UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

DELMIRO DALADIER SAMPAIO NETO

New Histogram-Based User and Item Profiles for Recommendation Systems

Recife

2021

DELMIRO DALADIER SAMPAIO NETO

New Histogram-Based User and Item Profiles for Recommendation Systems

Dissertation presented to the Graduate Program in Computer Science of the Informatics Center of the Federal University of Pernambuco, as a partial requirement for obtaining the title of Master in Computer Science.

Concentration Area: Computational intelligence.

Supervisor: Renata Maria Cardoso Rodrigues de Souza

Co-supervisor: Telmo de Menezes e Silva Filho

Recife

2021

Catalogação na fonte
Bibliotecária Nataly Soares Leite Moro, CRB4-1722

S192n Sampaio Neto, Delmiro Daladier
New histogram-based user and item profiles for recommendation systems/ Delmiro Daladier Sampaio Neto. – 2021.
73 f.: il., fig., tab.

Orientadora: Renata Maria Cardoso Rodrigues de Souza.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2021.
Inclui referências.

1. Inteligência computacional. 2. Sistemas de recomendação. 3. Dados simbólicos. 4. Histogramas. I. Souza, Renata Maria Cardoso Rodrigues de (orientadora). II. Título

006.31 CDD (23. ed.) UFPE - CCEN 2022 – 53

Delmiro Daladier Sampaio Neto

“New Histogram-Based User and Item Profiles for Recommendation Systems”

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação. Área de Concentração: Inteligência Computacional

Aprovado em: 07/12/2021.

BANCA EXAMINADORA

_____

Prof. Dr. Adriano Lorena Inácio de Oliveira
Centro de Informática / UFPE

_____

Prof. Dr. Yuri de Almeida Malheiros Barbosa
Departamento de Computação Científica / UFPB

_____

Profa. Dra. Renata Maria Cardoso Rodrigues de Souza
Centro de Informática/ UFPE
(Orientadora)

I dedicate this work to God, my family, my friends and the teachers who supported me throughout.

ACKNOWLEDGMENTS

First of all, to God, who made sure that my goals were achieved during all my years of studies.

To my supervisors, Dr. Renata Souza and Dr. Telmo Menezes for the corrections and teachings that allowed me complete this job.

To my family, for their unconditional friendship, for their support throughout the entire period of time that I dedicated myself to this work and for their patience.

To my friends for their help and who contributed in some way to this work.

ABSTRACT

Recommendation systems play an important role in businesses such as e-commerce, digital entertainment and online education. Most recommendation systems are implemented using numerical or categorical data, that is, traditional data. This type of data can be a limiting factor when used to model complex concepts where there is internal variability or internal structure in the data. In order to overcome these limitations, symbolic data are used, where values can be intervals, probability distributions or lists of values. Symbolic data can benefit recommendation systems and this work introduces a methodology to construct recommendation systems using symbolic descriptions for users and items.

The proposed methodology can be applied in the implementation of recommendation systems based on content or based on collaborative filtering. In the content-based approach, user profiles and item profiles are created from symbolic descriptions of their features and a list of items are matched against a user profile. In the approach based on collaborative filtering, user profiles are built and users are grouped to form a neighborhood, products rated by users of this neighborhood are recommended based on the similarity between the neighbor and the user who will receive the recommendation. Experiments are carried out to evaluate the effectiveness of the methodology proposed in this work in relation to existing methodologies in the literature for the two recommendation system approaches. In the experiments, it was shown that the methodology proposed in this work is able to produce ranked lists with higher quality than the methodologies in the literature, i.e., lists where items with greater relevance appear in the first positions. A movie domain dataset is used in these experiments and their results show the usefulness of the proposed methodology.

Keywords: recommendation systems; symbolic data; histograms.

RESUMO

Os sistemas de recomendação desempenham um papel importante em negócios como e-commerce, entretenimento digital e educação online. A maioria dos sistemas de recomendação são implementados usando dados numéricos ou categóricos, ou seja, dados tradicionais. Esse tipo de dado pode ser um fator limitante quando usado para modelar conceitos complexos onde há variabilidade interna ou estrutura interna nos dados. Para superar essas limitações, são utilizados dados simbólicos, onde os valores podem ser intervalos, distribuições de probabilidade ou listas de valores. Dados simbólicos podem beneficiar sistemas de recomendação e este trabalho apresenta uma metodologia para construir sistemas de recomendação usando descrições simbólicas para usuários e itens.

A metodologia proposta pode ser aplicada na implementação de sistemas de recomendação baseados em conteúdo ou baseados em filtragem colaborativa. Na abordagem baseada em conteúdo, perfis de usuários e perfis de itens são criados a partir de descrições simbólicas de seus recursos e uma lista de itens é comparada a um perfil de usuário. Na abordagem baseada em filtragem colaborativa, os perfis dos usuários são construídos e os usuários são agrupados para formar uma vizinhança, os produtos avaliados pelos usuários desta vizinhança são recomendados com base na semelhança entre o vizinho e o usuário que receberá a recomendação. Experimentos são realizados para avaliar a eficácia da metodologia proposta neste trabalho em relação às metodologias existentes na literatura para as duas abordagens de sistema de recomendação. Nos experimentos, foi mostrado que a metodologia proposta neste trabalho é capaz de produzir listas ordenadas com qualidade superior às metodologias da literatura, ou seja, listas onde os itens com maior relevância aparecem nas primeiras posições. Um conjunto de dados de domínio de filme é usado nesses experimentos e seus resultados mostram a utilidade da metodologia proposta.

Palavras-chaves: sistemas de recomendação; dados simbólicos; histogramas.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

CB-SDA	Content-based filtering using symbolic data
CBFA-SDA	Content-based filtering approach using symbolic data
CF-SDA	Collaborative filtering using symbolic data
HD-CFR	Histogram descriptions for collaborative-filtering recommendations
HD-CBR	Histogram descriptions for content-based recommendations

CONTENTS

1 INTRODUCTION

1.1 MOTIVATION

Nowadays, multiple services are running on the internet, such as e-commerce, entertainment platforms, online banking, e-learning and social networks. The massive amount of data running through these services leads to an information overload. Users have difficulty choosing items or services given the large list of options available, as stated in the work of Afsar, Crump e Far (2021). Companies also find it difficult to display the best products, items that are more compliant with the consumer's needs or even offer new items to the user. In this context, recommendation systems arise as relevant software tools used to filter information when their users have a large number of options at their disposal. These systems support decisions in various domains ranging from simple items such as books and movies to more complex items such as financial services, telecommunication equipment, and software systems (FELFERNIG et al., 2021).

With the popularization of the internet in the 1990s, recommendation systems received a lot of attention from researchers interested in applying them to tasks such as recommending movies, books or web pages. Their scope has gradually expanded since their introduction in the mid-1990s (RICCI; WERTHNER, 2006). Today, they appear as a core business item in companies of different sizes. Its most popular applications are e-commerce, digital entertainment, web pages and news, where they provide users with increasingly personalized recommendations and, consequently, increasing the online engagement, profit or another metric of interest for the companies. In their simplest form, personalized recommendations are offered as ranked lists of items. In performing this ranking, recommendation systems try to predict what the most suitable products or services are, based on the user's preferences and constraints (RICCI et al., 2011).

To perform an efficient ranking, a recommendation system must learn the relationship between items and user preferences, based on their historical interaction data, item data and implicit/explicit ratings of the items in question. Customer data includes four types of data: demographic data, classification data, behavior pattern data, and transaction data (WEI; HUANG; FU, 2021). In modern systems there is the possibility of extracting data related to the users experience when interacting with the application (behavior pattern) and the types of transactions carried out (transactions), but mostly the data are related to users (demographic

data), or they represent the characteristics of the items and scores assigned to items (ranking data). Examples of the types of data often used in recommendation systems can be found in Table 1.

Table 1 – Data types present in recommendation systems

Data Type	Explanation
Demographic Data	name, age, sex, professions, date of birth, telephone, address, hobbies, salary, educational experience and so on.
Classification Data	classification labels, such as multi-class discrete classification and continuous classification; latent comments, e.g. better, good, bad, worse and so on.
Behaviour Pattern Data	browsing duration, click times, site links; save, print, scroll, delete, open, close, web update; select, edit, search, copy, paste, mark and even web content download and so on.
Transaction Data	purchase date, purchase quantity, price, discount and so on.
Item Data	for movies or music, these can include actors or singers, topic, release date, price, brand and so on. For web documents, these can be content descriptions using keywords, links to other documents, exhibition time, topic and so on.

Source: (WEI; HUANG; FU, 2021)

Most recommendation systems work with standard data, that is, the values present in their datasets are either numeric or categorical. However, sometimes this representation is not enough to represent the true complexity of a variable. Additionally, a standard database can be too large and might need to be summarized without loss of information. It becomes a task of first importance to summarize these data in terms of their underlying concepts in order to extract new knowledge from them (DIDAY; BOCK, 2000). Due to these limitations, another type of data was defined, symbolic data, which is structured and contains internal variance. In this context, we have a rapidly increasing need to extend standard data analysis methods (exploratory, graphical representations, clustering, factorial analysis, discrimination, ...) to these symbolic data (DIDAY; BOCK, 2000)

Clearly, symbolic data brings datasets to another level of complexity, aggregating more information. Symbolic data are not only used to summarize large datasets. They also arise from many other sources. They lead to more complex data tables called symbolic data tables because a cell of such a data table does not necessarily contain just a single quantitative or categorical value, but several values which can be weighted and linked by logical rules and taxonomies (DIDAY; BOCK, 2000).

Among the aforementioned approaches, content-based filtering and collaborative filtering already have implementations using histogram-valued symbolic data to represent users and profiles in Bezerra e Carvalho (2004) and Bezerra e Carvalho (2010). The implementations already defined show that symbolic data can be employed to create better recommendation systems and they also can be improved providing better results.

However, existing implementations have some limitations. They have multiple sub-profiles for each user profile created; they use complex similarity functions with two components; and the methodologies used to build content-based systems and systems based on collaborative filtering are very different. In addition, their methodology for recommendation systems based on collaborative filters seeks to group users using only the scores assigned to items, i.e. at no point the item content is taken into account in the calculation of the similarity between users based on their item ratings. Finally, regarding the evaluation metric, existing recommendation systems that use symbolic data employ the Breese criterion (BREESE; HECKERMAN; KADIE, 2013) to evaluate the produced ranked lists. This evaluation estimates the likelihood of each item being visited by the user. This metric does not use an ideal ranked list as a reference, so items are evaluated individually. Additionally, it requires the definition of a half-life parameter α, i.e, the number of the item on the list such that there is a 50% chance the user will review the item.

Our main contributions are: (i) a single methodology for building profiles for recommendation systems using symbolic data, whether content-based or collaborative filtering approaches; (ii) our methodology makes use of a single profile for each user; (iii) we use the same dissimilarity function to calculate distances between items and users or between users; (iv) our collaborative filtering approach takes into account the content of the items when grouping users; (v) and finally, we evaluate the ranked lists using the normalized discounted cumulative gain (NDCG) (JäRVELIN; KEKäLäINEN, 2002), an information retrieval metric that assigns higher scores to systems that produce ranked lists with the most relevant items in the top positions.

1.2 OBJECTIVE

The objective of this work is to propose a single methodology to build user profiles or item profiles, using the proposed methodology to build two new recommendation system approaches using symbolic data, one for content-based systems and one for systems based on collaborative filters, which solve the previously mentioned limitations of existing implementations in the literature. To this end, both approaches will use a well-known dissimilarity function for histograms (Wasserstein distance). In addition, the content-based and collaborative filtering systems will both use a single approach to build user profiles and the calculation of similarities between users will take into account the content of the items they evaluated. These contributions aim to obtain better recommendation results than those observed in the literature, according to NDCG.

More specifically, this works aims to:

- Propose a new approach to build user profiles and item representations;
- Apply distances between probability distributions to calculate the dissimilarity between users and items;
- Employ symbolic data to build a content-based recommendation system;
- Apply distances between probability distributions to calculate the dissimilarity between pairs of user profiles;
- Employ symbolic data to build a collaborative filter recommendation system;
- Apply the NDCG metric instead of Breese Criterion, in order to better evaluate ranked lists.

1.3 METHODOLOGY

The methodology applied to carry out this work is formed by the following points:

- Literature review on recommendation systems, symbolic data analysis and recommendation systems using symbolic data;
- Pre-processing of data in the Movielens movie database;

- Creating user and item profiles;
- Implementation of a content-based recommendation system;
- Implement an appropriate similarity/dissimilarity metric to compare items and profiles and compare profiles to each other.
- Implementation of a collaborative-filter for recommendations;
- Evaluation of the results using the appropriate metrics, NDCG.

1.4 EXPECTED RESULTS

Implement recommendation systems, content-based and collaborative filters, using a new approach to create user and item profiles. Such systems must have a satisfactory performance for the NDCG metric, i.e., they must achieve a NDCG score equal or greater than the existing implementations in the literature.

1.5 STRUCTURE OF THE DOCUMENT

This dissertation comprises this introductory chapter and four more chapters. In Chapter 2, the theoretical foundation of the recommendation systems and symbolic data is presented; Chapter 3 presents the methodology, which will explain the entire process of creating profiles, items and definitions of dissimilarity metrics for recommendation systems. In Chapter 4, the analysis of the obtained results, evaluation of the proposed model and comparison with existing methods in the literature is carried out. Finally, in Chapter 5 the conclusions and final considerations will be presented, presenting future works.

2 LITERATURE REVIEW

2.1 RECOMMENDATION SYSTEMS

The decision-making process is an activity of great relevance in business and in everyday situations. Due to the overwhelming amount of options to choose, or how economically risky a choice might be, or which channel is more appropriate for communication or simply which item is most relevant to a particular type of user, the decision-making process could be costly, economically or with respect to time. This myriad of available choices is directly related to advances in the internet, mobile devices and IoT (Internet of Things) devices. However, the geometric growth of data makes it difficult for users to find information that meets their own needs in time, so "big data" leads to "information overload" problem, and makes a lot of irrelevant redundant information interfere with users' choice (CHEN et al., 2018).

The information overload is present since early 1990, with users being exposed to a huge load of emails and content. In order to solve problems generated from this exposition, in 1992, researchers at Xerox Palo Alto Research Center (PARC) developed an information filtering system called Tapestry (GOLDBERG; NICHOLS; TERRY, 1992). This system allowed users to filter electronic documents based on their content and also based on annotations provided by other users, a collaborative filtering. These information filtering systems came to be known as recommendation systems.

Mahmood e Ricci (2009) formally defined recommendation systems as intelligent applications which assist users in their information-seeking tasks, by suggesting those items (products, services, information) that best suit their needs and preferences. In the next years, recommendation systems were a constant interest of researchers, being employed in a variety of domains and developed with different approaches.

Tech giants such as Amazon, Facebook, Netflix and Google make use of recommendation systems in their businesses. These applications are beneficial for both users and companies. Users are aided to perform more assertive choices or purchases and companies can maximize their sales and retain customers. Moreover many media companies are now developing and deploying recommendation systems as part of the services they provide to their subscribers (RICCI et al., 2011). In addition, the subject has specialized conferences and journals dedicated exclusively to it.

Designing and implementing a recommendation system can be a complex task as the field

comprises many other fields of knowledge. Designing and developing recommendation systems is a multi-disciplinary effort that has benefited from results obtained in various computer science fields especially machine learning and data mining, information retrieval, and human-computer interaction (RICCI et al., 2011). Furthermore, recommendation systems must handle multiple access and availability when in a production environment.

Recommendations systems, in order to identify relevant items for a given user, work mostly on user or item data. User data can be used to identify similar users and perform a social approach to perform recommendations. Item data can be used to recommend similar products that were selected before. In addition, recommendation systems can use other kinds of data depending on the application domain, for example clicks, products added to cart, ratings or written reviews. Therefore, due to this sensitive data usage, this topic is also related with ethics and privacy issues. Therefore, there is a need to design solutions that will parsimoniously and sensibly use user data. At the same time these solutions will ensure that knowledge about the users cannot be freely accessed by malicious users (RICCI et al., 2011).

Recommendations can be built using different approaches depending on how the information is filtered in the input data. The most famous approaches are content-based filtering, collaborative filtering and hybrid filtering methods. Each of them has its own strengths and flaws.

2.1.1 Content-Based Filtering

The content-based filtering approach has its origins in information retrieval and information filtering (WEI; HUANG; FU, 2021). The main reasoning for this approach is the fact that items similar to those evaluated previously, frequently, have similar utility to a specific user. Following this idea, the methodology focuses on the items' contents and their properties to understand the relations between users and items.

Formally, let Content(i) be the information from item i, which can be interpreted as a table row describing an item. It is usually computed by extracting a set of features from item i (its content) which are used to determine the appropriateness of the item for recommendation purposes (ADOMAVICIUS; TUZHILIN, 2005). The information extracted can be textual, numerical or categorical. This information of previously evaluated items is used to build the user's profile.

The profile is a structured representation of user interests, adopted to recommend new interesting items (RICCI et al., 2011). To perform this task, classical techniques such as Neural

networks, clustering, decision trees and TF-IDF can be employed. Once the profile or model is built, the system can match new items to a specific user.

Let Profile(u) be the calculated profile for a user $\mathbf{u}$. The profile can be created by aggregating the features of each item evaluated before, according to the ratings received, i.e, items with better ratings have more relevance for a profile. The profile can also be visualized as a row of a table. Having the profile of a user, scores can be assigned to the other items in a database. So, the scoring function for an item $\mathbf{i}$ and user $\mathbf{u}$ is as follows:

$$\text{score}(u, i) = \text{Similarity}(\text{Content}(i), \text{Profile}(u)) \tag{2.1}$$

The Similarity function used in the scoring process can be either a similarity or a dissimilarity function. If the former is used, a higher value is desirable and if the latter, a lower value is desirable, as long as the metric works for two vectors as inputs. Often, cosine similarity (given by 2.2) or Euclidean distance (given by 2.3), which are widely used metrics, are used for this activity.

$$\text{cosine}(u, i) = \frac{u \cdot i}{\|u\| \times \|i\|} \tag{2.2}$$

$$\text{euclidean}(u, i) = \sqrt{(u_1 - i_1)^2 + ... + (u_n - i_n)^2} \tag{2.3}$$

In order to make the content-based recommendation concepts easier to understand, item information, user ratings and the user profile creation process are presented. As an example of item content, Table 2 shows items information for a book recommendation system.

Another fundamental component in this recommendation system is the evaluations provided by the system users,i.e., the ratings given by the users to the books. The ratings provided for this example are displayed in Table 3. The ratings are on a scale from 1 to 5.

Taking user 1 as an example, the books previously evaluated by this user and their respective ratings are shown in Table 4. These items are used to build user 1's profile for future recommendations. These can be used to build a regression model, classification model or can be grouped, creating a cluster.

The remaining items in the base will be matched against the profile created. Items are evaluated in terms of how adherent they are to user preferences, i.e., how similar they are to the profile created. Taking the book recommendation task as an example, a possible ranked list generated for user 1 is showed in Table 5.

Table 2 – Relevant information about a set of books

book id	title	author	genre
1	Frankenstein	Mary Shelley	Horror, Classics
2	Foundation	Isaac Assimov	Science Fiction
3	Books of Blood	Clive Barker	Horror
4	Fluent Python	Luciano Ramalho	Tech, Python
5	Eloquent Javascript	Marijn Haverbeke	Tech, Javascript
6	The Exorcist	William Peter Blatty	Horror
7	Dracula	Bram Stoker	Horror, Classics
8	The War of the Worlds	H. G. Wells	Science Fiction
9	The Invisible Man	H. G. Wells	Science Fiction
10	Brave New World	Aldous Huxley	Science Fiction
11	It	Stephen King	Horror
12	Interview with the Vampire	Anne Rice	Horror
13	Neuromancer	William Gibsons	Science Fiction
14	Dune	Frank Herbert	Science Fiction
15	Effective Python	Brett Slatkin	Tech, Python

Source: The author (2021)

This approach produces significant results when applied, however, some inherent issues can affect the quality of the recommendations provided by the system. Cold start, overspecialization and limited content analysis are recurrent problems in content-based recommendation systems.

Cold start, in content-based filtering, refers to problems in providing initial recommendations. The cold-start problem occurs when it is not possible to make reliable recommendations due to an initial lack of ratings. We can distinguish three kinds of cold-start problems: new community, new item and new user (BOBADILLA et al., 2012). For the content-based filtering approach the types of problems that occur are the new user and the new community inserted into the system, due to the lack of evaluated items. New items do not present problems in this approach, as they can appear in recommendations even without previous evaluations.

Overspecialization is a problem related to the limitation of content-based filtering in displaying recommendations with some novelty. This is also called serendipity problem. Since recommendation systems try to maximize an evaluation metric, such as accuracy, they have no prior knowledge on items that could be unexpectedly satisfactory for the user and, consequently, end up recommending very similar items. Additionally accuracy-based algorithms limit the number of items that can be recommended to the user, which lowers user satisfac-

Table 3 – Relevant information about a set of books

User id	book id	rating
1	1	5
1	6	5
1	2	2
1	4	4
1	11	4
2	7	4
2	6	5
2	11	5
2	5	5
3	4	5
3	5	4
3	3	5
4	2	5
4	10	5
4	7	4
5	13	4
5	14	5
5	1	2
5	12	3

Source: The author (2021)

Table 4 – Relevant information about a set of books

book id	title	author	genre	ratings
1	Frankenstein	Mary Shelley	Horror, Classics	5
2	Foundation	Isaac Assimov	Science Fiction	2
4	Fluent Python	Luciano Ramalho	Tech, Python	4
6	The Exorcist	William Peter Blatty	Horror	5

Source: The author (2021)

tion (KOTKOV; VEIJALAINEN; WANG, 2016)

Another factor that brings potential harm to content-based filtering is the limitation created by the data available. This means the system cannot make good suggestions when the data is insufficient to discriminate the items. For example, content-based movie recommendation can only be based on written materials about a movie: actors' names, plot summaries, etc,

Table 5 – Relevant information about a set of books

book id	title	author	genre
3	Books of Blood	Clive Barker	Horror
7	Dracula	Bram Stoker	Horros, Classics
11	It	Stephen King	Horror
5	Interview with the Vampire	Anne Rice	Horror
8	Effective Python	Brett Slatkin	Tech Python

Source: The author (2021)

because the movie itself is opaque to the system (BURKE, 2002)

Subsequent researches brought new strategies to overcome some of these problems. Innovations in methodologies similar to Sheth e Maes (1993), which uses genetic algorithms to improve the personalization, the approach developed by Zhang, Callan e Minka (2002), which avoids redundancy in subsequent recommendations.

It is also worth mentioning the challenges that impact the recommendation systems area as a whole such as scalability. The scalability issue concerns the behavior of the system in the real world, handling multiple requests, dealing with real data and having a short time to perform the recommendations. In order to manage the vast increase in number of users and items, a trade-off between prediction performance and scalability is inevitable (ALMAZO et al., 2010).

There are also privacy issues, which are due to the usage of personal data to perform successful recommendations. In general, the more information individuals have about their recommendations, the better they will be able to evaluate those recommendations. However, people may not want their habits or views to be widely known (RESNICK; VARIAN, 1997).

2.1.2 Collaborative Filtering

Differently from content-based filtering, collaborative filtering considers that items that were well rated by similar users can be useful for a particular user, i.e this approach is focused on the relationship between users in a database or group of users U. A target user is matched against the database to discover neighbors, who historically, had similar interests to the target user. Items that neighbors liked are then recommended to the target user (LI; KIM, 2003).

The collaborative-filtering approach can overcome some problems related to lack of data and novelty in recommendations of content-based filtering, as stated by Ricci et al. (2011), e.g,

the active user can receive recommendations of unexpected products if their closest neighbors rated it as useful.

This filtering strategy can be achieved in two main categories of implementation: heuristic-based (memory-based) and model-based. Heuristic-based algorithms operate over the entire user database to make predictions. Model-based collaborative filtering, in contrast, uses the user database to estimate or learn a model, which is then used for predictions (BREESE; HECKERMAN; KADIE, 1998).

Model-based techniques tackle the problem as a regression or classification problem, depending on the rating system, e.g, the model has to either estimate scores or ratings for products, given the user. For this category, clustering methods, Bayesian networks, artificial neural networks or linear regressions are the most used methods.

In heuristic-based techniques, the estimated score s_{iu} attributed to an item *i* by a user *u* is calculated as a weighted sum of the individual votes of other users in the base (or a selected number of closest users, called neighborhood), as shown in Equation (2.4), where $w(v,u)$ are the weights for the other users with respect to the user u, $\overline{s}_u$ is the average score given by user u, k is a scaling factor, s_{vi} is the score given by user v to item *i* and $\overline{s}_v$ is the average score given by the user v.

$$s_{iu} = \overline{s}_u + k \sum_{v=1}^{n} w(v,u) * (s_{vi} - \overline{s}_v) \tag{2.4}$$

The weights calculated for the other users express relationships between the current user and others. The stronger the relationship, the higher the weight. The weights $w(v,u)$ can reflect distance, correlation, or similarity between each user v and the active user (BREESE; HECKERMAN; KADIE, 1998). Depending on the chosen strategy, the weights can be calculated using different expressions.

An alternative approach to solve the problem is using latent-factor techniques, which use latent variables to explain the ratings (observed variables) in a user-item matrix. Latent factor models are an alternative approach that tries to explain the ratings by characterizing both items and users on a number p of factors inferred from the ratings patterns (KOREN; BELL; VOLINSKY, 2009). Salakhutdinov, Mnih e Hinton (2007) implemented this approach using stochastic neural networks. Hofmann (2004) employs Probabilistic Latent Semantic Analysis (pLSA) to do a similar approach. Finally, the Latent Dirichlet Allocation (LDA), described by Blei, Ng e Jordan (2003), can also be employed to perform this task.

In order to clarify the collaborative-filtering recommendation concepts, item information, user ratings and the user profile creation process are presented. Again, a book recommendation system is used as an example. Item data is displayed in Table 2 and the ratings in Table 3.

In this approach, the key points are the similarities between the users. By inspecting user 1's ratings, it's clear they prefer books the horror genre followed by technical books. In order, the most similar users to the one in question are user 2, user 3, user 4 and user 5. Therefore, these users have a decreasing influence on the recommendations provided to user 1. A possible ranked list provided to user 1 is displayed on Table 6.

Table 6 – Relevant information about a set of books

book id	title	author	genre
7	Dracula	Bram Stoker	Horros, Classics
3	Books of Blood	Clive Barker	Horror
11	It	Stephen King	Horror
3	Books of Blood	Clive Barker	Horror
8	Effective Python	Brett Slatkin	Tech Python

Source: The author (2021)

Despite the efficiency and bringing some improvements over the problems presented by content-based filtering, collaborative filtering has its own problems. The main difficulties faced by them are cold start and sparsity.

The cold start problem affects both new users and new items as well. New items are affected as the system performs recommendations based in evaluations from other users, so these items lack significant evaluations. Therefore, until the new item is rated by a substantial number of users, the recommendation system would not be able to recommend it (ADOMAVICIUS; TUZHILIN, 2005). With respect to the new user problem, it works in the same way as content-based filtering.

Sparsity is the problem related to the fact that users frequently rate a subset of the items in the database. Consequently, the capacity of providing novel recommendations is affected. Users usually rate a small fraction of available items, meaning the RS would have insufficient rating data to cluster and, therefore, the quality of recommendations would be compromised (SILVA; JUNIOR; CALOBA, 2018).

Other works also increased the performance of collaborative-filtering techniques, making this approach useful in many scenarios. For example, the usage of Singular Value Decomposition (SVD) to find latent factors attracted a lot of attention in the Netflix Prize, due to

the ability to handle huge and sparse datasets and the capability of using implicit and explicit feedbacks. As the Netflix Prize competition has demonstrated, matrix factorization models are superior to classic nearest-neighbor techniques for producing product recommendations, allowing the incorporation of additional information such as implicit feedback, temporal effects, and confidence levels (KOREN; BELL; VOLINSKY, 2009).

2.1.3 Other approaches for recommendation systems

Another well know approach involves hybrid recommendation systems, which combine multiple techniques to overcome individual weaknesses. Hybrid recommendation systems put together two or more strategies with the goal of reinforcing their advantages and reducing their disadvantages or limitations as stated in Çano e Morisio (2019)

Another example of recommendation systems is knowledge-based recommendation systems. These handle knowledge about users and items in order to provide recommendations that fulfill user needs. Knowledge-based approaches are different in that they have functional knowledge: they have knowledge about how a particular item meets a particular user need, and can therefore reason about the relationship between a this need and a possible recommendation (BURKE, 2002).

Besides the aforementioned approaches to build recommendation systems, there are others that can be used depending on the problem, how the recommendations are displayed, the dataset size or the type of information available.

2.2 SYMBOLIC DATA ANALYSIS

Traditionally, data is represented using single numeric or categorical values. This type of representation, sometimes, is not enough to describe the complexity of some real world concepts. Also when dealing with aggregated data, classical data representation cannot represent internal variations or structural patterns that may be useful for the researchers. To overcome these limitations, symbolic data and symbolic data analysis were introduced. According to Diday (2003) the first is a type of data that contains internal variation and are structured. In Diday e Noirhomme-Fraiture (2008) the authors aim of symbolic data analysis is to generalize data mining and statistics to higher-level units described by symbolic data.

Instead of single valued representation, symbolic data can be represented by data distribu-

tions, intervals, lists or structured data. When a company performs clustering analysis of its customers data, for example, the resulting clusters can be interpreted as customer category. On each cluster, columns (or features) of customers can be expressed as one of the data types mentioned, e.g, the customer age of a cluster can be expressed as a interval or a data distribution. Hence, when a description of concepts, classes or categories of such complex objects is required, symbolic data can be used (DIDAY; NOIRHOMME-FRAITURE, 2008).

Symbolic data can be found when a large database is being aggregated, in order to perform analysis in different "resolutions". As an example, when a company decides to aggregate the customers database to work with data at a regional level. The company needs to use symbolic data to avoid loosing information about internal variation and structures.

2.2.1 Data Types

Again, supposing that a nationwide company has in its database data from customers across the country. The information recorded for each customer is defined by the columns: Id, Gender, City, Region, Age, Annual Expense. The data descriptions for each column are displayed in Table 7.

Table 7 – Database columns

Y_i	**Description**	**Possible Values**
Y_1	Id	≥ 0
Y_2	Gender	male, female, non-binary, NA
Y_3	City	
Y_4	Region	
Y_5	Age	≥ 18
Y_6	Annual Expense	≥ 0 (In thousands)

Source: The author (2021)

The data presented in Table 8 are represented using traditional data pattern, each row represent a customer (individual). Some problems can happen when there are a lot of rows and/or columns in the database. Some aggregations need to be done to extract information more accurately. For example, studying the Age distribution in one of the regions to to define the marketing strategy, or to segment the purchase history to improve the product distribution.

Frequently, when performing analysis it's interesting to understand how specific groups of our dataset behave. Each of these groups are formed by several individuals, therefore, they

have internal variability and a structure and can be defined as symbolic variables. These groups work as an example of high-level variables.

Taking the marketing campaign example, aggregation by region, the new units in the table represent each of regions, Table 9. In the original table, there are ten customers from region NE, seven for region SE, eleven for region N and five for region CO. As shown in table, the values can be presented in different ways, depending on the need or the concept to be expressed. Using the values for age for Northeast region (NE), they are [47, 24, 18, 38, 66, 32, 58, 41, 30, 64] these values can be represented by a interval-value variable AGE, such that AGE(NE)=[18,66], the categorical variable gender can be represented by a modal-valued variable GENDER, such that GENDER(NE)= {M(4/10); F(4/10); NB(2/10)}. Each one of this ways to express a symbolic variable is a specific type of symbolic data.

For further explanations, this work adopted the following notation. The value for a classical variable Y_j, with j in 1, ..., p, for the individual i, for i in 1, ...,n, will be denoted as x_{ij}. The symbolic variables are denoted by ξ_{ij}. That is $Y_{ij} = x_{ij}$ is a classical variable and $Y_{ij} = \xi_{ij}$ is a symbolic variable.

2.2.1.1 Multi-valued symbolic variables

As the name states, this type of variables can assume one or more values given a domain. In Diday e Billard (2006), multi-values symbolic variables are defined as is one whose possible value takes one or more values from the list of values in its domain $\mathcal{Y}$. The complete list of possible values in $\mathcal{Y}$ is finite, and values may be well-defined categorical or quantitative values.

For the customer values in the table 3, the values for city (Y_3) in the region NE can be described as a multi-valued variable. Formally, we have:

$$Y = \xi_3 = \{Recife, Salvador, Natal, Fortaleza, Aracajú, Teresina\} \tag{2.5}$$

That is, each of the individuals belonging to the NE region are located in one of the cities listed above.

2.2.1.2 Interval variables

In this type of variable, the possible values are within a interval, open or closed, given a certain domain $\mathcal{R}$. Formally, we have:

$$Y = \xi_3 = |a, b| \subset \mathcal{R}^1, where, a < b, a, b \in \mathcal{R}^1 \tag{2.6}$$

Using the table 3 as an example, the values for Age (Y_5) and Annual Expense (Y_6) in the region NE can be described as a interval variable. The intervals for Age and Annual Expense are shown below.

$$Y = \xi_5 = [18, 66] \tag{2.7}$$

$$Y = \xi_6 = [3.6, 180] \tag{2.8}$$

That is, the age of individuals from region NE are in the closed interval [18, 66].

2.2.1.3 Modal variables

Modal variables are variables that have weights, probabilities or frequencies associated with a list of values from a specific domain. As formally defined in Diday e Billard (2006), let Y_u be the random variable that assume the values η_k; k = 1, 2, ..., over a domain $\mathcal{Y}$. Then, the outcome is symbolic and modal valued if it takes the form:

$$Y_u = \xi_u = \{\eta_k, \pi_k; k = 1, ..., s_u\} \tag{2.9}$$

Where η_k is a value from the dominion $\mathcal{Y}$ and π_k is a non negative value associated with the value η_k. The measures π_k are typically weights, probabilities or relative frequencies corresponding to the respective outcome component η_k. However, they can also be capacities, necessities, possibilities, credibility, and related entities (DIDAY; BILLARD, 2006).

As an example, in the table 3, the values for the Gender variable (Y_2) in the region Ne can be described as a modal variable. The values and the weights associated are shown below:

$$Y_2 = \xi_2 = [M(4/10); F(4/10); NB(2/10)] \tag{2.10}$$

Table 8 – Original data

Y_1	Y_2	Y_3	Y_4	Y_5	Y_6
0	Male	Recife	NE	47	60
1	Female	Belém	N	36	84
2	Male	Natal	NE	24	36
3	Male	Curitiba	S	50	120
4	Female	Cuiabá	CO	47	140
5	Female	São Paulo	SE	60	96
6	Non-binary	Manaus	N	25	8.4
7	Female	Fortaleza	S	40	108
8	Female	Recife	NE	18	3.6
9	Non-binary	Belém	N	55	240
10	Male	Salvador	NE	38	144
11	Female	Recife	NE	66	96
12	Male	Macapá	N	34	48
13	Non-binary	Rio de janeiro	SE	27	72
14	Non-binary	Aracajú	NE	32	12
15	Male	São paulo	SE	67	36
16	Male	Belo Horizonte	SE	17	0
17	Female	Natal	NE	58	180
18	Male	Belém	N	26	84
19	Female	Fortaleza	NE	41	24
20	Male	Manaus	N	31	120
21	Male	Rio de Janeiro	SE	25	8.4
22	Female	Porto Alegre	S	19	6
23	Male	Florianópolis	S	20	6
24	Female	Curitiba	S	42	96
25	Non-binary	Teresina	NE	30	120
26	Male	Rolim de Moura	N	37	180
27	Female	Rio Branco	N	29	9.6
28	Male	Goiânia	CO	48	240
29	Male	Cuiabá	CO	34	36
30	Female	Campo Grande	CO	26	48
31	Female	São Paulo	SE	51	108
32	Female	Belo Horizonte	SE	28	24
33	Male	Recife	NE	64	120
34	Male	Manaus	N	41	120
35	Female	Goiânia	CO	23	0
36	Male	Palmas	N	33	84
37	Non-binary	Palmas	N	37	108
38	Male	Porto Alegre	S	29	72
39	Female	Curitiba	S	39	100

Source: The author (2021)

Table 9 – Symbolic table describing the concept Region

w_u	**GENDER**	**AGE**	**ANNUAL EXPENSE**
NE	[M(4/10); F(4/10); NB(2/10)]	[18, 66]	[3.6, 180]
CO	[M(2/5); F(3/5); NB(0/10)]	[23, 47]	[23, 48]
SE	[M(3/7); F(3/7); NB(1/7)]	[17, 67]	[8.4, 108]
N	[M(6/11); F(3/11); NB(2/11)]	[25, 55]	[8.4, 180]
S	[M(3/7); F(4/7); NB(0/7)]	[19, 50]	[6, 120]

Source: The author (2021)

2.3 RECOMMENDATION SYSTEMS USING SYMBOLIC DATA

This section will cover approaches to creating and evaluating recommendation systems using symbolic data. The approaches will be presented in chronological order of development.

2.3.1 First approach: Content-based recommendation systems using symbolic data

Although recommendation systems, in their majority, are built and maintained with classical data, Bezerra e Carvalho (2004) presented the first approach of recommendation systems employing symbolic data. The approach developed in the cited work is focused in content-based recommendation systems, in this work called Content-based filtering approach using symbolic data (CBFA-SDA). With respect to the symbolic variables, the system uses modal variables to represent the items and user profiles are build by aggregating the items rated by each user.

In the remainder of this section, the implementation developed in Bezerra e Carvalho (2004) will be detailed, focusing on the formal definitions of the concepts.

The process is divided in three main steps:

1. Construction of a symbolic user profile
2. Comparison between the user profile and items to be recommended
3. Ranked list generation based on the scores given by the system

The focus in the first step is to build a user profile using symbolic modal variables. Since this is a content-based approach, the profile is built using symbolic descriptions and grades from every item previously rated by a particular user. This process can also be divided in two steps:

- Pre-processing
- Generalization

The pre-processing step is responsible for creating the symbolic descriptions for each item in the dataset. These descriptions are created in order to build the profile and to perform comparisons between items and profiles.

For an item $\mathbf{x}_i$, its description can be expressed as $\mathbf{x}_i = (X_i^1, ..., X_i^p, C(i))$, where its variables X_i^j are such that $X_i^j \subseteq D_j$, with D_j being the domain of all possible values that the variable can assume, $j \in [1, ..., p]$, and $C(i)$ belongs to the set of possible ratings. For each category $m \in X_i^j$, the weight w is attributed as follows in Equation (2.11):

$$w(m) = \begin{cases} \frac{1}{|X_i^j|}, \text{if the variable is single or multi-valued} \\ \frac{f(m)IDF(m)}{\sum_{m \in X_i^j} f(m)IDF(m)}, \text{if the variable is textual.} \end{cases} \tag{2.11}$$

For textual features, the weights $w(m)$ are attributed to each word in the text. The term $f(m)$ in the equation above is the word frequency and the term $IDF(m)$ is the inverse document frequency of the m-th word. The item x_i has a symbolic description such that, $\xi_i = (\xi_i^1, ..., \xi_i^p, C(i))$ where $\xi_i^j = (\eta_j, \pi_j; j = 1, ..., p)$ with η_j is a value from the dominion † and π_j are the weights associated to η_j

As an example, we show the pre-processing step applied to the movie domain. Table 10 shows the classical representation for a single movie.

Table 10 – Classical representation for a movie domain

Variable	**Variable type**	**Description**
Director	Single-valued	D_1
Cast	Multi-valued	$\{A_1, A_2\}$
Synopsis	Textual	Residents of Bacurau, a small town ...
...	...	...
Grade	Single valued quantitative	5

Source: The author (2021)

Table 11 shows the symbolic representations resulting from the pre-processing step for the item described previously by Table 10.

Table 11 – Symbolic representation for movie domain

Variable	**Description**
Director	$(\{D_1\}, \{1.0\})$
Cast	$(\{A_1, A_2\}, \{0.5, 0.5\})$
Synopsis	({Residents, of, Bacurau, a, small}, {0.001, 0.001, 0.2, 0.001, 0.03 })
...	...
Grade	5

Source: The author (2021)

Next, is the generalization step. The main objective in this step is to use the symbolic descriptions of the items, generated in the previous step, to create suitable descriptions of user behavior, i.e. the profiles. Items with good and bad ratings are used to create two sub-profiles, u_+ and u_- respectively, which compose the final user profile. For a given user, the ratings of each item are used to differentiate between good and bad items, such that, good items are those with ratings 5 or 4 and bad items are those with ratings 1 or 2.

Formally, let σ be the set of sub-profiles for a given user, e.g, $\sigma \in \{+, -\}$. Now, let $\xi_{u_\sigma} = (\xi^1_{u_\sigma}, ..., \xi^p_{u_\sigma})$ be the symbolic modal description of the sub-profile u_σ, where $\xi^j_{u_\sigma} = (\eta_j, \pi_k; j = 1, ..., p)$, where η_j is a value from dominion $\mathcal{Y}$ and π_j are the weights associated to η_j.

Formally, the construction of the user profile is as follows. Let $\xi_i = (\xi^1_i, ..., \xi^p_i, C(i))$ be the symbolic description of an item i. The symbolic modal description for the item $i \in u_\sigma$ also contains the set of values η_i, formed according to Equation (2.12):

$$\eta_{u_\sigma} = \bigcup_{i \in u_\sigma} \eta_{u_i} \tag{2.12}$$

Let $m \in \eta_{u_\sigma}$ be a category from the dominion $\mathcal{Y}$. The weight associated with this category is $W(m)$, such that $W(m) \in \pi_{u_\sigma}$, is calculated as follows in Equation 2.13:

$$W(m) = \frac{1}{|u_\sigma|} \sum_{i \in u_\sigma} \delta(i, m) \tag{2.13}$$

$$\delta(i, m) = \begin{cases} w(m) \in \pi_j, \text{if } m \in \eta_j \\ 0, \text{Otherwise.} \end{cases} \tag{2.14}$$

Where $|u_\sigma|$ is the number of categories in the set u_σ. As an example, suppose that the positive profile of a user, u_+, is formed by two movies, the variable Cast is defined as shown in Table 12. Next the symbolic description of the positive sub-profile (u_+) is shown in Table 13.

Table 12 – Movies in the positive sub-profile u_+

Variable	**Movie 1**	**Movie 2**
Cast	($\{A_1, A_2\}$, {0.5, 0.5})	($\{A_3, A_2\}, \{0.5, 0.5\}$)
Grade	5	4

Source: The author (2021)

After the symbolic description for each profile variable is created, it is necessary to determine how each item will contribute with the profile, based on the scores given previously. Notably, items with a higher score should have more influence on the positive profile u_+ and lower scores tend to have stronger influence on the negative profile u_- of a user. In order to address this issue the authors, using a rating system from 1 to 5, repeated movies with ratings 5 and 1 three times and for ratings 4 and 2 two times. The items with ratings 3 were left out of the profile since they confuse the system.

Table 13 – Symbolic representation for movie domain

Variable	User sub-profile (u_+)
Cast	($\{A_1, A_2, A_3\}$, {0.25, 0.5, 0.25})
...	...

Source: The author (2021)

The last step in the process is the comparison between the items and the profiles created. The comparison is performed using a dissimilarity function that takes into account each sub-profile. For this measure, an item needs to have small dissimilarity with the positive sub-profile (u_+) and a high dissimilarity with the negative sub-profile (u). This function also can take into account the order of a symbolic modal variable, if it is ordered.

Let the symbolic modal description of an item i be $i = (\xi^1, ..., \xi^p)$, where $\xi^j = (\eta_j(u), \pi_j(u))$, j = 1, ..., p and $y_{u_\sigma} = (Y^1_{u_\sigma}, ..., Y_{u^p_\sigma})$, where $Y^p_{u_\sigma} = (\eta_j(y_{u^p_\sigma}), \pi_j(y_{u^p_\sigma}))$, the modal symbolic description of the sub-profile u_σ , $\sigma \in \{+, -\}$. The comparison between the item u and the profile y is given by the dissimilarity function given by Equation (2.15).

$$\Phi(i, y) = \frac{(1 - \phi(i, y_{u_-})) + \phi(i, y_{u_+})}{2} \tag{2.15}$$

Function $\phi(i, y_{u_\sigma})$ is formed by two components: a context free component, which compares the sets of categories $(\eta_j(u), \eta_j(y_{u^p_\sigma}))$, and a context dependent component which compares the weight distributions $q_j(u), q_j(y_{u_\sigma})$. The function $\sigma(u, y_{u_\sigma})$ is defined as follows.

$$\phi(i, y_{u_\sigma}) = \frac{1}{p} \sum_{j=1}^{p} [\phi_{cf}(\eta_j(i), \eta_j(y_{u^p_\sigma})) + \phi_{cd}(q_j(i), q_j(y_{u_\sigma}))] \tag{2.16}$$

In detail, the context dependent component ϕ_{cd} can be formally defined as:

$$\phi(i, y_{u_\sigma}) = \frac{1}{2}(\frac{\gamma + \delta}{\alpha + \gamma + \delta} + \frac{\gamma + \delta}{\beta + \gamma + \delta}) \tag{2.17}$$

Where α and β represent the agreement and δ and γ the discordance between the weight distributions, based on the weights associated to each class *m* calculated for the items, *w(m)*, and for the profiles *W(m)*. Their calculation is show on Table 14.

Table 14 – Symbolic representation for movie domain

	+(Agreement)	-(Disagreement)
+	$\alpha = \sum_{m \in \eta_u} \cap \eta_{y_\sigma} w(m)$	$\gamma = \sum_{m \in \eta_u \cap \overline{\eta_{y_\sigma}}} W(m)$
+	$\beta = \sum_{m \in \eta_u} \cap \eta_{y_\sigma} W(m)$	
-	$\delta = \sum_{\overline{m \in \eta_u} \cap \eta_{y_\sigma}} w(m)$	

Source: The author (2021)

The context free component ϕ_{cf} is defined as:

$$\phi_{cf}(\eta_j(u), \eta_j(z)) = \begin{cases} 0, \text{if } \eta_j(u) \cap \eta_j(z) \neq 0, \\ \frac{|\eta_j(i) \oplus \eta_j(y_\sigma)| - |\eta_j(z)| - |\eta_j(u_\sigma)|}{|\eta_j(u) \oplus \eta_j(z)|}, \text{otherwise.} \end{cases} \tag{2.18}$$

The experimental evaluation implemented in this work used the well known dataset EACH-MOVIE database, which was shut down in October 2004. A new dataset called Movielens was created based on the previous one (GROUPLENS, 2022).

2.3.2 Bringing other approaches to the universe of symbolic data

In the following work, Bezerra e Carvalho (2010) expanded the applications of symbolic data for recommendation systems and also improved the methodology developed for content-based recommendation systems. The innovation, specifically in the case of content-based system, resides in the fact that there is a new dissimilarity function and multiple sub-profiles that comprise all interest levels (ratings) for each user, therefore it is not necessary to leave out items with ratings in the middle of the rating scale, e.g. 3 on a scale from 1 to 5.

In the remainder of this section, the implementations developed by Bezerra e Carvalho (2010) will be detailed, focusing on the formal definitions of the concepts for each approach for recommendations systems.

2.3.2.1 Content-based filtering supported by SDA

For purposes of differentiating this methodology from the one presented above, this one will be identified as Content-based filtering using symbolic data (CB-SDA). This new approach includes the same steps as the one mentioned above:

1. Construction of a symbolic profile.
2. Comparison between the user profile and items.
3. Ranked list generation based on the scores

Also the profile construction step is composed by the pre-processing and generalization steps.

During the process of building a symbolic description for user profiles, the pre-processing step is done in the same way mentioned previously in section 2.3.1. Then, in the generalization process, each sub-profile is created by aggregating histogram-valued items of each interest level (ratings), for a specific user.

Let u_{g_k} be the sub-profile of user u, containing the items for an interest level g_k. Let the symbolic description of a user sub-profile for the interest level u_{g_k} be $y_{u_{g_k}} = (\xi^1_{u_{g_k}}, ..., \xi^P_{u_{g_k}})$, where $\xi^P_{u_{g_k}} = (\eta_j(u_{g_k}), \pi_j(u_{g_k}))$ with $\eta_j(u_{g_k})$ being the support measure for $\pi_j(u_{g_k})$.

Formally, let $x_i = (\xi^1_i, ..., \xi^P_i)$, $j = 1, ..., p$, be the symbolic description of an item belonging to the interest level u_{g_k}. The support measure $\eta_j(u_{g_k})$ of $\pi_j(u_{g_k})$ is calculated as follows:

$$\eta_j(u_{g_k}) = \bigcup_{i \in u_{g_k}} \eta_j(i) \tag{2.19}$$

Let $m \in \eta_j(u_{g_k})$ be a category in a domain $\mathcal{Y}$. The weight $W(m)$ attributed to the category m in the weight distribution $\pi_j(u_{g_k})$ is given by Equation (2.20):

$$W(m) = \frac{1}{|u_{g_k}|} \sum_{i \in u_\sigma} \delta(i, m) \tag{2.20}$$

where

$$\delta(i, m) = \begin{cases} w(m) \in \pi_j, \text{if } m \in \eta_j \\ 0, \text{otherwise.} \end{cases} \tag{2.21}$$

As an example using the movie domain, the profile of a user u is shown in Table 15.

Table 15 – Symbolic representation of an user sub-profiles

Sub-profiles	Director	Cast	Genre
Y_{lvl1}	$\emptyset$	$\emptyset$	$\emptyset$
Y_{lvl2}	$(\{D_5, D_7\},$	$(\{A_2, A_3, A_4, A_6, A_7, A_8, A_9\},$	$(\{G_2, G_3\},$
	(0.5,0.5))	(1/8, 1/8, 1/8, 1/8, 1/8, 1/4, 1/8)	(0.5, 0.5))
Y_{lvl3}	$(\{D_3\}, (1.0))$	$(\{A_1, A_3, A_4, A_5\}, (1/4, 1/4, 1/4, 1/4))$	$(\{G_1\}, (1.0))$
Y_{lvl4}	$\emptyset$	$\emptyset$	$\emptyset$
Y_{lvl5}	$(\{D_2\}, (1.0))$	$(\{A_1, A_2, A_7, A_8\}, (1/4, 1/4, 1/4, 1/4))$	$(\{G_3\}, (1.0))$

Source: The author (2021)

The last step is a suitable measure of similarity, which performs a comparison variable-wise and also takes into account all the multiple sub-profiles. Let $x_i = (\xi_i^1, ..., \xi_i^p)$, where $\xi_i^j = (\eta_j(i), pi_j(i))$, be the symbolic description of an item *i* and $y_{u_{gk}} = (\xi_{u_{gk}}^1, ..., \xi_{u_{gk}}^p)$, where $\xi_{u_{gk}}^j = (\eta_j(u_{gk}), \pi_j(u_{gk}))$, be the symbolic description of a user sub-profile, with interest level u_{gk}, $u_{gk} \in L$. The similarity between the item *i* and the user profile *u* is calculated as follows:

$$\Phi(u, i) = \frac{1}{|L|} * \sum_{gk \in L} \frac{2 * \rho_{gk} * (1 - \phi(y_{u_{gk}}, x_i))}{\rho_{gmax} - \rho_{gmin}}. \tag{2.22}$$

In Equation (2.22), the term ρ_{gk}, where $\rho_{gk} \in \mathcal{P}$, refers to the importance of a sub-profile u_{gk} in the similarity calculation. In the work by Bezerra e Carvalho (2010), $\mathcal{P} = [-2, -1, 0, 1, 2]$ and the function is normalized in the interval of $[-1, 1]$, with -1 being the lowest similarity and 1 the highest.

As in the earlier work by Bezerra e Carvalho (2004), the function $\phi(y_{u_{gk}}, x_i))$ has two components, the context-free component, ϕ_{cf}, and the context-dependent one, ϕ_{cd}. The context-free component works on the support measures $(\eta_j(i), \eta_j(y_{u_{gk}}))$, and the context-dependent one works on the weight distributions $(\pi_j(i), \pi_j(y_{u_{gk}}))$. The calculation of $\phi(y_{u_{gk}}, x_i))$ is shown in Equation (2.23).

$$\phi(y_{u_{gk}}, x_i)) = \frac{1}{p} * \sum_{j=1}^{p} [\phi_{cf}((\eta_j(i), \eta_j(y_{u_{gk}}))) + \phi_{cd}((\pi_j(i), \pi_j(y_{u_{gk}})))], \tag{2.23}$$

where p is the number of features in set $(\xi_1, ..., \xi_p)$. The function ϕ_{cf} is calculated, feature wise, by Equation (2.24).

$$\phi_{cf}(\pi_j(u_{gk}), \pi_j(i)) = \frac{1}{2}\left(\frac{\gamma + \delta}{\alpha + \gamma + \delta} + \frac{\gamma + \delta}{\beta + \gamma + \delta}\right). \tag{2.24}$$

Terms α, β and γ are calculated in the same way as in the work by Bezerra e Carvalho (2010), and are shown in Table 14. The second component, ϕ_{cd}, is calculated as follows:

$$\phi_{cd}(\eta_j(u_{gk}), \eta_j(i)) = \begin{cases} 0, \text{if } \eta_j(u_{gk}) \cap \eta_j(i) \neq 0, \\ \frac{|\eta_j(u_{gk}) \oplus \eta_j(i)| - |\eta_j(u_{gk})| - |\eta_j(i)|}{|\eta_j(u_{gk}) \oplus \eta_j(i)|}. \end{cases} \tag{2.25}$$

The joint $\eta_j(u_{gk}) \oplus \eta_j(i)$ is defined as

$$\eta_j(u_{gk}) \oplus \eta_j(i) = \begin{cases} \eta_j(u_{gk}) \cup \eta_j(i), \text{if the symbolic variable is set-valued,} \\ min(m_L, c_L), max(m_U, c_U), \text{if the variable is ordered.} \end{cases} \tag{2.26}$$

Finally terms m_L, c_L, m_U and c_U are defined as follows:

$$m_L = min(\eta_j(i)), \tag{2.27}$$

$$m_U = max(\eta_j(i)), \tag{2.28}$$

$$c_L = min(\eta_j(u_{gk})), \tag{2.29}$$

$$c_U = max(\eta_j(u_{gk})). \tag{2.30}$$

In order to illustrate the item-profile comparison process, item i, displayed in Table 16, will be matched against the profile described in Table 15, y_u. The process goes through all the features in item i's description and each user sub-profile.

Table 16 – Symbolic representation of an user sub-profiles

Item	**Director**	**Cast**	**Genre**
i	(D_3, (1.0))	(A_3, A_5, A_6, A_7, (1/4, 1/4, 1/4, 1/4))	(G_1, (1.0))

Source: The author (2021)

The first step is to calculate the context free component ϕ_{cf}. The values of α, β, γ and δ are calculated as in Table 14. The calculation of these values, for the attribute *cast* and the sub-profile 2, is shown in the following equations.

$$
\begin{aligned}
\eta_{cast}(y_{u2}) \cap \eta_{cast}(i) &= \{A_2, A_3, A_4, A_6, A_7, A_8, A_9\} \cap \{A_3, A_5, A_6, A_7\} \\
&= \{A_3, A_6, A_7\} \\
\eta_{cast}(y_{u2}) \cap \overline{\eta_{cast}(i)} &= \{A_2, A_4, A_8, A_9\} \\
\overline{\eta_{cast}(y_{u2})} \cap \eta_{cast}(i) &= \{A_5\} \\
\alpha = \sum_{\{m \in A_3, A_6, A_7\}} w(m) &\Rightarrow \alpha = \left(\frac{1}{4} + \frac{1}{4} + \frac{1}{4}\right) = \frac{3}{4} \\
\beta = \sum_{\{A_3, A_6, A_7\}} W(m) &\Rightarrow \beta = \left(\frac{1}{8} + \frac{1}{8} + \frac{1}{8}\right) = \frac{3}{8} \\
\delta = \sum_{\{A_5\}} w(m) &\Rightarrow \delta = \frac{1}{4} \\
\gamma = \sum_{\{A_2, A_4, A_8, A_9\}} W(m) &\Rightarrow \gamma = \left(\frac{1}{8} + \frac{1}{8} + \frac{1}{4} + \frac{1}{8}\right) = \frac{5}{8}
\end{aligned}
\tag{2.31}
$$

From the calculated values for α, β, γ and δ we have

$$
\phi_{cf}(y_{u2}, i) = \frac{1}{2}\left(\frac{\frac{1}{4} + \frac{5}{8}}{\frac{3}{4} + \frac{1}{4} + \frac{5}{8}} + \frac{\frac{1}{4} + \frac{5}{8}}{\frac{3}{8} + \frac{1}{4} + \frac{5}{8}}\right) \cong 0.619 \tag{2.32}
$$

Next, it is necessary to calculate the context-dependent component, ϕ_{cd}. To illustrate the process, the same comparison done previously is done using the variable *Cast* in order to compare the item *i* against the sub-profile 2 (y_{u2}), previously defined. As $S(y_{u2}) \cap S(i) = A_3, A_6, A_7 \neq \emptyset$, Equation (2.25) leads to $\phi_{cd}(S(y_{u2),S(i)}) = 0$.

Putting everything together, the final result for the *Cast* variable is given by the equation below.

$$
\begin{aligned}
\phi(y_{u2}, i) &= \frac{1}{3}\sum_{j=1}^{3}[\phi_{cd}(\eta(y_{u2}), \eta(i)) + \phi_{cf}(\pi(y_{u2}), \pi(i))] \\
&= \frac{1}{3}(1 + 1 + 0.62 + 0.625, 1 + 1) \cong 0.874.
\end{aligned}
\tag{2.33}
$$

.

In order to understand the whole relevance that item i has in the whole profile u, the comparison needs to be done with respect to all sub-profiles of this specific user and the values used in the ϕ equation. The scores for the other profiles are show below.

$$\begin{aligned}
\phi(y_{u1}, i) &= \frac{1}{3}(1+0+1+0+1+0) \approx 1 \\
\phi(y_{u3}, i) &= \frac{1}{3}(0+0+\frac{2}{3}+0+0+0) \approx 0.222 \\
\phi(y_{u4}, i) &= \frac{1}{3}(1+0+1+0+1+0) \approx 1 \\
\phi(y_{u5}, i) &= \frac{1}{3}(1+0+\frac{3}{7}+0+1+0) \approx 0.809
\end{aligned} \tag{2.34}$$

Finally, the calculated scores must be applied to Equation (2.22), to evaluate the whole relevance of the item *i* to the user *u*. So, according to Equation (2.22) we have:

$$\Phi(u, i) = \frac{1}{5}\left(\frac{0.0}{4} + \frac{0.254}{4} + \frac{0.0}{4} + \frac{0.0}{4} + \frac{0.764}{4}\right) = 0.0225 \tag{2.35}$$

The value found reflects the relevance of item *i* for user *u*. This approach improves the capacities from the previous one by making the profile building process more intuitive, allowing the system to handle different interest levels and also working independently of the user community, a well know feature of content-based recommendation systems. That is, even an item that has never been evaluated by another user in the community may be recommended to a target user (BEZERRA; CARVALHO, 2010).

2.3.2.2 Collaborative filtering supported by SDA

In the same work by Bezerra e Carvalho (2010), the authors also define an approach for building recommendation systems with collaborative filters using symbolic data. As stated before, this type of recommendation system has a social aspect, where item evaluations made by similar users are more expressive in the current user's recommendation.

This section details the implementation developed by Bezerra e Carvalho (2010) will be detailed, focusing on the formal definitions of the concepts for collaborative filtering using symbolic data. In this work, this methodology will be identified as Collaborative filtering using symbolic data (CF-SDA).

This approach has some different steps compared to CB-SDA. First it is necessary to build a symbolic description of the user profiles, then the users need a weight associated with their similarity to the current user, the *h* closest users have to be selected and finally a ranked list is generated from the weighted combination of the ratings of the *h* closest users.

Similar to the content-based approach, the building of the symbolic descriptions of user profiles have the same two steps, pre-processing and generalization, but with some differences.

The pre-processing step, as the content-based approach, aims to build significant symbolic descriptions of the items. Let the possible interest levels for the items be represented by $L = [g_1, ..., g_k]$. Let the set of all users be U, such that given a user *u*, $u \in U$. Let $\mathcal{F}_i^k = \{u \in U$: the grade given by the user u to the item i is $g_k \in L\}$.

Given these definitions, the symbolic description of an item *i* is $x_i = (\xi_i)$ with $\xi_i = (\eta(i), \pi(i))$, where $\forall i, \eta(i) = L = \{g_1, ..., g_k\}$ is the support for the weight distribution $q(i) = (q_{g1}(i), ..., q_{gk}(i))$. The weights are calculated as follows.

$$q_{gk}(i) = \frac{|\mathcal{F}_i^k|}{\sum_{h=1}^{k} |\mathcal{F}_i^h|} \tag{2.36}$$

In Equation (2.36) $|\mathcal{F}_i^k|$ represents the size of the set $\mathcal{F}_i^k$. Table 17 shows the symbolic descriptions for four items.

Table 17 – Symbolic representation for four items for collaboration-filtering

item	ξ_i
i_1	$(\{1, 2, 3, 4, 5\}, (0.5, 0.0, 0.5, 0.0, 0.0))$
i_2	$(\{1, 2, 3, 4, 5\}, (0.75, 0.0, 0.0, 0.0, 0.25))$
i_3	$(\{1, 2, 3, 4, 5\}, (0.0, 1.0, 0.0, 0.0, 0.0))$
i_4	$(\{1, 2, 3, 4, 5\}, (0.0, 0.0, 0.50, 0.0, 0.50))$

Source: The author (2021)

In the generalization step the goal is to build symbolic descriptions for each level of interest of a specific user, corresponding to the sub-profiles that compound the final profile that summarizes the whole interest of this specific user.

Let the sub-profile of user *u* for the interest level g_k be represented by u_{gk}. Now let the symbolic representation of the sub-profile u_{gk} be $y_{gk} = (\xi_{gk})$, such that $\xi_{gk} = (\eta(u_{gk}), \pi(u_{gk}))$, with $\forall u_{gk}$, $\eta(u_{gk}) = L = [g_1, ..., g_k]$ and the weight distribution $\pi(u_{gk})$. Additionally, let $|u_{gk}|$ be the cardinally of the set u_{gk}. Given such definitions, the weight assigned to each level of interest, $W(gk)$ is given by

$$W(gk) = \frac{1}{|u_{gk}|} \sum_{i \in u_{gk}} \pi_{gk}(i) \tag{2.37}$$

For a better understanding of the concepts detailed above, Table 18 shows examples of symbolic descriptions of user profiles.

Next is the user profile similarity step, which evaluates how similar two profiles are, in order to later attribute bigger weights for the ratings given by the closest users, neighbors.

Table 18 – Symbolic representation for users for collaboration-filtering

User	Symbolic profile description
User 1	({1, 2, 3, 4, 5}, (0, 0, 0, 0, 0))
	({1, 2, 3, 4, 5}, (0, 0.50, 0, 0.25, 0.25))
	({1, 2, 3, 4, 5}, (0, 0, 0, 0, 0))
	({1, 2, 3, 4, 5}, (0, 0, 0, 0, 0))
	({1, 2, 3, 4, 5}, (0.125, 0, 0.375, 0.125, 0.375))
User 2	({1, 2, 3, 4, 5}, (0, 0, 0, 0, 0))
	({1, 2, 3, 4, 5}, (0, 0.50, 0, 0.375, 0.125))
	({1, 2, 3, 4, 5}, (0.25, 0, 0.25, 0.25, 0.25))
	({1, 2, 3, 4, 5}, (0, 0, 0, 0, 0))
	({1, 2, 3, 4, 5}, (0, 0, 0, 0, 1.0))
User 3	({1, 2, 3, 4, 5}, (0.25, 0, 0.25, 0.25, 0.25))
	({1, 2, 3, 4, 5}, (0, 0, 0, 0, 0))
	({1, 2, 3, 4, 5}, (0, 0, 0.50, 0, 0.50))
	({1, 2, 3, 4, 5}, (0, 0.50, 0, 0.50, 0))
	({1, 2, 3, 4, 5}, (0, 0.25, 0, 0.125, 0.625))
User 4	({1, 2, 3, 4, 5}, (0, 0, 0, 0, 0))
	({1, 2, 3, 4, 5}, (0, 0, 0, 0, 0))
	({1, 2, 3, 4, 5}, (0, 0, 0, 0, 0))
	({1, 2, 3, 4, 5}, (0.125, 0.25, 0.125, 0.25, 0.25))
	({1, 2, 3, 4, 5}, (0, 0, 0, 0, 1))

Source: The author (2021)

Let the sub-profile for user u be represented by $y_{u_{gk}} = (\eta(u_{gk}), \pi(u_{gk}))$ and a neighbor candidate sub-profile be represented by $y_{v_{gk}} = (\eta(v_{gk}), pi(v_{gk}))$. The similarity between user *u* and candidate *v* is given by following Equation (2.38).

$$\Psi(u, v) = \frac{1}{|L|} \sum_{gk \in L} (1 - \varphi(y_{u_{gk}}, y_{v_{gk}})), \tag{2.38}$$

where $|L|$ is the cardinality of the set of the possible interest levels L and function $\varphi(y_{u_{gk}}, y_{v_{gk}}))$ measures the similarity between two sub-profiles. Function $\varphi(y_{u_{gk}}, y_{v_{gk}}))$ can be interpreted as a version of the *Euclidean Distance*, as presented in the next equation.

$$\varphi(y_{u_{gk}}, y_{v_{gk}})) = \sqrt{\sum_{g_k \in L} (W(u_{gk}) - W(v_{gk}))^2}. \tag{2.39}$$

With Equation (2.39), the system must calculate the similarity between the user in question and everyone else present in the base. The similarities between the profiles from Table 13 were

calculated and show in Table 19.

Table 19 – Distances between users

	User1	**User2**	**User3**	**User4**
User1	1.0000	0.7146	0.3886	0.6340
User2	0.7146	1.0000	0.4376	0.6790
User3	0.3886	0.4376	1.0000	0.5715
User1	0.6340	0.6790	0.5715	1.0000

Source: The author (2021)

Moving forward to the building of the recommendation list step, the ratings of the *h* closest users are weighted for each item not evaluated by the current user. Function $\Pi(u, i)$ is used to estimate scores given by the user *u* to the item *i*. The function is defined as follows

$$\Pi(u, i) = \overline{r}_u + \frac{\sum_{v=1}^{h}(r_{v,i} - \overline{r}_v) * \varphi(u, v)}{\sum_{v=1}^{h} \varphi(u, v)}, \tag{2.40}$$

where h stands for the size of the neighborhood of user *u*, *v* is one of the *h* neighbors, $\overline{r}_u$ is the average rating given by user *u*, $\overline{r}_v$ is the average rating given by user *v* and $\overline{r}_{v,i}$ is the rating for item *i* by neighbor *v*.

The score for item 4, from Table 12, and user 1, from Table 13, is calculated as follows:

$$\begin{aligned}
\Pi(u_1, i_4) = \overline{r}_1 &+ \frac{\sum_{v=1}^{3}(r_{v,i4} - \overline{r}_v) * \Psi(u_1, v)}{\sum_{v=1}^{3} \Psi(u_1, v)} = 3 \\
&+ \frac{\sum_{v=1}^{3}(r_{u_2,i4} - \overline{r}_{u_2}) * \Psi(u_1, u_2)}{\Psi(u_1, u_2) + \Psi(u_1, u_3 + \Psi(u_1, u_4))} \\
&+ \frac{\sum_{v=1}^{3}(r_{u_3,i4} - \overline{r}_{u_3}) * \Psi(u_1, u_3)}{\Psi(u_1, u_2) + \Psi(u_1, u_3 + \Psi(u_1, u_4))} \\
&+ \frac{\sum_{v=1}^{3}(r_{u_4,i4} - \overline{r}_{u_4}) * \Psi(u_1, u_4)}{\Psi(u_1, u_2) + \Psi(u_1, u_3 + \Psi(u_1, u_4))} \\
\Pi(u_1, i_4) = 3 &+ \frac{(5-4) * (0.71)}{0.71 + 0.39 + 0.63} + \frac{(3 - \frac{8}{5}) * 0.39}{0.71 + 0.39 + 0.63} + 0 \cong 3.275
\end{aligned} \tag{2.41}$$

This approach doesn't need any prior knowledge of the products being recommended, only the ratings attributed by the users. This could be interesting to deal with the previously mentioned *cold-start* problem.

In the same work the authors Bezerra e Carvalho (2010) also define the whole structure for a Hybrid approach for recommendation systems that inherit the aspects from the content-based approach and collaborative filtering as well. As this work doesn't implement any upgrade using this hybrid approach, it will be left out.

2.4 CHAPTER CONCLUSION

The first two sections of the chapter introduce challenges related to recommendation systems and some of their most famous approaches. Then, an introduction was made about symbolic data, the possible data types and the advantages of using symbolic data. Finally, the problem of building recommendation systems using symbolic data and its first approaches is presented.

Due to the fact that it is a relatively new area, the analysis of symbolic data has a lot of open problems. Being a very rich field for studies and practical applications that make use of its many advantages.

As we can see the last topic presented is not as explored as its counterpart using classical data. The need to further explore recommendation systems using symbolic data is evident. The next chapter of this work describes with details a new approach to build recommendation systems using symbolic data.

3 METHODOLOGY

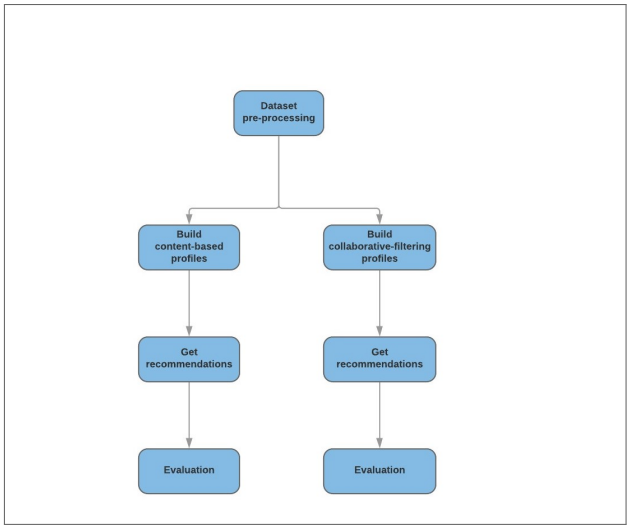
This chapter aims to present in detail the proposed methodology to implement and evaluate content-based recommendation systems and collaborative filtering-based recommendation systems using symbolic data.

3.1 METHODOLOGICAL STEPS

The code developed during the research was implemented using the python language and making use of its Pandas (MCKINNEY, 2011), Scipy (VIRTANEN et al., 2020) and Numpy (OLIPHANT, 2007) libraries used to manipulate dataframes and matrices and perform calculations. The code is stored in the Github repository <https://github.com/DelmiroDaladier/recommender_system_using_symbolic_data/settings>.

The research conducted in this work follows the steps shown in Figure 9. Each step depicted in the flowchart will be explained in detail in the following sections.

Figure 1 – Flowchart of the steps in the development of this work.



Source: The author (2021)

3.2 DATASET

The articles that preceded this work developed their methodologies using data from the film domain, specifically the EACHMOVIE Database , therefore we decided to validate our

work using the same domain. The dataset was created based on the Full MovieLens Dataset. [1]

This dataset is composed by 45000 movies belonging to the Full MovieLens Dataset. The information covers the ratings given by users to movie metadata that includes cast, crew, director, genre and other information. The whole data is divided among the following files:

movies_metadata.csv: This file contains information about the movie itself, including classification, film collection, genres, budget, overview, original language and others.

keywords.csv: Contains the movie keywords in json format.

credits.csv: Contains information about movie crew and cast as a "stringified" json object.

links.csv: Contains the TMDB and IMDB of all movies in the dataset.

links_small.csv: Contains the TMDB and iMDB for a subset of 9000 movies from the original dataset.

ratings_small.csv: The file contains 100000 ratings from 700 users on 9000 movies.

In addition to belonging to the same domain as the one used in the articles that inspired this work, the dataset is well known and is frequently used in the construction of recommendation systems. However, the classical data present in the files need to be converted to the symbolic data domain, so we can apply the techniques developed in this work. Therefore a series of steps need to be applied to have the required data format in order to build recommendation systems with symbolic data.

3.3 DATA PRE-PROCESSING

Differently from the approaches for content-based and collaborative filtering using symbolic data that employed two different pre-processing strategies, this work uses a single step for both approaches. This step is the same observed in the (BEZERRA; CARVALHO, 2010) for content-based recommendations, once the methodology proposed in this work focus on the items features and in its ratings.

The purpose of this step is to convert the classical data present in the original dataset into modal symbolic values, so it should be applied to all suitable features. Let the description of

[1] more information can be found in the link: <https://grouplens.org/datasets/movielens/latest/>

an item i be $x_i = (X_i^1, ..., X_i^p)$ and X_i^j be a set of categories. We assign a weight $w(m)$ to each category $m \in X_i^j$, such that:

$$w(m) = \begin{cases} \frac{1}{|X_i^j|}, \text{if the variable is single or multi-valued,} \\ \frac{f(m)\text{IDF}(m)}{\sum_{m \in X_i^j} f(m) IDF(m)}, \text{if the variable is textual,} \end{cases} \tag{3.1}$$

where $f(m)$, commonly known as TF(m), and IDF(m) represent the term frequency and the inverse document frequency respectively. The product TF$(m)*$IDF(m) is an important information retrieval metric.

At the end of this step the item x_i can be represented as $x_i = (\xi_i^1, ..., \xi_i^j)$ such that $\xi_i^j = (\eta_j(i), \pi_j(i))$ is a modal symbolic variable, where $\eta_j(i)$ is a set of categories and $\pi_j(i)$ is the weight distribution associated with the categories listed in $\eta_j(i)$.

The desired features of each movie, distributed among the various files in the dataset, were merged into a single dataset, called *movie_data.csv*, which contains the features movie_id, genres, director, cast and synopsis. The dataset containing the ratings, movie_id and user_id is the same used in the original dataset, ratings.csv.

Once the movie_data.csv content goes through the process of creating symbolic descriptions for each of its resources, it is possible to start the construction of user profiles in order to obtain recommendations, that is, the methodologies developed in this work to build profiles and compare items/users can be applied. Such techniques are described in the following sections.

3.4 CONTENT-BASED RECOMMENDATION SYSTEMS

In this section, the proposed methodology for creating content-based recommendation systems is described, which is identified as Histogram descriptions for content-based recommendations (HD-CBR). The user profile creation process and the matching between the base items and the created profiles will be detailed in each of the following sections.

3.4.1 User profile creation

After the pre-processing step, the symbolic descriptions of the movies are ready to be aggregated in order to build the user profiles. Differently from the other works, the user profiles

created using this methodology are not composed of sub-profiles. The profiles are created by taking into account the scores of each individual item evaluated previously, the weights of their symbolic item descriptions and the set of scores given by an user.

The symbolic description of a user is built using the *n* items previously evaluated. These *n* items are obtained from a stratified sampling for each of the interest levels leaving out the lowest ratings, i.e. 0 and 1, since the system is focused on producing recommendations with the potential to receive higher scores. Therefore, the modeling focus is on the preferences of each user.

Formally, user u has a symbolic description $x_u = (\xi_u^1, ..., \xi_u^j)$, where $\xi_u^j = (\eta_j(u), \pi_j(u))$, $\eta_j(u)$ is a subset of categories of the domain D_j, i.e. $\eta_j(i) \in D_j$, and $\pi_j(i)$ is a weight distribution associated with the categories in $\eta_j(i)$. The weight associated to each category *m*, $W(m)$, in $\eta_j(i)$ is calculated as follows

$$W(m) = \frac{\sum_{i=0}^{n}(r_i * w_i(m))}{\sum_{i=0}^{n} r_i}, \tag{3.2}$$

where r_i is the rating given by a user to the item *i* and $w_i(m)$ is the weight associated to the category *m* for the item *i*. The set of categories for a feature in the user profile, $\eta_j(u)$, is calculated as follows

$$\eta_j(u) = \bigcup_{i \in u} \eta_i(u) \tag{3.3}$$

In order to illustrate the calculation for building user profiles, Table 20 contains symbolic descriptions for the *Cast* variable of four items.

Table 20 – Symbolic descriptions for the *Cast*.

Item	Symbolic profile description	Score
M_1	$([A_1, A_2, A_3, A_4], \{0.25, 0.25, 0.25, 0.25\})$	5
M_2	$([A_1, A_2, A_5, A_6], \{0.25, 0.25, 0.25, 0.25\})$	4
M_3	$([A_5, A_6, A_7, A_8, A_9], \{0.20, 0.20, 0.20, 0.20, 0.20\})$	3
M_4	$([A_9, A_8, A_7, A_3, A_4], \{0.20, 0.20, 0.20, 0.20, 0.20\})$	4

Source: The author (2021)

The calculation for the weights is as follows

$$W_{A_1}(Cast) = \frac{\sum_{i=0}^{n}(r_i * w_i(m))}{\sum_{i=0}^{n} r_i} = \frac{(5 * 0.25) + (4 * 0.25)}{5 + 4 + 3} = \frac{2.25}{12} = 0.1875$$
$$W_{A_5}(Cast) = \frac{\sum_{i=0}^{n}(r_i * w_i(m))}{\sum_{i=0}^{n} r_i} = \frac{(4 * 0.25) + (3 * 0.20)}{5 + 4 + 3} = \frac{1.6}{12} = 0.1333 \quad (3.4)$$

Once this process is executed for all users and all features of the chosen database, the matching process can be started. In this process, the dissimilarity between items and profiles must be calculated to attribute scores to each item, given an user.

3.4.2 Dissimilarity Calculation

Since both item symbolic description and user profile symbolic descriptions are histograms, a suitable measure of dissimilarity should be one that can handle data distributions as inputs.

Among the metrics of distance between probability distributions, the chosen one was the Wasserstein Distance. As an analogy, the idea is to calculate the minimum energy required to turn the shape of a sand pile into the shape of a second sand pile, where each sand pile represents a probability distribution. Due to this analogy the distance is also called Earth Mover's distance.

Formally, let μ and ν be two probability distributions defined in $\mathcal{R} \times \mathcal{R}$ and p is the number of measures taken from these distributions. The Wasserstein distance between μ and ν, $W_p(\mu, \nu)$ is calculated as follows

$$W_p(\mu, \nu) = \inf_{\pi \in \Gamma(\mu,\nu)} \left(\int_{\mathcal{R} \times \mathcal{R}} |x - y| d\pi(x, y) \right) \quad (3.5)$$

Where $\Gamma(\mu, \nu)$ is the set of distributions whose marginals are μ and ν.

3.4.3 Ranked list generation

After the dissimilarity calculation between items and the user profile a ranked list is generated using the item-user dissimilarity as scoring function, in ascending order.

3.5 COLLABORATIVE-FILTERING RECOMMENDATION SYSTEMS

In this section, the proposed methodology for creating collaborative-filtering based recommendation systems is described. The next sections detail how we create user profiles, build

neighborhoods and generate ranked lists.

3.5.1 User profile creation

For the collaborative-filtering approach, the user profile building process is the exact same process described for the content-based approach. The difference between the approaches lies in how items are evaluated and how a ranked list with recommendations is created.

Once the profiles are created, it's necessary to identify the most similar users to the user who will receive the recommendation, in order to calculate weights for the ratings given by them. This is called the user's neighborhood.

3.5.2 Neighborhood creation

The evaluation of user similarity, i.e. the formation of a neighborhood, also makes use of the symbolic representations created for user profiles. In this case the histograms describing each profile variable are compared and their distances used to calculate the total similarity between profiles.

As all features in the user profiles are histograms, distance functions that can handle probability distributions are suited for this task. The same distance functions used in the content-based approach were used to create neighborhoods, but Wasserstein distance had better results in our experiments.

Formally, given a user *u*, whose profile is described by $x_u = (\xi_u^1, ..., \xi_u^p)$ where $\xi_u^j = (\eta_j(u), \pi_j(u))$ such that $\eta_j(u)$ is the set of categories and $\pi_j(u)$ is the weight distribution associated to $\eta_j(u)$. Also let the neighbor candidate *v* be described by $x_v = (\xi_v^1, ..., \xi_v^p)$ where $\xi_v^j = (\eta_j(v), \pi_j(v))$ such that $\eta_j(v)$ is the set of categories and $\pi_j(v)$ is the weight distribution associated to $\eta_j(v)$. The dissimilarity between *u* and *v*, $D(u, v)$, can be measured as follows

$$D(u, v) = \frac{\sum_{i=0}^{p} W_i(u, v)}{p} \tag{3.6}$$

Where *p* is the number of features in the profile and $W_p(u, v)$ is the Wasserstein distance between the users u and v.

To illustrate the process, let Table 21 describe a group of six user profiles with values for variables *Cast* and *Genres*. The first row contains the profile for active user *u*, which will

receive recommendations, and the others contain five possible neighbors.

Table 21 – Symbolic representation of user and the candidate neighbors

User	Cast	Genres
u	$(\{[A_1, A_2, A_3],$	$(\{[G_1, G_2, G_3]$
	$[0.33, 0.33, 0.33]\}$	$[0.5, 0.4, 0.1]\})$
n_1	$(\{[A_3, A_5, A_1, A_9]$	$(\{[G_5, G_2, G_7, G_4]$
	$([0.2, 0.05, 0.25, 0.5]\}$	$[0.3, 0.4, 0.1, 0.2]\})$
n_2	$(\{[A_4, A_5, A_7]$	$(\{[G_5, G_1, G_4]$
	$([0.33, 0.33, 0.33]\}$	$[0.3, 0.4, 0.3]\})$
n_3	$(\{[A_3, A_5, A_1, A_9, A_2]$	$(\{[G_1, G_2, G_7, G_4, G3]$
	$([0.2, 0.2, 0.2, 0.2, 0.2]\}$	$([0.3, 0.4, 0.1, 0.1, 0.1]\})$
n_4	$(\{[A_2, A_7, A_9, A_1]$	$(\{[G_1, G_2, G_5]$
	$([0.25, 0.25, 0.25, 0.25]\}$	$([0.33, 0.33, 0.33]\})$
n_5	$(\{[A_9, A_7, A_3])$	$(\{[G_1, G_2]$
	$([0.33, 0.33, 0.33]\}$	$([0.7, 0.3]\})$

Source: The author (2021)

Applying Equation (3.6) to the user profile *u* and the profiles of possible neighbors, it is possible to obtain the dissimilarity values between them and then form a neighborhood. Neighbors ranked according to their dissimilarity with user u are shown in Table 22.

Table 22 – Neighbors ranked according to their dissimilarity with user u.

Neighbor	Score
n_3	1.45
n_4	1.91
n_5	2.32
n_2	2.42
n_1	2.90

Source: The author (2021)

Therefore, ratings assigned by user n_3 have a greater impact than grades assigned by user n_1, for items recommended for user *u*.

With the dissimilarity values for each user in the base, they are ranked and a number *h* of closest users is selected to compound the neighborhood for the active user. In this work the value adopted for *h* is 30.

3.5.3 Ranked list generation

To evaluate the score of each item, the scores given by the users in the neighborhood are weighted such that ratings from closest users have more influence on the final score.

Formally, let the dissimilarity between a user *u* and a neighbor *v* be defined as $D(u, v)$. Let r_u be the average ratings of the user *u*, r_{vi} the rating given by the neighbor *v* for the item i and $\overline{r}_v$ the average ratings given by the neighbor *v*. So, the rating for item *i* and user *u*, given their neighborhood with size *h* is calculated as follows

$$S(u, i) = \frac{\sum_{v=0}^{h}(r_{vi} - r_v)}{\sum_{v=0}^{h} D(u, v)} \tag{3.7}$$

The scores calculated by Equation (3.7) are used to perform an ascending ranking for the items in the base, creating the recommendation list for the active user.

3.6 METRICS

Usually the objective in the recommendation process is to create a list containing relevant items for a specific user. Given the task of creating a ranked list, pertinent questions include: how to evaluate a ranked list quality and how to measure the quality at different ranked list sizes. To evaluate this task, a suitable metric must answer these questions.

The metric chosen in this work to evaluate the quality of the ranking generated by the recommendation systems is the NDCG, which stands for Normalized Discounted Cumulative Gain defined by Järvelin e Kekäläinen (2002). This metric is originally from the Information Retrieval area and belongs to a group of measures that estimate the relevance gain of a ranked list by evaluating the result up to a given rank.

While traditional metrics treat highly relevant items and items with lower relevance equally, the NDCG assigns higher scores to ranked lists containing items with higher relevance in the first positions. Since all items are not of equal relevance to their users, highly relevant documents should be identified and ranked first for presentation (JäRVELIN; KEKäLäINEN, 2002). The NDCG measures the rank quality, by calculating the usefulness of an item in the ranking list using its position in the list and the adopted relevance scale.

The fundamental idea, in this measure, is that relevant items should appear at the first positions of the ranked list and less relevant items appear at the end of the list.

The NDCG for a list with size *k* is calculated according to Equation (3.8).

$$NDCG_k = \frac{DCG_k}{IDCG_k}, \tag{3.8}$$

where DCG_p is the discounted cumulative gain at position *p* given byÇ

$$DCG_p = \sum_{i=1}^{p} \frac{rel_i}{log_2(i+1)}, \tag{3.9}$$

where rel_i is the relevance of the item at position *p*. IDCG stands for Ideal Discounted Cumulative Gain, so it's the value of the DCG when items are ranked according to their real relevance. Thus better ranked lists are those whose the NDCG is closer to 1.

As an example, let li be a list containing the item scores of an ideal ranking such that $l_i = [5, 5, 5, 4, 4, 4, 3, 3, 2, 1]$, and two lists with the item scores generated by two different recommendation systems l_1 and l_2 such that, $l_1 = [5, 5, 4, 5, 4, 3, 4, 3, 2, 1]$ and $l_2 = [5, 1, 2, 3, 3, 5, 5, 4, 4, 4]$.

For the list l_1, the $NDCG_{10}$ value is calculated as follows

$$NDCG_{10} = \frac{17.8012}{18.1871} = 0.9787, \tag{3.10}$$

and the value calculated for the list $l2$ is calculated as follows

$$NDCG_{10} = \frac{15.2661}{18.1871} = 0.8393, \tag{3.11}$$

Therefore, list l1 presents a greater gain in relation to list l2, since the first presents items with greater relevance in its first positions.

3.7 EXPERIMENTAL SETUP

In order to evaluate the proposed methods for both of the approaches, content-based and collaborative-filtering, an experimental methodology was designed. The comparison is made between the proposed methods and the methods developed in the work of (BEZERRA; CARVALHO, 2010) as reference method.

The methodology comprises three steps. First the user selection, which followed by the profile building step and then the evaluation.

3.7.1 User selection

First step is to select user profiles that are suitable for the evaluation process. The users should have at least 200 evaluations in the base, since part of their evaluations will be used to build profiles and the remaining evaluations will be used to create a ground-truth set. Altogether, due to processing time constraints, 100 users were randomly collected for the evaluation.

3.7.2 Profile building

After selecting the users from the base, the user profiles used for evaluations are created using 20, 40, 60 and 100 randomly selected evaluations using the proposed methods,HD-CBR and Histogram descriptions for collaborative-filtering recommendations (HD-CFR), and the reference methods, CB-SDA and CF-SDA, as well.

Therefore, the content based approaches, HD-CBR and CB-SDA, are compared when it's user profiles are built using 20, 40, 60 and 100 evaluations. The same process is applied to the collaborative filtering approaches, HD-CFR and CF-SDA.

3.7.3 Evaluation

The profiles built are used to perform recommendations on a set of 200 remaining items, which were not used to build the profiles. The comparison is performed in pairs, i.e. for the same user, a profile built with the proposed method using 20 items should be compared with a profile built with the reference method also using 20 items, and so on.

For each user, the ranked lists produced in this step have their $NDCG_{10}$ calculated and the proposed method is compared to the references.

3.7.4 Hypothesis testing

To ensure that the methodologies produce significantly different results, a Wilcoxon hypothesis test was used in this step. The Wilcoxon test is used to evaluate if the samples belong or not to the same distribution. This test can be interpreted as a non-parametric version of the T-Test.

4 RESULTS

This Chapter presents an experimental evaluation of HD-CBR, CB-SDA, HD-CFR and CF-SDA, using the pre-processed dataset described in Section 3.3. All methods were evaluated with profiles based on 20, 40, 60 and 100 evaluations.

4.1 CONTENT-BASED RECOMMENDATION SYSTEMS

Table 23 presents the values for $NDCG_{10}$ obtained in the experiments with respect to content-based recommendation systems. Column *Items used* is related to the number of items used to build the user profiles, column HD-CBR contains the average *NDCG* at top 10 obtained by HD-CBR, column CB-SDA contains the average *NDCG* at top 10 obtained by the CB-SDA methodology. We used one-sided paired Wilcoxon tests to check whether HD-CBR significantly outperformed CB-SDA, which was confirmed by the resulting p-values shown in the last column of Table 23.

Table 23 – Average of $NDCG_{10}$ for content-based approach

Items used	HD-CBR	CB-SDA	p-value
20	0.7278	0.7022	0.0001
40	0.7476	0.7032	$2.3342e^{-08}$
60	0.7643	0.7009	$3.1697e^{-13}$
100	0.7832	0.7073	$3.2505e^{-13}$

Source: The author (2021)

As shown in Table 23, HD-CBR achieves higher values for $NDCG_{10}$, i.e. the method can recommend items that are more interesting to the users, regardless of the number of evaluations used to build the profiles. It is worth noting that even the lowest average $NDCG_{10}$ obtained by HD-CBR (using 20 item evaluations) outperformed the highest average $NDCG_{10}$ obtained by CB-SDA (using 100 item evaluations).

As an example, recommendations were generated for a random user applying the methodologies CB-SDA and HD-CBR, with the different numbers of items used to build the user profiles (20, 40, 60 and 100 items). In order to test the quality of the recommendations generated we used a set of 200 movies and ratings, given by this user.

Table 24 shows the top 10 movies ranked list generated by CB-SDA, when the user profile

was constructed using 20 items, the $NDCG_{10}$ obtained for this ranked list was 0.7554. In contrast, the Table 25 shows the top 10 movies ranked list using the methodology HD-CBR, the $NDCG_{10}$ obtained for this list was 0.7794. The $NDCG_{10}$ difference shows that the ranked list presented by Table 25 has a better quality than the one presented in Table 24, i.e., items with higher scores tend to appear at the first positions.

Table 24 – Recommendation list produced by CB-SDA - Using 20 items to build the profile

Title	Rating
Notes on a Scandal	5
Street Kings	4
48 Hrs.	4
M	3
The Last Castle	2
Killing Zoe	4
The Way of the Gun	4
Run Lola Run	2
Scarface	3
Blind Man	3

Source: The author (2021)

Table 25 – Recommendation list produced by HD-CFR - Using 20 items to build the profile

Title	Rating
Killing Zoe	4
Monsoon Wedding	4
My Tutor	4
The Hunchback of Notre Dame	2
Jack & Sarah	4
Gandhi	2
Soul Assassin	4
Run Lola Run	2
Belle Époque	3
The Man with the Golden Arm	4

Source: The author (2021)

Following the example, the recommendations generated for the same user with profiles built using 40 items. Table 26 shows the top 10 movie recommendations by the user generated by CB-SDA when the user profile was built using 40 items, the $NDCG_{10}$ obtained for this list was 0.7104. In contrast, the Table 27 shows the recommendations provided to the same

user using the methodology HD-CBR, the $NDCG_{10}$ obtained for this list was 0.7964. The $NDCG_{10}$ difference shows that the ranked list presented by Table 27 has a better quality than the one presented in Table 26.

Table 26 – Recommendation list produced by CB-SDA - Using 40 items to build the profile

Title	Rating
Notes on a Scandal	5
Street Kings	4
48 Hrs.	4
M	3
The Last Castle	2
Killing Zoe	4
The Way of the Gun	4
Run Lola Run	2
Scarface	3
Blind Man	3

Source: The author (2021)

Table 27 – Recommendation list produced by HD-CFR - Using 40 items to build the profile

Title	Rating
48 Hrs.	4
The Talented Mr. Ripley	3
Syriana	2
Sleepless in Seattle	4
Amélie	4
Bread and Tulips	3
Y Tu Mamá También	3
Romeo + Juliet	3
Tough Enough	3
Jack & Sarah	4

Source: The author (2021)

Next, the recommendations generated for the same user with profiles built using 60 items. Table 28 show the recommendations generated by CB-SDA when the user profile was built using 60 items, the $NDCG_{10}$ obtained for this list was 0.7111. In contrast, the Table 29 shows the recommendations provided to the same user using the methodology HD-CBR, the $NDCG_{10}$ obtained for this list is 0.8072. The $NDCG_{10}$ difference shows that the ranked list presented by Table 29 has a better quality than the one presented in Table 28.

Table 28 – Recommendation list produced by CB-SDA - Using 60 items to build the profile

Title	Rating
Notes on a Scandal	5
Street Kings	4
48 Hrs.	4
M	3
The Last Castle	2
Killing Zoe	4
The Way of the Gun	4
Run Lola Run	2
Scarface	3
Blind Man	3

Source: The author (2021)

Table 29 – Recommendation list produced by HD-CFR - Using 60 items to build the profile

Title	Rating
The Talented Mr. Ripley	3
Amélie	4
Y Tu Mamá También	3
Battle Royale	4
Jack & Sarah	4
Sleepless in Seattle	4
Tough Enough	3
Syriana	2
American Pie	2
Bread and Tulips	3

Source: The author (2021)

Finally, for content-based recommendations, the recommendations generated for the same user with profiles built using 100 items. Table 30 show the recommendations generated by CB-SDA when the user profile was built using 100 items, the $NDCG_{10}$ obtained for this list was 0.6871. In contrast, the Table 31 shows the recommendations provided to the same user using the methodology HD-CBR, the $NDCG_{10}$ 0.7921. The $NDCG_{10}$ difference shows that the ranked list presented by Table 31 has a better quality than the one presented in Table 30.

Another interesting way to compare both methodologies is to analyze the distribution of their $NDCG_{10}$ scores. In Figure **??**, we can see a slight shift of values to the right indicating a slight increase in the observed values. A similar trend can be seen in Figures **??**, **??**, and **??**,

Table 30 – Recommendation list produced by CB-SDA - Using 100 items to build the profile

Title	Rating
M	3
The Last Castle	2
Killing Zoe	4
The Way of the Gun	4
Run Lola Run	2
Scarface	3
Blind Man	3
Brake	3
Syriana	2
Comanche Station	3

Source: The author (2021)

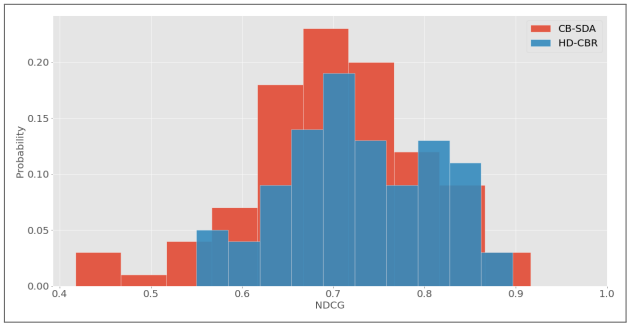
Table 31 – Recommendation list produced by HD-CFR - Using 100 items to build the profile

Title	Rating
The Talented Mr. Ripley	3
48 Hrs.	4
Amélie	4
Y Tu Mamá También	3
Sleepless in Seattle	4
Donnie Darko	2
Almost Famous	3
Sister Act	4
American Pie 2	2
Aguirre: The Wrath of God	4

Source: The author (2021)

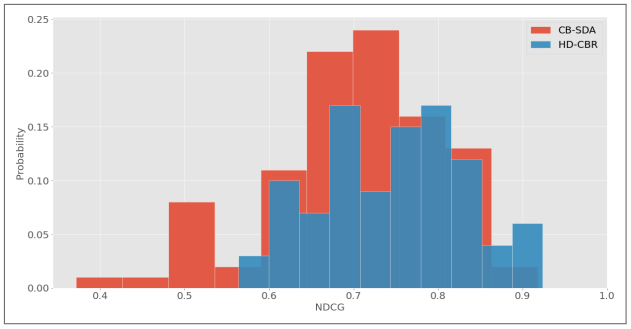
showing an improvement in $NDCG_{10}$ values.

Figure 2 – Comparison beetween DH-CBR and CB-SDA with profiles built using 20 items



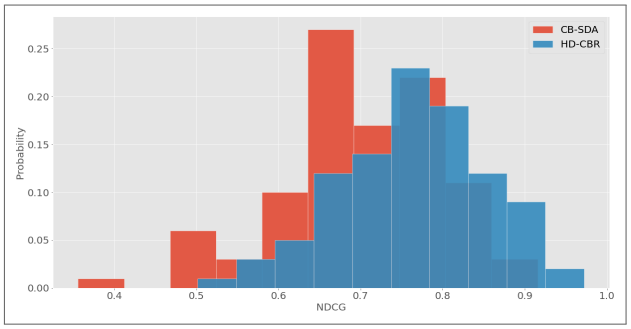
Source: The author (2021)

Figure 3 – Content-based approach with profile built using 40 items



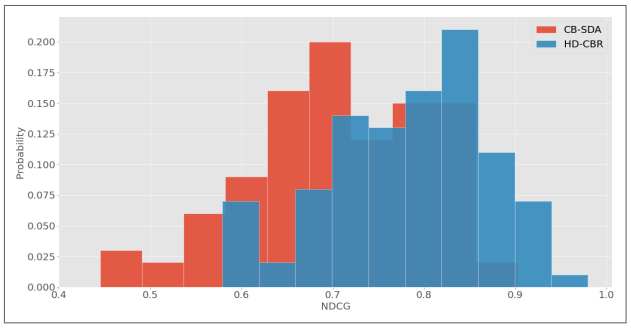
Source: The author (2021)

Figure 4 – Content-based approach with profile built using 60 items



Source: The author (2021)

Figure 5 – Content-based approach with profile built using 100 items



Source: The author (2021)

4.2 COLLABORATIVE FILTERING RECOMMENDATION SYSTEMS

Table 32 is relative to the experiments for collaborative filtering based recommendation systems. Column *Items used* contains the number of items used to build each user profile, column $NDCG_{10}$ *Proposed* shows the $NDCG_{10}$ values obtained by the HD-CFR method, column $NDCG_{10}$ *Reference* contains the $NDCG_{10}$ values obtained by the reference method, CF-SDA, and the last column contains the p-values from the one-sided paired Wilcoxon tests applied to the $NDCG_{10}$ values. Since all p-values are below 0.05, we conclude that HD-CFR

outperformed CF-SDA according to $NDCG_{10}$.

Table 32 – Average of $NDCG_{10}$ for collaborative filtering approach

Items used	HD-CFR	CF-SDA	p-value
20	0.8220	0.8055	0.0008
40	0.8286	0.8133	0.0012
60	0.8349	0.8139	0.0004
100	0.8494	0.8171	$3.2505e^{-13}$

Source: The author (2021)

The data presented in table 32 show that the proposed methodology presents better results compared to the CF-SDA method, by a smaller margin compared to the content-based approach.

As an example, recommendations were generated for another random user applying the collaborative filtering methodologies, CF-SDA and HD-CFR, with the different numbers of items used to build the user profiles (20, 40, 60 and 100 items). In order to test the quality of the recommendations generated a set of 200 movies and ratings, given by this user, was used.

Table 33 shows the top 10 movies ranked list generated by CF-SDA, when the user profile was constructed using 20 items, the $NDCG_{10}$ obtained for this ranked list was 0.7926. In contrast, the Table 34 shows the top 10 movies ranked list using the methodology HD-CFR, the $NDCG_{10}$ obtained for this list was 0.8386. The $NDCG_{10}$ difference shows that the ranked list presented by Table 34 has a better quality than the one presented in Table 33.

Table 33 – Recommendation list produced by CF-SDA - Using 20 items to build the profile

Title	Rating
Monsoon Wedding	4
The Million Dollar Hotel	5
The Conversation	5
Trois couleurs	5
The 39 Steps	4
La passion de Jeanne d'Arc	4
Bridge to Terabithia	4
A Nightmare on Elm Street	3
Die Hard 2	4
Back to the Future Part II	5

Source: The author (2021)

Table 34 – Recommendation list produced by HD-CFR - Using 20 items to build the profile

Title	**Rating**
Terminator 3	5
The Million Dollar Hotel	5
Sleepless in Seattle	4
Men in Black II	4
Sissi	4
Bridge to Terabithia	4
Blood: The Last Vampire	4
Point Break	5
Notes on a Scandal	5
Once Were Warriors	5

Source: The author (2021)

Moving forward with the example, Table 35 shows the top 10 movies ranked list generated by CF-SDA, when the user profile was constructed using 40 items, the $NDCG_{10}$ obtained for this ranked list was 0.8231. In contrast, the Table 36 shows the top 10 movies ranked list using the methodology HD-CFR, the $NDCG_{10}$ obtained for this list was 0.8348. The $NDCG_{10}$ difference shows that the ranked list presented by Table 36 has a better quality than the one presented in Table 35.

Table 35 – Recommendation list produced by CF-SDA - Using 40 items to build the profile

Title	Rating
The 39 Steps	5
Terminator 3	5
Blood: The Last Vampire	4
Monsoon Wedding	4
A Nightmare on Elm Street	3
Sleepless in Seattle	3
The Conversation	5
Trois couleurs	5
The Million Dollar Hotel	5
Fools Rush In	2

Source: The author (2021)

Next, Table 37 shows the top 10 movies ranked list generated by CF-SDA, when the user profile was constructed using 60 items, the $NDCG_{10}$ obtained for this ranked list was 0.7999. In contrast, the Table 38 shows the top 10 movies ranked list using the methodology HD-CFR,

Table 36 – Recommendation list produced by HD-CFR - Using 40 items to build the profile

Title	Rating
The Million Dollar Hotel	5
Bridge to Terabithia	4
Solaris	5
Rope	4
Light of Day	3
Monsoon Wedding	4
Terminator 3	5
Men in Black II	4
A Nightmare on Elm Street	3
Rain Man	4

Source: The author (2021)

the $NDCG_{10}$ obtained for this list was 0.84. The $NDCG_{10}$ difference shows that the ranked list presented by Table 38 has a better quality than the one presented in Table 37.

Table 37 – Recommendation list produced by CF-SDA - Using 60 items to build the profile

Title	Rating
The 39 Steps	5
The Million Dollar Hotel	5
Monsoon Wedding	4
Trois couleurs	5
Men in Black II	4
A Nightmare on Elm Street	3
The Conversation	5
Bridge to Terabithia	4
Blood: The Last Vampire	4
Fever Pitch	5

Source: The author (2021)

Finally, Table 39 shows the top 10 movies ranked list generated by CF-SDA, when the user profile was constructed using 100 items, the $NDCG_{10}$ obtained for this ranked list was 0.7999. In contrast, the Table 40 shows the top 10 movies ranked list using the methodology HD-CFR, the $NDCG_{10}$ obtained for this list was 0.8333. The $NDCG_{10}$ difference shows that the ranked list presented by Table 40 has a better quality than the one presented in Table 39.

Also the $NDCG_{10}$ distribution was evaluated. The analysis of the graphs present in figures 6, 7, 8, and 9 leads to a similar conclusion, there is an improvement in the metric, but with a

Table 38 – Recommendation list produced by HD-CFR - Using 60 items to build the profile

Title	Rating
Sleepless in Seattle	3
Terminator 3	5
The Million Dollar Hotel	5
Men in Black II	4
Point Break	5
Bridge to Terabithia	4
Monsoon Wedding	4
Blood: The Last Vampire	4
Rope	4
Fools Rush In	2

Source: The author (2021)

Table 39 – Recommendation list produced by CF-SDA - Using 100 items to build the profile

Title	Rating
The 39 Steps	5
Monsoon Wedding	4
The Million Dollar Hotel	5
Blood: The Last Vampire	4
The Conversation	5
When Saturday Comes	4
Terminator 3	5
La passion de Jeanne d'Arc	4
A Nightmare on Elm Street	3
Die Hard 2	4

Source: The author (2021)

slightly smaller margin. As in the previous approach, to ensure that the data actually differed a Wilcoxon test was applied to the $NDCG_{10}$ values obtained during the experiment.

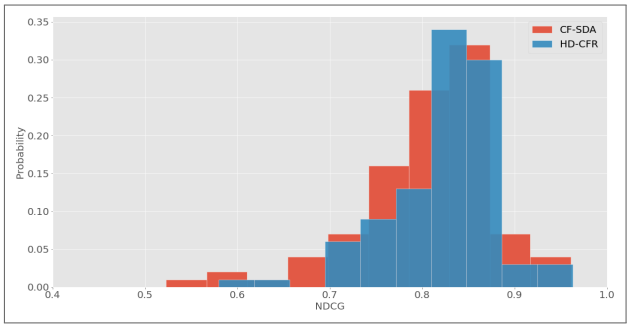
Thus, considering the experiments presented, it can be seen that the method proposed in this work is able to produce more efficient recommendations for users. In addition to making use of unique profiles, present a single profile type for both content-based and collaborative filter-based approaches and have a more intuitive similarity metric.

Table 40 – Recommendation list produced by HD-CFR - Using 100 items to build the profile

Title	Rating
Solaris	5
Terminator 3	5
Men in Black II	4
Bridge to Terabithia	4
Monsoon Wedding	4
Three Colors	5
Sleepless in Seattle	3
Young and Innocent	5
Sissi	4
Aguirre: The Wrath of God	4

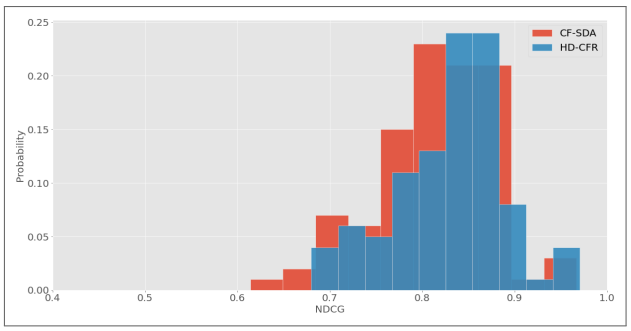
Source: The author (2021)

Figure 6 – Collaborative filtering approach with profile built using 20 items



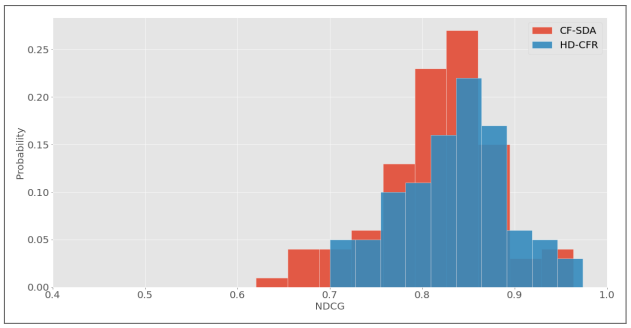
Source: The author (2021)

Figure 7 – Collaborative filtering approach with profile built using 40 items



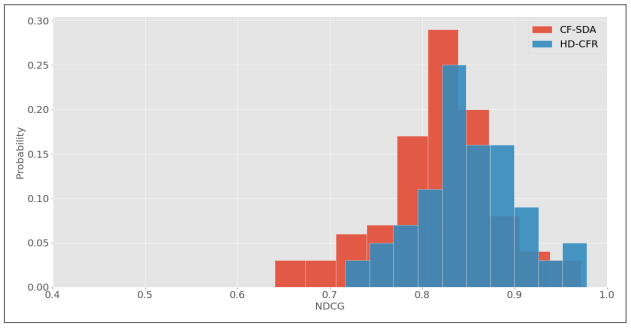
Source: The author (2021)

Figure 8 – Collaborative filtering approach with profile built using 60 items



Source: The author (2021)

Figure 9 – Collaborative filtering approach with profile built using 100 items



Source: The author (2021)

5 CONCLUSION

Recommendation systems have been a well-studied topic since the popularization of the internet in the early 1990s due to its many applications in online businesses. This topic still remains relevant due to the large amount of information used in online services. In addition symbolic data was chosen due to its ability to model variability of real-world concepts. So both are topics of practical interest.

Even though there is already an approach to make use of symbolic data by recommendation systems, this work seeks to develop a simpler technique, which uses unique profile building strategies, that can be used in different approaches to build recommendation systems and also achieve better results. Besides this, the collaborative filtering developed in this work takes into account the similarity between user preferences, thus grouping users with similar preferences, unlike current implementations that aggregate users based only on the values assigned to the items evaluated.

For that, modeling strategies were developed for the items present in the database and user profiles, which could be used both for recommendation systems based on content and for recommendation systems based on collaborative filters. To measure the dissimilarity between items and user profiles, the Wasserstein distance is used as part of the similarity calculations between users.

The experiments carried out show that the developed methods present recommendation lists with more relevant items in the first positions, both for recommendation systems based on content and for systems based on collaborative filtering.

The contributions of this work were:

- A methodology to build content-based recommendation systems.
- A methodology to build collaborative filtering recommendation systems.
- Both methodologies use a single profile for each user.
- A paper that will be submitted to a specialized journal.

5.1 FUTURE WORKS

During the development of this work, some points of improvement were identified and ideas for future work as well. The first point of improvement observed is the study of different ways to combine recommendation systems, aiming to aggregate content-based systems and collaborative filters in order to overcome their individual flaws.

Other possible improvement point is to analyze the influence of different clustering techniques in the collaborative filtering-based recommendation systems, since different clustering techniques might create better neighborhoods for users.

Future works also include comparisons between recommendation systems using symbolic data and systems built using conventional data.

Finally, taking a data centered approach, we can use deep learning models for natural language inference (MLI) to extract symbolic features from review texts, corresponding to the possible topics covered in the text. This could lead to a large dataset for symbolic data analysis, by extracting possible topics from user reviews and turning these topics into symbolic descriptions.

REFERENCES

ADOMAVICIUS, G.; TUZHILIN, A. 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*, v. 17, n. 6, p. 734–749, 2005.

AFSAR, M.; CRUMP, T.; FAR, B. Reinforcement learning based recommender systems:a survey. 2021.

ALMAZO, D.; SHAHATAH, G.; ALBDULKARIM, L.; KHEREES, M.; MARTINEZ, R.; NZOUKOU, W. A survey paper on recommender systems. 2010.

BEZERRA, B. L. D.; CARVALHO, F. de A.T. de. A symbolic approach for content-based information filtering. *Information Processing Letters*, v. 92, p. 45–52, 10 2004.

BEZERRA, B. L. D.; CARVALHO, F. de A.T. de. Symbolic data analysis tools for recommendation systems. *Knowl. Inf. Syst.*, v. 26, p. 385–418, 03 2010.

BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent dirichlet allocation. *J. Mach. Learn. Res.*, JMLR.org, v. 3, n. null, p. 993–1022, mar. 2003. ISSN 1532-4435.

BOBADILLA, J.; ORTEGA, F.; HERNANDO, A.; BERNAL, J. A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-Based Systems*, v. 26, p. 225–238, 2012. ISSN 0950-7051. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0950705111001882>.

BREESE, J. S.; HECKERMAN, D.; KADIE, C. Empirical analysis of predictive algorithms for collaborative filtering. In: *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998. (UAI'98), p. 43–52. ISBN 155860555X.

BREESE, J. S.; HECKERMAN, D.; KADIE, C. M. Empirical analysis of predictive algorithms for collaborative filtering. *CoRR*, abs/1301.7363, 2013. Disponível em: <http://arxiv.org/abs/1301.7363>.

BURKE, R. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, v. 12, 11 2002.

ÇANO, E.; MORISIO, M. Hybrid recommender systems: A systematic literature review. *CoRR*, abs/1901.03888, 2019. Disponível em: <http://arxiv.org/abs/1901.03888>.

CHEN, R.; HUA, Q.; CHANG, Y.-S.; WANG, B.; ZHANG, L.; KONG, X. A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks. *IEEE Access*, v. 6, p. 64301–64320, 2018.

DIDAY, E. An introduction to symbolic data analysis and the sodas software. *Intelligent Data Analysis*, 2003.

DIDAY, E.; BILLARD, L. Symbolic data analysis: Conceptual statistics and data mining. *Symbolic Data Analysis: Conceptual Statistics and Data Mining*, 12 2006.

DIDAY, E.; BOCK, H.-H. *Analysis of Symbolic Data Exploratory Methods for Extracting Statistical Information from Complex Data*. [S.l.]: Springer-Verlag Berlin Heidelberg, 2000. ISBN 978-3-642-57155-8.

DIDAY, E.; NOIRHOMME-FRAITURE, M. *Symbolic Data Analysis and the SODAS Software*. USA: Wiley-Interscience, 2008. ISBN 0470018836.

FELFERNIG, A.; LE, V.; POPESCU, A.; UTA, M.; TRAN, T. N. T.; ATAS, M. An overview of recommender systems and machine learning in feature modeling and configuration. *VaMoS, 2021*, 2021.

GOLDBERG, D.; NICHOLS, D.; TERRY, B. M. O. e D. Using colaborative filtering to weave an information tapestry. *Communications of the ACM*, v. 35, p. 61–70, 1992.

GROUPLENS. *Eachmovie dataset*. 2022. Disponível em: <https://grouplens.org/datasets/eachmovie/>.

HOFMANN, T. Latent semantic models for collaborative filtering. *ACM Trans. Inf. Syst.*, Association for Computing Machinery, New York, NY, USA, v. 22, n. 1, p. 89–115, jan. 2004. ISSN 1046-8188. Disponível em: <https://doi.org/10.1145/963770.963774>.

JäRVELIN, K.; KEKäLäINEN, J. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, Association for Computing Machinery, New York, NY, USA, v. 20, n. 4, p. 422–446, out. 2002. ISSN 1046-8188. Disponível em: <https://doi.org/10.1145/582415.582418>.

KOREN, Y.; BELL, R.; VOLINSKY, C. Matrix factorization techniques for recommender systems. *Computer*, v. 42, n. 8, p. 30–37, 2009.

KOTKOV, D.; VEIJALAINEN, J.; WANG, S. Challenges of serendipity in recommender systems. In: *WEBIST*. [S.l.: s.n.], 2016.

LI, Q.; KIM, B. M. An approach for combining content-based and collaborative filters. In: *in Proceedings of the Sixth international workshop on Information retrieval with Asian languages (ACL-2003*. [S.l.]: In Press, 2003. p. 17–24.

MAHMOOD, T.; RICCI, F. Improving recommender systems with adaptive conversational strategies. In: *HT '09: Proceedings of the Twentieth ACM Conference on Hypertext and Hypermedia*. New York, NY, USA: ACM, 2009.

MCKINNEY, W. pandas: a foundational python library for data analysis and statistics. *Python High Performance Science Computer*, 01 2011.

OLIPHANT, T. Python for scientific computing. *Computing in Science Engineering*, v. 9, p. 10–20, 06 2007.

RESNICK, P.; VARIAN, H. R. Recommender systems. *COMMUNICATIONS OF THE ACM*, 1997.

RICCI, F.; ROKACH, L.; SHAPIRA, B.; KANTOR, P. B. *Recommender Systems Handbook*. New York: Springer, 2011. ISBN 978-0-387-85819-7.

RICCI, F.; WERTHNER, H. Introduction to the special issue: Recommender systems. In: *International Journal of Electronic Commerce*. [S.l.: s.n.], 2006.

SALAKHUTDINOV, R.; MNIH, A.; HINTON, G. Restricted boltzmann machines for collaborative filtering. In: *Proceedings of the 24th International Conference on Machine Learning*. New York, NY, USA: Association for Computing Machinery, 2007. (ICML '07), p. 791–798. ISBN 9781595937933. Disponível em: <https://doi.org/10.1145/1273496.1273596>.

SHETH, B.; MAES, P. Evolving agents for personalized information filtering. In: *Proceedings of 9th IEEE Conference on Artificial Intelligence for Applications*. [S.l.: s.n.], 1993. p. 345–352.

SILVA, J. F. G. da; JUNIOR, N. N. de M.; CALOBA, L. P. Effects of data sparsity on recommender systems based on collaborative filtering. In: *2018 International Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2018. p. 1–8.

VIRTANEN, P.; GOMMERS, R.; OLIPHANT, T.; HABERLAND, M.; REDDY, T.; COURNAPEAU, D.; BUROVSKI, E.; PETERSON, P.; WECKESSER, W.; BRIGHT, J.; WALT, S.; BRETT, M.; WILSON, J.; MILLMAN, K.; MAYOROV, N.; NELSON, A.; JONES, E.; KERN, R.; LARSON, E.; VáZQUEZ-BAEZA, Y. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature Methods*, v. 17, p. 1–12, 02 2020.

WEI, K.; HUANG, J.; FU, S. A survey of e-commerce recommender systems. International Working Conference on Variability Modelling, 2021.

ZHANG, Y.; CALLAN, J.; MINKA, T. Novelty and redundancy detection in adaptive filtering. In: *SIGIR '02*. [S.l.: s.n.], 2002.