continuous purchasing. Addressing these objectives is crucial due to the significant challenge of customer retention and the need of interesting content to maintain those customers engaged.

RS are classified into three categories:

- Content-based

- Collaborative
- Hybrid

2.2.1 Content-based recommender systems

On this category of RS, description of items and customer profiles with their preferences are used to make the recommendations because it is based on the similarity between the item descriptions and the customer profile. Although it can allow recommendations to be created for new customers, avoiding the cold start problem, it also has a limitation because it needs good item descriptions and detailed customer profiles. The profile can be constructed explicitly with the information of the users being collected through questionnaires, to gather detailed information about the customer preferences. It can also be built implicitly, searching for similarities in items rated by the customer. And it can also be model based, modeling the customer profile based on a learning method that uses items descriptions as input on a supervised algorithm and the ratings as output. Each method has its strengths or trade-offs, and the choice depends on the availability of data and the specific goals of the recommender system. (Lakshmi & Bhavani, 2021).

On the content-based RS, users will be recommended similar items that they rated positively in the past, e.g. when on a streaming platform, as Netflix, a user will be recommended movies similar to the ones that he/she preferred in the past. Being this also its limitation because it will only recommend something similar to the content already rated and do not introduce the user to new type of content that could be of his/her preference (Adomavicius & Tuzhilin, 2005).

2.2.2 Collaborative filtering recommender systems

Collaborative filtering RS, according to Farashah et al. (2021), recommends to customers, using data mining techniques, content that was rated by other customers that are considered similar to them, turning this type of RS one of the most used, using the advantages of the networks created by the customer-item interactions.

One big advantage of collaborative filtering systems, compared to content-based ones, is their independence from product content descriptions. They do not rely on detailed item descriptions, making them versatile for various types of products and content. The limitation of these RS is the cold start problem because it arises when there is no historical data to generate predictions for new users and also the data sparsity can be a big problem

due to insufficient ratings, potentially resulting in suboptimal recommendations. To address the data sparsity problem, techniques such as Principal Components Analysis (PCA) and Singular Value Decomposition (SVD) can be employed. These methods help in reducing the dimensionality of the data and enhance the quality of the recommendations by uncovering latent patterns and relationships within the user-item interaction data (Lakshmi & Bhavani, 2021).

According to (Adomavicius & Tuzhilin, 2005), they can be divided into two classes:

- Memory-based: make the predictions based on all the previously rated items by the users.
- Model-based: makes the predictions based on a model used to learn the data using ML techniques.

2.2.3 Hybrid recommender systems

Most recommender systems use a hybrid approach by combining collaborative and content-based methods, which helps to avoid certain limitations of these two systems. Different ways to combine collaborative and content-based methods into a hybrid recommender system exist. One of them is implementing collaborative and content-based methods separately and combining their predictions, other is to incorporate some content-based characteristics into a collaborative approach, or incorporating some collaborative characteristics into a content-based approach and constructing a model that incorporates both content-based and collaborative filtering characteristics (Adomavicius & Tuzhilin, 2005).

However, according to Burke (2002), most commonly, collaborative filtering is combined with some other technique to avoid the ramp-up problem, that states that until there is a sufficiently large number of customers, to use for the recommendations, the recommender system cannot make meaningful predictions to the customers.

2.3 Graph Theory

We inhabit a world where a vast array of things can be effectively represented using graph structures. This is because many real-world relationships are inherently interconnected. Actually, graphs or networks offer a powerful framework for capturing and modeling complex relations among different nodes within a network. In this context, recommender systems benefit significantly from graph structures. RS inherently involve objects, such as customers and items, which can be represented within a graph, reflecting the direct connections

and relationships between them. This graph-based representation allows RS to harness the full potential of these intricate connections, enabling more accurate and personalized recommendations (Wang, Hu, Wang, He, Sheng, Orgun, Cao, Ricci & Yu, 2021).

Graphs serve as a universal language for representing complex systems. At their core, a graph consists of a collection of objects interconnected between them. For instance, in social networks, individuals are depicted as nodes, while edges symbolizing their connections. However, graphs extend beyond mere visual representations, they offer a robust mathematical foundation that facilitates the comprehension, analysis, and learning from real-world systems. (Majeed & Rauf, 2020).

An important challenge is that with mass information created every day, we need to unlock the potential that all this data offers us, so Machine Learning (ML) can play a crucial role in allowing us to model and analyze this complex graph data, with an increasing scale that we need to understand. Different types of graphs can be used to better model real world situations (Bessy, 2013).

Graphs are structures made up of a set of nodes and links. Network analysis plays a crucial role in the realm of recommender systems because it provides a framework to represent various systems as networks, where nodes represent customers or organizations, and edges capture their interactions. The study of these networks, often referred to as Graph Theory in mathematical literature, is essential for comprehending and analyzing the interactions that serve as the base of the design of effective recommender systems (Li & Chen, 2011).

Graphs can be multi-relational, meaning that they represent networks with multiple types of nodes or links between those nodes. The multi-relational graphs can also be multiplex graphs, that are graphs with multiple layers where all the nodes are replicated across the layers and each layer represents a particular aspect of the connection between the nodes. Graphs that do not have all the nodes in all the layers are multilayer networks and this type of network is particularly useful to represent RS (Li, Ng, Xu & Yip, 2023). In a movie recommender system, for example, the different genres of the movies can be represented by different layers.

To facilitate accurate recommendations, it is crucial to employ the most effective measures. We have at our disposal both node-level and graph-level statistics, each playing a vital role in enhancing the accuracy of the recommendations.

Node-level statistics provide valuable insights into individual nodes within the network. These measures include, among others:

Node Centrality- Centrality measures the importance of a node within the network. Metrics such as degree centrality, betweenness centrality, and closeness centrality quantify how many neighbors a node has and its relative position within the network. High centrality nodes can play critical roles in the flow of information.

Clustering Coefficients- Clustering coefficients evaluate the degree to which nodes tend to cluster together. They help identify nodes that are part of clusters within the network because nodes with high clustering coefficients may have strong local influence.

In addition to node-level statistics, we also have access to graph-level features, which allow us to compute measures on a global scale. One approach is the use of Graph Neural Networks (GNNs). GNNs are powerful tools for learning representations of nodes and graphs, making them well-suited for recommender systems in complex networks. By leveraging these node-level statistics, as well as the capabilities of GNNs for graph-level analysis, we can create a framework that makes accurate recommendations. This approach considers both local and global network characteristics, enabling us to harness the full potential of the underlying data to enhance the recommendations accuracy (Hamilton, 2020).

Graph theory has various types of walks, each representing a different way of traversing from one node to another within a graph. These walks are fundamental concepts in graph theory and play a significant role in designing recommender methodologies that utilize graph structures. As defined by Wilson (1996), a graph serves as a representation of a set of nodes and the connections (or edges) between them. This representation captures the relationships and interactions between different entities, making it a valuable tool for modeling various real-world scenarios, including recommender systems. Using these various types of walks and the inherent structure of a graph, recommender systems can extract valuable insights, identify relevant items or customers, and generate recommendations that are more accurate. Wilson's definition of a graph as a representation of points and their connections underscores the foundation upon which recommender methodologies can be built within the framework of graph theory.

Below we can see some of the mathematical structures used to represent graphs.

Let N be a set of nodes and E a set of edges from $N \times N$, the pair G = (N, E) is called a graph, as can be observed on the right-side graph of figure 2.2. This network is what is called an undirected graph because its edges do not have a defined direction. When the

edges have a direction, it is called a directed graph. More formally, in a undirected graph, two nodes v and w represent an unordered pair {v,w}. On the other hand, in a directed graph, the pair is ordered as (v,w) or (w,v), as depicted in figure 2.1.

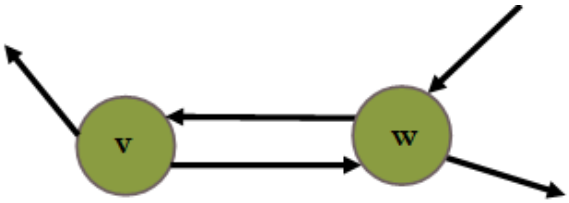

Figure 2.1 – Representation of two nodes from a directed graph.

Given a set of two different types of nodes, as represented on equation 2.1, where no edges exist between the nodes of the same type, we say that G is a bipartite network.

$$N = N1 + N2 \tag{2.1}$$

When the connection between each pair of nodes represents only the existence or non-existence of a connection, the graphs are called unweighted or binary graphs. In turn, when the connection between each pair of nodes is represented by weighted values that represent the strength or intensity of the relationship, these graphs are called weighted graphs.

These graph structures can be represented by a matrix called adjacency matrix, which represents all the nodes on the network and the connections between them. An unweighted network can be represented by its adjacency matrix, such that:

$$A = (A_{i,j})_{1 \le i,j \le n} \tag{2.2}$$

With

$$A_{i,j} = \begin{cases} 1 \; if \; \{i,j\} \in E \\ 0 \; otherwise \end{cases} \tag{2.3}$$

Weighted networks can be represented in terms of the weighted adjacency matrix, such that:

$$W = (w_{i,j})_{1 \leq i,j \leq n}$$

(2.4)

With

$$W_{i,j} = \begin{cases} w_{i,j} \; if \; \{i,j\} \in E \\ 0 \; otherwise \end{cases}$$

(2.5)

Where $\{i, j\}$ represent a pair of nodes in the network (Barber et al, 2008).

Networks can have different types and that significantly influences the analysis approach. Static networks maintain a constant structure over time, with nodes and edges remaining unchanged. In contrast, dynamic networks are characterized by continuous changes over time. In dynamic networks, both nodes and edges can be added or removed, making them a dynamic and evolving entity (Samad, Qadir, Nawaz, Islam & Aleem, 2020).

Machine learning (ML) models are a valuable tool for solving specific challenges within graphs. Node classification tasks involve training models to categorize nodes. Relation prediction, or link prediction, is crucial for forecasting future connections between nodes, a fundamental aspect of building recommender systems using graph data. Additionally, ML helps in clustering and community detection, which identifies subgraphs composed of nodes with similar features. These applications demonstrate the broad utility of ML in solving complex problems within graph structures. (Kannaiyan, Pappula & Veerubommu, 2020).

2.3.1 Bipartite networks

Many of the systems studied to design recommender systems can be represented as a bipartite network, which is a network with two different types of nodes linked by edges that represent the interactions between them, on recommender systems, it is usually called a user-item network where the edges connect two different types of nodes (Gupta & Pravin, 2023).

According to Xue, Yang, Rajan, Jiang, Wei & Lin (2018), a bipartite network is a graph structure with two different types of nodes and with edges that only exist between those two types of nodes, examples of bipartite graphs include author-paper, customer-item purchases, user-song playlists, and user-movie connections. A formal representation can be observed on equation 2.1.

Bipartite networks with the two distinct types of nodes offer a precise representation of various interaction patterns (see Figure 2.2). They effectively capture the details of different interactions on the networks. From bipartite networks, researchers often derive unipartite networks as projections. This transformation enables the application of specialized measures and statistics designed for unipartite networks, enhancing the analysis of these complex structures (Barber et al, 2008). Moreover, bipartite networks are a vital and efficient form of representation for the analysis and modeling of complex networks, as they can unveil patterns unrepresentable on more simple networks (Tarissan, 2015).

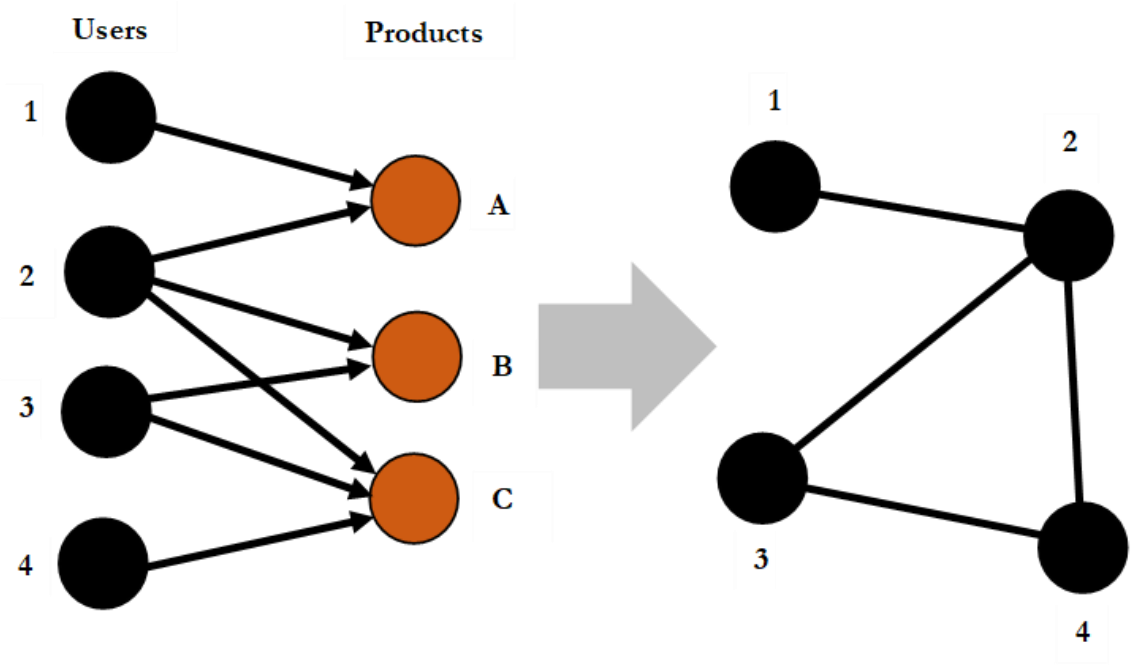

Figure 2.2 - Depiction of a bipartite network featuring two distinct types of nodes is shown on the left, alongside a projection of the users in a one-mode network displayed on the right.

2.4 Multilayer networks

To accurately represent the complexity of interactions in the real world, networks can be separated into different layers, those are separated networks which have nodes that belong to different layers (see Figure 2.3), for example, different genres of movies. (Hamilton, 2020).

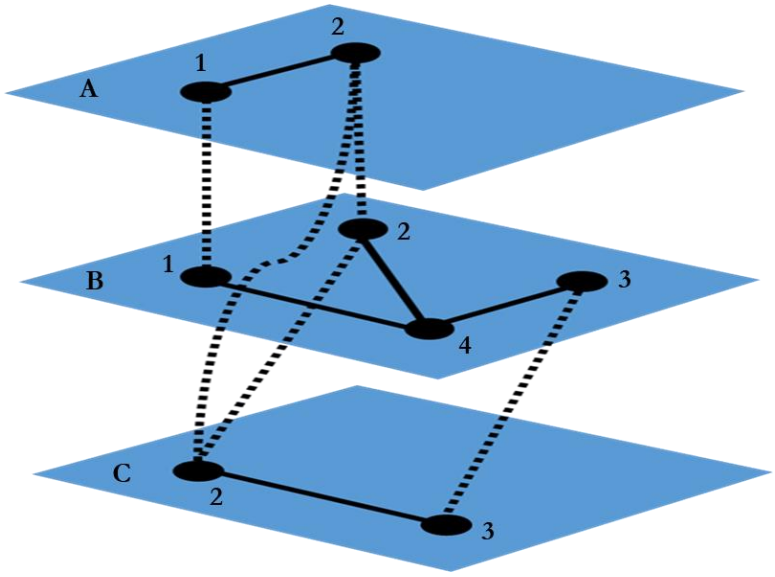

Figure 2.3 - General example of a multilayer network with four nodes divided between three layers. Based on source: (Kivelä, Arenas, Barthelemy, Gleeson, Moreno & Porter, 2014).

Complex network research initially focused on single-layer networks, but recent years have witnessed growing interest in multilayer networks. Within this domain, exists a specific type known as multiplex networks. In a multiplex network, the same set of nodes is represented across multiple layers, with each layer signifying a distinct type of relationship between these nodes. On the other hand, multilayer networks represent a broader concept encompassing networks with multiple layers, but they do not necessarily have a shared set of nodes on all the layers. In multilayer networks, different layers may involve different sets of nodes and these layers can represent diverse aspects of the network. While all multiplex networks are inherently multilayer networks due to their multiple layers, not all multilayer networks qualify as multiplex networks, as some may have non-overlapping node sets. This distinction clarifies the relationship between these two network structures (Jafari, Abdolhosseini-Qomi, Asadpour, Rahgozar & Yazdani, 2021).

With the increasingly higher efforts to analyze networks that have several types of connections, that can be called networks of networks, we can apply constraints that will help us to get the information we need more accurately. Multilayer networks can help to discover patterns that cannot be represented by single-layer networks (also known as one-mode networks). To harness the full potential of multilayer networks, it is essential to consider not only the connections within each layer (intralayer connections) but also the interactions between layers (interlayer connections). For instance, when analyzing the presence of a certain

company on social media, we can use the different social media platforms as the feature used to divide the network into multiple layers (Kinsley, Rossi, Silk & VanderWaal, 2020).

The first way to gather information for multilayer networks is to generalize the graph measures used for single-layer networks. Those measures are based on degrees, neighborhoods, walks, clustering coefficients, centrality measures and network models. It is also possible to project the multilayer network to a single-layer network and with it, we can use standard network techniques, but that aggregation can discard precious information that only networks with layers can have.

Several measures for multilayer network were created (Kivelä, Arenas, Barthelemy, Gleeson, Moreno & Porter, 2014):

- Node degree and neighborhood are the number of edges that are linked to a certain node.
- Walks, paths and distances, walks and path length are important measures because they allow the generalization for multilayer networks to know the network distance, components that are connected, betweenness centralities, random walks, or clustering coefficients.
- Centrality measures, measures the importance of a node in the network.
- Interlayer measures, one way of developing interlayer measures is to compare intralayer network measures of at least two layers.
- Communities, is a very used measure where densely connected nodes, comparing to the rest of the network nodes, are found.

According to Najari et al (2019), multilayer networks have a strong correlation between the nodes in different layers, this indicates that ignoring the importance of layers to predict links on the network, could lead to loss of important information to, for instance, make accurate recommendations on a recommender system.

2.5 Link Prediction

Transforming real world interactions into graph representations allows us to convert recommendation problems to a link prediction task. While many link prediction problems traditionally occur on single-layer networks, recommendation problems, which involve matching node pairs within a network, are more accurately modeled in bipartite multilayer networks. This multilayer approach aligns more closely with the complexities of recommendation tasks, enhancing their effectiveness. (Li & Chen, 2013).

According to Samad et al. (2020), link prediction problems can be put into two categories, missing and future links prediction. Being the main goal to predict the new or missing links between pairs of nodes, link prediction tasks have a lot of useful applications, being one of the most used the recommender systems (Jafari et al., 2021).

As stated by Lü and Zhou (2011), the work of recommending content to users can be considered a link prediction problem using bipartite networks.

The link prediction problem is traditionally applied to one-mode networks. While link prediction algorithms can be generalized to bipartite networks, they may not perform as effectively due to the unique characteristics of bipartite graphs. Bipartite graphs exhibit a distinctive feature: two connected vertices belong to different types of nodes and do not share common neighbors. This divergence from traditional one-mode networks makes common neighbor-based approaches less effective for link prediction. A potential solution to address this challenge is to project the bipartite network into a one-mode network. This projection involves creating a new network where nodes of the same type are connected if they share a common neighbor in the original bipartite graph. This transformation enables the more effective application of conventional link prediction methods. While projecting a bipartite network into a one-mode network can simplify link prediction task, this projection that simplifies the link prediction may result in information loss. Therefore, the decision should be based on the specific requirements of the problem, considering whether information preservation is critical or not (Kunegis, De Luca & Albayrak, 2010).

The various approaches to link prediction can be categorized into different classes, similarity based, learning based, probabilistic, and preprocessing approaches, being the most used the two first ones. On the similarity-based approach, for each pair of not connected nodes a similarity measure is computed and the top ranked ones, are the ones most likely to be linked in the future. In the learning-based approach, ML techniques are used and, in many cases, achieve better predictions compared to similarity-based predictions, although they are more computationally heavy (Najari et al., 2019).

Different link prediction approaches exist and are depicted in figure 2.4.

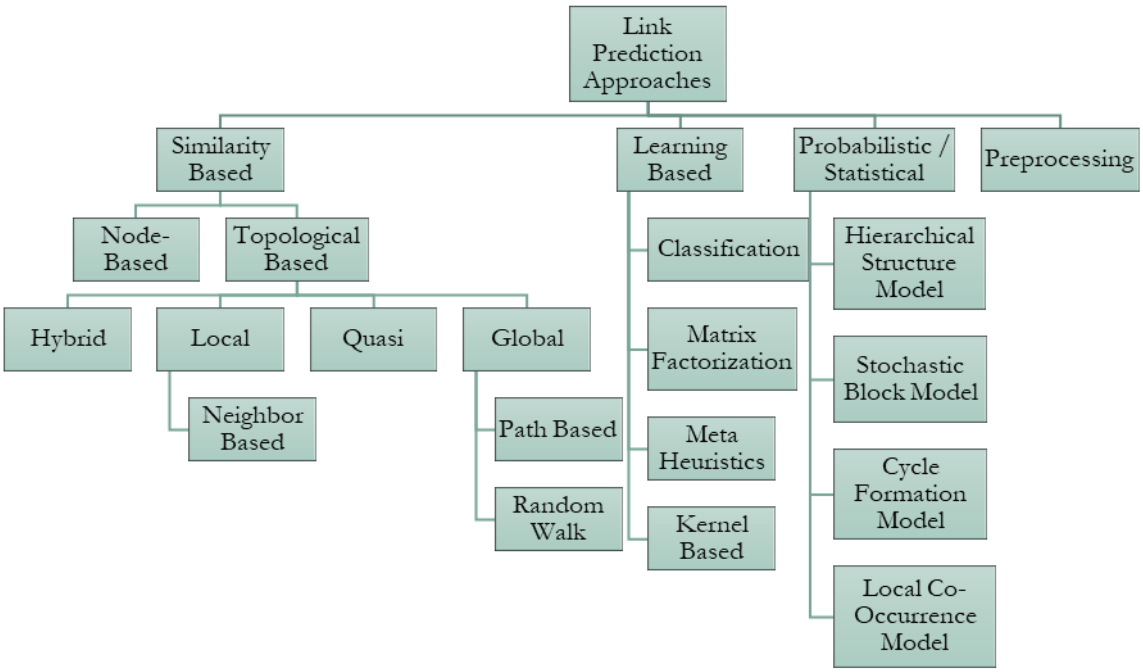

Figure 2.4 - Taxonomy o link prediction approaches. Based on source: Samad et al. (2020).

2.5.1 Similarity-based approaches

Similarity-based approaches in link prediction assume that nodes attempt to form connections with other nodes that exhibit some similarity. Node similarity is typically determined by the presence of a common connected node or the shortest distance between nodes in the network. These approaches can be categorized as either node-based or topological. In node-based similarity approaches, it's assumed that the higher the similarity between two nodes, the greater the likelihood of a link forming between them. This concept reflects the idea that individuals tend to connect with others who share similar characteristics. Node-based approaches consider node or edge attributes to quantify similarity. Topological-based approaches utilize various metrics without considering node or edge attributes. These metrics can be further classified into local, global, or quasi-based categories. Local metrics focus on determining the similarity of each node with respect to its neighboring nodes. These metrics are efficient, particularly for predicting links in dynamic networks, as they rely on local neighborhood information. However, they have a limitation because they may restrict nodes from forming connections beyond their immediate neighborhood, limiting the prediction of links at greater distances (Samad et al., 2020).

The equations below, that are used to compute the similarity of nodes in a network,

have the following notation:

- (u, v) represent a pair of nodes
- τ_u it is the set of connections of node u to its neighbors
- $\tau_u \cap \tau_v$ represents the matching neighbors of nodes u and v
- $\tau_u \cup \tau_v$ represents all the neighbors of both nodes u and v

Common neighbors (CN)

It's one of the most used methods for link prediction and it states that the more common neighbors a node has, the biggest the chances that a link will be formed in the future.

$$CN(u, v) = |\tau_u \cap \tau_v|$$

(2.6)

Where τ_u and τ_v are neighbor nodes of nodes u and v (Samad et al, 2020).

Jaccard Coefficient (JC)

As stated by Lakshmi et al. (2021), is the normalized the CN measure, it considers the common neighbors and the total neighbors of both nodes.

$$JC(u, v) = \frac{|\tau_u \cap \tau_v|}{|\tau_u \cup \tau_v|}$$

(2.7)

Adamic-Adar Index (AA)

This index gives a bigger importance to common neighbors with low degree, initially this method was proposed to find the similarity between two pages (Samad et al, 2020).

$$AA(u, v) = \sum_{z \in \tau_u \cap \tau_v} \frac{1}{\log |\tau_z|}$$

(2.8)

Preferential attachment (PA)

This method proposed by Barabâsi and collaborators (Barabâsi, Jeong, Néda, Ravasz, Schubert & Vicsek (2002)), states that a node will be connected with another node with a

high degree (Samad et al., 2020)

$$PA\ (u, v) = |\tau_u| \cdot |\tau_v|$$

(2.9)

The global methods use the whole network, although because of their complexity, they can be hard to use on very large networks.

Shortest Path (SP)

It is the simplest measure because it computes the similarity between nodes u and v considering the shortest path between them (Samad et al. 2020).

$$SP\ (u, v) = \min(|Pu \rightarrow v|)$$

(2.10)

Katz (KZ)

This measure computes the total number of paths between u and v, penalizing the connections made with distant neighbors. It is similar to Google's PageRank and eigenvector centrality (Lakshmi et al., 2021).

$$KZ\ (u, v) = \sum_{l} \beta^l |P_l(u, v)^l|$$

(2.11)

Where l is the path length between u and v and $P_l(u, v)^l$ is all paths between u and v with length l.

FriendLink (FL)

This method runs all the paths between two nodes, assuming that all the paths between them can be used, but the bigger the path the worse the performance of the measure (Samad et al., 2020).

$$FL\ (u,v) = \sum_{i=1}^{l} \frac{1}{i-1} \cdot \frac{|P_l(u,v)^l|}{\prod_{j=2}^{i}(n-j)}$$

(2.12)

Where n is the size of the network, l is the path length between u and v and $P_l(u,v)^l$ is all paths between u and v with length l.

Page Rank (PR)

Represents the significance of a node based on the significance of its neighbors, it was designed to rank web pages in Google search. Today PR is a popular tool to analyze different types of networks (Coppola, Guo, Gill & Croon, 2019).

$$R(u) = \sum_{v \in B_u} \frac{R(v)}{N_v}$$

(2.13)

Where B_u represents the set of all nodes connecting to u, N_v is the number of edges departing of node v and $R(v)$ is the Page Rank of node v.

Quasi approaches combine local and global methods, trying to take advantage of the qualities of both methods.

Local path index (LP)

Takes into consideration the local paths but with a wider horizon than CN, it uses information of paths with length 2 and 3 (Samad et al. 2020).

$$LP\ = A2 + \alpha A3$$

(2.14)

Where LP is the adjacency matrix of the nodes with length 2 and 3 and because the neighbors with length 2 are more important than the neighbors with length 3, α will be used as an adjustment factor. On the equation, A2 is the adjacency matrix of nodes with length 2 and A3, is the adjacency matrix of nodes with length 3.

Local random walk (SRW)

When it measures the random walk from a start node to an end node, it restricts the random walks to a small number (Samad et al., 2020)

$$SRW\ (u,v) = \frac{|\tau(u)|}{2|E|}\overrightarrow{p_v^u(t)} + \frac{|\tau(v)|}{2|E|}\overrightarrow{p_v^u(t)} \quad (2.15)$$

Where $\overrightarrow{p_v^u(t)}$ is the probability vector estimated on the iteration t.

2.5.2 Learning based approaches

According to Li & Chen (2013), learning based methods, compared with other methods, generally have a more stable performance across different datasets and require a bigger computational power.

Learning-based link prediction models learn a group of parameters by processing the input graph and use a certain methodology to create the output. These models often have better results than the similarity-based ones, but that does not mean that they cannot be used. On the one hand, similarity-based models provide a better understanding of the underlying characteristics of the networks, and they often take less computational effort, making them well suited for online predictions without the need for resource-intensive training procedures or elaborate feature selection stages. (Jafari et al., 2021).

According to Samad et al. (2020), the learning-based approaches can be modeled as a classification problem, we can have every pair of nodes as an instance with a class label, if the nodes are connected, the label says it is positive, otherwise says it is negative. This kind of approach has to deal with a problem, that is class imbalance. Matrix factorization approach extracts and utilizes additional features for link prediction, commonly employed in many recommender systems. The complexity of link formation, influenced by numerous factors, often calls for meta-heuristic methods. These approaches makes hypothesis in the network and aim for higher prediction accuracy compared to other methods. Kernel-based methods for link prediction involve integrating different graph kernels and dimensionality reduction techniques. What sets them apart is their ability to learn a function that outputs an adjacency matrix, enhancing their adaptability and predictive power.

2.5.3 Probabilistic or statistical approaches

These approaches solve the link prediction task on the base of probability and statistical analysis. These probabilistic methods usually suppose that the network that is going to be studied has a known structure and a set of model parameters that are estimated in order to build a model. For each missing link, formation probability is computed on the base of these parameters. The formation probability values sort the important links as it is done in similarity-based approaches. Different models exist, e.g. the Hierarchical Structure Model, because most of the real networks are organized hierarchically, where lower degree nodes are expected to have higher clustering coefficient than higher degree nodes. It can be assumed that the nodes in the network are distributed in blocks and communities, where nodes that belong to the same group or community have the same status. The chances of link formation between two nodes depends on the community or block they belong. The Cycle Formation Model, which is based on the hypothesis that networks have the inclination towards close cycles in their link formation process. This hypothesis is the same as other methods, like common neighbors, which consider the number of cycles that would be created if the link existed. Moreover, this approach tries to detain longer cycles by increasing clustering coefficients to make it more generalized (Samad et al., 2020).

2.5.4 Preprocessing approaches

Preprocessing approaches, often referred to as meta-approaches, or high-level approaches, are designed to work in conjunction with other methods to enhance their performance and reduce noise in the network data. According to Samad et al. (2020), three different preprocessing approaches play a vital role in refining network data and improving the performance of subsequent link prediction methods. **Low Rank Approximation:** This method simplifies network structure by solving the low-rank approximation problem. It leverages an adjacency matrix to make the network noise-free. The optimization process minimizes a cost function that estimates the fit between the original and approximated matrices with reduced rank. **Handling Unseen Bigrams**: Unseen bigrams, valid but unobserved pairs, are addressed in various applications, as speech recognition and cryptography. In link prediction, a strategy similar to bigrams can be applied to reduce noise by replacing similar nodes. **Filtering or Clustering**: Another noise reduction method involves filtering or clustering, which targets the removal of weak ties between nodes to improve link prediction results. Weak ties are characterized by links with few or no shared neighbors. This approach assigns similarity

scores to connected pairs, allowing the removal of the weakest links to clean the network and enhance link prediction accuracy.

2.6 Interlayer similarity measures

With these measures, the goal is to calculate the similarity between layers in multilayer networks, some of the measures most used can be defined as below for a network with two layers (Najari et al., 2019).

Degree-degree correlation (DDC)

Because the nodes have different degrees on the layers, DDC measures the interlayer correlation of the degrees across the layers (Najari et al., 2019).

$$DDC = \frac{\sum_{k_{L1}} \sum_{k_{L2}} (k_{L1} K_{L2} (p(k_{L1}, k_{L2}) - (\sum_{k_{L1}} p(k_{L1}, k_{L2}))(\sum_{k_{L2}} p(k_{L1}, k_{L2}))))}{\sum_{k_{L2}} k_{L2}^2 \sum_{k_{L1}} p(k_{L1}, k_{L2}) - (\sum_{k_{L2}} k_{L2} \sum_{k_{L1}} p(k_{L1}, k_{L2}))^2}$$

(2.16)

Where $p(k_{L1}, k_{L2})$ is the probability that a random node has degree k_{L1} in layer L1 and K_{L2} in layer L2.

Betweenness (BW)

Betweenness similarity measures the importance of a node, and it represents the number of times a node appears on the shortest paths of the network (Najari et al., 2019).

$$D_{BW_i} = |BW_i^{(L1)} - BW_i^{(L2)}|$$

(2.17)

Where $BW_i^{(L1)}$ is the betweenness similarity measure of node i in layer L1 and $BW_i^{(L2)}$ is the betweenness similarity measure of node i in layer L2. The normalized measure can be computed as it is on the equation below.

$$S_{BW} = \frac{\sum_{i=1}^{N} S_{BW_i}}{N}$$

(2.18)

With $S_{BW_i} = 1 - D_{BW_i}$

2.7 Performance evaluation measures

When a recommender system recommends the top ranked items, those recommendations that used a link predictions task, need to be evaluated. To do it the data needs to be divided into training set and test set, being the training set the known data. On the test set the results need to be evaluated and the metric used to do it depends on the goal of the recommender system (Lü, Medo, Yeung, Zhang, Zhang & Zhou, 2012).

The results of a classifier that labels the results as positive or negative can be represented in a matrix structure called confusion matrix or contingency table (see table 2.1), this matrix has four different categories: the true positives (TP), with the label correctly classified as positive. The false positives (FP), that are the negative examples incorrectly labeled as positive. True negatives (TN) are the labels correctly classified as negative. Finally, the False negatives (FN), are the positive labels classified incorrectly as negative (Davis & Goadrich, 2006).

	Actual Positive	Actual Negative
Predicted Positive	TP	FP
Predicted Negative	FN	TN

Table 2.1 – Confusion Matrix

From the confusion matrix we can get valuable information about the performance of the model, below are represented the equations to compute several measures that can be taken from the confusion matrix, mainly the ones used to plot the Receiver Operating Characteristic curve (ROC), the F1 score and the Accuracy.

.

$$Recall\ or\ True\ Positive\ Rate = \frac{TP}{TP + FN} \quad (2.19)$$

$$Precision = \frac{TP}{TP + FP} \quad (2.20)$$

$$True\ Negative\ Rate = \frac{FP}{FP + TN} \quad (2.21)$$

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

(2.22)

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

(2.23)

Evaluation measures can be classified into two categories, threshold curves, as the Receiver Operating Characteristic curve, that is the false positive rate versus the true positive rate, or the Area Under the Curve (AUC), that is the area under the ROC curve, where the best results are represented by a high AUC. And the fixed threshold curves, where it is used the accuracy, recall, precision, or the F1 score, that is the harmonic mean of recall and precision (Samad, 2020).

The ROC curve uses the results presented on the confusion matrix, it plots the false positive rate (FPR) on the horizontal axis, that measure the fraction of negative labels that are incorrectly classified as positive, and on the vertical axis it plots the true positive rate (TPR), that measures the fraction of positive labels that are correctly classified as positive (Davis et al., 2006).

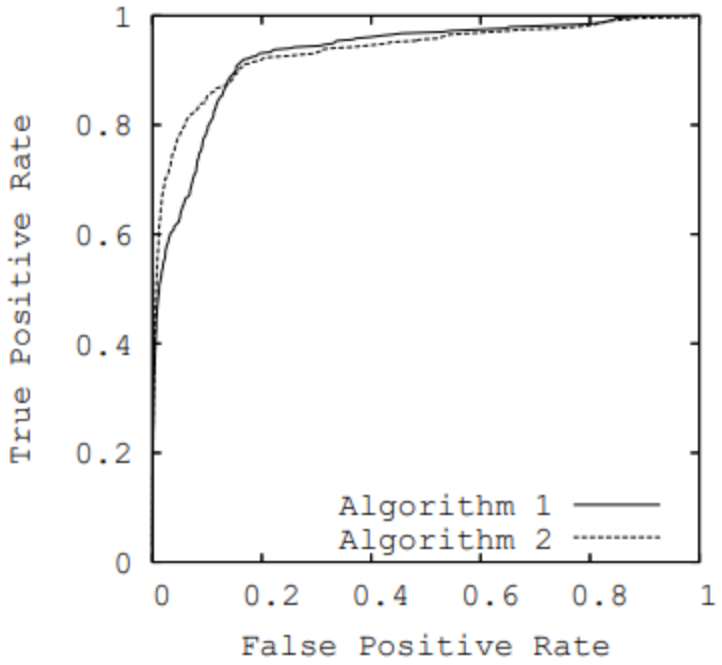

Figure 2.5 - ROC representation for two different algorithms. Source: Davis et al. (2006).

3. Methodology and Data

3.1 Related Work

Several methodologies were studied that achieve good results and use different techniques to solve the link prediction problem, so we can have a broad perspective of the different possibilities available. Four of those methodologies studied are presented below.

Lakshmi & Bhavani (2021) proposed an approach that achieved good results to solve the link prediction problem on bipartite networks. First, they filter the data to have only the customers that gave more than 20 scores to the products. Then, they get the temporal bipartite graph and compute the central neighborhood set in the bipartite graph (BCNS) and apply the breadth first search algorithm (BFS) to obtain the paths between the products and users. For the BFS application, it was set the maximum path length of 10 connections, eliminating some connection that are not important. Filtering the graph according to the most important connections, the authors apply an algorithm to get the B-clique for the products. For that purpose, they consider the users that gave a score to the products and then, for each user, the products they bought. To get the top N scores, referred as B-COP, by each product for each user, it is applied a junction tree algorithm and then they have the products sorted by the score they have.

Najari et al. (2019), proposed a framework based for link prediction accounting interlayer similarity, which is based on using both intralayer features and interlayer similarity, for multiplex networks. In the proposed framework, the intralayer predictor calculates the probability of link existence by using intralayer features and then these probabilities along with interlayer similarity are given to a synthesizer to calculate the final probability values for link existence. The intralayer link predictor uses only intralayer information for the link prediction and it can use classification or probabilistic methods as the intralayer link predictor. In classification-based methods, the link prediction problem is considered as a classification problem with two classes, then several features are considered and a proper classifier such as SVM, Naive Bays, KNN or logistic regression is used to solve the problem. In probabilistic models, latent features are used to obtain a probabilistic model, resulting in the probability of link existence. The proposed framework includes a synthesizer that combines intralayer and interlayer information to produce meaningful information for the link prediction task. When the network has more than two layers, the similarity of the link is obtained across all layers by obtaining the similarities for all layer pairs.

Xue et al. (2018), presented the called Dual HyperGraph Convolutional Networks that demonstrated good results regarding the robustness to varying sparsity levels, node attribute initialization strategies and handling of imbalanced classes. Given an input multiplex bipartite network, they first transformed it into two sets of homogeneous hypergraphs. The model architecture comprises a hypergraph convolutional network that assumes these dual homogenous hypergraphs as inputs, with additional inter and intra-message passing layers to enable information sharing across the networks. Finally, the entire model is trained using a gradient descent-based optimizer. A hypergraph generalizes the notion of an edge in simple graphs to a hyperedge, which can connect more than two nodes. It models heterogeneous interactions, for example, in e-commerce networks, multiple items can form an hyperedge with a user if they are bought or clicked by the same user, this reflects the idea that a user's interaction with multiple items simultaneously, can be represented as a single hyperedge. Multiple users can be connected by a hyperedge to an item, this occurs when multiple users collectively interact with a specific item.

Tang, Chen, Wei, Li, Wang, Wang, & Wang (2022), propose an interlayer link prediction model that uses several attributes of the network, it measures the number of possible common close triads, the similarity of the number of the intralayer links, the number of common matched neighbors and the similarity of neighbors on intralayer links to produced accurate prediction of links.

In table 3.1 we summarize the methodologies introduced in this section

Author(s)	**Year**	**Title**	**Methodology**
Lakshmi & Bhavani	2021	Link Prediction Approach to Recommender Systems	Alternative methodology adapted to bipartite networks using the breadth first search algorithm
Najari, Salehi, Vahid Ranjbar & Jalili	2019	Link prediction in multiplex networks based on interlayer similarity	Framework based for link prediction, which is based on using both intralayer features and interlayer similarity
Xue, Yang, Rajan, Jiang, Wei & Lin	2018	Multiplex Bipartite Network Embedding using Dual Hypergraph Convolutional Networks	Model architecture that comprises a hypergraph convolutional network, that assumes dual homogenous hypergraphs as inputs, with additional inter- and intra-message passing layers to enable information sharing across the networks.
Tang, Chen, Wei, Li, Wang, Wang	2022	Interlayer link prediction based on multiple	Calculates the matching degree of unmatched nodes once by leveraging the

& Wang		network structural attributes	information of a closed triad, intralayer links, matched neighbors and intralayer links of neighbors simultaneously to guarantee accuracy while reducing time consumption

Table 3.1 - Summary of the methodologies presented.

3.2 SPROUT algorithm

In this section we introduce SPROUT – (a Supervised link PRedictiOn in mUltilayer bipartite neTworks), developed in the scope of the dissertation. SPROUT is a supervised algorithm for link prediction on multilayer networks, based on the LPIS algorithm developed by Najari et al (2019). It consists of link prediction using both interlayer and intralayer information to make predictions of new links between a pair of nodes in a certain layer of a multiplayer network. SPROUT allows the link prediction on a multilayer network considering all the nodes on all the layers, even those that do not exist on the layer being studied. One of the advantages of SPROUT over the ones we presented earlier, is that it can be used in multilayer networks that are not specifically multiplex networks.

This algorithm is implemented using a function that calculates the probability of a link being formed between a pair of nodes, as presented in equation 3.1. To compute this probability on a certain layer *Lm*, an intralayer similarity measure is used, in this case it is the Jaccard index, as described in section 2.5.1. After that, an interlayer probability is computed between layer *Lm* and all the other layers, using the pairs of nodes that are matching on each pair of layers. For this purpose, the Jaccard index of the other layer is used and an interlayer similarity measure, for each pair of nodes, is also computed, that in this case was used the node betweenness, introduced on section 2.6.

For the total probability, that represents the probability of the link creation between each pair of nodes in a certain layer of the network, the input parameters are weighted by a value *a* that gives the best combination of interlayer and intralayer probabilities that provides the most accurate results for the links being predicted. The new links considered are the ones that achieved at least a certain threshold β, defined for the minimum probability that gives the most accurate predictions.

$$p_{Lm}^{total}(i,j) = (1-\alpha).\, p_{Lm}^{intra}(i,j) + \alpha.\, p_{Lm}^{inter}(i,j) \quad (3.1)$$

The first step involves creating the probability of link existence by considering the information from other layers of the network where a specific pair of nodes exists. To do this, a matrix is computed, which contains the differences in the betweenness similarity measure for all matching nodes between the two layers, taking into account that the lowest that difference is, the bigger its predicting power. Additionally, the intralayer similarity measure for the other layer is also calculated. This measure incorporates the internal information of the matching pair of nodes in the other layers, in conjunction with the interlayer similarity measure.

The algorithm created by Najari et al. (2019) is used based only on layers that have the same number of nodes in all the layer of the network. To be able to work with a larger scope of networks and more realistic ones, SPROUT is able to work with multilayer networks, where the layers can have different sizes, with different number of nodes, using the information of the whole layers, even if they do not have matching nodes.

Algorithm 1: Supervised link PRediction fOr mUltilayer neTworks (SPROUT) pseudocode

Input

Lm: Layer of the multilayer network where the links are being predicted

$Lk_{k\in\{1,2,\ldots,n\},\ m\neq k}$: Layers of the multilayer network used for the link prediction in Lm

A_{Lk}: Adjacency matrix of layer Lk

J_{Lk}: Jaccard index of layer Lk

J_{Lm}: Jaccard index of layer Lm

B_{LmLk}: Betweenness similarity measure between the nodes of layers Lm and Lk

$0 \leq \alpha \leq 1$: Control parameter that gives the weight for the intralayer and interlayer probabilities that maximize the $p_{Lm}^{total}(i,j)$ predicting power

$0 \leq \beta \leq 1$: Minimum threshold used for the $p_{Lm}^{total}(i,j)$ to consider or not the link creation

Output

$p_{Lm}^{total}(i,j)$: Probability of link existence between nodes i and j on layer Lm

ALGORITHM: Link Prediction for Multilayer Networks

1. For all $(i,j) \in$ Lm and Lk in a multilayer network

1.1 IF $A_{Lk}\ (i,j) = 1$ THEN

$$p_{Lm}^{inter}(i,j) = \sum_{k\in\{1,2,\ldots,n\},\ m\neq k} J_{Lk}(i,j) \times B_{LmLk}(i,j)$$

1.2 ELSE

$$p_{Lm}^{inter}(i,j) = \sum_{k\in\{1,2,\ldots,n\},\ m\neq k} (1 - J_{Lk}(i,j)) \times (1 - B_{LmLk}(i,j))$$

2. Normalize $p_{Lm}^{inter}(i,j)$

$$p_{Lm}^{inter}(i,j) = p_{Lm}^{inter}(i,j) \div MAX\ p_{Lm}^{inter}$$

3. Compute the total probability using intralayer and interlayer measures

$$p_{Lm}^{total}(i,j) = (1-\alpha) \times J_{Lm}(i,j) + \alpha \times p_{Lm}^{inter}(i,j)$$

4. $Return\ the\ (i,j)\ with\ p_{Lm}^{total}(i,j) \geq \beta$

SPROUT uses all the layers on the multilayer network, where the pair of nodes (i,j) exists to compute the probabilities of link existence on layer *Lm*. It uses the intralayer similarity measure of the other layers using the Jaccard index, to measure the centrality of each node in the layer and it also uses an interlayer similarity measure named node betweenness, that measures the node vitality, counting the number of times a node appears on the shortest paths between the nodes in the network, in the case those two nodes are linked or not on layer *Lk*.

In the case the network has more than two layers, the probability is measured across all the pairs of layers where the nodes (i,j) are present. If a link between the pair of nodes (i,j) exists in layer *Lk* and there is a high similarity with layer *Lm* then there is a high probability that this link also exists on layer *Lm*. The confirmation that the link between the pair of nodes (i,j) exists in layer *Lk* is given by the adjacency matrix of layer *Lk*.

Having the probability of link existence using the information of the other layer and then using also the probability of link existence in layer *Lm*, a synthesizer combines the intralayer and interlayer information of layer *Lm* using a weight parameter, that needs to be optimized each time the model is implemented in different datasets, so it can have the optimal mix between the interlayer and intralayer probabilities to maximize the predicting power of the model. This optimization process helps ensure that the model's parameters are fine-tuned to provide the most accurate and effective predictions for the given data.

Because the model has the possibility to predict new links in a layer using information of other layers, we can add a new node to a layer and predict new links for that node in that new layer, using only the interlayer probability. We achieve that, setting the value of $a=1$ on the equation of the total probability, as can be observed on equation 3.2.

$$p_{Lm}^{total}(i,j) = \alpha . p_{Lm}^{inter}(i,j) \quad (3.2)$$

With this possibility, we can have a more complete scope of SPROUT because a user, that is not present in a certain layer, can be added to that layer using the information of the other layers where that user is present, so the recommender system can recommend to a user, movies of a genre that was not rated by that user.

3.3 A small illustrative example of SPROUT

A "toy example" was created to analyse all the steps of the implementation of the SPROUT methodology. We will consider an example involving 10 different users who rated movies. These 10 users are divided into 3 layers, each representing movies of distinct genres: Action, Adventure and Crime. In total, there are 30 different movies, with 10 movies in each genre. In figures 3.1, 3.2 and 3.3 we can see the network representations of the three layers.

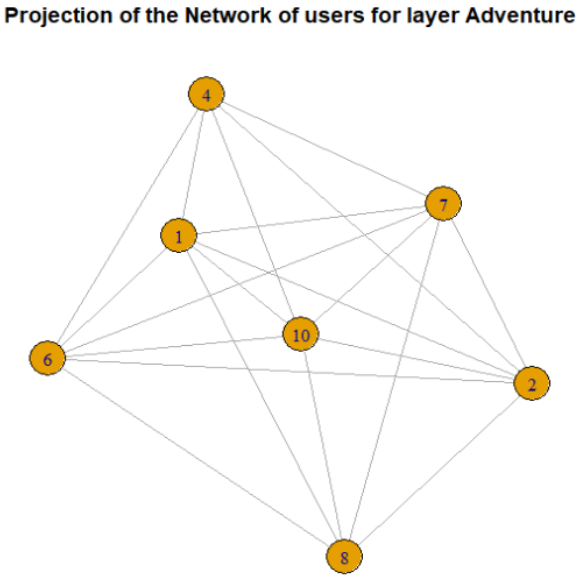

Figure 3.1 - Layer Adventure of the network from the small example

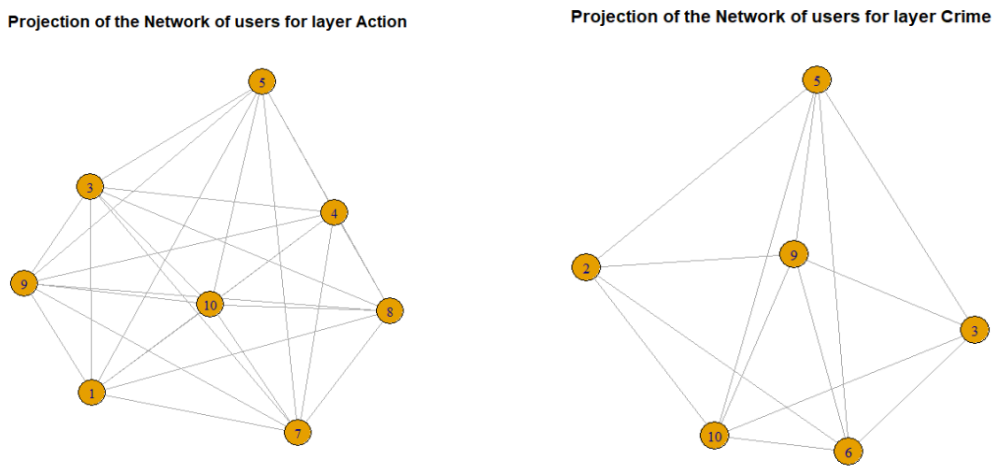

Figure 3.2 - Layers Action and Crime of the network from the small example.

In table 3.2 the results for the p_{Lm}^{Total} are presented for the small example, with three layers and with different number of nodes. Only the probabilities for the matching nodes between layer *Lm* (layer adventure on the example) and the other layers are computed. For the rest, the probability is zero.

	4	**10**	**1**	**8**	**6**	**7**	**2**
4		0.4523810	0.4523810	0.6666667	0.2857143	0.3968254	0.2857143
10			0.5238095	0.4523810	0.5052910	0.4682540	0.5052910
1				0.3634921	0.3571429	0.4904762	0.3571429
8					0.2857143	0.7301587	0.2857143
6						0.3571429	0.3682540
7							0.3571429
2							

Table 3.2 – Symmetric matrix for the results of p_{Lm}^{Total} for the example (probabilities of link formation from node i (in rows), for node j (in columns).

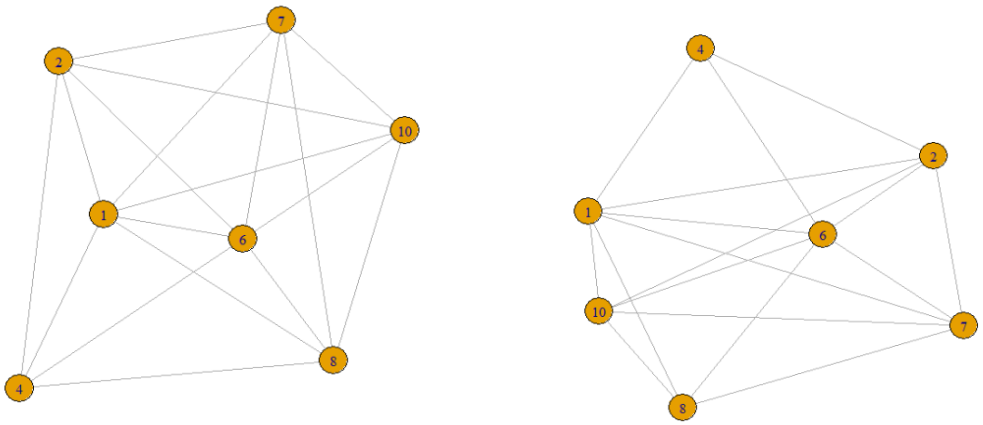

Figure 3.3 - Layer *Lm* with 30% of the connections removed (left), Layer *Lm* with the predicted links (right).

The process followed to predict the links can be seen on figure 3.3 where from the initial connections between the nodes on layer *Lm*, 30% of those links were removed. Then

the algorithm computes the probabilities of link existence on layer *Lm* using both interlayer and intralayer probabilities.

All the values of p_{Lm}^{Total} were computed using the *a*, that is found giving it values from 0 to 1 with 0.1 increments, that maximized the results of the evaluation measures, as can be seen on the first table on APPENDIX A, for different proportions of the interlayer and intralayer probabilities the evaluation measures have different values.

Similar to the optimization process carried out for *a*, there was a necessity to optimize *β* as well. *β* represents the minimum total probability threshold used to determine whether a new link should be predicted or not. This optimization step is crucial to maximize the performance of the model, ensuring that the model's parameters are finely adjusted to deliver the most accurate and effective predictions for the specific dataset in use. The table with all the values of the AUC, F1 score and accuracy, that are explained in more detail in section 3.3, for each value of *a* and *β*, from 0 to 1 with increments of 0.1, can be found on APPENDIX A, as stated before.

Because the similarity measures used were created for networks with only one type of node, the layers of the bipartite network were projected into layers with one type of node. Figure 3.4 represents the bipartite network of layer *Lm*, that in the case of the small example is the layer with the users that rated movies with the genre adventure.

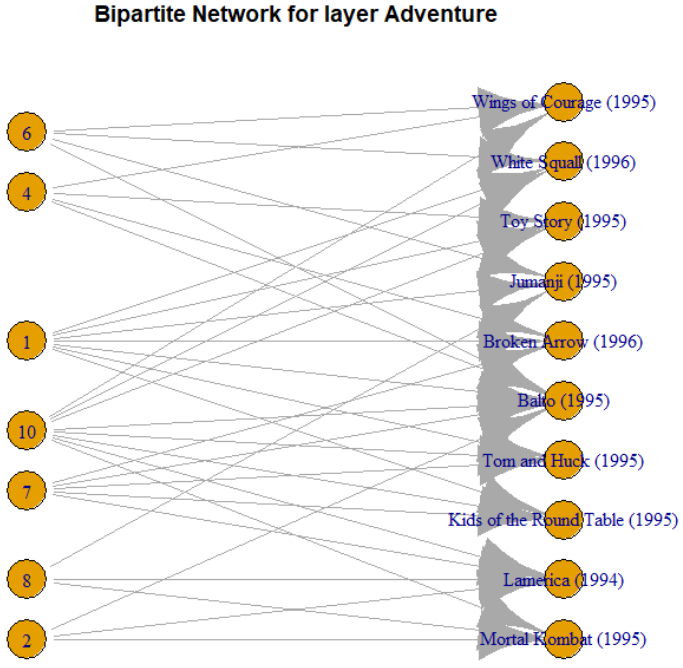

Figure 3.4 - Bipartite projection of layer *Lm*.

Once the projection for a network layer with a single type of node is established (one-mode network), a network comprising only the movies that were rated with the adventure genre can also be created, as depicted in figure 3.5, being the projection for a one-mode network of the users already represented on figure 3.1.

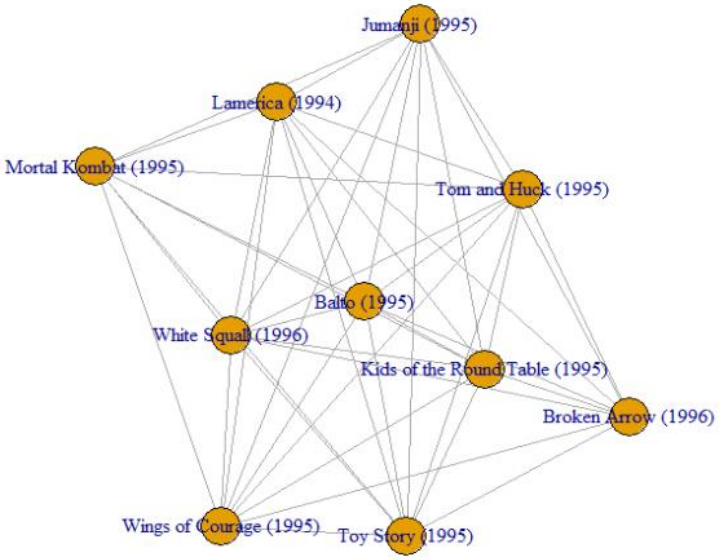

Figure 3.5 - Projection of layer adventure for only the movies.

3.3 Evaluation Metrics

Measuring the quality of the links predicted is a crucial step to evaluate the methodology implemented. To evaluate the quality of the methodology created, a receiver operating characteristic curve (ROC) was created and the respective area under the curve (AUC) was computed, to compare the links that were deleted from the network with the ones that were predicted, the F1 score and the accuracy of the model are also computed.

The evaluation of the implemented methodology is made using the adjacency matrix of the layer L_m with the new links created and the adjacency matrix with the real links on the layer L_m. The ROC curve is created and the AUC is computed, based on the results of model SPROUT, using the probability threshold that was optimized for the most accurate results.

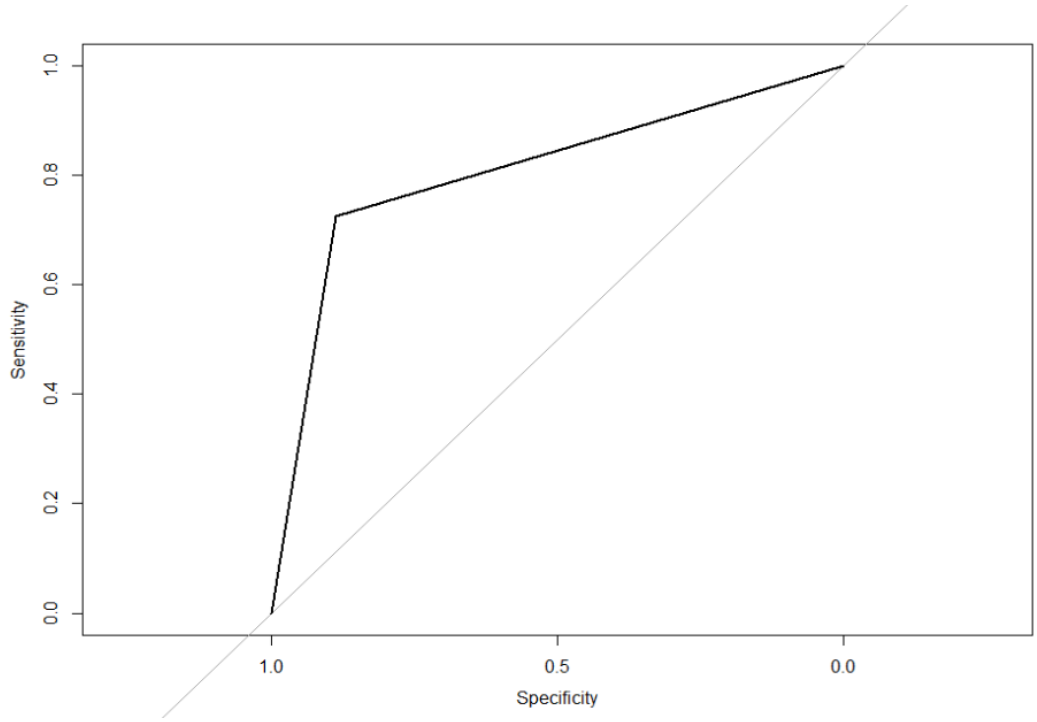

Figure 3.6 - ROC for the predictions for layer Lm of the toy example.

For the example used, the AUC of the respective ROC, is 0.87, which is an excellent result, allowing us to conclude that the model is working with a high accuracy for the predictions made and therefore, it can be projected on a larger network. Because this is a small example, the best result was obtained with β=0.0, using all the links predicted, and three values of *a,* α=0.5, 0.6 and 0.8. The values of β and *a* were obtained by computing the AUC for all the combinations of these two values, from 0 to 1 with 0.1 increments, that maximized it. The F1 score of the model implementation for the small example is 0.89, which is also a good result, and with an good accuracy of 0.81.

3.4 Data: MovieLens 100k movie ratings

The data used for the model implementation is a dataset of movie ratings made available by MovieLens[3]. It is a web-based recommender system and virtual community that recommends movies for its users to watch, based on their film preferences. It is administrated by GroupLens, a research lab at the University of Minnesota. So, as in the small example of

[3] https://grouplens.org/datasets/movielens/latest/

the previous section, the nodes are users, that establish a link with a movie they rated – it is therefore a bipartite network.

This dataset was generated on November 21st, 2019, with the data of movie ratings and also tag activities generated on the MovieLens recommender system. The network created to implement SPROUT uses this database with 100836 ratings, created by 610 users, for 9742 different movies, that was created between March 29th, 1996 and September 24th, 2018. The users selected for analysis are only identified by their *userId*, that was anonymized. We know that they rated at least 20 movies, with no more information provided about them. Only movies with at least one rating or tag were included in the database. The database has six different files that are linked between them with common variables that can be considered as the keys of the tables on a relational database, as can be analysed on table 3.3.

File	**Description**
genome-scores.csv	Contains movie-tag relevance data
genome-tags.csv	Provides the tag descriptions for the tag IDs in the genome file
links.csv	Each line represents one movie with a link to IMDB and TMDB
movies.csv	Each line represents one movie with the genres
ratings.csv	Each line represents one rating of one movie by one user
tags.csv	Each line represents one tag applied by one user to one movie

Table 3.3 - Files of the MovieLens database.

Table 3.4 contains the relations between each file: the six files work as a relational database with some variables that work as the primary key in each table and as the secondary key on other tables.

links.csv	**movies.csv**	**ratings.csv**	**tags.csv**	**genome-scores.csv**	**genome-tags.csv**
movieId	movieId	userId	userId	movieId	tagId
imdbId	title	movieId	movieId	tagId	tag
tmdbId	genres	rating	tag	relevance	
		timestamp	timestamp		

Table 3.4 - Variables of all the files in the database.

The files *tags.csv*, *genome-scores.csv* and *genome-tags.csv* have information about the tag structure, where each movie that was tagged has a value for its relevance. That classification informs about how relevant certain properties are in each one of the movies tagged (thought-provoking, realistic, etc.). The tags are not used for the link prediction task, and therefore these three files are not considered when implementing the methodology proposed, but these tags could be used for more advance recommendations in a future improvement of SPROUT.

The *links.csv* file contains the links for two websites, Internet Movie Database[4] (IMDB) and The Movie Database[5] (TMDB), which could potentially offer information about the movies in the database. However, since no web scraping was required, this file was not utilized in the creation of the network for the link prediction.

The files used to create the bipartite multilayer network to implement the link prediction model were the files *movies.csv* and *ratings.csv*. The *movies.csv* file not only contains the *movieId* variable, that will link with the *ratings.csv* file, but also includes the movie's *title*, including the respective release year. Additionally, it features the *genres* variable, which is a pipe-separated list indicating the movie's genres (e.g. Crime | Drama | Romance). This variable ensures the construction of the network layers based on the movie genres.

The dataset encompasses 19 different genres for potential layer creation, however, the genre "IMAX", which corresponds to only one movie, was excluded. Thus, the network was built with the remaining 18 genres, each representing a unique network layer. The significance of these 18 genres, determined by the number of movies associated with each one of them, is depicted in figure 3.7, utilizing a treemap visualization. Table 3.5 provides the number of movies for each genre, categorizing them based on their position within movies with multiple genres and presenting the respective totals. Movies lacking genre information were not considered in the model implementation.

[4] https://www.imdb.com/

[5] https://www.themoviedb.org/

Genres	Count of genre 1	Count of genre 2	Count of genre 3	Count of genre 4	Count of genre 5	Count of genre 6	Total
No genres listed	34						34
Action	1828	1768	1409	689	189	47	5930
Adventure	653	641	494	230	92	24	2134
Animation	298	268	183	76	20	3	848
Children	197	193	90	23	3		506
Comedy	2779	1833	679	118	17	1	5427
Crime	537	525	319	94	15	1	1491
Documentary	386	47	5				438
Drama	2226	1173	357	86	12	1	3855
Fantasy	42	38	24	11			115
Film Noir	12	9	5	1			27
Horror	468	301	91	5			865
Musical	23	8	2				33
Mystery	48	45	12	2			107
Romance	38	17	2				57
Sci Fi	62	25	1				88
Thriller	84						84
War	4						4
Western	23						23
Total	9742	6891	3673	1335	348	77	22066

Table 3.5 - Totals for the movies with a certain genre in the database.

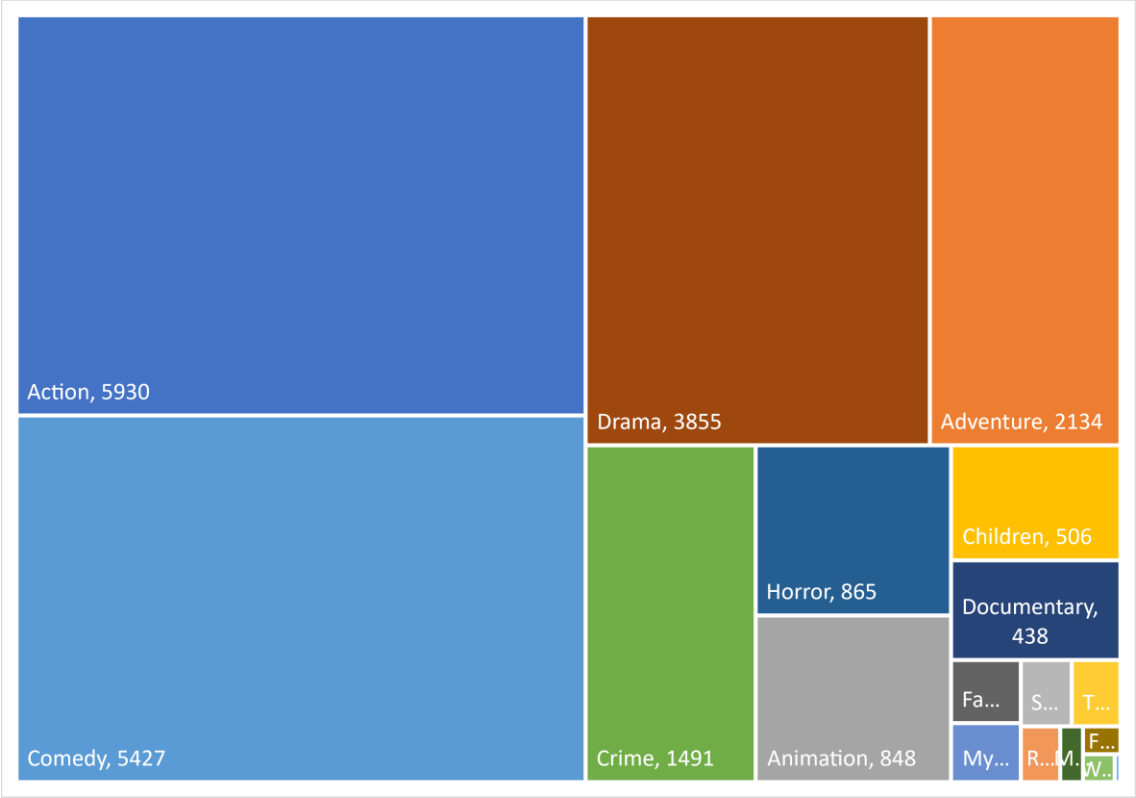

Figure 3.7 – Treemap for the different genres based on the number of movies that have a certain genre.

The other file used for the creation of the bipartite multilayer network was *ratings.csv*. It has the variable *timestamp*, that represents the seconds since midnight of January 1st, 1970, in UTC time zone, when the ratings were created. This variable will not be used for the link prediction tasks. The other variable is the *movieId* and as stated before, it links to the file *movies.csv* that has the movie titles and the respective *genres*. It also contains the variable *userId,* that is an anonymized identification of every user that rated at least 20 movies. The last relevant variable is the *ratings* that each user created for a movie. This categorical variable works in a scale from zero to five, with half-star increments, so the users have 11 different rating possibilities to classify the movies.

This variable will serve the purpose of suggesting movies to users in the cases where a new connection has been established in the network. Specifically, it will recommend movies with ratings equal to or greater than 4, as this threshold distinguishes the quality films worthy of recommendation from those that should not be recommended.

In figure 3.8, it is evident that a rating of 4 was the most frequently assigned, accounting for over a quarter of all ratings. Additionally, approximately half of the ratings are equal to or greater than 4.

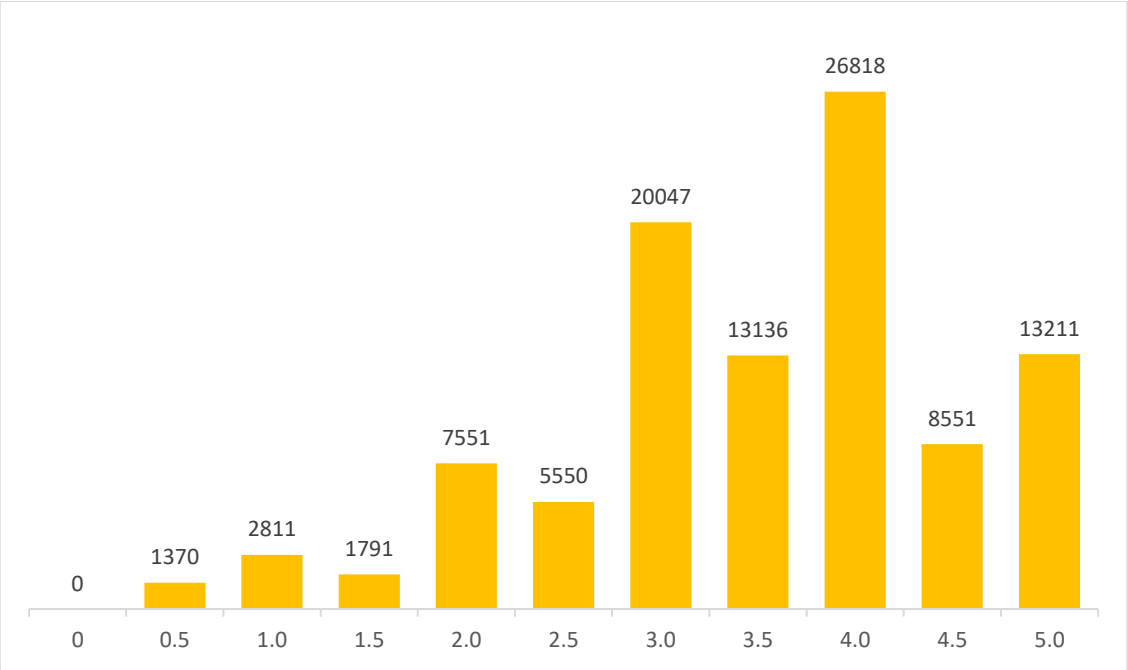

Figure 3.8 - Number of ratings for each rating level.

With the data about the movies and users that rated them, we can create the bipartite network to implement the link prediction model. In order to utilize the Jaccard index for both intralayer and interlayer probabilities, as well as the node Betweenness for interlayer probability, the bipartite network underwent a projection into a one-mode network. This one-mode network exclusively consists of the users from each layer for which we intend to predict the links.

Different metrics, that can be observed on APPENDIX B, were computed to analyse the 18 layers in the network to better understand which are the most important ones to make the predictions. The biggest layers that will contribute more to the interlayer similarity measure are layers *Drama* with 143853 nodes, *Comedy* with 131841 nodes, *Thriller* with 130593 nodes, *Action* with 120655 nodes and *Crime* with 122485 nodes. The smaller layers in the network are layers *Documentary* with 4412 nodes, *Film-Noir* with 11525 nodes, *Western* with 30473 nodes, *Musical* with 45744 nodes and *Horror* with 68711 nodes.

The average degree of each layer gives the understanding of how dense most of the layers are, namely the centrality of a node, that is influenced by the number of edges that in average connect to each node. The layers with more nodes are the ones that have the highest average degree, being the layer *Drama* the one with the highest one, with 471.64 connections

for each node, in average, and the layer *Documentary*, the one with the lowest with an average of 39.56 connections, for each node.

The density of each layer was also studied, and we were able to understand that more than half of the layers have nodes that are very well connected with a density higher than 0.5, with special attention to layers *Drama*, *Comedy*, *Thriller* and *Action*, with a density higher than 0.7. Layers *Documentary* and *Western* have nodes that are not very well connected in the network, with density of 0.17 and 0.34 respectively.

The high value of the global clustering coefficient for all the layers shows that the nodes of the network have the tendency to group very well with each other, forming a tight group with a high density of ties between the nodes.

Below four of the most important layers can be observed, as examples of the diversity of the different types of layers that are part of the network, with the four metrics for each one of them to better understand their composition.

Layer Action

Nº of nodes	**Average Degree**	**Density**	**Clustering Coefficient**
129 655	426.49	0.70	0.86

Table 3.6 - Network metrics for layer action.

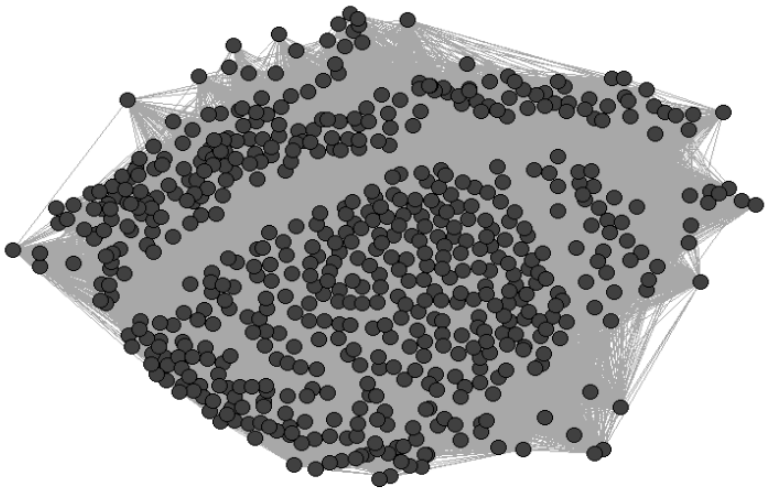

Figure 3.9 - Representation of layer Action.

Layer Comedy

Nº of nodes	Average Degree	Density	Clustering Coefficient
131 841	432.97	0.71	0.86

Table 3.7 - Network metrics for layer Comedy.

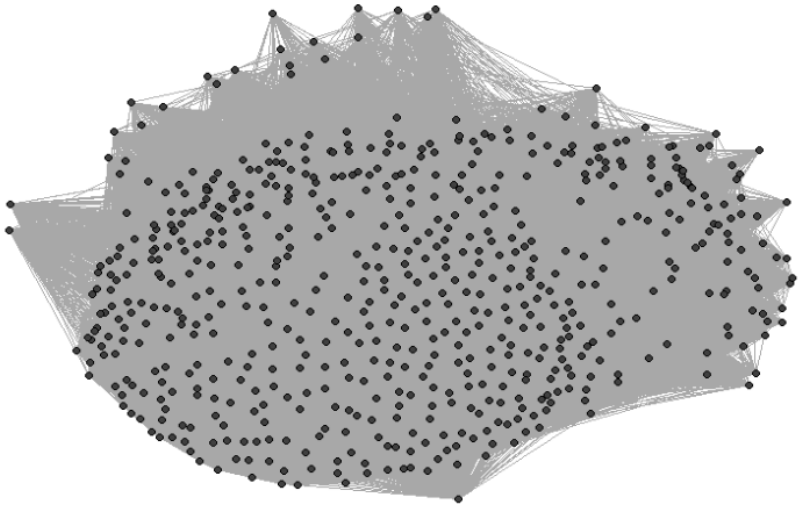

Figure 3.10 - Representation of layer Comedy.

Layer Drama

Nº of nodes	Average Degree	Density	Clustering Coefficient
143 853	471.64	0.77	0.88

Table 3.8 - Network metrics for layer Drama.

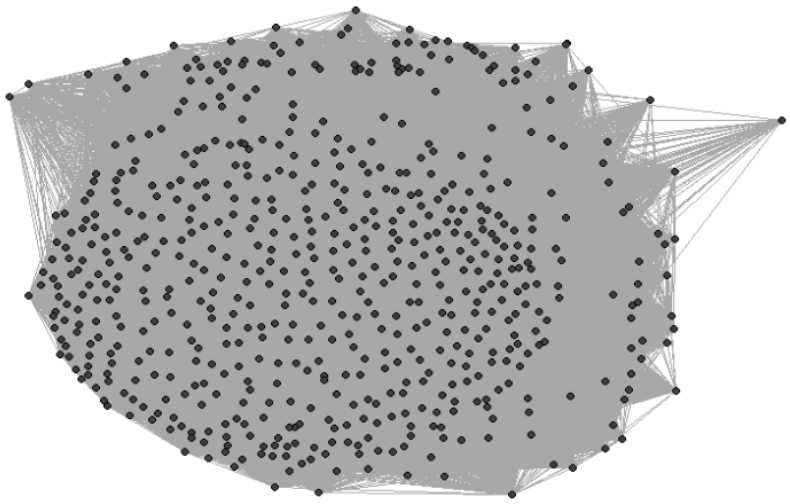

Figure 3.11 - Representation of layer Drama.

Layer Fantasy

Nº of nodes	Average Degree	Density	Clustering Coefficient
88 257	302.76	0.52	0.82

Table 3.9 - Network metrics for layer Fantasy.

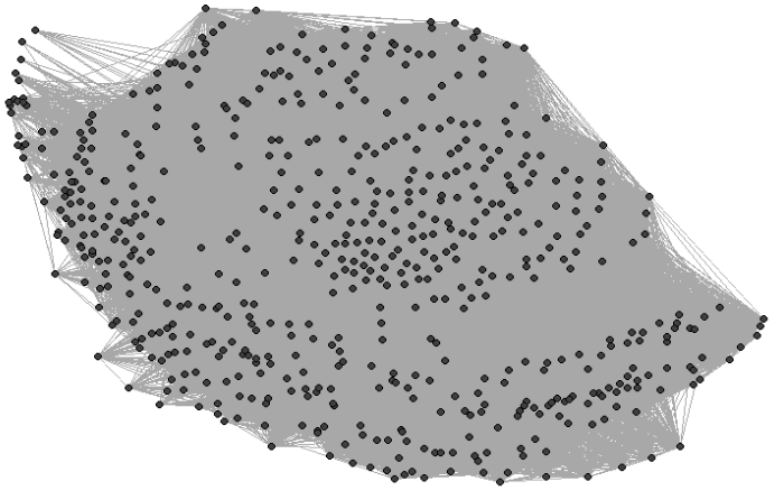

Figure 3.12 - Representation of layer Fantasy.

4. Results and analysis

To implement the model that was created, we used RStudio[6], to prepare the data and also to develop and test the methodology. Excel was also used mainly for some small data analysis tasks.

An implementation of SPROUT is made using the MovieLens dataset. This data, as described before, is related to 100836 ratings, across 9741 movies and created by 610 users of the recommendation platform, the data was created by the users between 29th of March of 1996 to 24th of September of 2018.

The data used was the one related with the movies and ratings files, being the genres of the movies the condition to separate the data into different layers on the network that was created. 18 layers were created with different sizes and connections between the nodes, a list with information about the 18 layers is presented in APPENDIX B.

As described before on chapter 3, the layers of the network are formed projecting the bipartite network into a one-mode network, so specific metrics can be used because most of the similarity measures developed are for networks with only one type of node.

The model SPROUT was implemented using the layer of the genre crime as the layer represented as L_m on the pseudocode of the model, on figure 4.1 is represented layer crime and as can be analysed, although some nodes do not have a lot of connections with other nodes, most of the nodes are very well connected. On table 4.1 can be observed the high density and the high average degree of the nodes on the network, this layer is not one of the biggest on the network, but as the other layers, the clustering coefficient is high.

Nº of nodes	**Average Degree**	**Density**	**Clustering Coefficient**
122 485	406.25	0.67	0.87

Table 4.1 - Network metrics for layer Crime.

[6] RStudio is a free and open-source integrated development environment (IDE) for R, a programming language for statistical computing and graphics (R Core Team, 2020).

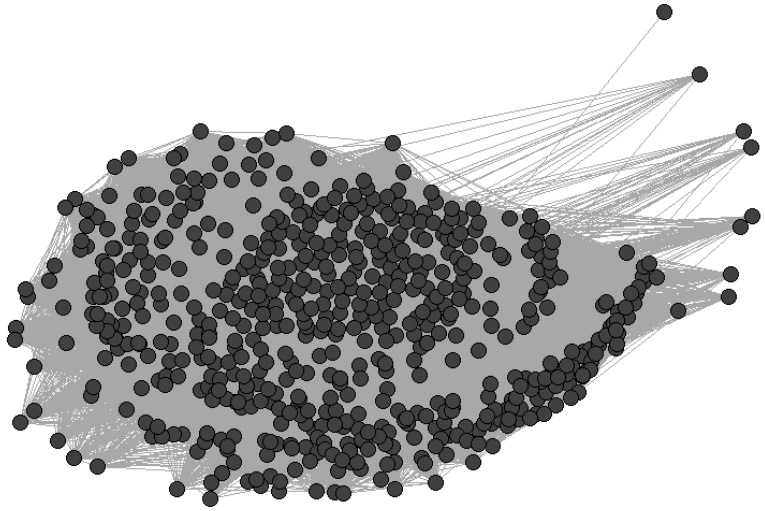

Figure 4.1 - Representation of layer Crime chosen for layer *Lm*.

The Jaccard index of layer *Lk* and the node Betweenness for the interlayer similarity measure were computed to have the values for the interlayer similarity probability. For this, it was taken into account if the nodes were connected on the other layers being analysed, to decide if the probability to use is the probability of the pair of nodes being connected or not, according to the information on the adjacency matrix. Then, to compute the p_{Lm}^{Total}, the Jaccard index of layer *Lm* and the interlayer probability were used and to have the optimal values of the total probability. Different values of *a* were used in order the AUC could be maximized, we could note that an equal value of *a* for both interlayer and intralayer probabilities was the more appropriate to use, letting us conclude that both the interlayer, as well as the intralayer information, were equally important to create the predictions of links. β was also optimized in order the best minimum threshold for the total probability could be used to decide about a link formation or not.

4.1 Results and analysis for four pairs of nodes

To better understand the output of SPROUT, we analysed several pairs of nodes that had their connection deleted from the layer *Lm*. The pair of nodes selected were (6, 150), (1, 62), (375, 68) and (474, 45), on table 4.2 we can see the number of layers in which each pair of nodes appears.

Pairs of nodes	Nº of layers
(6, 150)	12
(1, 62)	11
(375, 68)	12
(474, 45)	9

Table 4.2 - Table with the number of layers in which each pair of nodes appear.

To start the model implementation the interlayer probability was computed summing the Jaccard index and the node betweenness of the pairs layers where these pairs of nodes are present, besides the layer of the genre crime, that is the layer chosen to make the link prediction, and after that the normalization of results were done.

In table 4.3 we can observe the values of the normalized probabilities of the inter-layer probability, we can observe that for the pair of nodes (375, 68) the probability of link formation based on the information provided by other layers is very good.

Pairs of nodes	Np_inter_Lm
(6, 150)	0.05
(1, 62)	0.055
(375, 68)	0.83
(474, 45)	0.09

Table 4.3 - Pairs of nodes and the respective normalized interlayer probability.

As we can observe on table 4.4, the intralayer similarity measure on the layer for the genre crime (layer *Lm*), the pairs of nodes (1, 62) and (474, 45) have a very good probability, although on the other layers where they are present, these two pairs of nodes did not had a good result, for a value of *a* that optimizes the results of the model, with more weight to intralayer than interlayer features, a connection between these two nodes could be predicted.

For the node pair (375, 68), the Jaccard index exhibits a favourable value, particularly when comparing with the interlayer similarity probability. Consequently, it is likely that a link will be predicted for this node pair. In contrast, when evaluating the node pair (6, 150), it becomes evident that a new link is unlikely to be predicted. This conclusion is drawn from the poor interlayer probability value and the relatively modest intralayer probability value on layer *Lm*.

Pairs of nodes	Jaccard index of layer Crime (*Lm*)
(6, 150)	0.46
(1, 62)	0.86
(375, 68)	0.61
(474, 45)	0.92

Table 4.4 - Intralayer probability on layer crime.

To compute the total probability using the equation 3.1 presented on chapter 3, there was the need to optimize the *a* parameter, that allows the model to have the best weights for the interlayer and intralayer probabilities. To attain the optimal value of β, which represents the minimum total probability threshold for link prediction, it should be optimized in conjunction with *a*. The goal is to find the combination of β and *a* that yields the best evaluation measures for the model, when applied to the specific dataset in use. This optimization process helps ensure that the model's parameters are fine-tuned to provide the most accurate and effective predictions for the given data. For the results that can be observed below in table 4.5, α=0.5 and β=0.3 are the values that optimize the results of the model, as can be analysed on APPENDIX A.

In table 4.5 can be observed that only one new link will be created to the four pairs of nodes considered, considering the threshold for the minimum total probability that is used. The new link will be created between nodes with *userIDs* 375 and 68, the movie recommendations will be done on chapter 5, as a practical example of the deployment of a RS using the results provided by SPROUT.

Pairs of nodes	p_{Lm}^{total}
(6, 150)	0.25
(1, 62)	0.46
(375, 68)	0.72
(474, 45)	0.50

Table 4.5 - Values of the total probability used for the link prediction.

With these final values of the link prediction probabilities, we can conclude that with an equal distribution for the interlayer and intralayer probabilities for the total probability and a minimum threshold β of 0.3 for the total probability, we will decide if a new link should be created or not. We created on ROC curve with the results of the model, that can be analysed on fig 4.2, and an AUC of 0.93. The F1 score is 0.9 and the accuracy is 0.93, which are

excellent results, that let us conclude that this model has a very good ability to predict new links in a certain layer of the network.

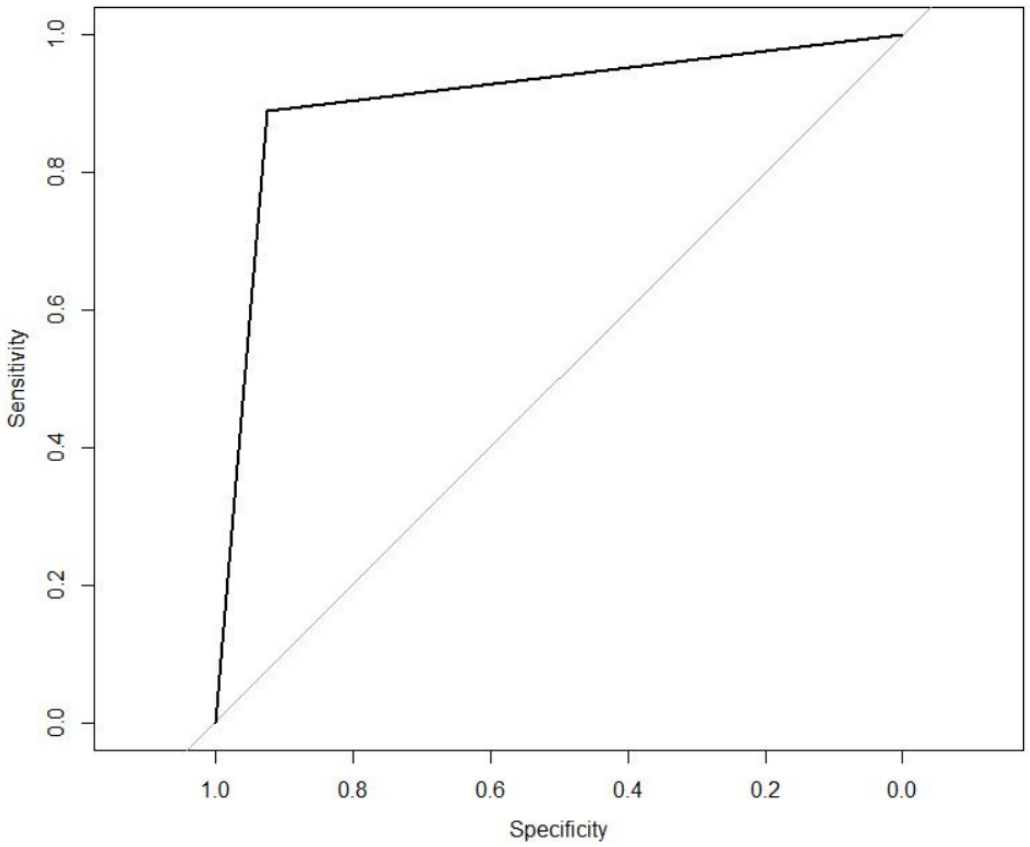

Figure 4.2 - ROC for the results after the implementation of the model SPROUT to the MovieLens dataset.

5. Recommendation of movies based on the new link predicted

Because the final goal of SPROUT is to predict new links between pairs of users and recommend to the users the movies with a rating superior or equal to 4, in this chapter a simple recommendation of movies to the users for which a new link was created in the layer crime, is proposed. The new link created between each pair of nodes will be used to recommend movies to both users. On chapter 4 was analysed the implementation of the model for four pairs of nodes and in the end a new link was predicted between nodes 375 and 68.

For user 68 we can recommend the movies user 375 rated with a value above or equal to 4 with the genre crime, as observed on table 5.1, 5 movies can be recommended to user 68.

Movie title	**Genres**	**Ratings given by user 375**
Shawshank Redemption, The (1994)	Crime | Drama	5
Hamlet (1996)	Crime | Drama | Romance	5
Patriot Games (1992)	Action | Crime | Drama | Thriller	5
Wild Things (1998)	Crime | Drama | Mystery | Thriller	4
Lethal Weapon 3 (1992)	Action | Comedy | Crime | Drama	4

Table 5.1 - Movies that will be recommended to user 68 based on the ratings of user 375.

For user 375 we can predict the movies user 68 rated with a value above or equal to 4 with the genre crime, as observed on table 5.2.

Movie title	**Genres**	**Ratings given by user 68**
Office Space (1999)	Comedy|Crime	5
Scarface (1983)	Action|Crime|Drama	5
After the Sunset (2004)	Action|Adventure|Comedy|Crime|Thriller	5
Sin City (2005)	Action|Crime|Film-Noir|Mystery|Thriller	5

Eastern Promises (2007)	Crime|Drama|Thriller	5

Table 5.2 - Movies that will be recommended to user 375 based on the ratings of user 68.

For simplicity only 5 movies, from the 41, that will be recommended to user 375 that were rated by user 68, are presented in table 5.2, the complete list is displayed in APPENDIX C.

6. Conclusions and Future Work

Developing a model to predict new links between two nodes in a multilayer network using the information not only of the layer where the model is being implemented, but from all the other layers where those two nodes are present proved to be both feasible and accurate. It allowed the creation of new links with meaningful information, which can be employed as an input of a recommender system that can be tailored for diverse recommendation needs.

The application of link prediction tasks on bipartite multilayer networks revealed to be a topic not very well developed. Lakshmi & Bhavani (2021) proposed an approach to a single-layer bipartite network where they get the temporal bipartite graph and compute the central neighborhood set in the bipartite graph (BCNS) and apply the breadth first search algorithm (BFS) to obtain the paths between the products and users. Filtering the graph according to the most important connections, the authors apply an algorithm to get the B-clique for the products. To get the top N scores, called B-COP, by each product for each user, it is applied a junction tree algorithm and then they have the products sorted by the score they have.

Several studies using multiplex networks were made, with authors Najari et al. (2019), proposing the LPIS model for link prediction accounting interlayer similarity, which is based on using both intralayer features and interlayer ones, but only being applied on multiplex networks.

The two models performed very well for the types of networks for which they were designed, several features proposed could be used in a more complete model that can analyse not only bipartite single-layer networks or bipartite multiplex networks, but also networks with several layers with a different number of nodes. Because SPROUT uses the internal information of the layer being studied as well as the internal information of all the other layers, despite having unmatched nodes, computing the similarity measure for a specific pair of nodes using the information of the whole layer. That revealed to be a good advantage because the model will not have a significant loss of information, ignoring nodes that do not match between the pairs of layers studied. The information comparing a certain pair of nodes between two layers is also taken to know if a certain pair of nodes has the same importance on all the layers where it is present.

The primary objective of this work is to accurately establish new connections using a model that is sufficiently flexible to accommodate networks with layers of varying typologies

and sizes. For this, the control parameter *a* used on the synthesizer allowed the model to always have a good balance between the use of intralayer and interlayer information that gave the best predictions. The parameter β defined the minimum threshold for the usage of the most meaningful results of the model and it being maximized together with α, ensures that the model's parameters are fine-tuned to provide the most accurate and effective predictions for the given data., this model could be adapted for different types of data that create networks with different number of layers and different types of layers. For smaller networks with only two or three layers, the intralayer information of layer *Lm* was more important for the link creation decision and a lower threshold defined by β, that represents the minimum total probability was also used to consider more possibilities of links creation, showing the flexibility of the model developed.

A future application of the SPROUT model should be made using similarity measures specifically developed for bipartite networks, e.g. odd path counting, to avoid some of the information loss that can happen when the bipartite network is converted into a one-mode network.

A challenge faced with this model applied for a large network is the computational power needed to create the network with all the layers, but also the computation of the similarity measures and probabilities needed to decide on the link creation. Because of this, servers with high computational power are needed, as it was the case of this dissertation where the FEP's R server needed to be used to run the large MovieLens dataset. This disadvantage could be improved using clustering techniques to try to decrease the size of the network, without compromising too much on the efficiency of the model.

Future developments of the SPROUT model should take into account other information that can be used to feed the model, besides the structural information used. The usage of weights on the existing links could be an important factor that could be used on the stage when it is decided which link should be created. Tags can also be used as an importance measure for the link prediction, text mining tasks can be performed, for instance, a sentiment analysis considering the types of words used on the opinion given by the users can be performed to understand which users and movies are associated with positive feedback.

New models could also be created using the big developments of deep learning models, that could create high level abstractions of the data using multiple processing layers, allowing their usage on increasing complex data structures being created, e.g. using graph neural net-

works. Dynamic network structures are also something to be considered, with the link prediction tasks being performed regularly, to take advantage of the fast increase of information that online platforms have.

Temporal link prediction is also an important topic, it is a task of predicting new links that will be formed in the future, based on different snapshots of the network through time (Qin & Yeung, 2023).

With this complexity, another challenge are the evaluation methods. Evaluating the quality of a complex model is difficult because there is the need to understand which layers contribute more to a meaningful result. Because the algorithms could perform better depending on the network topology, but worse in terms of node characteristics, making it difficult to determine which method performs better in the future.

With the objective of sharing our research findings and ideas, as well as receiving feedback and constructive criticism about our work, we submitted an extended abstract with the main points of this dissertation to the Complex Networks 2023 conference[7]. The conference is scheduled to take place from the 28th to the 30th of November 2023. Presenting this extended abstract can provide us valuable feedback for further improvements and offer alternative perspectives on the subject under study.

[7] https://complexnetworks.org/

7. References

Adomavicius, G. & Tuzhilin, A. (2005). Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on knowledge and data engineering, vol. 17, nº 6, June 2005.

Barabâsi, A.L., Jeong, H., Néda, Z., Ravasz, E., Schubert, A. & Vicsek, T. (2002) Evolution of the social network of scientific collaborations. Physica A: Statistical mechanics and its applications, 311(3-4): 590– 614.

Barber, M.J., Faria, M., Streit, L., & Strogan, O. (2008). Searching for Communities in Bipartite Networks. AIP Conference Proceedings 1021, 171.

Bessy, S. (2013). Some problems in graph theory and graphs algorithmic theory. Discrete Mathematics [cs.DM]. Université Montpellier II - Sciences et Techniques du Languedoc, 2012. tel-00806716. https://theses.hal.science/tel-00806716

Burke, R. (2002). Hybrid Recommender Systems: Survey and Experiments. Article in User Modeling and User-Adapted Interaction. doi: 10.1023/A:1021240730564

Choi, S.-M., Lee, D., Jang, K., Park, C. & Lee, S. (2023). Improving Data Sparsity in Recommender Systems Using Matrix Regeneration with Item Features. Mathematics 2023, 11, 292. https://doi.org/10.3390/math11020292

Coppola, M., Guo, J., Gill, E. & Croon, G. (2019). The PageRank algorithm as a method to optimize swarm behavior through local analysis. Swarm Intelligence (2019) 13:277–319. https://doi.org/10.1007/s11721-019-00172-z

Daher, J.B., Brun, A. & Boyer, A. (2017). A Review on Explanations in Recommender Systems. [Technical Report] LORIA - Université de Lorraine. 2017. hal-01836639.

Davis, J. & Goadrich, M. (2006). The Relationship Between Precision-Recall and ROC Curves. Proceedings of the 23rd International Conference on Machine Learning,

Pittsburgh, PA, 2006.

Deepjyoti, R. & Mala, D. (2022). A systematic review and research perspective on recommender systems. Journal of Big Data (2022) 9:59. https://doi.org/10.1186/s40537-022-00592-5

Edmunds, A. & Morris, A. (2000). The problem of information overload in business organizations: a review of the literature. International Journal of Information Management 20 (2000) 17-28.

Farashah, M.V., Etebarian, A., Azmi, R. & Dastjerdi, R.E. (2021). A hybrid recommender system based-on link prediction for movie baskets analysis. Journal of Big Data volume 8, Article number: 32. https://doi.org/10.1186/s40537-021-00422-0

Ferreira, L., Silva, D.C. & Itzazelaia, M.U. (2023). Recommender Systems in Cybersecurity. Knowledge and Information Systems. https://doi.org/10.1007/s10115-023-01906-6

Fuller, R.M., Harding, M.K., Luna, L. & Summers, J.D. (2022). The impact of E-commerce capabilities on online retailer performance: Examining the role of timing of adoption. Information & Management, Volume 59, Issue 2, March 2022, 103584.

Gupta, A. & Pravin, S. (2023). Link Prediction based on bipartite graph for recommendation system using optimized SVD++. International Conference on Machine Learning and Data Engineering. Procedia Computer Science 218 (2023) 1353–1365.

Hamilton, W.L. (2020). Graph representation learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, Vol. 14, No. 3, Pages 1-159.

Han, C., Castells, P., Gupta, P., Xu, X. & Salaka, V. (2022). Addressing Cold Start in Product Search via Empirical Bayes. In Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21,

2022, Atlanta, GA, USA. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3511808.3557066

Harper, F. & Konstan, J. (2015). The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4: 19:1–19:19. https://doi.org/10.1145/2827872

Hendricks, S. & Mwapwele, S.D. (2023). A systematic literature review on the factors influencing e-commerce adoption in developing countries. Data and Information Management, https://doi.org/10.1016/j.dim.2023.100045

Isinkaye, F.O., Folajimi, Y.O. & Ojokoh, B.A. (2015). Recommendation systems: Principles, methods and Evaluation. Egyptian Informatics Journal Volume 16, Issue 3, November 2015, Pages 261-273. https://doi.org/10.1016/j.eij.2015.06.005

Jafari, S.H., Abdolhosseini-Qomi, A.M., Asadpour, M., Rahgozar, M. & Yazdani, N. (2021). An information theoretic approach to link prediction in multiplex networks. Scientific Reports volume 11, Article number: 13242. https://doi.org/10.1038/s41598-021-92427-1

Jain, V., Malviya, B. & Arya, S. (2021). An Overview of Electronic Commerce (e-Commerce). Journal of Contemporary Issues in Business and Government Vol. 27, No. 3, 2021. doi: 10.47750/cibg.2021.27.03.090

Kabrilyants, R., Obeidat, B.Y., Alshurideh, M. & Masa'deh, R. (2021). The role of organizational capabilities on e-business successful implementation. International Journal of Data and Network Science 5 (2021) 417–432. doi: 10.47750/cibg.2021.27.03.090

Kannaiyan, G.N., Pappula, B. & Veerubommu, R. (2020). A Review on Graph Theory in Network and Artificial Intelligence. Journal of Physics: Conference Series 1831 (2021) 012002. doi:10.1088/1742-6596/1831/1/012002

Khoo, V., Ahmi, A. & Saad, R.A. (2018). A Comprehensive Review on E-Commerce Research. Proceedings of the 3rd International Conference on Applied Science and Technology (ICAST'18). AIP Conf. Proc. 2016, 020069-1–020069-10. https://doi.org/10.1063/1.5055471

Kinsley, A.C., Rossi, G., Silk, M.J. & VanderWaal, K. (2020). Multilayer and Multiplex Networks: An Introduction to Their Use in Veterinary Epidemiology. Front. Vet. Sci. 7:596. doi: 10.3389/fvets.2020.00596

Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J.P., Moreno, Y. & Porter, M.A. (2014). Multilayer networks. Journal of Complex Networks (2014) 2, 203–271. doi:10.1093/comnet/cnu016

Kunegis, J., De Luca, E.W. & Albayrak S. (2010). The Link Prediction Problem in Bipartite Networks. Conference: Computational Intelligence for Knowledge-Based Systems Design, 13th International Conference on Information Processing and Management of Uncertainty, IPMU 2010, Dortmund, Germany, June 28 - July 2, 2010. doi: 10.1007/978-3-642-14049-5_39

Lakshmi, T.J. & Bhavani, S.D. (2021). Link Prediction Approach to Recommender Systems. arXiv:2102.09185 [cs.IR].

Li, S. & Wang, Y. (2022). Research on Knowledge Transfer on Multilayer Networks Based on Link Prediction Algorithm. Journal of Physics: Conference Series, Volume 2224, 2021 2nd International Symposium on Automation, Information and Computing (ISAIC 2021). doi: 10.1088/1742-6596/2224/1/012015

Li, X. & Chen, H. (2013). Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach. Decision Support Systems 54 (2013) 880–890.

Li, X. & Chen, L. (2011). Recommendations based on Network Analysis. ICACSIS 2011.

ISBN: 978-979-1421-11-9.

Li, X., Ng, M.K., Xu, G. & Yip, A. (2023). Multi-relational graph convolutional networks: Generalization guarantees and experiments. Neural Networks 161 (2023) 343-358.

Lü, L., Medo, M., Yeung, C.H., Zhang, Y.-C., Zhang, Z.-K. & Zhou, T. (2012). Recommender systems. Physics Reports 519 (2012) 1–49. doi:10.1016/j.physrep.2012.02.006

Lü, L. & Zhou, T. (2011). Link prediction in complex networks: A survey. Physica A 390 (2011) 1150–1170. doi:10.1016/j.physa.2010.11.027

Majeed, A. & Rauf, I. (2020). Graph Theory: A Comprehensive Survey about Graph Theory Applications in Computer Science and Social Networks. Inventions 2020, 5, 10. doi: 10.3390/inventions5010010

Najari, S., Salehi, M., Ranjbar, V. & Jalili, M. (2019). Link prediction in multiplex networks based on interlayer Similarity. Physica A 536 (2019) 120978.

Pandey, D. & Agarwal, V. (2014). E-commerce Transactions: An Empirical Study. International Journal of Advanced Research in Computer Science and Software Engineering. Volume 4, Issue 3, March 2014.

Parikshith, G. & Natesan, G. (2023). Exploring the Benefits of E-commerce Applications for Efficient Online Operations. International Journal of Scientific Research in Computer Science, Engineering and Information Technology. January-February-2023, 9 (2): 158-162. doi: 10.32628/CSEIT2390212

Qin, M. & Yeung, D.-Y. (2023). Temporal Link Prediction: A Unified Framework, Taxonomy, and Review. 1, 1 (June 2023), 47 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Rita, P. & Ramos, R.F. (2022). Global Research Trends in Consumer Behavior and Sustainability in E-Commerce: A Bibliometric Analysis of the Knowledge Structure. Sustainability 2022, 14, 9455. https://doi.org/10.3390/su14159455

RStudio Team (2020). RStudio: Integrated Development for R. RStudio, PBC, Boston, MA

URL http://www.rstudio.com/

R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/

Saeed, S. (2023). A Customer-Centric View of E-Commerce Security and Privacy. Appl. Sci. 2023, 13, 1020. https://doi.org/10.3390/app13021020

Samad, A., Qadir, M., Nawaz, I., Islam, M.A. & Aleem, M. (2020). A comprehensive survey of link prediction techniques for social network. EAI endorsed Transactions on Industrial Networks and Intelligent Systems. doi: 10.4108/eai.13-7-2018.163988

Singh, P.K., Pramanik, P.K.D., Dey, A.K. & Choudhury, P. (2021). Recommender systems: an overview, research trends, and future directions. International Journal of Business and Systems Research 15(1):14–52. doi: 10.1504/IJBSR.2021.10033303

Taher, G. (2021). E-commerce: advantages and limitations. International Journal of Academic Research in Accounting Finance and Management Sciences, 11(1), 153-165. doi: 10.6007/IJARAFMS /v11-i1/8987

Tang, R., Chen, X., Wei, C., Li, Q., Wang, W., Wang, H. & Wang, W. (2022). Interlayer link prediction based on multiple network structural attributes. Computer Networks Volume 203, 11 February 2022, 108651. https://doi.org/10.1016/j.comnet.2021.108651

Tarissan, F. (2015). Comparing Overlapping Properties of Real Bipartite Networks. ISCS 2014: Interdisciplinary Symposium on Complex Systems , 2014, Florence, Italy. pp.309-318, 10.1007/978-3-319-10759-2_32. hal-01208320.

Wang, S., Hu, L., Wang, Y., He, X., Sheng, Q.Z., Orgun, M.A., Cao, L., Ricci, F. & Yu, P.S. (2021). Graph Learning based Recommender Systems: A Review. Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21).

Wilson, R.J. (1996). Introduction to graph theory. Fourth edition. Prentice Hall. ISBN 0-582-

24993-7.

Xin, Y. (2015). Challenges in Recommender Systems: Scalability, Privacy, and Structured Recommendations. Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science and Engineering at the MASSACHUSETTS INSTITUTE OF TECHNOLOGY, June 2015.

Xue, H., Yang, L., Rajan, V., Jiang, W., Wei, Y. & Lin, Y. (2018). Multiplex Bipartite Network Embedding using Dual Hypergraph Convolutional Networks. Gaze Detection, June 03–05, 2018, Woodstock, NY. ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/1122445.1122456

Appendix A

AUC for the small example.

α \β	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1**
0.0	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.1	0.85	0.8333	0.8333	0.8333	0.8333	0.8333	0.8333	0.8333	0.8333	0.8333	0.7944
0.2	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.3	0.85	0.8333	0.8333	0.8333	0.8333	0.8333	0.8333	0.8333	0.8333	0.8333	0.7944
0.4	0.8625	0.8472	0.8472	0.8472	0.8472	0.8472	0.8472	0.8472	0.8472	0.8472	0.8069
0.5	**0.875**	0.8611	0.8611	0.8611	0.8611	0.8611	0.8611	0.8611	0.8611	0.8611	0.8194
0.6	**0.875**	0.8611	0.8611	0.8611	0.8611	0.8611	0.8611	0.8611	0.8611	0.8611	0.8194
0.7	0.85	0.8333	0.8333	0.8333	0.8333	0.8333	0.8333	0.8333	0.8333	0.8333	0.7944
0.8	**0.875**	0.8056	0.8056	0.8056	0.8056	0.8056	0.8056	0.8056	0.8056	0.8056	0.7944
0.9	0.85	0.8333	0.8333	0.8333	0.8333	0.8333	0.8333	0.8333	0.8333	0.8333	0.7944
1	0.85	0.8583	0.8583	0.8583	0.8583	0.8583	0.8583	0.8583	0.8583	0.8583	0.8194

AUC for the MovieLens dataset.

α \β	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1**
0.0	0.8496898	0.849016	0.849773	0.8489812	0.8490099	0.8494592	0.8492643	0.8494674	0.8490961	0.8492212	0.8476567
0.1	0.849853	0.849902	0.8499102	0.8499735	0.8501123	0.8504511	0.8501021	0.8496367	0.8497653	0.8505388	0.8501327
0.2	0.8501449	0.8497632	0.849951	0.8495183	0.8498469	0.8496183	0.8504511	0.850398	0.8500531	0.8494652	0.8499694
0.3	0.8501225	0.850051	0.8506123	0.8497061	0.8498061	0.8501939	0.8498714	0.8499408	0.8503347	0.8497979	0.8503572
0.4	0.8498612	0.8497714	0.8500388	0.8500388	0.8500653	0.8498286	0.8502633	0.8503654	0.8497285	0.8497857	0.8498428
0.5	0.573171	0.7273808	0.8565057	**0.9265592**	0.9294761	0.85556602	0.8505201	0.8502364	0.849651	0.8499184	0.8497673
0.6	0.8499857	0.8497469	0.8497265	0.8499857	0.8498061	0.8499775	0.8500857	0.8500245	0.8497265	0.8495018	0.8501184
0.7	0.8498265	0.8497551	0.8498592	0.8500551	0.8495816	0.8500551	0.8500245	0.8502408	0.8495346	0.849853	0.850151
0.8	0.725685	0.8521117	0.9033977	0.8522628	0.8498743	0.8500458	0.8500793	0.8504245	0.8503037	0.8496693	0.8499653
0.9	0.7957022	0.9068123	0.8559327	0.8503325	0.8497816	0.8503982	0.8498508	0.8497559	0.849775	0.8496551	0.8501163
1	0.8458613	0.8592511	0.8498785	0.8494402	0.8494096	0.8505311	0.8501731	0.8498313	0.849679	0.8503017	0.8498367

F1 score for the MovieLens dataset.

α \β	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1**
0.0	0.7631456	0.7608501	0.761756	0.7608084	0.7608427	0.7613801	0.7611469	0.76139	0.7609457	0.7610954	0.7613531
0.1	0.7613342	0.763401	0.7634108	0.7634869	0.763654	0.7640622	0.7636417	0.7630818	0.7632364	0.764168	0.7636786
0.2	0.7636934	0.763234	0.7634599	0.7629395	0.7633347	0.7630597	0.7640622	0.7639983	0.7635828	0.7628757	0.763482
0.3	0.7636663	0.7635803	0.7642567	0.7631653	0.7632856	0.7637524	0.7633641	0.7634476	0.763922	0.7632757	0.7639491
0.4	0.7633519	0.7632438	0.7635656	0.7635656	0.7635975	0.7633126	0.763836	0.7639589	0.7631923	0.763261	0.7633298
0.5	0.2561073	0.6255513	0.8258586	**0.8999308**	0.8768471	0.7712284	0.7641653	0.7638068	0.763099	0.7634206	0.7632389
0.6	0.7635017	0.7632144	0.7631898	0.7635017	0.7632856	0.7634919	0.7636221	0.7635484	0.7631898	0.7630278	0.7636614
0.7	0.7633101	0.7632242	0.7633494	0.7635852	0.7630156	0.7635852	0.7635484	0.7638089	0.7629591	0.763342	0.7637007
0.8	0.6224883	0.7667029	0.8391358	0.7668872	0.76358	0.7636343	0.763634	0.7640303	0.7638879	0.7631211	0.7634771
0.9	0.7402997	0.858574	0.7723894	0.7643972	0.7634584	0.7640646	0.7633579	0.763235	0.7632525	0.7631039	0.763659
1	0.8008191	0.7805595	0.7644706	0.7632536	0.7629981	0.7642403	0.7637471	0.7633268	0.7631371	0.7638855	0.7633224

Accuracy for the MovieLens dataset.

α \β	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1**
0.0	0.7975	0.7975	0.7975	0.7975	0.7975	0.7975	0.7975	0.7975	0.7975	0.7975	0.7965
0.1	0.7977	0.7978	0.7978	0.7978	0.798	0.7985	0.798	0.7974	0.7976	0.7986	0.7981
0.2	0.7981	0.7976	0.7978	0.7972	0.7977	0.7974	0.7985	0.7984	0.798	0.7972	0.7978
0.3	0.798	0.798	0.7987	0.7975	0.7976	0.7981	0.7977	0.7978	0.7983	0.7976	0.7984
0.4	0.7997	0.6995	0.7979	0.7979	0.798	0.6997	0.7982	0.7007	0.7975	0.7976	0.7977
0.5	0.7212	0.82	0.899	**0.9344**	0.9096	0.8076	0.7986	0.7982	0.7974	0.7978	0.7976
0.6	0.7979	0.7975	0.7975	0.7979	0.7976	0.7979	0.798	0.7979	0.7975	0.7973	0.798
0.7	0.7977	0.7976	0.7977	0.798	0.7973	0.798	0.7979	0.7982	0.7973	0.7977	0.7981
0.8	0.8183	0.8024	0.8778	0.8026	0.7983	0.7981	0.798	0.7985	0.7983	0.7974	0.7978
0.9	0.8606	0.9006	0.81	0.7996	0.7981	0.7986	0.7977	0.7976	0.7976	0.7974	0.798
1	0.8775	0.8235	0.8006	0.7982	0.7976	0.7988	0.7982	0.7977	0.7975	0.7983	0.7977

Appendix B

Layer	Nº of nodes	Average Degree	Density	Clustering Co-efficient
Action	129655	426.4967	0.7026305	0.8652497
Adventure	124650	411.3861	0.6799771	0.8551254
Animation	66892	253.8596	0.4826228	0.8321869
Children	75181	268.9839	0.48205	0.8269766
Comedy	131841	432.9754	0.7121305	0.862854
Crime	122485	406.2521	0.6748373	0.8788807
Documentary	4412	39.56951	0.178241	0.8271624
Drama	143853	471.6492	0.774465	0.8861074
Fantasy	88257	302.7684	0.5202207	0.8200815
Film-Noir	11525	96.44351	0.4052249	0.8428466
Horror	68711	256.8636	0.4810179	0.8512107
Musical	45744	194.6553	0.4150433	0.8275679
Mystery	80276	276.8138	0.4780895	0.8231026
Romance	112767	372.1683	0.6151542	0.84359
Sci-Fi	113082	373.8248	0.6189152	0.8486585
Thriller	130593	428.8768	0.7053896	0.8751154
War	90292	327.7387	0.5958885	0.8741121
Western	30473	145.4558	0.3479805	0.8149004

Appendix C

Complete list of movies of user 68 that will be recommended to user 375.

Movie title	**Genres**	**Ratings given by user 68 ≥ 4**
Heat (1995)	Action|Crime|Thriller	4
Leon: The Professional (a.k.a. The Professional) (Leon) (1994)	Crime|Drama|Thriller	4
Batman (1989)	Crime|Thriller	4
Godfather, The (1972)	Crime|Drama	4
Psycho (1960)	Crime|Horror	4.5
U.S. Marshals (1998)	Action|Crime|Thriller	4
Suicide Kings (1997)	Comedy|Crime|Drama|Mystery|Thriller	4
Few Good Men, A (1992)	Crime|Drama|Thriller	4
Office Space (1999)	Comedy|Crime	5
General's Daughter, The (1999)	Crime|Drama|Mystery|Thriller	4
Boiler Room (2000)	Crime|Drama|Thriller	4
Miss Congeniality (2000)	Comedy|Crime	4
Blow (2001)	Crime|Drama	4
Scarface (1983)	Action|Crime|Drama	5
Training Day (2001)	Crime|Drama|Thriller	4.5
Transporter, The (2002)	Action|Crime	4
Starsky & Hutch (2004)	Action|Comedy|Crime|Thriller	4.5
Man on Fire (2004)	Action|Crime|Drama|Mystery|Thriller	4
After the Sunset	Action|Adventure|Comedy|Crime|Thriller	5

Movie title	**Genres**	**Ratings given by user 68 ≥ 4**
(2004)		
Sin City (2005)	Action|Crime|Film-Noir|Mystery|Thriller	5
Proposition, The (2005)	Crime|Drama|Western	4
District 13 (Banlieue 13) (2004)	Action|Crime|Sci-Fi	4
Inside Man (2006)	Crime|Drama|Thriller	4
LucKy Number Slevin (2006)	Crime|Drama|Mystery	4
Hot Fuzz (2007)	Action|Comedy|Crime|Mystery	4
Mr. Brooks (2007)	Crime|Drama|Thriller	4.5
Bourne Ultimatum, The (2007)	Action|Crime|Thriller	4
3:10 to Yuma (2007)	Action|Crime|Drama|Western	4.5
Eastern Promises (2007)	Crime|Drama|Thriller	5
In Bruges (2008)	Comedy| Crime|Drama|Thriller	4
Dark Knight, The (2008)	Action|Crime|Drama	5
Taken (2008)	Action|Crime|Drama|Thriller	5
Slumdog Millionaire (2008)	Crime|Drama|Romance	4.5
Girl with the Dragon Tattoo, The (2009)	Crime|Drama|Mystery|Thriller	4.5
Hangover, The (2009)	Comedy| Crime	4.5
Stoning of Soraya M., The (2008)	Crime|Drama	4.5
Ninja Assassin (2009)	Action|Crime|Drama|Thriller	4.5
Inception (2010)	Action|Crime|Drama|Mystery|Sci-Fi|Thriller	5

Movie title	**Genres**	**Ratings given by user 68 ≥ 4**
Dark Knight Rises, The (2012)	Action|Adventure|Crime	4
21 Jump Street (2012)	Action|Comedy|Crime	5
Kingsman: The Secret Service (2015)	Action|Adventure|Comedy|Crime	4