

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Estado de México

School of Engineering and Sciences



**Hybrid Recommender System for a Context Aware Recommendation
in the Film Domain**

A thesis presented by

Nora Patricia Hernández López

Submitted to the
School of Engineering and Sciences
in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Science

Atizapán de Zaragoza, Estado de México, Jun, 2020

To my mom.

Acknowledgements

I wish to express my sincere appreciation and gratitude to my supervisor, Dr. Jorge Uresti for his patience and wise guidance. He encouraged me throughout this project and always motivated me to be better each time.

I want to thank all the people whose assistance was a milestone in the completion of this project. First and foremost to Isabel and Héctor, who sacrificed time and priceless effort for a somewhat unconventional endeavor. Without you, this project would definitely not have been possible. To Gina, for her enthusiasm and friendship; thank you for letting me stole a few minutes of the best movie talks I could ever been part of.

I wish to acknowledge the great love and support my family always give me. To my mom, for being my greatest source of inspiration; if life would have been different, I probably would not be writing this. To my dad, for always encouraging my crazy ideas and offering a helping hand. To my sister, my deepest admiration for showing me how to be brave and always follow my dreams.

I would like to recognize the invaluable support I received from my lifetime friends, and the friends I found in the past two years. To Maru, for being the best listener and always being there whenever I felt my spirit weakened; I treasure every moment we've been together and the thought-provoking endless talks. To Polo, for standing by me in my toughest times and offering his kind friendship from the first day. To Ana, for being like my sister; I've always found kindness and wise advice when I came to you. To Arturo, the best friend I could ever wish; thank you for all the tea sessions, music exchange and enlightening talks. To all my fellow classmates, and specially to my dorm mates, for the countless laughs and amazing shared experiences.

Last but not least, I would like to thank Tecnológico de Monterrey and CONACyT for the grant and funding for completing the master's program. This opportunity has changed my life.

Hybrid Recommender System for a Context Aware Recommendation in the Film Domain

by

Nora Patricia Hernández López

Abstract

Recommendation systems aim to offer personalized help in discovering relevant content. Several approaches have been designed for providing better recommendations that satisfy users' needs. Based on ratings, on content, or on knowledge, isolated recommendation techniques often lack some good properties of other methods. Hence, hybrid combinations are able to compensate for those differences. Furthermore, the information to include in the recommendation is most of the time limited to the set of ratings users assigned to the items. By including additional information on where and when the recommendation is taking place, can improve the overall performance. Nevertheless, combining all these features into one single model is rather a daunting task due to its complexity, and often is disregarded as it might require some degree of domain knowledge.

We propose a recommender system based on a model that captures the human understanding of how to produce a personalized recommendation. Moreover, by including context information, we try to enhance the overall user's experience. This system is able to produce recommendations even under uncertainty. Hence, we used an explicit model which is in fact a Bayesian network, that directly encodes the relationships between users' preferences, item attributes, and context information. The final recommendation is obtained by a two stage process, a combination of two recommendation strategies that complement each other. Such model is the Contextual Hybrid Bayesian Model.

List of Figures

2.1	Simple Bayesian network	11
2.2	Knowledge Engineering with Bayesian Networks (KEBN) methodology .	20
3.1	Matrix relating users and items through ratings	23
5.1	Structures of the different Bayesian Networks generated using bootstrapping	50
5.2	Bayesian Network defined by the experts	52
5.3	BN obtained with ML methods and expert knowledge assessment	54
5.4	BN structure learned from data using HC algorithm with AIC	56
5.5	Loss results from the cross validation with different scores for the models	58
5.6	CMPR obtained with ML methods including the original variables	59
5.7	Arc strength for the model defined by film experts	60
5.8	Arc strength for the model obtained with expert knowledge as seed	61
5.9	Arc strength for the model learned from data and vague expert assessment	62
5.10	Arc strength for the CMPR	63
5.11	ROC curves for the different models	65
5.12	Cascade Hybrid Model for Personalized Recommendations	70
6.1	Beta version of the user interface	73
6.2	CineScope: User interface developed to test the CHYBAM	74
6.3	Comparison between CineScope and MovieLens results for scalar variables	79
6.4	<i>How motivated</i> are users to watch at least some of the recommendations?	80
6.5	Evaluation of <i>goodness</i> of the recommendations	81
6.6	Perception of the influence of the context	82
6.7	Overall perceived relevance of a CARS	82
6.8	Equivalence test for the observed effect in <i>attractiveness</i>	85

List of Tables

3.1	Metrics used to assess prediction accuracy in recommendation systems . . .	31
4.1	Hybrid implementations in the literature	35
4.2	Summary comparison	40
5.1	Total number of observations for the different stages of data collection . .	44
5.2	Variables included in the different models	46
5.3	Relevant measures for comparing the different graphical models	64
5.4	Loss and AUC values for the different models	66
5.5	Variables to include in the conditional probability query of each BN . . .	67
6.1	Pilot study results and expected measures	77
6.2	Comparison of RS choice	83
6.3	Summary statistics for the tests comparing the perceived attributes of the RSs	84
A.1	Experimental process defined for the study	99

Contents

Abstract	v
List of Figures	vi
List of Tables	vii
1 Introduction	1
1.1 Problem Statement and Motivation	1
1.2 Hypothesis and Research Questions	3
1.3 Objective	4
1.4 Scope	5
1.5 Thesis contribution	5
1.6 Document organization	6
2 Probabilistic Graphical Models	8
2.1 Probabilistic Reasoning	8
2.1.1 Notion of dependence and Bayesian inference	9
2.2 Representation	10
2.3 Inference	12
2.4 Learning	13
2.4.1 Constraint-based Structure Learning Algorithms	14
2.4.2 Score-based Structure Learning Algorithms	15
2.4.3 Hybrid Structure Learning Algorithms	16
2.4.4 Parameter Learning	17
2.5 Model elicitation	19
3 Recommender Systems	21
3.1 General definition	21
3.2 Common approaches	22

3.3	Context Aware Recommender Systems	24
3.3.1	Definition	24
3.3.2	Context Representation	25
3.3.3	Common approaches	26
3.4	Hybrid Recommender Systems	27
3.4.1	Common approaches	27
3.5	Evaluation	29
3.5.1	Measuring recommendation system properties	29
3.5.2	User centric evaluation	30
4	Related Work	33
4.1	Comprehensive literature review	33
4.1.1	Hybrid implementations	33
4.1.2	Context aware systems	36
4.1.3	Bayesian networks in recommendation systems	37
4.1.4	Miscellaneous	38
4.1.5	Summary	39
4.2	Recommendation strategies used as reference	40
5	A Context Hybrid Bayesian Model for Recommendations	42
5.1	Expert knowledge elicitation	43
5.2	Data acquisition	44
5.3	Data pre-processing	45
5.4	Development of the models	49
5.5	Knowledge-based models	51
5.5.1	Pure expert knowledge	51
5.5.2	Expert knowledge as seed	53
5.5.3	Vague expert knowledge	55
5.6	Replication of the CMPR	57
5.7	Analysing the models	60
5.8	Evaluation	63
5.8.1	Measures from graphical models	63
5.8.2	Measures from algorithmic performance	64
5.9	Inference to obtain recommendations	66
5.10	Hybridization Method	68

6	Experiments and Results	71
6.1	User interface for testing the CHYBAM	71
6.1.1	Early versions	72
6.1.2	CineScope	73
6.2	Experiment setup	75
6.3	Results	77
6.3.1	Statistical analysis	83
6.4	Discussion	85
7	Conclusion and Future Work	88
7.1	Conclusion	88
7.2	Limitations	89
7.3	Future Work	90
A	Questionnaires	93
A.1	Questionnaire for initial data collection	93
A.2	User evaluation	98
A.2.1	Experimental procedure	98
A.2.2	Questionnaire for evaluating CineScope	99
A.2.3	Questionnaire for evaluating MovieLens	101
A.2.4	Questionnaire for comparing CineScope and MovieLens	101
	Bibliography	112

Chapter 1

Introduction

Recommendation systems aim to offer personalized help in discovering relevant content [53]. Several approaches have been designed for providing better recommendations that satisfy users' needs. Based on ratings, on content, or on knowledge, isolated recommendation techniques often lack some good properties of other methods. Hence, hybrid combinations are able to compensate for those differences. Furthermore, the information to include in the recommendation is most of the times limited to the set of ratings users assigned to the items. By including additional information on where and when the recommendation is taking place, can improve the overall performance. Nevertheless, combining all these features into one single model is rather a daunting task due to its complexity, and often is disregarded as it might require some degree of domain knowledge.

This chapter presents the introduction to the main topics that guide the research. The statement of the problem and motivation for the thesis is presented in Section 1.1. The hypothesis and research questions are described in Section 1.2, followed by the definition of the objectives in Section 1.3. Section 1.4 delimits the scope of the research, and Section 1.5 details the contribution of the thesis. Finally, Section 1.6 briefly outlines the content of the following chapters.

1.1 Problem Statement and Motivation

A recommendation system is most commonly described as a tool that help users in finding what they need given the overwhelming amount of information available in the Internet [53, 84]. Some have even stated that the Internet has not only brought us more information and choice, but has also increased the burden of making a choice [13]. To address this problem, several techniques have been proposed to describe user's needs and

how to address them. Some of these techniques attempt to give recommendations based on similarities among users, others on the similarities among items [87], maybe some consider the content of the item, or the knowledge about how the user and the item relate to each other beyond the rating.

Nonetheless, the majority of these rely on a score assigned by the user to an item [45]. This rating-system schema has produced a generalized trend to create algorithms more powerful every time, capable of computing hundreds, maybe thousands, of thousands of user-item-rating tuples in order to more accurately predict user's rating on a item. But as McNee [64] beautifully puts it, *being accurate is not enough*. Moreover, this paradigm creates systems that penalize the discovery of new items, and rewards on the other hand systems that find items the user already likes. Prediction-based recommendations contradict the exact objective of recommender systems, because if a recommender tells the user what she already knows, then it is not a *good* recommender.

It is clear that the goal of a recommender system nowadays goes beyond of the traditional definition found in the literature. The information overload described earlier to be a burden on the user, is now a threat for streaming services. Competitive technologies must be developed in order to engage an audience. Perhaps the best known example is the Netflix Prize [54], which granted 1 million dollars for an algorithm that could improve the accuracy metric in the well known streaming service. Similarly, Google recently made public the revenue reported from YouTube's advertising [60], which directly comes from users effectively consuming the content, powered by engaging recommendations.

To alleviate this problem, alternative approaches have been studied. From considering the personality of the users [43] to explaining the recommendations [24], researchers have tried to offer more understandable recommendations for the user. This can be understood as an attempt to emulate the real-life process of seeking advise from trusted sources [13], but yet users struggle to convert the information into meaningful actions, as most of the items in a recommendation remain unexplored [111].

Evaluation centered on users' experience has proven to yield better results and easier adoption [50]. After all, regardless of what many might think, users do know what to expect from a recommendation, and are more than capable to identify what is missing. When asked how to improve the recommendations in the well regarded movie recommendation provider, MovieLens, users identified that considering the mood could yield better experience when browsing recommendations [59].

The mood, and many other additional factors can be introduced into the recommendation via context variables. These might include location, companion, weather, among other relevant information. Context aware systems can adapt the recommendation to the specific setting of the user, and offer more reliable information.

Another technique used to improve the recommendations is the product of the

combination of two or more different techniques, each of which compensate for the weaknesses of the other. This is better known as a hybrid recommender system, and usually aims to solve the ramp-up and/or the cold start problem [17]. The first one refers to the impossibility of a system to recommend a new item, and the later refers to the problem of crafting recommendations for new users.

With the method proposed in this research, we aim to generate a recommender system based on a model that captures the human understanding of how to produce a personalized recommendation. Moreover, by including context information, we try to enhance the overall user's experience. This system is able to produce recommendations even under uncertainty. Hence, we used an explicit model which is in fact a Bayesian network, that directly encodes the relationships between users' preferences, item attributes, and context information. The final recommendation is obtained by a two stage process, a combination of two recommendation strategies that complement each other. Such model is the Contextual Hybrid Bayesian Model.

1.2 Hypothesis and Research Questions

We propose a hybrid recommender system that combines two recommendation strategies, one derived from expert-knowledge and the other obtained with a data-driven approach. This hybrid model describes the user and her interactions with the items given the context. The recommendations produced by the hybrid system will be able to reflect the specific context setting in which the interaction user-item is taking place.

Will the proposed recommendation technique produce attractive recommendations that are equivalent to the ones from traditional recommendation methods? The main hypothesis of this research is to investigate if the mean attractiveness of the recommendations provided with the new method, as perceived by users, are equivalent, within a statistically significant confidence interval, to the ones from other recommendation methods. To test the hypothesis, we will measure how appealing the recommendations provided by the proposed strategy are, thus assessing user satisfaction, to verify if they are statistically equivalent to those provided by traditional recommendation strategies, which are neither hybrid nor context aware.

We establish the following research questions:

1. What are the causal relations between users' preferences and specific context variables? *This will be addressed by domain experts.*
2. What are the strengths of each of the recommendation models proposed that will serve to compensate for the deficiencies of the other? *Once we have the models,*

we will be able to analyze the them to characterize their capabilities to produce recommendations and distinguish such advantages.

3. What will be the best method to combine the models into one hybrid system? *Based on the previous answer, relevant methods will be evaluated.*
4. How accurately will the model predict user's preferences for an item? *When the hybrid model is completed, we will be able to evaluate it exhaustively both for accuracy metrics and subjective evaluation by the users.*
5. How relevant will the final recommendation be for the user given a specific context? *A comprehensive study will be designed to test the perceived goodness of the recommendation, as well as the relevance of the context.*

1.3 Objective

The main objective of this research is to generate a user model based on the knowledge and ability of an expert in the film domain, such that it can be used to generate personalized recommendations for a specific user request, considering context variables.

The specific objectives are:

1. Generate a data base to support the recommendation process. *It will be obtained from a questionnaire designed by the domain experts to collect explicit information about the users' movie-going past experiences and preferences.*
2. Create a knowledge-based model assessed by the experts, which establishes the causal relationships between users' characteristics, items and context variables. *The model elicitation process will determine the core structure of the Bayesian network.*
3. Analyze the models to discover the similarities between the models, and to identify the weaknesses of each model. *Based on an exhaustive performance evaluation, the recommendation capabilities for each model will be determined.*
4. Combine the models to create a hybrid system able to provide context aware recommendations. *The strengths of each model will help us determine the best hybridization method based on the output of each model.*
5. Obtain feedback from the users to evaluate the model, and assess the real relevance of the recommendation provided. *This will allow us to determine the true advantages of the recommendation system as perceived by the users.*

1.4 Scope

This research focus on developing a hybrid model capable of producing recommendations in the film domain. The recommendations will be the product of a two stage process. The first output are recommended genres, which are used to retrieve movies that match that specific combination of genres (intermediate stage), and then those movies are *scored* by eliciting the probability for each of their genres. All the stages are performed by direct inference with Bayesian networks. No further personalization will be provided beyond the possibility to include or exclude some specific genres for the intermediate stage. There is no user profile available to store user's interactions with the application.

The evaluation of the system will be performed by final user evaluation. The aspects to evaluate will only consider the ease of use, the attractiveness of the recommendations, and how the context is perceived. The study does not consider the visual design of the application, or any other design element. Neither any aspect regarding the intentions or personal motivations and interest of the users will be considered in the evaluation. The evaluation process was designed with an informed background and specific goals, but was not assessed by any expert on human-computer interaction.

The samples, both for collecting the data set and evaluating the system, belong to the college community, mostly composed by students within 18 and 21 years old, and from the upper and upper-middle class in the metropolitan area of Mexico City. Some of the participants in the study (evaluation) also participated in the first stage of data collection. Some of the participants, both for data collection and final evaluation, are part of a film club.

1.5 Thesis contribution

To fully describe the contributions of the method proposed in this document, it is convenient to remark the individual strengths of each of its components. First, we must stress the importance of the data set, which was constructed specifically to address the specific objectives of the approach. It includes context-specific information about how users interact with movies, and can be interpreted as a description of their movie-going experience. With (slightly) more than 800 observations, it provides a solid ground to draw rich conclusions, from which the subsequent steps in the recommendation rely. The complete data set is available at https://bit.ly/cinescope_data.

Second of all, the recommendation approach is based on context. It provides recommendations that adapt to different situations, and aims to satisfy specific needs accordingly. It was proven that the model is highly context responsive, and can produce adequate recommendations for each scenario. Even more, experimental results show that

users find relevant the consideration of the context into the recommendation, which strengthen our prior assumption.

In addition, the hybrid structure of the model allows to combine two stages in the recommendation, each of which focus on different aspects of the same recommendation. In the first step, the system obtains the recommended genres, so that the second step *rates* an appropriate set of films matching these characteristics. The cascaded method is proposed as a strategy to enhance the final set of recommended items, and provide a set that fully reflects the interests and needs of the user.

Moreover, each of the recommendation stages is performed by a Bayesian network. The nature of Bayesian networks facilitates the identification of casual dependencies between the variables explicitly. The core relationships that must be captured by the network were elicited by domain experts, which provides a true understanding of how the variables interact. Additionally, based on the probabilistic distributions obtained from the data set, Bayesian networks provide a method to obtain information under uncertainty. Through reliable inference methods, the recommendation is based on the direct relationships described in the structure of the network.

All of these elements are encompassed into one single model able to produce recommendations based on contextual factors. The interactions between these factors, user's characteristics and item information are contained explicitly in a Bayesian model which provides the methods to produce the recommendation based on the joint probability distribution of the underlying data.

Despite the hybrid recommendation model was tested exclusively in the film domain, with promising results, the main contribution we recognize is that of the model to be domain-independent. Such approach can be applied to any domain by following the same procedure described along this document. The final product will be a Context Hybrid Bayesian Model (CHYBAM) that captures the relevant variables' dependencies in the domain of choice, and capable of producing recommendations accordingly.

1.6 Document organization

This document delves in the development of a context aware hybrid recommender system, which has been specifically tested in the film domain. The theoretical foundations, as well as all the stages involved in the research process are fully described in the following chapters. More precisely, the content of each chapter is as follows:

- **Chapter 1:** Provides a general overview of the relevance of the research, including the hypothesis and objectives. A brief presentation of the problem and previous works in the area is introduced to recognize the relevance of the proposed solution.

- **Chapter 2:** Presents the theory behind probabilistic graphical models, and more specifically, Bayesian networks. The concepts introduced here will allow us to properly develop the model, from which useful and relevant information can be drawn to generate the recommendation.
- **Chapter 3:** Similar to Chapter 2, this chapter introduces the key concepts concerning recommendation systems, its core characteristics, and the general approaches in the field. Moreover, the techniques and procedures to conduct a proper evaluation of the system are described here.
- **Chapter 4:** Discusses the relevant works that have been developed to solve the same problem as we do, and performs a comprehensive analysis of their strengths and similarities to our proposed solution.
- **Chapter 5:** Explains in its entirety the model developed to produce recommendations based on context information, and a combination of two stages in the recommendation process. We have named this method *Contextual Hybrid Bayesian Model*.
- **Chapter 6:** Describes the experimental procedure implemented to evaluate model with a user-centric strategy, followed by a extensive report of the findings.
- **Chapter 7:** Examines carefully the results of the research process, evaluating the advantages and main restraints of the application. This serves to determine the key factors that could be improved, and the importance on continuing the work.

Chapter 2

Probabilistic Graphical Models

Probabilistic graphical models are a representation of a set of probabilistic distributions. The main definitions required to formally describe a Bayesian network will be described in this chapter.

First, the basic notions of probabilistic inference and dependence will be presented in Section 2.1. Details on the main characteristics of Bayesian networks are described in section 2.2. The processes for obtaining information from the model as a form of probabilistic inference are examined in Section 2.3. In Section 2.4, the available methods for finding the best structure and parameters of the network are discussed. Finally, Section 2.5 describes the produces for effectively producing reliable Bayesian models when domain knowledge is available.

2.1 Probabilistic Reasoning

In many of the real-world decision scenarios it is necessary to deal with randomness and uncertainty. The intrinsic uncertainty^a about the *real* state of the world, imposes on us the need not to only consider what is possible, but also what is probable, to draw meaningful conclusions. Probability theory is by far the most studied and well defined tool for dealing with these questions.

With clear semantics that allow a declarative representation, probability theory is useful to find powerful reasoning patterns by conditioning under the available evidence.

^aQuantum mechanics aside

2.1.1 Notion of dependence and Bayesian inference

Perhaps the best known formula in probability theory is the inversion formula, and is in fact the core of the Bayesian methods [75].

$$P(H|e) = \frac{P(e|H)P(H)}{P(e)} \quad (2.1)$$

states that the belief about hypothesis H based on the evidence e can be computed by the *prior* belief $P(H)$ and the *posterior* (likelihood) $P(e|H)$ that e will be automatically true if H happens to be true. The denominator $P(e)$ is a normalizing factor that assures that $P(H|e) + P(\neg H|e) = 1$. Also can be computed by

$$P(e) = P(e|H)P(H) + P(e|\neg H)P(\neg H) \quad (2.2)$$

Nevertheless, human performance cannot handle computing numerical representations of probabilistic information. Simple tasks as computing the impact of a piece of evidence over a hypothesis, as in Eq. 2.1, require immense amount of computations that do not rely on any familiar mental processes [76].

For example, the illustrative proposition by Korb and Nicholson [73] in which the goal is to compute the probability of having cancer given a positive test result, and assuming that the long term probability of having cancer is 0.01, the test used in the detection has a 0.2 false positive rate, and a 0.1 false negative rate, leads to the prompt conclusion that a positive test given cancer will occur with 90% probability. The later is then refuted by applying the proper mathematical computations:

$$\begin{aligned} P(\text{Cancer}|\text{Pos}) &= \frac{P(\text{Pos}|\text{Cancer})P(\text{Cancer})}{P(\text{Pos})} \\ &= \frac{P(\text{Pos}|\text{Cancer})P(\text{Cancer})}{P(\text{Pos}|\text{Cancer})P(\text{Cancer}) + P(\text{Pos}|\neg\text{Cancer})P(\neg\text{Cancer})} \\ &= \frac{0.9 \times 0.01}{0.9 \times 0.01 + 0.2 \times 0.99} \\ &= \frac{0.009}{0.009 + 0.198} \\ &\approx 0.043 \end{aligned}$$

Similar to this one, several examples address the notion of independence, which is not a natural formulation in everyday language. On the contrary, people can easily detect dependencies. For example, knowing the time of the last bus is undoubtedly relevant for assessing how long it's necessary to wait for the next one to come; however, knowing the location of the next bus, the previous knowledge provides no relevant information. Despite common judgements like this are made qualitatively, they reflect the notion of conditional dependency.

Therefore, any reasoning language used for representing probabilistic information must allow to express statements about dependency relationships explicitly, directly, and qualitatively [75]. Such is the nature of Bayesian networks.

2.2 Representation

Probabilistic models offer the possibility to liberate the need to formally specify each and every possibility, and consider raw approximations to model the behaviour of a complex system, which in fact results in a faithful representation of reality [52]. Specifically, probabilistic graphical models (PGM) use a graph based representation for describing compactly complex distributions over a high dimensional space.

One key property in many of these type of distributions is that variables tend to interact directly only with a few other variables. This allows a natural and direct encoding in a graphical model.

This leads to the definition of a Bayesian network as a directed acyclic graph (DAG) $\mathcal{G} = (V, E)$, in which the nodes V represent the random variables X_1, \dots, X_n in the domain, and the edges E can be intuitively identified as the influence of one variable over another. The structure satisfies both, the graphical representation of a joint distribution compactly factorized and, a compact representation for the conditional independence assumptions in the distribution.

Formally, a Bayesian network represents a joint distribution via the chain rule

$$P(X_1, \dots, X_n) = \prod_i P(X_i | \text{Par}_{\mathcal{G}}(X_i)) \quad (2.3)$$

where $\text{Par}_{\mathcal{G}}(X_i)$ are the parents of X_i in the graph \mathcal{G} . This assumption of independence of X_i from any other random variable which are not its parents, its what makes feasible to specify conditional probabilities and efficiently perform inference with Bayesian networks [44].

Figure 2.1 shows a simple Bayesian network. It encodes the joint distribution $P(A, B, C, D, E)$ such that

$$P(A, B, C, D, E) = P(B)P(E)P(A|B, E)P(C|E)P(D|A) \quad (2.4)$$

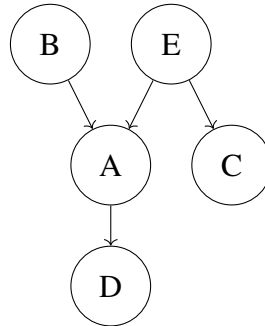


Figure 2.1: Simple Bayesian network

Equation 2.4 is the chain rule for Bayesian networks applied to the graph in Figure 2.1. It is possible to identify the three types of connections:

- **Serial:** $B \rightarrow A \rightarrow D$. This connections allow the flow of information from B to D and vice versa unless the state of A is known.
- **Diverging:** $A \leftarrow E \rightarrow C$. The flow of information flows in this connections from C to A , or vice versa, when E is unknown.
- **Converging:** $B \rightarrow A \leftarrow E$. Also known as V -structure, allows the flow of information only if information of A or one of its descendants is known.

These reasoning patterns, and the conditional independencies implied, can be generalized to formulate statements of relevance and irrelevance relations between two (sets of) variables given a third (set of) variables.

To further describe and analyze Bayesian networks, it is necessary to introduce the key concepts. The following paragraphs give proper definitions to these concepts.

d-separation

The d-separation criterion (d stands for directional) states that two vertices u and v are d-separated if for each path between them there is a vertex w such that the edges of the path meet head to head at w ; otherwise, u and v are d-connected [44].

Formally, a path $\pi = \langle u, \dots, v \rangle$ in a DAG $\mathcal{G} = (V, E)$ is said to be blocked by $S \subseteq V$ if π contains a vertex w such that either

- $w \in S$ and the edges of π do not meet head to head at w , or
- $w \notin S, desc(w) \cap S = \emptyset$ and the edges of π meet head to head at w

Markov Blanket

For Bayesian networks, the Markov blanket of a node u is said to be the set of parents of u , the children of u and all the other nodes sharing a child with u [66]. It can be understood as the smallest set of elements that shields u from the influence from other elements [76].

To be precise, the Markov blanket of a node $u \in V$ is the minimal subset S of V such that

$$u \perp\!\!\!\perp_P V - S - u | S \quad (2.5)$$

where $\perp\!\!\!\perp_P$ indicates probabilistic independence.

Neighborhood and branching factor

Given a directed graph $\mathcal{G} = (V, E)$, and a undirected graph $\mathcal{H} = (V, E')$, where $E' = \{u - v : u \rightleftharpoons v \in E\}$, whenever we have $u_i - u_j \in E$ we say that u_i is a neighbor of u_j in \mathcal{H} , and vice versa [52].

The branching factor refers to the number of descendants of a node- $desc(u)$, also known as out degree.

2.3 Inference

Inference over a BN is known to be a NP-hard problem [27], therefore many approaches have been made to perform approximate inferences over the networks. Given the characteristics of a the structure of a graphical model, inferences can be drawn effectively to obtain answers using the distributions. Efficient algorithms have been developed to compute the posterior probability of some variables given the evidence on others.

Specifically, *particle-based methods* can be understood as an approximation of the joint distribution by means of the instantiations to all or some variables in the network, which are also called *particles*. There are simple methods to perform forward sampling to generate particles, which allows sampling nodes given the values of all of its parents (once a topological ordering of the variables has been made to establish the BN structure).

This approach addresses the problem of estimating marginal probabilities $P(\mathbf{Y} = y)$ of specific events relative to the original joint distribution. However, the conditional

probabilities of the form $P(y|\mathbf{E} = e)$ are the crucial form in which information can be retrieved from any BN. There are two main approaches to perform conditional probability queries: *rejection sampling* and *likelihood weighting (LW)* [52].

Rejection sampling obtains samples from the posterior probability $P(\mathcal{X}|e)$. The procedure by which these samples are obtained includes first to generate samples \mathbf{x} from $P(\mathbf{X})$ and then, reject all those which are incompatible with the evidence e . The main disadvantage of this method is that most of the times the number of rejected particles is quite small. Typically the number of expected particles that are not rejected is proportional to the number of observations M and the probability of the evidence $P(e)$, such that at least $M/P(e)$ are necessary to obtain M^* unrejected particles.

LS is a form of rejection sampling that attempts to make samples based on the logical test of the evidence.

On the other hand, likelihood weighting is more sensible with the evidence for generating the samples. This implies that the evidence nodes are set to their corresponding observed values. Hence, it is important to consider the probability of each of the observed nodes to have resulted in the observed values, and thus the weight of each sample should be the product of the probabilities of each evidence node separately. In other words, the weights of the different samples are the accumulated likelihood of the evidence.

2.4 Learning

The task of learning a Bayesian network can be subdivided into two different subtasks: learning the parameters for a given network topology, and identifying the network itself. This translates into first eliciting the structure of the network, and then its parameters, aiming always for simple structures (sparse networks) with the minimum number of parameters and the minimum set of possible dependencies [77]. The inability to estimate parameters reliably as the dimensionality of the parent set grows is one of the key limiting factors in learning Bayesian networks from data [52].

The seminal work by Heckerman, Geiger and Chickering [35] defined a new paradigm in the process for automatically discovering causal structures that satisfies the joint probability as observed in empirical data [77]. The following sections are devoted to define most of the approaches for learning the structure of the network (Sections 2.4.1 to 2.4.3) and also the appropriate parameterization (Section 2.4.4).

2.4.1 Constraint-based Structure Learning Algorithms

Verma and Pearl [101] provided the foundations for learning structure of Bayesian networks using conditional independence tests. They defined the inductive causation (IC) algorithm, which begins by considering a complete graph (fully connected), and then pruning based on statistical tests for conditional independence. This process is also known as backward selection [66].

The first step in the algorithm is to identify which pairs of variables are connected by an arc, regardless of its direction. These variables cannot be independent given any other subset of variables because they cannot be d-separated. Secondly, the V -structures among all the non-adjacent pairs with common neighbor must be identified.

At the end of the first and second steps, the skeleton and the V -structures of the network are known, and the equivalence class is uniquely identified.

Finally, the the third step of the IC algorithm is to identify compelled arcs and orient them recursively to obtain the completed partially DAG (CPDAG).

The practical implementation of the first two steps of the IC algorithm as described here is almost impossible due to the exponential number of possible conditional independence relationships. Nonetheless, many improved versions have been implemented [66]:

- **PC.** Is the first practical application of the IC algorithm. Is based on the notion that if any two nodes $u, v \in V, \mathcal{G} = (V, E)$ are d-separated, they must be d-separated either by $Par_{\mathcal{G}}(u)$ or $Par_{\mathcal{G}}(v)$. Hence, it is necessary only to consider subsets of the neighborhoods of either u or v in \mathcal{G} to determine if u and v are d-separated. The PC algorithm does this by increasing the subset size by one each time, starting from 0 [67].
- **Grow-Shrink.** Based on the Grow-Shrink Markov blanket algorithm. It considers which members of the blanket of each node are neighbors by a series of dependent tests [62].
- **Incremental Association.** Based on the Incremental Association Markov blanket algorithm [99], is a two phase selection scheme. It first applies a forward selection which considers all the nodes that might belong in the Markov blanket of u , and then a backward conditioning to remove all the nodes that do not satisfy the independence from u given the rest of the nodes in the probable Markov blanket set.
- **Fast Incremental Association.** Is a variant of IAMB which uses speculative step-wise forward selection to reduce the number of the conditional independence tests [107].

- **Interleaved Incremental Association.** Another variant of IAMB which uses forward step-wise selection as IAMB to avoid false positives in the first step for determining the Markov blanket [99].

By deriving from the Markov blanket for each node, all the algorithms listed above (except for PC) greatly simplify the identification of its neighbors, which translates into a significant reduction in computational complexity [66].

2.4.2 Score-based Structure Learning Algorithms

Score-based structure algorithms are also known as search-and-score algorithms. They are an implementation of a general optimization heuristic to the problem of finding the best structure for a Bayesian network. Each candidate network is assigned a network score reflecting its goodness of fit, which is used by the algorithm as a maximization function.

The heuristics that have been applied to this task are:

- **Greedy search.** This includes algorithms such as Hill Climbing (with random restarts) and Tabu search [14]. These algorithms explore the search space starting from a network structure (usually an empty graph) and adding, deleting, or reversing one arc at a time until the score can no longer be improved.
- **Genetic algorithms.** Mimic the natural evolution process through the iterative selection of the *fittest* models and the combination of its features [58]. In this case, the crossover operator combines the structures of two networks (n -point crossover), and the mutation operator introduces random variations in the structures by modifying the edges.
- **Simulated annealing.** This algorithm performs a stochastic local search by accepting changes that increase the network score and at the same time, allowing changes that decrease it with a probability inversely proportional to the (allowed) score decrease.

Network scores

The most common scores found in the literature are:

- **Bayesian Dirichlet equivalent (BDe) score.** The posterior density associated with a uniform prior over both the space of the network structures and of the parameters of each local distribution [35].

- **Bayesian Information criterion (BIC).** A penalized likelihood defined as

$$\text{BIC} = \sum_{i=1}^n \log P(X_i | \text{Par}_{\mathcal{G}}(X_i)) - \frac{d}{2} \log n \quad (2.6)$$

where d represents the total number of parameters in the global distribution.

- **Akaike Information Criteria (AIC).** Is not a score of the model itself, but an estimation of the quality of each model relative to other models. Is defined as [4]

$$\text{AIC} = -2 \log \hat{L} + 2d \quad (2.7)$$

where \hat{L} is the maximum likelihood of the model.

- **K2.** Relies on several assumptions of the model, such as parameter independence and uniformity of the distribution, given the network structure, and represents the entropy of a distribution based on the concept of conditional entropy [37]:

$$H_{BN} = \text{K2} = \sum_{X \in V} \sum_{\pi_X} P(\text{Par}_{\mathcal{G}} = \pi_X) H_{X|\pi_X}, \quad (2.8)$$

$$\text{given } H_{X|\pi_X} = \sum_x P(X = x | \text{Par}_{\mathcal{G}}(X) = \pi_X) \ln P(X = x | \text{Par}_{\mathcal{G}}(X) = \pi_X) \quad (2.9)$$

where π_X is a specific assignment for the parents of X in \mathcal{G} .

2.4.3 Hybrid Structure Learning Algorithms

Hybrid structure learning algorithms combine constraint-based and score-based algorithms to compensate the weaknesses of each other, and produce reliable network structures. The two best known algorithms of this kind are:

- **Max-Min Hill Climbing (MMHC).** Is based on two steps called *restrict* and *maximize*. In the first one, the candidate set of parents of each node is reduced to a smaller set whose behaviour has been shown to be related in some way to the target node. This simplifies the search space, and then the algorithms maximizes the score function based on the restrictions imposed in the first step [100]. In the implementation, the Min-Max Parents Children (MMPC) heuristic is used to learn the candidate sets for each node, and hill climbing is applied in the optimization stage.

- **Sparse Candidate (SC).** Follows the same steps as MMHC, but they are repeated iteratively until there is no change in the network, or no network improves the network score [32].

2.4.4 Parameter Learning

Once the structure for the BN has been established, the task for estimating and updating the parameters of the global distribution benefits from the Markov property. This results relevant in practice, since we consider that local distributions concern only a reduced set of variables, reducing significantly the complexity of the algorithms. There are two main approaches to the estimation of parameters, one based on maximum likelihood estimation (frequentist), and another one based on Bayesian estimations ^b.

Maximum Likelihood Estimation

Here we show a simplified description of the Maximum Likelihood Principle to exemplify the process that can be generalized to learning the parameters of a Bayesian network. First, the learning problem can be defined as follows: assume several samples of a set of random variables X from an unknown distribution $P^*(X)$, and a known sample space. Let \mathcal{D} be the training set consisting of M instances of $X : \xi [1], \dots, \xi [M]$. Then, assume a parametric model $P(\xi : \theta)$ that assigns a probability to ξ given a particular set of parameter values θ . It is necessary to satisfy that for each assignment to the parameters θ there is a legal (non negative) distribution and $\sum_{\xi} P(\xi : \theta) = 1$.

Then, the likelihood function is defined by

$$L(\theta : \mathcal{D}) = \prod_m P(\xi [m] : \theta) \quad (2.10)$$

It is important to emphasize that the likelihood function measures the effect of the choice of parameters in the training data.

Formally, the Maximum Likelihood Estimation (MLE) chooses parameters $\hat{\theta}$ given a data set \mathcal{D} that satisfy:

$$L(\hat{\theta} : \mathcal{D}) = \max_{\theta \in \Theta} L(\theta : \mathcal{D}) \quad (2.11)$$

^bThe concepts introduced here are the minimum necessary to grasp the notion of parameter learning for BNs. For a more extensive description and derivation of the formulas, please refer to Koller and Friedman (2009) [52].

Therefore, for multinomial distributions, the maximum likelihood is in fact the probability of each value of X given its frequency in the data. Similarly, for a continuous variable, the maximum likelihood is obtained when μ and σ correspond to the empirical mean and variance of the training data.

The structure of Bayesian networks allows to decompose the likelihood estimation in isolated problems, one for each variable. Each of these terms is known as a local likelihood function that measures how well the variable is predicted by its parents.

Hence, each conditional probability distribution (CPD) is parameterized by a separate set of parameters that do not overlap, as the likelihood decomposes as a product of independent terms (one for each CPD in the network).

Bayesian Parameter Estimation

The Bayesian approach draws upon the prior knowledge about θ , which encodes the probability, or how likely, we are a priori to believe the different choices of parameters. The main difference with the MLE approach resides in the use of the posterior. Here, the entire θ is used to make predictions of the probability of the event, rather than just selecting one single value.

On the other hand, the challenge is to pick a prior distribution that compactly represents the continuous space Θ , and that can be updated efficiently. Generally, the *Beta* distribution is used due to the convenience of obtaining also a *Beta* distribution in the posterior distribution.

Then, we define the learning problem assuming a data set \mathcal{D} contains M samples of a set of random variables \mathcal{X} from an unknown distribution $P^*(\mathcal{X})$, and a parametric model $P(\xi|\theta)$ where its possible to choose parameters from a parameter space Θ .

The Bayesian approach states that the subjective probabilities we assign to values of θ must be updated after we have seen the evidence. The Bayes rule provides such probabilistic tool. First, it is possible to define a joint distribution over the θ by accounting our knowledge (or lack of) about the possible values of θ .

Intuitively, it is possible to see that the Bayesian prediction converges to the MLE estimate when $M \rightarrow \infty$, regardless of the starting point (prior). In this case, the effect of the prior is negligible and the prediction is dominated by the frequency. It is possible also to state that the Bayesian estimate is more stable than its MLE counterpart, because with few instances, even small single samples will change the MLE estimate dramatically. This property leads to more robust estimates when not enough data are available to provide definite conclusions.

Applying this concepts to the parameterization of a Bayesian network relies in the concept of global parameter independence, which must be carefully considered when

applied to a specific domain.

This general assumption leads to the conclusion that complete data d-separates the parameters for different CPDs. Thus, if two parameter variables are independent a priori, they are independent a posteriori.

Generalizing the conclusion to the task of Bayesian network learning, and assuming global parameter independence, we can represent the posterior as a product of local terms. This implies that it is possible to solve the prediction problem for each CPD independently, and then combine the results.

It is now necessary to address how to assess an adequate selection of the parameter priors required for a Bayesian network. One option is to use a separate Dirichlet prior, with hyperparameters determined by a domain expert. Nevertheless, this is rather a burdensome task, and a contradiction!. A more sensitive approach is to define a set of independent marginals over the variables X_i 's.

2.5 Model elicitation

As Bayesian networks began to rise in popularity, and the inference was not anymore an issue due to its feasible computation, researchers realized the bottleneck was the knowledge engineering required to design the models. The efforts then focused on automating the learning process (see Section 2.4). Nevertheless, in some circumstances domain expert knowledge may be available, which will require some knowledge engineering to elicit the model. Some other times, one part of the model elicitation might require a combination of both expert and data driven methods.

Given that there are multiple ways to represent the relevant attributes in any domain, choosing the variables to include in the model is often one of the hardest tasks, and has direct implications throughout the rest of the process for determining the model.

Similarly, there are many structures that will fit consistently the same set of independencies. In general, a good approach is to select a structure that reflects the causal ordering of variables and its dependencies. This type of causal graphs tend to be sparser, but they must convey the causality *in the world*, and not in the *way we see the world*, which is more a reflection of our personal inference process [52]. And an overall good rule of thumb is not to include weak or specific dependencies that may increase the complexity of the model.

When it comes to parameterize the network, eliciting probabilities from experts is one of the most challenging tasks. One common approach is to use abstract concepts such as *common*, *rare* and *surprising* and assign each a predefined number. Still, this method is imprecise and can lead to misinterpretation. Another issue of the process is that people's

estimates can vary if the question is rephrased, which adds another layer of complexity. However, the most important consideration is to never assign a probability equals to zero: no matter how unlikely the event might be, it is not impossible, and a zero probability may cause the elimination of an effect when factorizing the joint distribution.

Korb and Nicholson [73] propose a method to construct Bayesian models under a variety of circumstances. Knowledge Engineering with Bayesian Networks (KEBN) deals with the major tasks when constructing a BN from defining the network itself to its full deployment in working environments.

Figure 2.2 illustrates the KEBN lifecycle model. It comprises the development of the Bayesian model from the first stage, which includes defining the variables and their possible values, establishing the appropriate graph structure and parameters. Followed then by verifying if the model satisfies the requirements of the task it was designed for. Specifically, sensitivity testing refers to determining how much does the output of the network responds to changes in the input and parameters, which is very helpful to understand how to best apply the model in the field. The BN must also comply with usability requirements, and the corresponding field evaluation must be completed before its full deployment in the assigned task. Further statistical analysis is required to refine the model, hence useful for later improvements.



Figure 2.2: Knowledge Engineering with Bayesian Networks (KEBN) methodology

In general, a probabilistic graphical model is a good option to represent the characteristics of a domain problem if it has well defined variables and identifiable cause-effect relations with some degree of uncertainty. Ideally, the model will tackle a repetitive task, and its main purpose is to support decision making [44].

Chapter 3

Recommender Systems

Recommender systems have the fundamental purpose of suggesting relevant items to users, and help them in the decision making process. These can be applied to a wide variety of domains, and different strategies are available to meet specific users' needs, and also designers' goals. The basic and most relevant notions regarding the methods and characteristics, as well as the evaluation process are described in this chapter.

First, the definition and the core characteristics of a recommendation system are presented in Section 3.1, followed by the description of the strategies most commonly implemented in Section 3.2. A comprehensive characterization is made for the essential approaches pursued with the research, context aware systems in Section 3.3 and hybrid implementations in Section 3.4. Finally, different evaluation techniques are detailed in Section 3.5.

3.1 General definition

Recommender Systems (RSs) can be defined as those software tools and techniques whose purpose is to provide items suggestions useful for an user [84]. This recommendations can be related to any decision making process, from buying an item, listening to a song or read a book, up to more difficult choices such as buying a car, a real state property, or how to make and investment or auction. The main features of the RS can vary depending on the requirements of the task; therefore, there are a great variety of techniques that implement different approaches focused on resolving in a specific manner the recommendations process.

The general components of a RS include:

- Background data: all the information from which the recommendation process

begins.

- Input data: all the information the user must provide to the system before generating a recommendation.
- An algorithm that combines background and user specific data to produce the recommendation.

3.2 Common approaches

Based on the origin of data used to produce the recommendation, five different types of RS can be identified [17]

Collaborative techniques

These are the most mature of all the approaches. They are based upon a rating system, that can be either binary or continuous, from which they identify similarities among users to determine which item (or set of items) and how will satisfy user's request. Figure 3.1 shows a classical representation of the rating matrix and how users' and items' similarities are composed.

The model user in this kind of RS consists of a vector correlating items and ratings that is continuously updated based on new interactions. It may include also detriment of preferences, accounting for the user's change of taste over time. Their strongest characteristic is their independence from any interpretation of what is being recommended, what makes them suitable for recommending complex items where variation in preferences is intimately linked with taste variation.

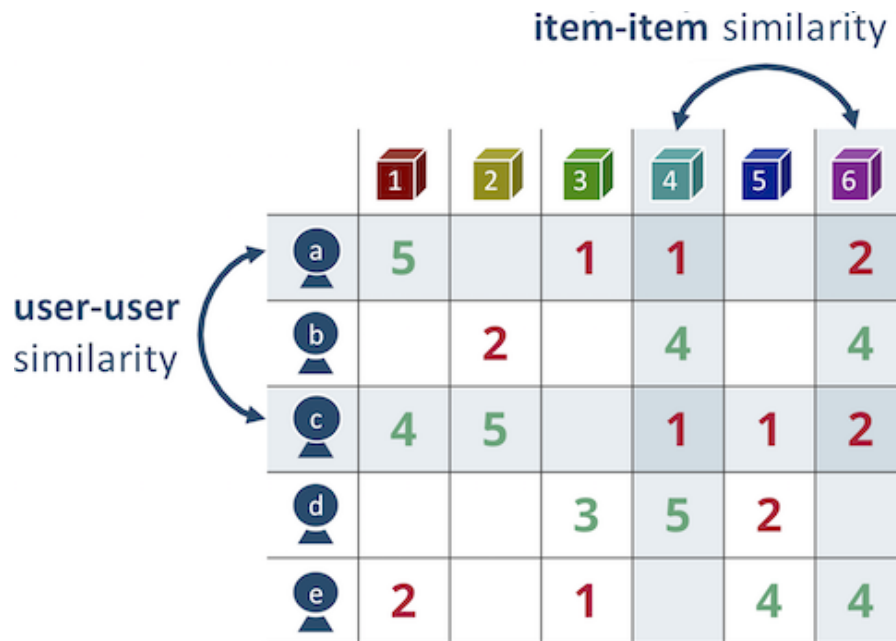


Figure 3.1: Matrix relating users and items through ratings

Content-based recommendations

These are based on specific features of items. They construct a profile for the user depending the items that have been rated, formulating the recommendation according to the similarities between items' features. They are a good approach to construct a long term profile for the user that can be updated continuously.

Demographic systems

These methods aim to classify the user based on specific characteristics, and then produce a recommendation comparing demographic classes. They are good to implement recommendations where no background data about the user is given, contrary to what collaborative or content-based techniques require.

Utility-based recommendations

These techniques follow a set of utility rules depending on features of items that will better meet the preference of the user. Constraint satisfaction methods are used to generate an appropriate recommendation, since the model for the user are the set of utility functions based on the ranked items. This allows to match specific requirements not

directly related to the item, such as availability and purchase options. In contrast of what previous techniques pursue, utility-based techniques do not aim to provide a long term representation of the user.

Knowledge-based recommendations

With these methods, recommendations are constructed from inferences made on user's needs and preferences. They have a solid foundation on functional knowledge, such that they are capable of providing a relationship between the specific need and how a possible item will fulfill it. The recommendations are static, since the knowledge base is not updated over time, but they are very flexible due to the variety of forms that the knowledge base can take.

3.3 Context Aware Recommender Systems

A broad conception of context includes the information about the location, the companion, and the nearby resources available to the user. Similarly, Dey [29] defines context as any information that can be used to describe the situation, which generally includes all relevant interactions between the user and the application.

Consequently it is possible to define a context-aware system as the one that provides relevant information for the user's task based on the context [29]. Hence, a Context Aware Recommender System (CARS) will ideally be able to classify each user interaction in to a particular category depending on the context to provide adequate recommendations that satisfy completely the need for this situation [5]. However, is important to note that a RS can be context aware, while not being personalized.

3.3.1 Definition

Based on the information that the CARS has available, it is possible to identify three levels of knowledge about environmental factors [2]

- Fully observable. All relevant factors involved in the recommendation process are well known, along with their values and structure or relationships.
- Partially observable. In this scenario, only some of the information is known explicitly. There can be many cases, where maybe the factors are known but the structure or values are not, or vice versa.

- Unobservable. It is not possible to obtain information about contextual factors, then the recommendation is made upon latent knowledge about the environment, and may be based on implicit predictive models.

According to the behavior of the context variables over time, the information of the environment can also be separated in two different classes: (i) static, where the structure of context factors remains the same; and (ii) dynamic, that means that contextual factors change somehow over time.

3.3.2 Context Representation

The relevance of context relies in its potential capability to encapsulate the relationship between the computation process and the situation in which such technology is used. Despite the applications may vary, it is possible to identify two main approaches that define the basic characteristics of the entity we understand as context: representational and interactional [30].

First, the representational view deals with how the context can be encoded and represented. Hence, context is assumed to be a delineable from of information, that is independent from the activity itself but stable from instance to instance. Second, the interactional approach instead considers that context is a relational property that holds between objects and activities, and whose relevance and scope are defined dynamically relevant to a particular activity, from which is originated.

Most of the context aware systems adopt the representational view, as it is more similar to the traditional computation methods, and thus its implementation is smoother. Nevertheless, one of the biggest challenges when designing a context aware system is the definition of how the context will be encoded. For example, it can be defined as an unobservable dynamic environment, so the RS will be modeled upon latent variables with machine learning methods and refined with consecutive interactions. The opposite case will be to consider a static and fully observable environment, where the system knows everything relevant to the recommendation, and is usually referenced as the representational view [2].

Similarly, the representation of the context usually requires the intervention of a domain expert, since the problem of the context specification is, generally, domain specific. In this scenarios, knowledge engineering must be applied to identify the contextual variables that must be defined as part of the core system. Most of the time it is not possible to define them, and therefore, collect the pertinent data beforehand.

Another challenge when collecting contextual information, is choosing a convenient method, either explicit or implicit. The first one comes directly from the users or from the sensors available, for example, the application may require the user to input some

information to begin with, and if the interaction is on a mobile device, additional information can be retrieved from the GPS or other applications. In contrast, the latter requires inference from previous collected data, and in most cases, a model.

However, in the film domain it is rare to ask the user to rate an item in repeated situations according to the context, and most users will not do it willingly. However, user feedback can also be used to modify context factors and values, and not only profiles. This research will focus on the specific situation users define explicitly, as a way to test the relevance to the implementation of the method. Later, with the results of this study, a real ubiquitous and personalized experience can be implemented in a fully working environment that includes the context representation we define here.

3.3.3 Common approaches

The information of the context can be included in different stages of the recommendation process, hence, three main paradigms have been defined [2]:

Contextual prefiltering

The contextualization is made before the recommendation. The information about the context is used to narrow the data from which the recommendation based on user-item ratings is made. This allows to take full advantage of all the recommendation techniques described before. However, if the search space is narrowed too much by the contextual information used as filter, the output may result over specific and the final recommendation won't be relevant to the user.

Adomavicius et al. [2] highlight the relation of this approach with the task of constructing local models, and reviewed item-splitting versus microprofiling. The first one splits the item into several fictitious ones based on the context in which it can be consumed, whereas the second technique produces subprofiles that represent the user in different situations according to the context. For both scenarios, the recommendation is based on these different submodels instead of a single one.

Contextual postfiltering

Can be understood as contextualization of the recommendation output, where the result from the traditional 2D recommendation is then rearranged using the context knowledge, and similar to prefiltering, allows to fully deploy any of the RS methods. The rearrangement can be of two kinds: (i) filtering the recommendations to match the given context, and (ii) adjustment of the ranking of recommendations.

They can be classified into two categories: heuristic and model-based. While heuristic postfiltering focus on finding similar items based on the characteristics preferred by the user in this context and then rearrange the recommendation, model-based postfiltering calculates the probability for each item based on the user's preference given a context to adjust the recommendation.

Contextual modeling

The recommendation function is contextualized from its core, and information from the context is used directly in the estimation of the rating. Therefore, the recommendation function is multidimensional, that can be also represented in predictive models or heuristics.

Solutions of this kind appear on recent approaches, and were made popular by the Netflix Prize competition, and the CARS [54].

3.4 Hybrid Recommender Systems

Many researchers have tried to combine different techniques to achieve higher performance [80] [61]. The use of two recommendation techniques may be sufficient to overcome the deficiencies of individual approaches. Most of them generally include collaborative filtering (CF) as the main technique, and the proposed combination aims to diminish the sparsity of data that is present at the very beginning of the recommendation process, trying to alleviate the ramp-up problem.

3.4.1 Common approaches

The different ways in which two or more recommendation techniques can be combined are described as follows:

Weighted

The scores for the recommended item, product of all recommenders included in the system are combined into a single recommendation. The most common and simple combination is in a linear manner; this allows to adjust the relative valued assigned to each score.

Switching

This is a strategy that selects the recommendation technique according to some recommendation criteria. This usually represents an increase in the complexity of the recommendation process, but it brings sensitivity to some specific parameter that would be imperceptible by individual techniques.

Mixed

This system presents multiple recommendations from different techniques at the same time. This approach clearly avoids the ramp-up problem since it can relay on multiple methods, that are suggested usually ranked.

Feature combination

In this system, features from different data sources are combined into one single recommendation process. It is a way to achieve content and collaborative association by including feature data of each item, and use content-based techniques over this augmented information. This process allows to reduce the sensitivity of the system to the items that have been rated by users discovering inherit data similarity that would have remained unseen by collaborative filters [17].

Cascade

This is a clearly staged process, where the outcome of one recommendation technique is the base for the next one, that refines the recommendation. The main advantage is that it avoids the use of the second RS into items that would not have been sufficiently good for being recommended.

Feature augmentation

This process incorporates the rating information product of the first technique into the second one, allowing the increase of the performance of the main technique.

Meta-level

This system is determined by the use of the model of one technique as input for the second. The main improvement that this combination provides is that the learned model is a more specific representation of the user's interest. In a content-based/collaborative

hybrid, the first model will represent a more dense representation of the information that can be processed more easily and from which better results can be generated.

3.5 Evaluation

The task of designing a RS is mostly overgeneralized by the process of developing new recommendation algorithms that support the goals of the designer. There are plenty of approaches that can fit the requirements as seen in previous sections, and also several other considerations such as type, availability of data, reliability, computational restrictions, etc. There exists a diverse set of evaluation techniques to provide some kind of ranking, and aid the designer to choose the algorithm that best fits the specific application [94].

There are different experimental setups that can be considered when evaluating a new recommendation algorithm:

- **Offline experiments.** These are performed over a controlled set of data, which allows to measure the predictive power of an algorithm. The goal of these experiments is to filter out inappropriate algorithms, and cannot be used to measure user behaviour under the new algorithm.
- **User studies.** These tests aim to properly evaluate the system through real user interactions. Many qualitative or quantitative information can be obtained from these experiments, and a more precise idea can be made about the influence of the RS on users' behaviour.
- **Online evaluation.** This type of experiments provides the strongest evidence about the true value of a RS. They test the real effect on users given some utility measure that servers to conclude superiority of one system over another.

All of these different strategies have a different purpose, and some considerations must be made prior the evaluation of the system. These range from appropriate selection of the data to evaluate the system with, to possible expenditures when performing user or online evaluation. And despite all of those are correctly assessed, there is the slight possibility that the results are due to a statistical mishap. Therefore, it is important to design a study based on the desired confidence level (usually $\alpha = 0.05$) and feasible statistical power (depending on the sample size and predicted effect size).

3.5.1 Measuring recommendation system properties

Depending on the goals of the application, different evaluation techniques may suit better the purposes of the evaluation [36]. Prediction accuracy is by far the most discussed in

the literature, and even the Netflix Prize awarded 1 million dollars to the designers of an algorithm which improved this single metric by 10% [54]. It is completely independent from user interaction. and usually relies solely on ratings.

The different scores that can be calculated based on predictions are summarized in Table 3.1. To identify the common factors, r_{ui} is the rating of the item i for a user u , \mathcal{T} is the set of user-item pairs in the test set, the $\hat{(\cdot)}$ represents the predicted values, and $\bar{(\cdot)}$ the average.

To understand the nature and purpose of this metrics, it must be clear that metrics from 1 to 6 are directly related to precision over algorithmic performance as is usually understood in machine learning environments.

On the other hand, metrics from 7 to 9 consider rank as a measure of relative preference of the users over the ordered list of recommendations for a given user $\mathcal{O}(u)$. Other rank sensitive measures are Mean Reciprocal Rank (MRR), normalized DCG (NDCG) which normalizes the DCG by the maximum achievable utility, fraction of concordant pairs, rank effectiveness, R-score, among others [94, 31].

The evaluation process to obtain the rank for every item in the recommendations set is a laborious task since it requires the users to specify the *correct* order for each item. Nevertheless, this kind of evaluation metrics have proven to yield better and more diverse recommendations while maintaining accuracy [1]. This rises the question whether predictive accuracy based only on ratings translates into a compelling recommendation experience. Moreover, traditional evaluation procedures based on hidden-data evaluation (such as the traditional partitioned sets for machine learning training) assume that items the user has already seen (and liked) are automatically better than items the user hasn't already seen (an potentially liked).

This trade-offs with diversity and coverage lead to recommender systems that recall what the users already know, rather than encouraging the discovery of new and interesting items. The traditional accuracy metrics are prone to create overfitted models, which penalize the recommendation of unseen items, and may not perform well in real life scenarios [31].

3.5.2 User centric evaluation

There are significant challenges and considerations when performing user-centered research, such as maintaining a consistent user community [53], that properly reflects the target population of the system, as well as counteracting biased results that may come from paid experiments, or users trying to unconsciously satisfy the hypothesis that is being tested.

Most importantly, when users evaluate system properties it is important to consider

	Metric	Definition
1	Root Mean Squared Error (RMSE)	$\sqrt{\frac{1}{ \mathcal{T} } \sum_{(u,i) \in \mathcal{T}} (\hat{r}_{ui} - r_{ui})^2}$
2	Mean Absolute Error (MAE)	$\sqrt{\frac{1}{ \mathcal{T} } \sum_{(u,i) \in \mathcal{T}} \hat{r}_{ui} - r_{ui} }$
3	Precision	$\frac{TP}{TP+TN}$
4	Recall (True positive rate)	$\frac{TP}{TP+FN}$
5	False positive rate	$\frac{FP}{FP+TN}$
6	Area under the ROC curve (AUC)	-
7	Spearman's ρ	$\frac{\frac{1}{T} \sum_i (r_{iu} - \bar{r})(\hat{r}_{ui} - \bar{\hat{r}})}{\sigma(r)\sigma(\hat{r})}$
8	Mean Average Precision (MAP)	$\frac{1}{U} \sum_{\mathcal{U}} AP(\mathcal{O}(u))$
9	Discounted Cumulative Gain (DCG)	$\sum_i \frac{r_{ui}}{\max(1, \log_2 i)}$

Table 3.1: Metrics used to assess prediction accuracy in recommendation systems

whether her behaviour changed as a response to the change in that variable rather than if the user noticed the effect after all. This implies a deeper and conscious consideration of the subjective evaluation users assign to both the system as a whole, and to the recommendations [94].

This concept brings to light the idea that different users may be looking for something different in their recommendations [46], and satisfaction can be intrinsically related to how the interaction with the system is performed, and the different levels of personalization and preference elicitation.

Therefore, a comprehensive study based on user experience must rely on stable hypothesis to be able to statistically analyse the results [49]. Useful experimental manipulations can only yield consistent results if the appropriate considerations regarding target audience and sampling procedure were determined based on the purposes of the

evaluation.

Knijnenburg, Willemsen and Kobsa [48] propose a framework for user-centric evaluation. They suggest that first is important to randomly assign each participant a condition, depending on what is being tested. The minimum information necessary to complete the procedure without influencing their behaviour must be provided as a mean of introduction to the experiment. Additional personal and situational characteristics are clearly beyond the scope of any reasonable study regarding the interactions with a RS, but nevertheless they are important moderators to the overall experience and are relevant enough to collect them [50], under the corresponding privacy considerations.

The framework relates objective system aspects, such as algorithm interaction and presentation, directly with more subjective aspects, such as perception (usability, quality, and appeal), experience (of the system, of the process, of the outcome), interaction (rating, consumption, retention). Additionally, personal and situational characteristics of the user are interrelated with all of the above [47].

Pu, Chen and Hu [81] propose ResQue as a set of recommendations to the RS developers, and comes from the idea that useful technology must also be easy to use and easy to understand for the user. They structure a four-layers construct based on perceived system qualities, users' beliefs, subjective attitudes and behavioural intentions. The framework considers the trade offs that come with balancing accuracy and design, and mainly discuss qualities of the recommended items (perceived accuracy, novelty, attractiveness, diversity), preference elicitation, layout, and the support provided for the user's decision as well as the ability to explain the results [82].

The user-centric design and evaluation is a trend in recent developments in recommendation systems. The tasks include setting a set of solid guidelines, based on usability and user preference, for the successful implementation a RS for a target audience which includes only college students [70], comparing complex machine learning algorithms for recommendations given explicit or implicit action prediction [110], assessing similarity of the recommended items [106], and designing more persuasive and credible RS based on core characteristics of the RS [108].

Chapter 4

Related Work

As the amount of available information continues to grow, recommendation techniques have taken more importance than ever. Their capacity to retrieve and prioritize personalized information has been exploited in a wide variety of fields [11]. Relevant and similar approaches will be examined in the following paragraphs. Section 4.1 presents the most relevant and recent advances in recommendation strategies, and Section 4.2 presents a detailed description of the reference works upon which this research is born.

4.1 Comprehensive literature review

Many trends are present in current research in the recommendations field, however the analysis made here will be focused on three fundamental lines, which concern the fundamental aspects of this research. First, combinations of two or more recommendation strategies into one hybrid system are discussed in Section 4.1.1, how information about the context is integrated into the recommendation is examined in Section 4.1.2, the different ways Bayesian networks have been used to offer personalized recommendations are presented in Section 4.1.3. Also, other approaches are considered in Section 4.1.4.

4.1.1 Hybrid implementations

Most of the past and current research in the field of hybrid recommendations is targeted to solve the ramp-up problem [112]. For instance, linear combinations of the recommendation scores is implemented for news recommendations (P-Tango [25]) where collaborative and content-based recommendations were weighted with gradual modifications until the user confirms the predictions. Other implementations perform

some kind of voting system, or consensus, among the different recommendation strategies [74].

Mixed hybrid implementations have been made to combine recommendations based on item content with preference from the users obtained from collaborative techniques (PTV [95]), or simply to present recommendations side by side (ProfBuilder [104] and PickAFlick [18]).

The switching strategy has been used to act as a salvation net when one system fails, so the other one can come to the rescue (DailyLearner [10]). Users' past ratings can be set as switching criteria for the system to choose the best technique to employ for the next recommendation [98].

Item features play a significant role when combined with a collaborative strategy. This combination leads to a decrease in the sensitivity of the system to the number of ratings (either high or low) for any given item, and at the same time makes explicit the similarity between items. It has proven good results in the film domain given a manual selection of the item features [9].

Fab [7] is an interesting hybrid recommender. It uses a term vector model to describe users' preferences which is elicited from user-specific selection agents based on filters through item content. Additionally, performs a cascade combination of collaborative and content-based recommendations for the final selection of the items.

EntreeC [17] is perhaps one of the most famous hybrid recommender systems, which based on knowledge and item-content recommends restaurants by applying a critique mechanism. GroupLens [88] is a well known recommendation engine, and has used feature augmentation to enhance the ratings matrix by using knowledge-based "filterbots".

Libra [65] uses a naïve Bayes classifier to make content based recommendations based on content augmented from collaborative techniques. A more complex implementation of content-based recommendations is applied by LaboUr [89], where instance-based user profiles are derived from item-content, which are later compared in a collaborative fashion.

Table 4.1 summarises many relevant hybrid implementations, and classifies them according to the corresponding combination of recommendation techniques and the hybridization method employed. Some cells are shaded indicating the redundant (light color) and not possible (dark color) combinations as suggested by Burke [17]. In addition, the strategy described in this document is marked in green, specifying the combination of knowledge and content-based recommendations into a cascaded hybrid system.

	Weighted	Mixed	Switching	Feature comb.	Cascade	Feature Aug.	Meta-level
CF+CB	P-Tango [25]	PTV [95], ProfBuilder [104]	DailyLearner [10]	Basu et al. [9]	Fab [7]	Libra [65]	
CF+DM	Pazzani [74]						
CF+KB	Twole & Quinn [97]		Tran & Cohen [98]				
CB+CF	Kaminskas [41]				Karatzoglou et al. [42]	Baltrunas et al. [8], Oku et al. [68]	Fab [7], LaboUr [89]
CB+DM	Pazzani [74]			Condiff et al. [26]			
CB+KB		PickAFlick [18]					
DM+CF							
DM+CB					Huang & Biang [38]		
DM+KB							
KB+CF					EntreeC [17]	GroupLens [88]	
KB+CB					CHYBAM		
KB+DM							

Table 4.1: Hybrid implementations in the literature

NOTATION. **CF**: Collaborative Filtering, **CB**: Content-based, **DM**: Demographic, **KB**: Knowledge based.

4.1.2 Context aware systems

Context can be introduced into the recommendation in various ways (see Section 3.3), and definitely can improve the perceived performance of the recommendation as it enhances user's models [2] and provides an additional level of personalization. The relevance of the context has been proven [71, 5], and even it is possible to infer the contextual information by using Bayesian methods (BN or naïve classifiers).

A wide variety of domains can benefit from the inclusion of context as part of the recommendation process. For example, personalized shopping assistance was studied by Sae-Ueng et al. [86], where the behaviours of the customers were classified as different context situations. The recommendations are based upon ratings inferred from behaviour patterns, and use collaborative techniques. The system was tested in a controlled environment where real users tested the recommendations.

Similarly, Jin et al. [40] present an approach for modelling the behaviour of users while browsing the web. The main objective is to discover how web pages are associated with various tasks. This information is combined with user's navigation history into a maximum entropy engine, which makes a top- N recommendation associated distribution.

MusicSense [19] suggests music when users read. The recommendations are based on a Emotional Allocation modeling which characterizes songs and documents (or any text source) by a set of words, which are associated with a mixtures of emotions. The inference of the emotions is the core aspect of the recommendations, since serves to match the items. The similarity between the songs and the documents is used to generate the set of recommendations.

Also in the field of music recommendations, Kaminskas [41] propose a method to match points of interest (POI) to specific music. The system initially filters the tracks and ranks the POIs according to user preference profile. Then, the individual scores are linearly combined into the final recommendation.

The MoMa-system [16] offers proactive recommendations for mobile advertising. The systems considers private and public context; the first one refers to mobile terminal parameters, while public context include variables such as traffic, weather, time of day, etc. Filtering techniques are applied to identify the matching item-user pair, and recommendations are delivered directly to the mobile device of the user.

In a more general task for the film recommendations domain, the LDOS-CoMoDa dataset [55] collects information from more than 90 users, 900 items and 1600 ratings, each one related to specific context conditions and consumption time. Variables include season, location, companion, and also user emotions before and after the consumption.

Given the enormous quantities of variables that can be used to represent context,

tensor factorization has been used to deal with the complex relationships in a user-item-context N -dimensional matrix. This model can be then combined with a collaborative technique to produce the recommendation [42].

In a similar effort to make more accessible the usage of context information, Oku and colleagues [68] propose a SVM capable of modeling user's preferences based on context. Then, this information is combined with a collaborative approach to find the final set of recommendations. While the model accuracy is astonishing (100%), users perceived the recommendations as useful at least half of the time.

Similarly, the relevance of different context variables, and its influence over the recommendation process is studied by Baltrunas et al. [8]. Rich relationships were found among context variables and items, which were used to enhance the recommendation through a collaborative technique. With a study on the recommendation of tourist attractions, they proved that 95% of the users consider the context aware recommendations more appropriate.

4.1.3 Bayesian networks in recommendation systems

Bayesian networks offer the possibility to graphically represent the explicit relationships between variables, and assess their influence by a joint probability distribution [51]. The application of a BN to the recommendation process allows to combine multiple task into one system. For instance, it has been used as a flexible method to model trust in a peer-to-peer network which allows to communicate recommendations directly [103].

One of the earliest examples of the usage of Bayesian networks as recommendation strategies is the one proposed by Breese, Herckerman and Kadie [15]. They formulate a probabilistic collaborative filtering based on a BN with a node representing each item in the domain. Tree-based conditional probabilities based on the ratings for each movie were employed, and the experiments showed that the larger the tree for each node, the better the performance.

In the field of intelligent tutoring systems, Bayesian networks have been used to assess the knowledge of the student [83]. The next set of questions that will be presented to the student are based on a personalized user model corresponding to the current knowledge, and learning material can be adapted accordingly.

Bayesian networks can be also implemented to infer the context of the recommendation. Park and colleagues. [72] designed a music recommender based on user's preference given the inferred context. Scores for all the items are calculated as the sum of the conditional probabilities for each attribute given the fuzzy evidence from the BN, and the final recommendation is given by the top- N items.

Recommendations for groups have been studied also with the use of Bayesian

networks as user models. Huang and Biang [38] used a BN to describe group behaviour in a travel environment. Tourist attractions are the recommended items, which are scored by an analytic hierarchy process (AHP).

He and Chu [34] used a Bayesian network to model social interactions in the recommendation environment. By using information from social networks, user's preferences, perceived item's relevance, and influence from social friends. User's preference over a given item is calculated using the naïve Bayes assumption given its attributes and immediate friends ratings.

Ono and colleagues [69] used a BN to model user preferences. The conditional probability distribution is used to formulate an item preference model of an item for a target (set of) user(s) given some specific condition; this probability estimation is used as ranking score for the final set of recommended items. The conditioning variables can be related to the context, the item, the rating, or a characteristic of the target user (gender, age, etc.).

De Pessemier and Deryckere [79] present a Bayesian Classifier that combines user preferences with context information. The main task of the system is to compute the conditional probability that a given user will like an unseen item given its features and specific environmental conditions. The recommendation set is conformed by those items whose probability is above some specified threshold.

A broader deployment of the inference capabilities of a Bayesian network is proposed by Yuan et al. [109]. They designed a context aware system which relies on the prediction of contextual situations and user's behaviour based on location and other characteristics. To exploit these characteristics, they studied the user's spatial-temporal behavior and preferences in Twitter to produce both recommendations and search results.

4.1.4 Miscellaneous

Knowledge based recommendations suffer from the problem of knowledge elicitation. Since most of the times is difficult to obtain explicit assessment from a domain expert, the knowledge can be inferred from experts *within* the system. Phuksenga and Sodseeb [80] propose a method in which experts are identified by social networks, number of ratings, or other users' recognition. This information is then incorporated into a collaborative recommendation.

Bonhard and Sasse [13] suggest that seeking and receiving a recommendation is an intrinsic social activity, and RSs must consider this to enhance the reliability of the recommendations. Collaborative techniques clearly help reveal similarities between users, but a more explicit user model could improve overall taste matching in the final recommendations. This idea was tested in a fictitious movie recommendation scenario

and confirmed that recommendations enhanced with profile similarity are preferred over those generated only by high rating similarity [12].

Explicit user models are rare given the complex task involved into its formulation. However, Aguilar et al. [3] offer a framework to develop such model, with which specific relations between item features and users are explicitly represented. In addition, a rating system can be used to discover interests that would not have been available otherwise.

Information describing the content of the items can improve the user model by adding specific preferences. The task of modeling this relationships involves dynamic behaviour analysis, whereby the approach is often disregarded. Cami et al. [21] propose the Dirichlet Process Mixture Model to effectively model user interests and preferences, and is able to adapt to the user's behaviour over time.

Similarly, Ayyaz et al. [6] developed a framework based on content filtering to learn user's profile from previous activity. A fuzzy system then matches user's profile to items based on similarities or dissimilarities to conform the set of recommendations.

4.1.5 Summary

After reviewing many relevant applications it is possible to identify the key aspects that will serve to guide to the present research process.

The work by Ono, Motomura and Asoh [69] is undoubtedly the most similar approach. The conditional probability product of the inference with the Bayesian network directly represents the recommendation score. This is also one of the key features of the recommendations made with the present study.

While context information is proven to be relevant, a conscious evaluation of the variables to include in the recommendation must be made. The representation of the context in the LDOS-CoMoDa dataset [55] will serve as main foundation, given that it has been constructed by hand, from the selection of the variables to the data collection.

The user model is also a key feature of the RS to develop, since it will encode the information for relating user to items and context information. The approaches proposed by Cami et al. [21] and Aguilar et al. [3] result very interesting, since they are conceived in a probabilistic and graphical framework, respectively. Small differences concerning the characteristics of the problem need to be considered, but they show relevant key points in the model elicitation that will surely serve the development of the new model.

It is interesting to notice that almost none of the approaches examined incorporates expert knowledge to elaborate the recommendation. It might be due to the laborious task of knowledge elicitation, or inaccessibility to reliable sources. Nevertheless, it surely is one of the strengths of the proposed hybrid strategy.

Table 4.2 shows an element-wise comparison of the most relevant features of the model developed with this thesis, and some remarkable works discussed previously.

Method	Hybrid	Uses expert knowledge	Uses item content	Includes context	Based on ratings	Explicit user model	Focus on user	Predicts context
CHYBAM	✓	✓	✓	✓	✗	✓	✓	✗
Park et al. [72]	✗	✗	✓	✓	✓	✗	✓	✓
Huang et al. [38]	✓	✗	✓	✗	✗	✓	✗	✗
Baltrunas et al. [8]	✓	✗	✗	✓	✓	✗	✓	✗
Ono et al. [69]	✗	✗	✓	✓	✓	✓	✗	✗
Yuan et al. [109]	✗	✗	✗	✓	✗	✓	✓	✓
Breese et al. [15]	✗	✗	✗	✗	✓	✗	✗	✗
Cami et al. [21]	✗	✗	✓	✗	✓	✓	✗	✗
Aguilar et al. [3]	✗	✓	✓	✓	✓	✓	✗	✗

Table 4.2: Summary comparison

4.2 Recommendation strategies used as reference

The method proposed in this document details the process for offering personalized recommendations based on user model that considers user preferences, context data and content information of the items. This methodology was inspired by the previous work on a Context Model for Personalized Recommendations by Gonzalez [33].

The recommendation output is designed with a user-centric mindset. To test the hypothesis stated in this document, the new method is compared against one of the landmarks in the film recommendations domain, MovieLens, to assess the utility and goodness of the recommendations as perceived by the users.

Both of the above mentioned references that inspired and guided the study are detailed in the following paragraphs.

Context Model for Personalized Recommendations

The Context Model for Personalized Recommendations (CMPR) [33] proposes a method for modeling user's preferences and provide personalized results. It outlines the process by which explicit information about the user interactions with the items is collected and prepared for the subsequent steps. Information about the context is one of the main components of the model, and so is the content attributes of the items. Both are included

explicitly in the models, which allows to find relationships of how an item satisfies user's need in a specific situation.

The user model is represented graphically in a Bayesian Network, whose structure and parameters are defined by learning algorithms. Direct inference over the network is used to obtain the probability of user preference, or satisfaction, given explicit evidence.

The method is employed as the core recommendation engine in this research, and the complete process for obtaining an adequate model is detailed in Section 5.6.

MovieLens

MovieLens is an online service that provides movie recommendations, and is part of the GroupLens Research Lab at the University of Minnesota. It uses collaborative filtering techniques to generate personalized predictions based on information from at least 27 million movie ratings and 1,100,000 tag applications for 58,000 movies by 280,000 users, along with 14 million relevance scores across 1,100 tags ^a.

The user interface offers users the possibility to choose among 4 possible recommendation algorithms. The user experiments were conducted using "The warrior", which employs an item-item collaborative strategy [87]. The recommendation process starts by eliciting user's preference by using groups of similar items [22] and then, the user can start rating movies to obtain more personalized recommendations. The user must have rated at least 15 movies in order for "The warrior" algorithm to be available. MovieLens also offers additional personalization by assessing popularity of the recommended movies. All this functionalities were evaluated to identify if the recommendation process itself influences the overall satisfaction of the user.

^aAccording to the largest data set made public in September, 2018: <https://grouplens.org/datasets/movielens/latest/>

Chapter 5

A Context Hybrid Bayesian Model for Recommendations

Given the theoretical and practical foundations for both probabilistic graphical models (see Section 2) and recommendation technologies (see Section 3), it is pertinent now to fully explain the design of the Context Hybrid Bayesian Model (CHYBAM) for recommendations. The CHYBAM, as said by its name, is a hybrid model that combines the knowledge from domain experts with the advantages of machine learning techniques into a model capable of producing context aware recommendations based on probabilistic inferences with Bayesian networks (BNs).

The process for developing the CHYBAM starts by consulting experts in the domain to define both the knowledge base from which the whole system is based on, and the core relationships among these variables (Section 5.1), followed by data acquisition (Section 5.2) and preprocessing (Section 5.3). Different models are then generated from this information; first, a raw evaluation of the methods available is performed (Section 5.4) to then apply the best practices to generate functional BNs (Section 5.5 and 5.6). A thorough analysis of the structure of the different models is necessary to understand the reasoning patterns present in the networks (Section 5.7), including also performance evaluation of the models given by formal metrics (Section 5.8). All this information is crucial to conceive the inference process that defines how recommendations are produced (Section 5.9), which will lead to finally combine all the elements into the desired hybrid model (Section 5.10).

5.1 Expert knowledge elicitation

As seen in Section 2.5, the construction of a Bayesian network by hand is a nontrivial task. It requires expert knowledge elicitation to define either the structure or the parameters of the model, or both, and usually involves several iterations to verify that the model reasonably represents the problem.

For completing this process, the expert assessment was provided by Ccinemedia ^a, a group devoted to culture film appreciation, with broad experience in film exhibition and thus, constantly involved with public approaches to cinema. The experts directly involved in the task are Isabel Jiménez, with a specialization in Latin American Film Studies by UNAM ^b, and Héctor Robles, with a degree in Screenwriting by CCC ^c.

The knowledge base was constructed from a series of interviews with the film experts, where the first conversations were intended to define whether or not the variables present in the current CMPR [33] were enough to represent the user and context characteristics involved in the experience of watching movies.

Once all necessary variables were established, the experts put to test their capability to abstract movie features to recognize relevant patterns that help in the recommendation. This critical mindset is essential for the next steps, which directly involve defining the relevant attributes (of all the user, the movie, and the context) and how they influence each other.

Taking as reference the original 34 variables included in the CMPR, a new augmented set was defined to incorporate additional variables, mainly related to the user, as well as different experiences when watching movies, such as streaming services and non-commercial exhibition environments. Many other variables are directly related to specific attributes of the film that may influence the choice to watch a specific movie, or the overall experience that comes with it. After these considerations, the total number of variables to be included in the research was 47, which can be grouped in three main categories:

- **Experience related.** These variables capture information regarding where the user watched the movie, her motivation to watch it, and other contextual information such as day and time of watch, companion, whether he consumed food or had to go to the exhibition place. Overall ratings are considered here.
- **Film related.** These variables capture specific features about the movie, to identify possible patterns that reflect which ones are preferred by the user and motivated her

^aCentro de Cine y Medios Audiovisuales: <http://ccinimedia.blogspot.com/>

^bUniversidad Nacional Autónoma de México

^cCentro de Capacitación Cinematográfica

to watch the movie. These include the cast, the director, the genre(s), if the movie was awarded by some external organism, and even if the user heard of recommendations or reviews before watching the movie.

- **User specific.** These variables were included in order to determine if demographic factors beyond age are relevant for generating a recommendation. The specific information to consider includes occupation, education level, and gender.

5.2 Data acquisition

Following the path described to establish the data set from which the CMPR was generated, a survey was deployed to collect the necessary information for the new models. To obtain the data, the survey that served to generate the CMPR was modified. Most of the questions remained unchanged, but several more were added in order to include the new variables (see Section 5.1).

There are 49 questions in the final survey, but some of them are segmented so that certain answers lead to more specific questions. Hence 49 is the maximum number of questions a given user can answer, and the minimum is 34. For more details see Appendix A.1.

In total 808 responses were collected, but one of the observations in the original data set showed an inconsistent value for the age. Two possibilities were considered to deal with this entry, either the value was manually set to the average age, or removed from the data set. The later was considered more convenient, thus the total number of observations included in the following steps is 807. Table 5.1 shows a summary of the number of entries in each data set. It is important to notice that the data set initially collected for the CMPR (indicated as Original in the table) was constructed with two iterations of the survey, hence the missing values come from the initial observations where some of the variables were not considered.

Data set	Missing data	Complete	Inconsistent data	Total
Original	128	333	1	461
Augmented	0	347	0	347
Total	128	680	1	808

Table 5.1: Total number of observations for the different stages of data collection

The data was obtained mainly from college students ranging from 18 to 22 years old, from the upper-middle class in Mexico City's metropolitan area. It is important to consider

that interests and behaviors depicted by the models will be greatly influenced by this age group. In general, the results will be inherently biased towards most popular practices, which will have a great impact on the final recommendations.

5.3 Data pre-processing

The process of preparing the data for the following steps requires special attention since it constitutes the outline of how information will be treated. Therefore, it must establish useful level identifiers for categorical variables, as well as identifying appropriate rules for discretizing the continuous variables, if applicable.

Table 5.2 shows how the variables were encoded, and in which model they will be included (KB indicates all the models derived from expert knowledge). Following the pattern set by the original data set, the variables that were also considered with the new survey are set with identical levels. For the new variables, suitable levels were defined. Only three variables have a continuous value, these are the two rating scales (for the movie and the experience) and age. Discretizing the age implies many assumptions and, probably some information loss. The initial approach for selecting relevant intervals was made according to the Scott's choice for a normal distribution [90]. However, the ranges were assessed by the film experts to reflect a better understanding of the needs and behaviours of the different age groups. With this in mind, the age can be grouped into 7 categories: users younger than 15 years old, between 16 and 18, 19 to 22, 23 to 27, 28 to 35, 36 to 54, and older than 54. This categorization matches the general audience definition in Mexico by IMCINE^d [39].

The other two continuous variables besides age are user ratings for the movie and for the experience, where 1 is the lowest score possible. The first one is a straightforward score for the movie. The later encompasses a net satisfaction score for the different factors involved in the experience of watching the movie, such as companion, food, place, among others. Another way of interpreting this value could indicate how likely the user is to repeat the same pattern when watching movies.

An important remark must be made regarding user ratings. Since these are completely subjective and are given arbitrarily, the same score assigned by two different users, or even by the same user in different moments, can represent completely different appreciations of the same variable. Consequently, ratings must not be treated as absolute scores, but instead as a general tool to identify overall appreciation of the evaluated attribute.

The categorical variables included in the study represent most of the contextual

^dInsittuto Mexicano de Cinematografía

factors. These identify the key elements users deal with when watching a movie, whether they are watching at home through a streaming service or at the movie theater. One of the most interesting variables to analyze is `motivation`, which specifies why the user decided to watch the movie, and can be decisive in the recommendation. Other variables answer to questions such as at what time did the user watch the movie (`movie_showtime`), whether she was alone or accompanied (`companion`), if the movie theater was far away (`distance`), or if she consumed any snacks while watching the movie (`bought_food`).

When it comes to the genre of the movie, two different approaches are considered. The first one assumes genres as individual attributes a movie has or not, and uses logical assignments to indicate the combined genre. The second one combines the possible genre and subgenre combinations into three categorical variables. The genre classification of the movies directly comes from the information available at The Movie Database (TMDb)^e.

Logical variables, beyond the special treatment of the genre previously mentioned, help identify which aspects of the movie the user considered relevant for choosing the movie, this include recommendations from friends or family (`recommendations`), the director, cast, `special_effects` or the awards, among others. Some other variables such as the plot, the runtime, the studio, or the music were included in the survey for consideration, but discarded at the end in the models. Further discussion on this regard is given in Section 5.4.

The complete data set product of this process is available at https://bit.ly/cinescope_data.

Table 5.2: Variables included in the different models

Variable	Description	Type	Levels / Range	KB	CMPR
age	User's age	Continuous	12 - 70	✓	✓
motivation	The reason to watch a movie	Categorical	DG - Distraction/hobby DM - Watch this movie DN - No special reason RQ - Recommendations UN - Only one available	✓	✓
language	The language spec. preferred by the user	Categorical	S - Subbed D - Dubbed O - Original	✓	✓

^e<https://www.themoviedb.org/>

CHAPTER 5. A CONTEXT HYBRID BAYESIAN MODEL FOR RECOMMENDATIONS 47

tickets	Method employed to by the tickets	Categorical	CW - Cinema web page ET - Electronic box office MA - Cinema mobile app. OA - Box office	X	✓
movie_showtime	At what did the user watch the movie	Categorical	BN - Before noon AN - Afternoon NT - At night LT - Late at night	✓	✓
first_time	Is this the first time the user watches the movie?	Categorical	t - True f - False	X	✓
companion	With whom the user watched the movie	Categorical	AL - By him self FA - Family FR - Friends SO - Significant other	✓	✓
movie_decision	Who decided which movie to watch	Categorical	EV - Everyone CO - Companion ME - User	X	✓
bought_food	Did the user consumed food while watching the movie	Categorical	t - True f - False	X	✓
food_source	Where the food was obtained from	Categorical	NA - No food consumed AC - Nearby store CI - Cinema HO - Home	X	✓
distance	Distance traveled to the exhibition place	Categorical	NA - No distance traveled FA - Far away NC - Not very close NE - Near RC - Really close	✓	✓
transportation	Transportation mean if applicable	Categorical	NA - No distance traveled BI - Bicycle CA - Car OF - By foot PT - Public transportation	✓	✓
available_time	How much time the user had available to watch the movie	Categorical	EES - More than enough JEA - Enough to watch any movie JEE - Enough to watch this specific movie	✓	✓

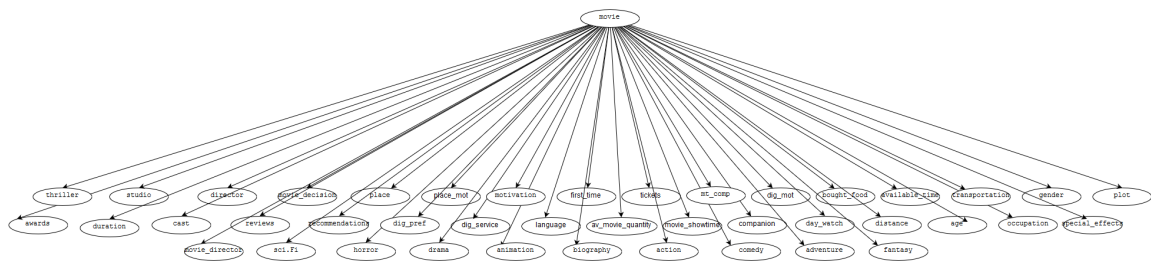
movie_rating	User's rating for the movie	Continuous	1-5	✗	✓
exp_rating	User's rating for the overall experience	Continuous	1-5	✗	✓
director	Whether the user takes into account the movie director	Categorical	t - True f - False	✓	✗
recommendations	Whether the user considers recommendations	Categorical	t - True f - False	✓	✗
cast	Whether the user takes into account the cast	Categorical	t - True f - False	✓	✗
special_effects	Whether the user watch a movie for its special effects	Categorical	t - True f - False	✓	✗
awards	Whether the user takes into account the movie awards	Categorical	t - True f - False	✓	✗
place	Where did the user watched the movie	Categorical	CM - Traditional movie theater NC - Non-commercial theater DP - Streaming service PH - Physical copy or TV OT - Other	✓	✗
regular_user	Does the user frequently watch movies in this place?	Categorical	t - True f - False	✓	✗
movie	Movie title	Categorical	Movie title	✓	✗
genres	Movie genres ^f	Categorical	Action, Adventure, Animation, Comedy, Crime, Documentary, Drama, Family, Fantasy, History, Horror, Music, Romance, Science.Fiction, TV.Movie, Thriller, War, Western	✓	✓

^fAs specified by TMDb

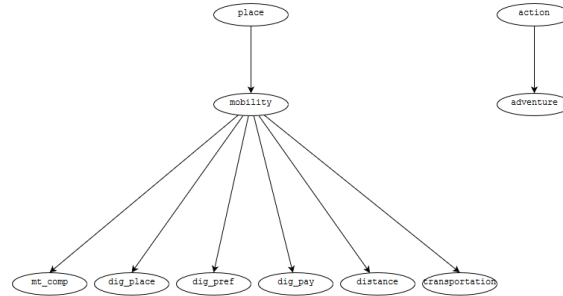
5.4 Development of the models

The first attempts to generate a Bayesian network structure which includes the complete augmented set of variables, 65 in total, resulted poorly connected graphs with causal relationships only associating the movie to some specific movie attributes. A visual analysis of this models in Figure 5.1 can compare them to Naïve Bayes Classifiers, in which the variable on top is strongly dependant on its children. Also, strong independent assumptions among the leaf nodes are implied by these models, which do not match the desired behaviour of the model.

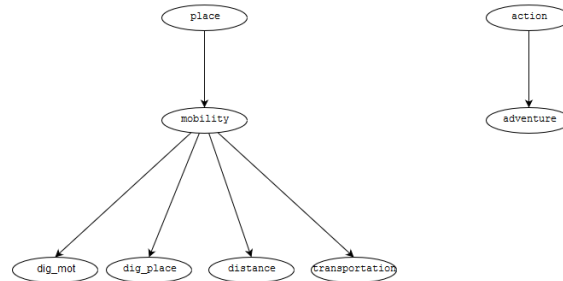
Bootstrapping was applied in order to verify the strength of the probabilistic relationships inferred from the data, also testing the validity of the direction. From a partition of the new data set with 250 observations, 200 estimations with sample size equals 30 were executed to inspect the effect of the objective function with different learning algorithms. The combination of the Hill Climbing algorithm aiming to maximize the conditional linear Gaussian log-likelihood estimation (loglik-cg) gave identical results to the Tabu search with the same score, as appears in Figure 5.1a. In this network, 42 variables are directly dependant on the movie, and the other 22 are not connected at all. The same experiment was held using Hill Climbing with Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC), but both generated structures with small number of connected nodes, as seen in Figure 5.1b and 5.1c. Consequently, the most presumable deduction is that the number of variables must be reduced in order to emphasize only the most relevant variables and its relations.



(a) Tabu search and Hill Climbing with loglik-cg



(b) Hill Climbing with AIC



(c) Hill Climbing with BIC

Figure 5.1: Structures of the different Bayesian Networks generated using bootstrapping

Defining variables to include in the models

The first step to ensure a congruent generation of the models is determining the variables to include in the recommendation process. Given the network assessed by the film experts (see Section 5.5.1), only 19 variables were included in the new models (identified by KB in Table 5.2), while the CMPR maintained its original 34.

From a follow-up consultation with the experts, one of the most influential conclusions is the relevance of associating each movie with a combination of three genres, which will be included in three categorical variables. Following the original setup for the CMPR, 19 genres were established as logical variables. Even though this

definition can be more complicated to deal with given the heterogeneity and increased variability, which even can be a diminishing factor for the quality and strength of the recommendation, is the core foundation where the whole recommendation strategy resides.

Also, to consider `motivation` and `place` as essential decision factors for the recommendation is decisive in the subsequent models. Mobility aspects are also important, and will be tested for their usefulness in the recommendation.

5.5 Knowledge-based models

Following the assessment provided by the film experts, a general Bayesian network was defined with the fundamental relationships involved in the recommendation process according to them. Such model is described in Section 5.5.1. Machine learning techniques were applied to obtain more robust models that inherently contain information from the data and reflect such conditions in their structure. First, the effort to combine directly the knowledge from both sources is described in Section 5.5.2. In Section 5.5.3 is defined a model which only follows the inherent relationships of the 19 variables defined by the experts, but without any direct influence over the structure.

5.5.1 Pure expert knowledge

The model defined by the experts comprise 18 variables, and is shown in Figure 5.2. It can be implied that the most influential aspect for the motivation to watch a specific movie is the information the user has about the it, since there are many root nodes which have a direct influence on the `motivation` node. Another interesting aspect can be noticed at the bottom-left of the network, where a connected component is relating where the user watch the movie (`place`) with the time at which he watched it (`movie_showtime`) and mobility related variables, which identify whether the user had to travel to watch the movie, and if so, how far (`distance`) and by which transportation mean (`transportation`).

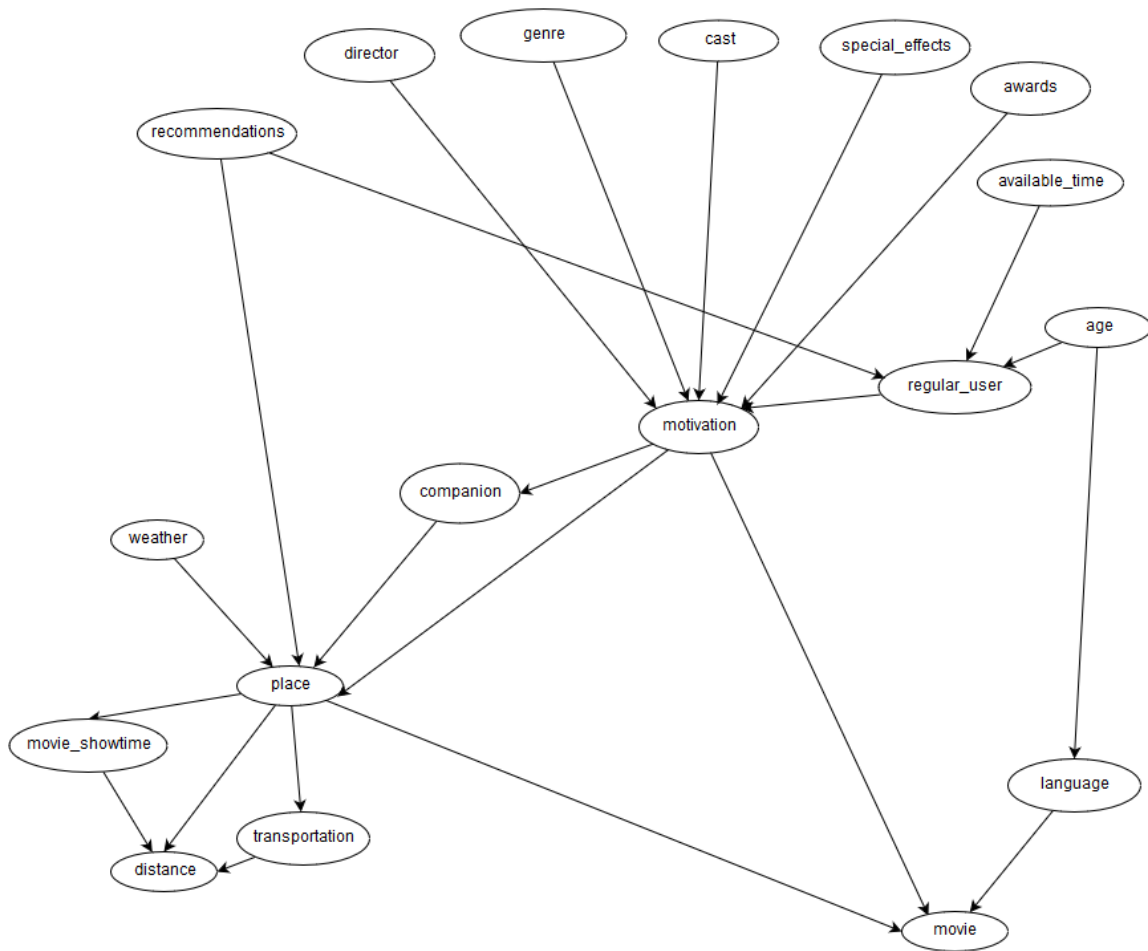


Figure 5.2: Bayesian Network defined by the experts

`weather` is a relevant variable in the model, but since the retrieval of accurate information to fit the data base is basically impossible, it was removed from the rest of the experiments. Nonetheless, it would be advisable to keep the variable when dealing with a situation where actual information about the weather in real time can be automatically fetched.

Usually, experts may assess the CPTs for each node, but given the high number of possible instances of each variable, the task became nearly impossible; thus, the definition of such crucial aspect of the BN was left for automated learning algorithms. The model is implemented in the programming language R using `bnlearn` package [92]. The first set of data in which the model was tested consisted only in 117 observations; with the available information, the fitting method failed to find support for two arcs in the original

structure. Those were the ones coming out from `age` to `regular_user` and `language`, respectively. This irregularity can be explained with the high variance of the `age`. However, with an increased data set, this issue disappeared, and all the arcs were maintained. A further revision of the performance of the network is given in Section 5.8

5.5.2 Expert knowledge as seed

The model obtained by assessment of film experts on the causal relations among variables serve as a core structure used in the development of a more robust and enhanced network, which reflects the expert knowledge and is validated by the observed data. The assessment of the network is made through explicit specification of the desired edges, i.e. directed connections that must be present in the output network are set as restriction for the learning algorithm.

The structure of the network obtained with such specifications is shown in Figure 5.3. The model was obtained through 10 times 10-fold cross validation, using Hill Climbing algorithm. Different scores were tested as the optimization variable to determine the fitness of the models. Given that all variables are discrete in this network, the score functions used in the learning process are AIC, BIC, K2. Additionally, the MMHC algorithm was tested using as utility function the BIC score.

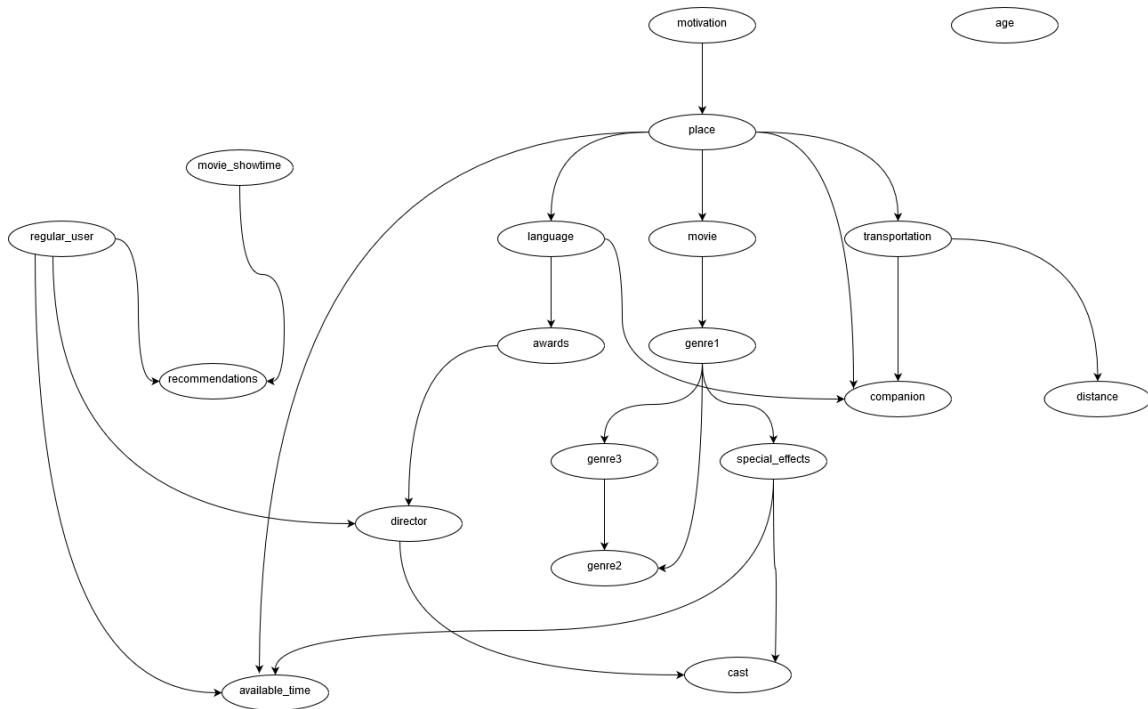


Figure 5.3: BN obtained with ML methods and expert knowledge assessment

After the structure was learned appropriately, the parameters had to be fitted. The structure learning process was executed considering only 347 data observations that correspond to the number of responses obtained for the new data set, assessed also by the experts to contain all the relevant information. Then, the parameters, or Conditional Probability Tables (CPTs) were fitted using the complete set of 807 observations, where some data had to be imputed to the original 460 observations. The fitting of the parameters was also accomplished by means of 10-runs of 10-fold cross validation with different score functions. For fitting the CPTs there are two methods available: Maximum Likelihood parameter Estimation (MLE) and Bayes parameter estimation (Bayes) (see Section 2.4.4). Given that the previous step, it was the $K2$ score which gave the best results, both parameter approximation methods were applied to this model. The loss results are shown in Figure 5.5a.

The selected network for performing the inference in the following stages of the recommendation process was the one obtained with the $K2$ score, and the Bayes estimation. Even though the loss values were lower with the MLE estimation or MMHC algorithm, the predictions performed with those networks showed some deficiencies, as the Bayes estimator yields better predictions (see Section 2.4.4).

5.5.3 Vague expert knowledge

A few methods were executed to develop a suitable structure for the BN that reflects the information available solely in the data, without any expert knowledge assessment. At the very beginning of the study, and before selecting the more relevant variables (as seen in Section 5.4), two greedy algorithms were compared to evaluate promising structures. Hill Climbing and Tabu search were tested with different scores as optimization functions: log-likelihood estimation, Akaike Information Criteria and Bayesian Information Criteria. The obtained structures showed poorly connected graphs, hence not viable for inference (see Section 5.4).

After the variable selection was made, several scoring functions were tested with the Hill Climbing algorithm to obtain a suitable model to compare against the assessed model. The results are listed as follows:

- Log-likelihood: not possible to determine a suitable model with the available data.
- Akaike Information Criterion: the structure was learned properly, but some important node were disconnected. This result allowed further tests with this score.
- Bayesian Information Criterion: the final graph showed various clusters among the variables and was not connected completely.
- Logarithm of the Bayesian Dirichlet equivalent: not possible to determine a suitable model with the observed data.
- Logarithm of the Bayesian Dirichlet sparse score: some errors were encountered while attempting to create the network related to the levels observed in the data.
- Logarithm of the modified Bayesian Dirichlet equivalent: not executed due to computational limitations.
- Logarithm of the locally averaged Bayesian Dirichlet: not executed due to computational limitations.
- Logarithm of the K2 score: the structure generated was excessively big, and further approximations with this model was impossible due to memory requirements.

With these considerations made, 10 runs of 10-fold cross validation were executed over the reduced set of variables selected by the experts, and the structures of the BNs were obtained without any expert knowledge assessment. AIC and BIC scores were the only ones capable of producing suitable networks, due to limitations in the computational

and memory requirements. The loss values for the model learned from data are shown in Figure 5.5b.

Considering the loss results, and a visual inspection of the edges present in the two different models, the one selected to implement the system was the one obtained with AIC. The BN structure of the such model is shown in Figure 5.4.

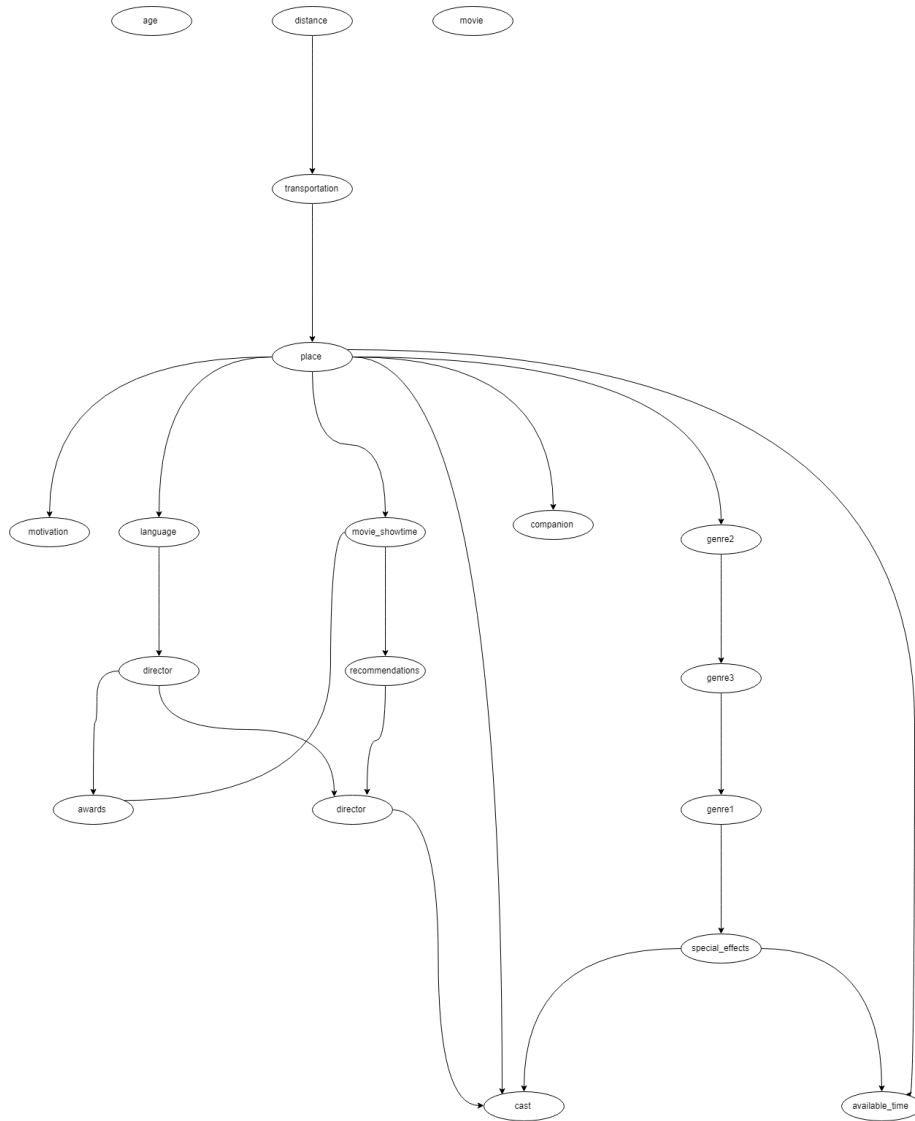


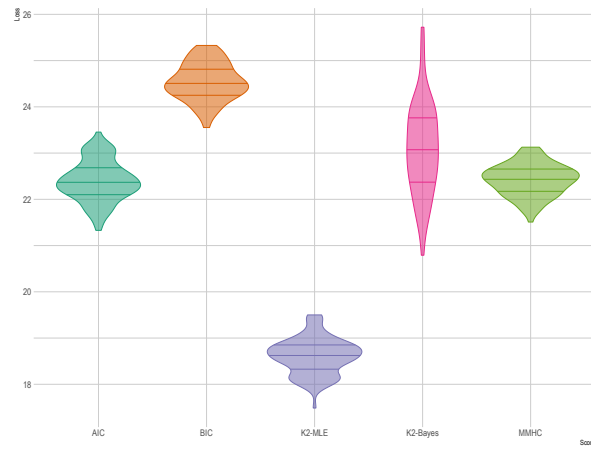
Figure 5.4: BN structure learned from data using HC algorithm with AIC

5.6 Replication of the CMPR

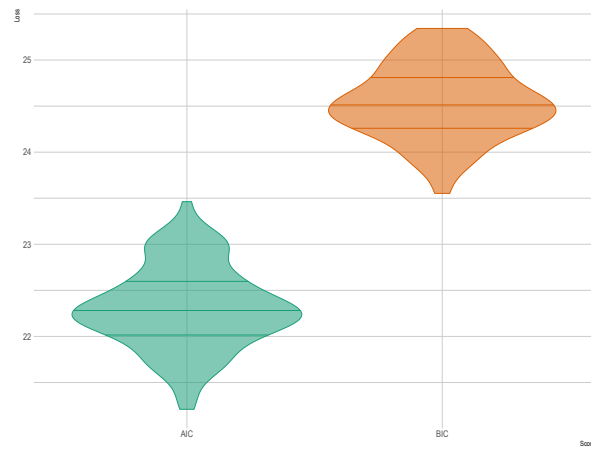
The structure of the CMPR was previously defined by González [33]. However, when trying to fit the new data into the same structure, the information available in the data showed violation of some of the restrictions imposed by the previously defined arcs. Therefore, the best solution was to proceed following the same method as for the model defined in Section 5.5.2, and create a new structure that better fits the data while also considering the previous structure to assess the learning process. Also 10 runs of 10-fold cross validation were executed for both learning the structure and fitting of the parameters. Given that this network includes some continuous variables (ratings and age), the scores used to calculate the loss of the networks are conditional Gaussian:

- Conditional Gaussian Akaike Information Criterion (AIC-CG)
- Conditional Gaussian Bayesian Information Criterion (BIC-CG)

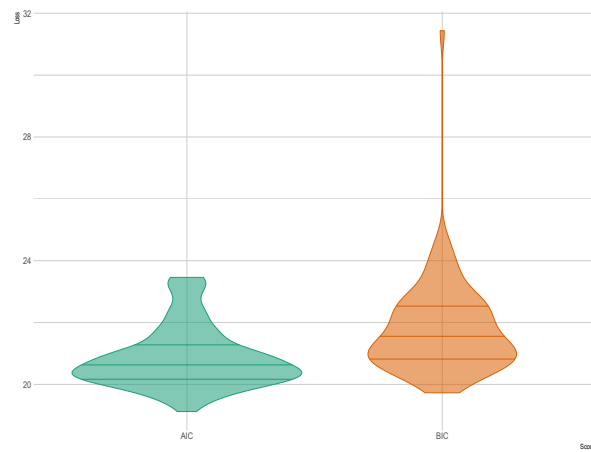
The loss results are shown in Figures 5.6. From this evaluation, the selected network structure is the one obtained with the AIC, and the corresponding network structure is shown in Figure 5.5c.



(a) From experts



(b) From data



(c) CMPR

Figure 5.5: Loss results from the cross validation with different scores for the models

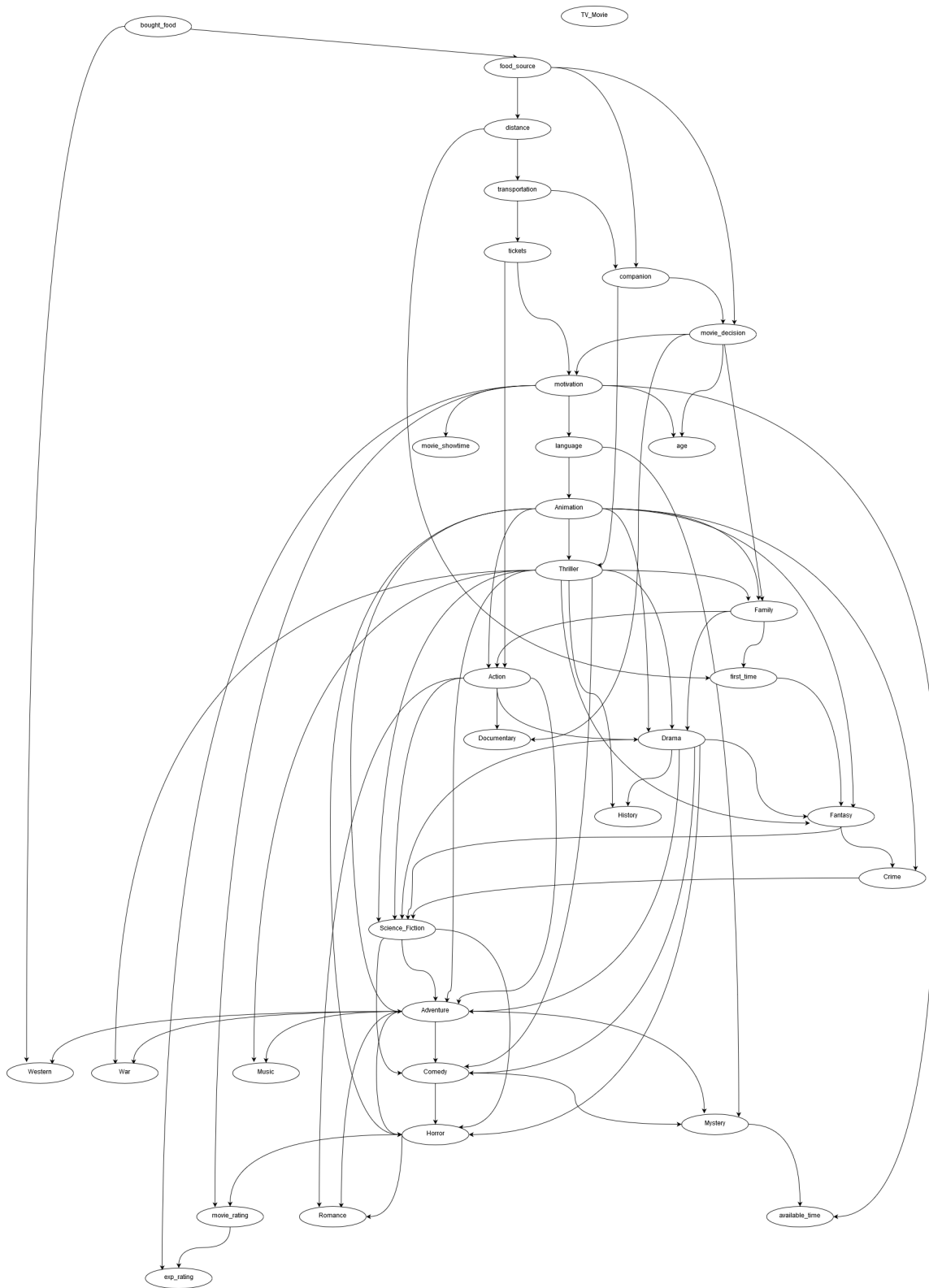


Figure 5.6: CMPR obtained with ML methods including the original variables

5.7 Analysing the models

Despite all the models have been obtained from the same information set, the different factors considered in each parameterization scenario have produced relatively different models. These differences mainly reside in the arcs connecting the variables and their intrinsic probabilistic relationship. To verify the validity of the graphs obtained, it is possible to compute a measure for the strength of each arc by conducting a confidence test while keeping the rest of the network fixed [91].

This procedure is performed by applying a conditional independence test where the null hypothesis would be that to drop each arc of the network, one at a time, producing then a p -value for each connection. In this sense, the lower the p -value, the stronger the relationship between the two variables linked by that edge. Figure 5.7 shows all the arcs present in the network defined by film experts. The stronger arcs are represented with a darker hue. For example, the path `place` \rightarrow `transportation` \rightarrow `distance` is confirmed to be a strong relationship when tested against the collected data. On the other hand, all the arcs that arrive to the node `motivation` do not find strong support in the data, therefore those p -values are close to 1.0.

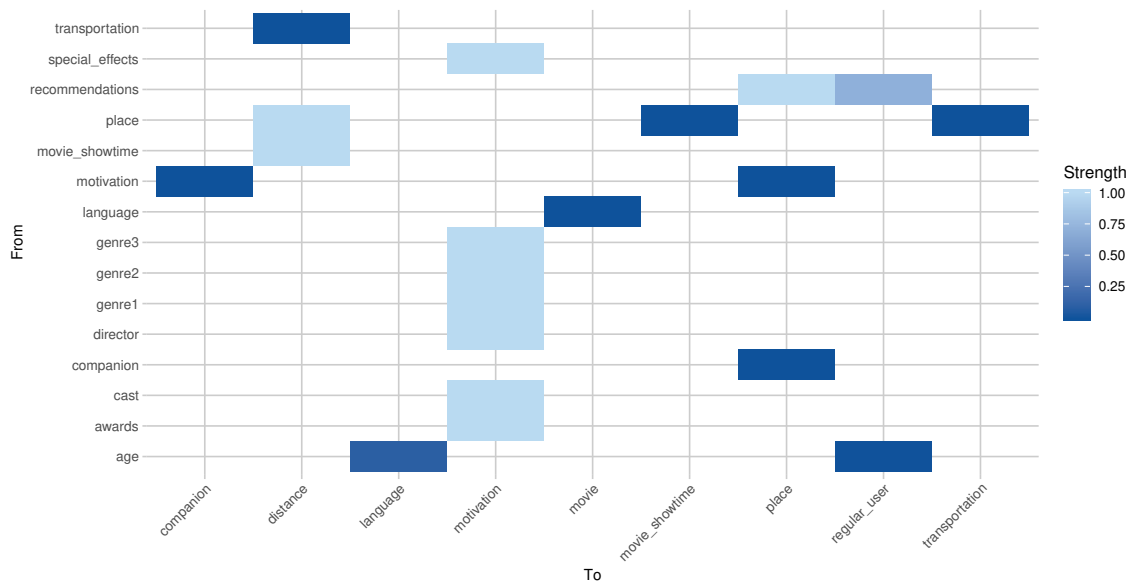


Figure 5.7: Arc strength for the model defined by film experts

When analyzing the two models derived from expert knowledge, the same test indicates overall stronger relationships. This improved overall relevance of the arcs

comes indeed from the learning techniques applied to obtain these models. In Figure 5.8 is shown the strength representation for the network obtained by using expert knowledge as learning seed (see 5.5.2). Some interesting relationships are brought up to light, most of which were not originally foreseen by the experts.

For instance, the trail `special_effects` \rightarrow `cast` shows an unexpected strong relationship indicating that as the user considers, or not, the special effects of a movie before watching it, she also does the same for the cast. In addition, the content attributes of the movie, such as the genre or the language, do not keep strong relation among them as could be expected. On the other hand, some of the original relationships defined by the experts are well identified in the data, such as `place` \rightarrow `transportation` \rightarrow `distance`.

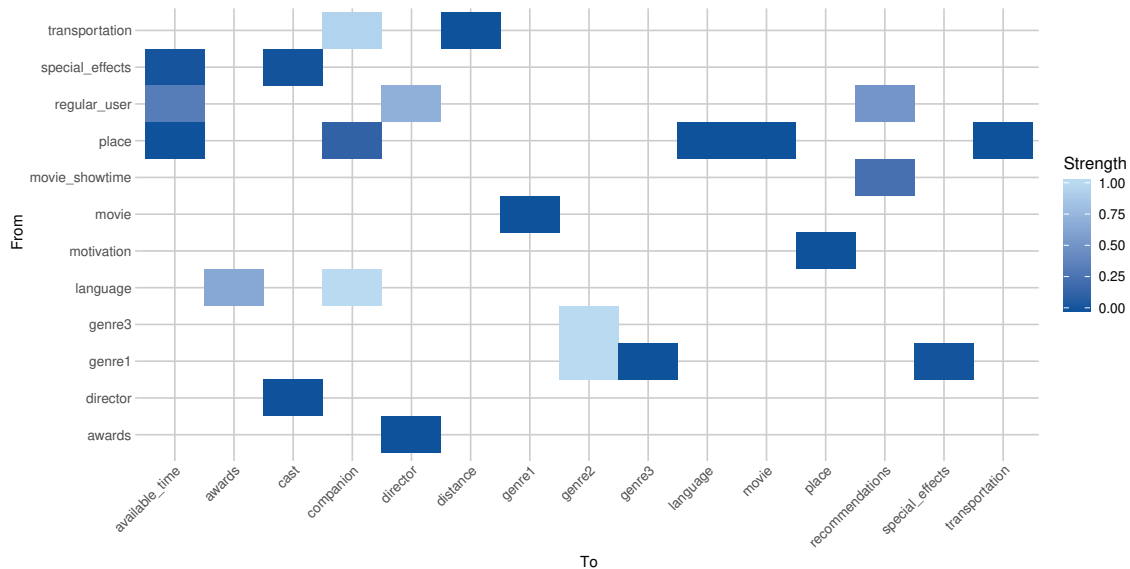


Figure 5.8: Arc strength for the model obtained with expert knowledge as seed

In the model obtained with vague knowledge assessment, as seen in Figure 5.9, the arcs present have a substantial increase in their relevance given the lower p -values of the tests. There is a prevailing trail among the three models, which continues to include the same variables but with an inverse direction in this model. `distance` \rightarrow `transportation` \rightarrow `place` implies that the place where the user chooses to watch a movie is not the first thing to consider, instead the means available to watch it define the subsequent considerations.

All the models that include some kind of expert knowledge present similarly dense networks. The explicit model defined by the experts has a total of 20 arcs, from which only

3 and 2 are common with the models with seed and vague assessment respectively. The model with the knowledge seed has 23 arcs in total, 8 of which are shared with the model with vague knowledge assessment which has a total of 21 arcs. It is possible to understand that these differences come from the learning algorithms used for producing the models. These techniques are designed to optimize the networks by successive additions, removals or reversals of the arcs, allowing to find interesting relations among the variables that improve the global performance score of the model.

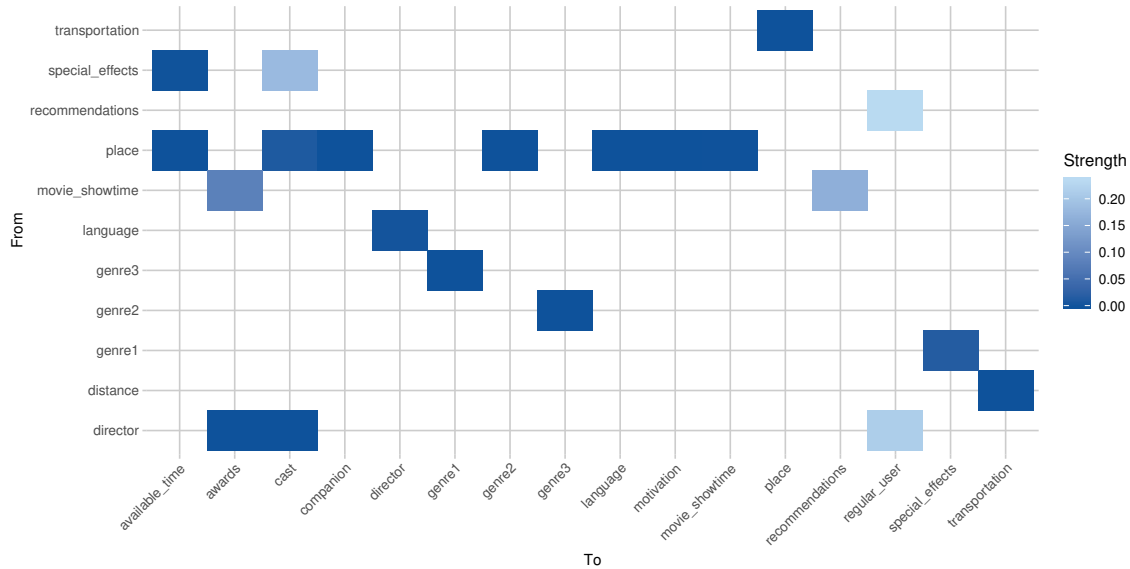


Figure 5.9: Arc strength for the model learned from data and vague expert assessment

As seen in Figure 5.10, the CMPR shows an arc set which is proven to be highly consistent with the natural distribution of the data. This is verified by the considerably small p -values obtained from the tests, with more than 90% proving to be significant below the confidence level $\alpha = 0.05$.

Given the variables included in this model, the relationships found between movie attributes and different context conditions result very interesting. For example, the connection `tickets` \rightarrow `Action` directly indicates that the mean to purchase the movie ticket is related to the genre of the watched movie. A more sensible analysis must consider that the most popular new releases are blockbusters, commonly mixing action, adventure and some other genres. Therefore the movies that are most watched at the cinema, which requires necessarily to buy tickets, are action movies. This link is more a result of the market status quo, or film industry strategies, than a real user inclination.

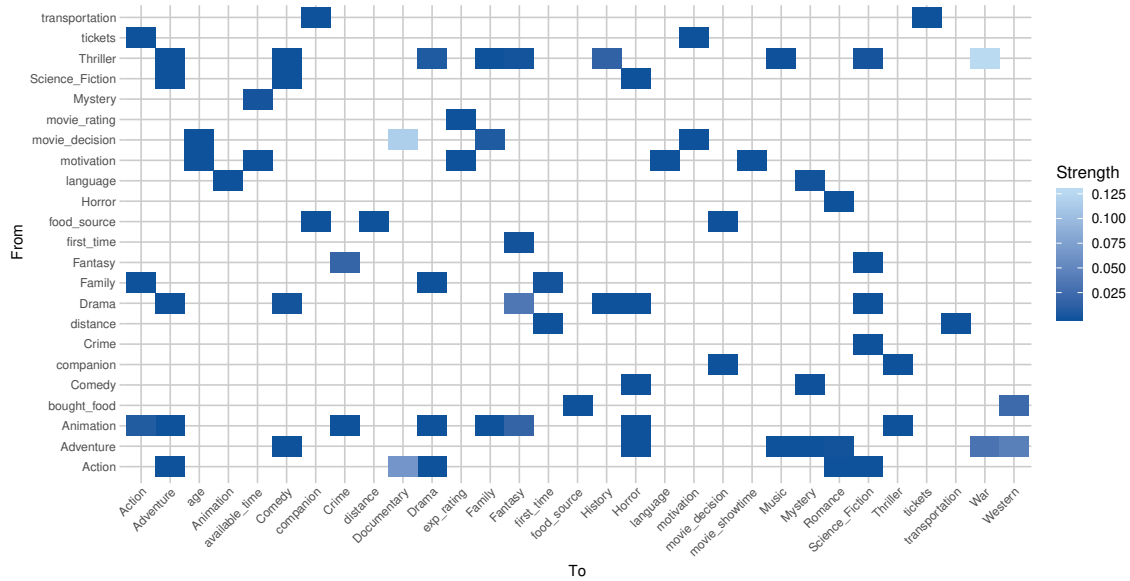


Figure 5.10: Arc strength for the CMPR

The characteristics of the movie, given by the genres, a more complex structure than the simple intuitive considerations that can be made through direct observation of the most common genres. The inclusion of all the different genres as individual variables allowed to fully understand how they interact with user-s preferences and context information. These relationships are the foundation of the conditional probabilistic reasoning behind the recommendation process, as described in Section 5.9.

5.8 Evaluation

Before continuing with the design of the hybrid recommendation system that combines the Bayesian networks developed in Sections 5.5 and 5.6, it is necessary to verify the validity of the information contained in the models. Considering the nature of the models, two different approaches can be taken to evaluate them, both of which are considered in sections 5.8.1 and 5.8.2, respectively.

5.8.1 Measures from graphical models

When describing graphical models, a pertinent measure to observe is the Markov blanket size. The Markov blanket of a node is defined as the set of the parents, the children, and all

the other nodes sharing a child with a given node [78]. For a complete graph, the average Markov blanket size is computed as the average of the individual Markov blanket size of each node.

Similarly, the branching factor and the neighborhood size are good indicators of the density of a given graph. More specifically, the average branching factor in a Bayesian network is defined as the average number of children of each node; and the average neighborhood size represents the average adjacent vertices to every node in the graph [93]. In Table 5.3 are shown the results for these measures for each of the different models.

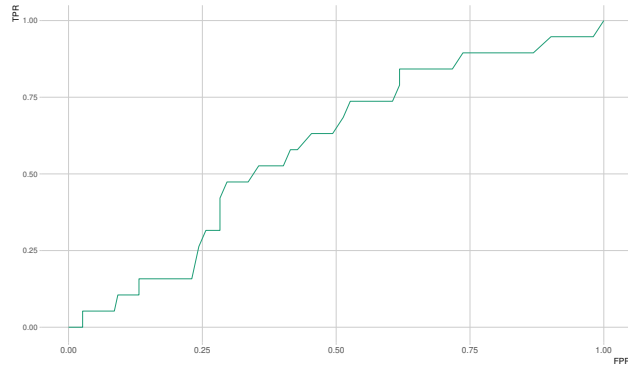
Model	Arcs	Avg. Markov blanket	Avg. neighborhood	Avg. branching factor
A	20	4.74	2.11	1.05
B	23	3.16	2.42	1.21
C	21	2.74	2.21	1.11
D	76	5.94	4.47	2.24

Table 5.3: Relevant measures for comparing the different graphical models
 NOTATION. **A:** Pure expert knowledge, **B:** Expert knowledge as seed, **C:** Vague expert knowledge, **D:** CMPR.

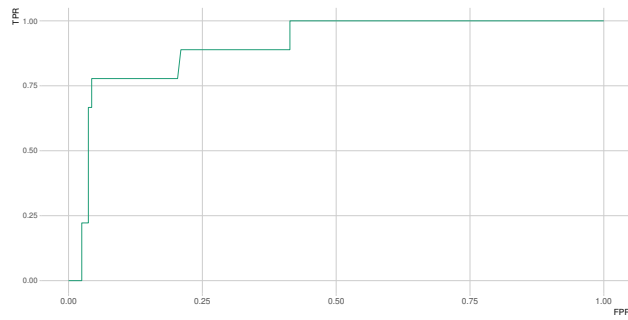
By comparing the metrics for the three knowledge-based models, it is easy to see that the models keep a close relationship with the original structure defined by the experts while also reducing the complexity, as the smaller number of nodes (in average) in the Markov blanket indicates. For the graph representing the CMPR, these metrics almost doubles the average value of the other networks, but considering that the number of arcs is almost the triple, the magnitude is reasonable and the comparison to the other networks cannot be done by simple looking at those numbers.

5.8.2 Measures from algorithmic performance

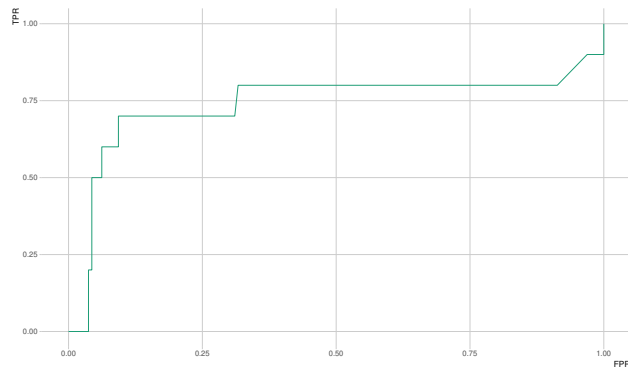
The learning algorithms that were used to produce the Bayesian networks allow to easily compare the models by the loss obtained using the associated score. Also, the predictive properties of the networks can be used to estimate the performance by analyzing the Receiver Operating Characteristic (ROC) curve. In Figure 5.11 are shown the corresponding ROC curves for all the models. Even though the experts designed the core model manually, it is possible to fit the model to the data and observe its behaviour.



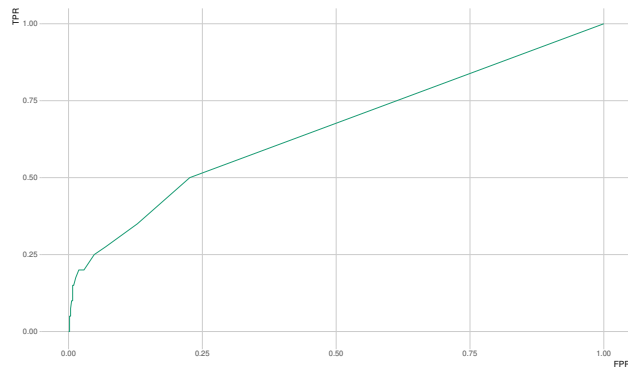
(a) Model defined by the experts



(b) Model with expert knowledge as seed



(c) Model with vague knowledge assessment



(d) CMPR

Figure 5.11: ROC curves for the different models

Similarly to the graph analysis (see Section 5.8.1), the two models derived from expert knowledge, in Figures 5.11b and 5.11c respectively, show significant improvements in the performance compared with the experts' structure. The AUC results confirm good predictive capabilities for both models as well. Table 5.4 shows a summary of the results for the loss and AUC obtained from the different models.

Model	Score	Loss	AUC
A	-	-	0.5855
B	K2	17.4866	0.9043
C	AIC	21.2091	0.7385
D	AIC	19.9229	0.6546

Table 5.4: Loss and AUC values for the different models

NOTATION. **A:** Pure expert knowledge, **B:** Expert knowledge as seed, **C:** Vague expert knowledge, **D:** CMPR.

5.9 Inference to obtain recommendations

As seen in Section 2.3 the most popular approximate methods to obtain reliable probability inference from the networks are logic sampling (*LS*) and likelihood weighting (*LW*).

Given that both models developed from expert knowledge are completely discrete networks (see Sections 5.5.2 and 5.5.3), *LS* is used to obtain the probability distribution of the desired node(s). On the other hand, *LW* is implemented to perform conditional queries over the CMPR, which benefits from the Gaussian distributions present in some of the nodes. In order to determine the best method to obtain valuable information from the models that leads to the final recommendations, different settings for the evidence and the hypothesis were tested.

The first tests were intended to obtain the movie prediction directly as a result from the inference process, using as evidence some context information. This evaluation led to very poor results since the available data with which the models were fitted consists of 807 observations, and only 291 different movies, which implies a high variability rate and low probability for each value. Thus, regardless of the sampling method employed, the inference over the `movie` node is impractical. Even if the conditional test results in a movie title with high probability, the set from which the movie can be selected is very small, and in the long run several users will find exactly the same recommendations.

Some other tests were executed to determine the effect of fixing the variables to their different levels. In this sensitivity analysis, conditional probability queries were performed using as evidence all the possible values for all the variables, one at a time,

and producing probability distributions for the movie attributes, i.e. the genres. While some of these combinations resulted in incongruousness for the reasoning with the Bayesian networks, some others demonstrated that many variables have no direct influence over the final probability distribution of the test node. This can be understood directly from the structure of the networks, since instantiating certain variables blocks the flow of information from one node to another.

For example, in the V -structure $recommendations \rightarrow place \leftarrow companion$ present in the original network defined by the experts (see Figure 5.2), $recommendations$ cannot influence $companion$, unless $place$ or one of its descendants are observed as evidence. On the contrary, if evidence is given for $companion$ information cannot flow from $motivation$ to $place$ in the trail $motivation \rightarrow companion \rightarrow place$. These type of considerations had to be taken into account in order to perform informative queries, which helped to define the nodes that influence the most the target variable.

There exists also the possibility that the combination of some specific instances of the evidence do not find support in the data. In these cases, the model fails to produce an informative outcome, and probability of the evidence is undefined. This flaw arises due to the lack of evidence in the data, which implies that not all the possible combinations, or context scenarios, are observed. Hence it is not possible to conduct exhaustive tests to verify the behaviour of the networks under different and rare conditions.

To select the nodes that produce a consistent and relevant outcome when tested as evidence in the query, it was necessary to evaluate each arc of the network to validate its strength (see Section 5.7). The relevant variables for each BN determined by those tests are listed in Table 5.5; these will be used as evidence when performing probability queries to the different models.

CMPR	Expert knowledge seed	Vague expert Knowledge
bought_food	place	distance
motivation	companion	motivation
tickets	language	companion

Table 5.5: Variables to include in the conditional probability query of each BN

Thus, directly including this information in the inference process, the models derived from expert knowledge can be used to obtain information about the relevant genres given the information in the context. For instance, using the model with expert knowledge as seed, the query

$$P(\text{genres} | \text{place} = \text{CM} \ \& \\ \text{language} = \text{D} \ \& \\ \text{companion} = \text{FA})$$

produces a probabilistic distribution of the genre nodes based on the evidence, resulting in an appropriate combination of the most likely genres to watch under those circumstances. Action, Adventure, Science Fiction and Thriller are the recommended genres for a user who chooses to go to the movie theater ($\text{place} = \text{CM}$) with her family ($\text{companion} = \text{FA}$), and watch a dubbed movie ($\text{language} = \text{D}$).

Accordingly, the queries performed with the CMPR can serve to evaluate the probability of watching any genre in a movie, and having a good time. A possible query to the model is defined by

$$P(\text{exp_rating} \geq 4 | (\text{Action} = \text{t} \ \& \\ \text{bought_food} = \text{t} \ \& \\ \text{motivation} = \text{DM} \ \& \\ \text{tickets} = \text{OA}))$$

where the probability of having a very enjoyable experience ($\text{exp_rating} \geq 4$) watching an action film ($\text{Action} = \text{t}$) is calculated given that the user did consumed food while watching the movie ($\text{bought_food} = \text{t}$), she specifically wanted to watch this movie ($\text{motivation} = \text{DM}$) and bought the tickets at the box office ($\text{tickets} = \text{OA}$). The estimated probability for this specific scenario is 0.6372, which can be broadly understood as an overall good experience. This is the type of queries that will be used in the recommendation process described in Section 5.10.

5.10 Hybridization Method

Once all the models are characterized and it is possible to perform informative queries to them, the next step is to consolidate the hybrid model which must take advantage of the individual strengths to produce a relevant recommendation given the user's contextual information.

For selecting the hybridization technique, two of the most common hybridization methods were considered: *cascade* and *switching*. As defined in Section 3.4, the main difference between these approaches is the evaluation of the known information. In the first one, consecutive passes through the different RS produces a more refined recommendation, while the second one considers from the beginning the fittest RS and only uses that one for producing the recommendation.

However, due to the capabilities of the inference queries over the different models (see Section 5.9), it is not convenient to implement the switching method. This approach would imply losing some information which is only present in one of the three models. For example, a query containing information about the tickets bought could not be answered by the models based on expert knowledge; or another query asking for the probability of watching a movie on a streaming service would not be resolved by the CMPR.

From this reasoning, follows the possibility of conforming a pipeline which consists of two main steps: first generate the appropriate combination of genres with the knowledge-based models, and then evaluate the probability that such combinations provides a satisfying overall experience for the user. Then, the two stages of the recommendation process will be fulfilled as follows:

1. Generate a list of items to recommend. This is commonly approached by most recommenders by a scoring function $s(i, u, q, x)$, where i is the item to recommend, u is the user, q is the query and x is the context.

In the proposed strategy, this stage corresponds to the inference of the combination of genres according to the contextual information.

2. Order the list of items according to specific criteria. The ordering function may depend on explicit attributes that are preferred, or by sorting the items according to the rating score or another representative measure. In the general form, this function is $O(I, u, q, x)$, where I is the set of items to order.

In the hybrid algorithm, the order is given by the inference of the probability of enjoying certain genres obtained from the CMPR query.

In other words, the proposed HRS first generates a list of genres that are more likely to be combined in a movie from the conditional probability queries to the knowledge based models. Then, a list of movies matching the specified criteria is generated by browsing a movie database. The next step is to assess the probability of enjoying the whole experience of watching each movie on the list, by applying queries to the CMPR. Finally, the list of movies is ranked according to these results. In this schema, the

combination of the recommendation models is closer to the *cascade* hybrid technique, which requires that the output of one RS is employed by another recommendation method to refine the final recommendations [17].

The general flow diagram of the hybrid recommendation process is shown in Figure 5.12. In this HRS, the personalization aspect is included at the first stage of the recommendation process. Beyond the responses given to perform the inference query on the BN, the user is allowed to comment on the preferred and undesirable genres to include in the final recommendation, thus giving an extra fine tuning to the search of matching items. All the stages in the recommendation process are carried out within the user interface described in Section 6.1.

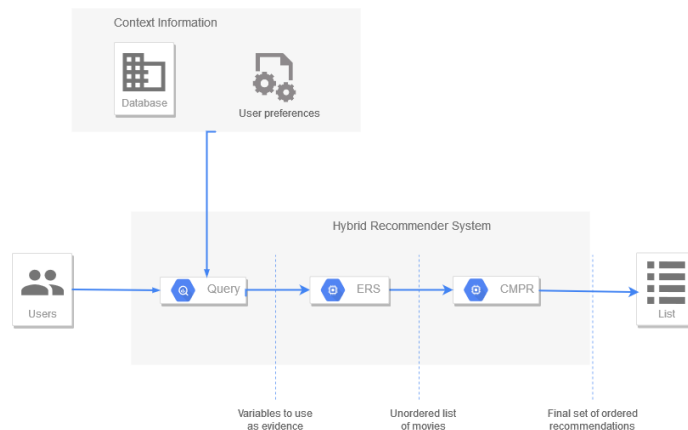


Figure 5.12: Cascade Hybrid Model for Personalized Recommendations

Chapter 6

Experiments and Results

The methodology proposed in Chapter 5 for developing a hybrid system capable of producing relevant recommendations for the user given certain context information is deployed and analyzed in the following sections. It is important to mention that this research aims to produce helpful and relevant recommendations that enhance the possibility of a great experience watching a movie, leaving behind the pursue of high algorithmic evaluation metrics. Thus, the experiments presented here follow a user-centric approach to asses user satisfaction, which is an indicator of the real value of the recommendations and how the whole process is perceived.

With this in mind, the user interface where the recommendations are generated is described in Section 6.1. Then the experiments that serve to test the relevance of the recommendations is detailed in Section 6.2; followed by the analysis of the findings in Section 6.3. Finally, a discussion on the method and the results is presented in Section 6.4.

6.1 User interface for testing the CHYBAM

For the test phase, it is necessary to implement a solution which facilitates the process of the recommendation. The relevance of a *good* design has been proven to be crucial in the perception of the recommendation [53, 108] and even can affect users opinions over the recommendations [28].

Considering the guidelines described in Section 3.5.2, the design of system must include as first step the collection of the necessary information about the user and the context, and then showing the recommended films. The most important consideration made for the design is that the information upon which the recommendation is made is explicit, i.e. the user must answer some specific questions regarding her preferences and particular context where the recommendation is taking place.

The solution was developed with programming language R using the `shiny` package [105], which combines graphical interface tools with all the benefits from the data analysis in R, and the possibility to publish the resulting environment in the web. This schema allowed for a relatively easy implementation of the main stages of the user experience, from collecting the information to showing within the same environment the set of recommendations.

It is relevant to mention that the set of movies that is available to recommend to the users is directly linked to The Movie Data base ^a (TMDb), with a total of 529,966 movies ^b, and increasing as more films are released. TMDb is a user based movie and TV data base, where all the information has been generated by users. The metadata for each item, either movie or TV show, includes release date, rating average, cast and crew members, production companies, genres, keywords, runtime, posters, among other variables. These attributes makes TMDb a perfect match for the HRS to obtain a set of movies that satisfies certain content characteristics. The ordering with which the movies are retrieved via an API can be set to follow the release date, popularity, revenue, or alphabetic, either in a ascending or descending fashion.

With these considerations, the application was developed, and had different versions throughout the process. The general aspects of the early versions are described in Section 6.1.1 and the final published version is defined in Section 6.1.2.

6.1.1 Early versions

The first prototype of the UI considered to collect the information for all the original questions defined by the experts, despite that most of them were not used in the following steps of the recommendation process. This led to a longer time required from the user, which may be translated in lower satisfaction level.

When developing the hybridization technique, the first phases of the method did not consider the agreement of the knowledge based models, and generated a set of movies which derived exclusively from one of these models. The top-3 movies from each set were presented to the user at the same time. One of the purposes of this strategy was to identify which model would satisfy better the users. An example of this experimental setup can be seen in Figure 6.1, which even shows an unexpected error in the first set of recommendations.

^a<https://www.themoviedb.org/>


^bas of the moment of writing this document

Escribe el nombre de la película:

Recomendación 1

Error: invalid argument type

- The General Died at Dawn
- To Have and Have Not
- Blood Alley



¿En dónde viste la película?

- Sin respuesta
- Sala de cine comercial
- Plataforma digital
- Sala de exhibición independiente
- Copia física (o TV)

¿Por qué decidiste ver la película?

- Sin respuesta
- Pasatiempo (o distracción)
- Quería ver esta película en específico
- Un amigo o familiar me la recomendó
- Tiene buenas críticas
- Forma parte de una exhibición especial
- Era la única disponible
- Sin motivo especial

¿Es la primera vez que te acercas a este formato/lugar?

- Sin respuesta
- Sí
- No

¿Es la primera vez que vez la película?

Recomendación 2

- Hells Angels on Wheels
- The Assassination Bureau
- Jungle Goddess




Figure 6.1: Beta version of the user interface

Other variables involved in the retrieval of the movies were evaluated. For example, as shown in Figure 6.1 the release date was bounded to contain movies from the origin of cinema (1895) until the day before the query was performed, with an ascending ordering relative to this attribute. This setting led to recommendations including mostly *classic* films. With a few test with real users, these strategies were dropped since the recommendations were not appealing to the users at all.

6.1.2 CineScope

CineScope is the name of the web application developed to produce recommendations using the hybrid method described in Section 5.10, and improving the UI based on the discoveries made with the first versions (see Section 6.1.1). The application is available at http://bit.ly/app_cinescope.

The recommendation process begins with the user answering a reduced set of questions, including only those which are truly indispensable for the inference with the models. These questions are based on the original questionnaire which served for the collection of the data set (see Section 5.2), but with slight modifications in the syntax so that they reflect the future possibility of watching a movie.

As seen in Figure 6.2, the left panel (1) is devoted to collect the user and context information that serves as input to the system. With these answers, the queries are performed on the models following the process described in Section 5.10, and a set containing the top-10 recommendations is presented to the user using both title and

poster for each movie in the main panel (2). The UI also collects the user evaluation of the recommendations (3), with a score in the range from 1 to 6.

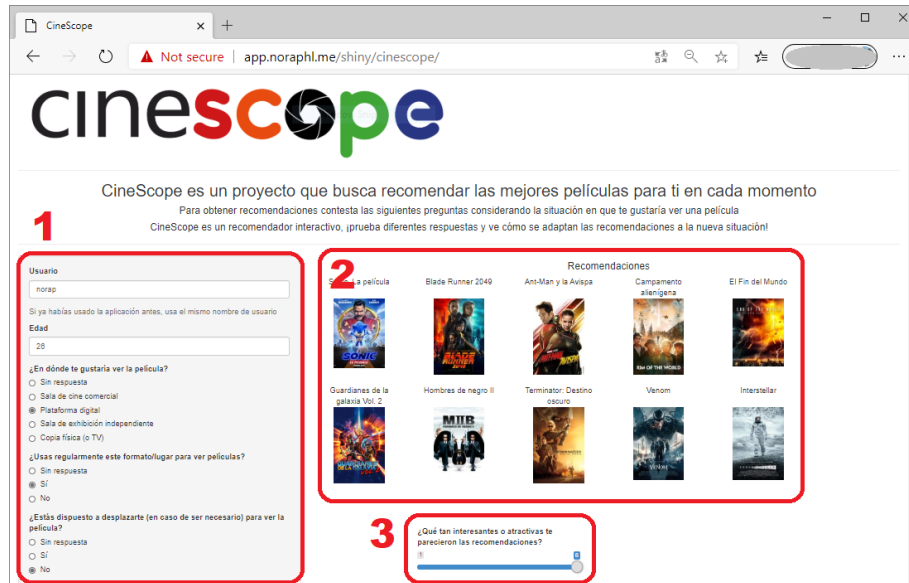


Figure 6.2: CineScope: User interface developed to test the CHYBAM

The retrieval process of the movies (from TMDb) is made considering two important aspects:

- The release date of the movies must be between 2000-01-01 and the day the recommendation is being produced; and the ordering is given by the popularity of the movies, thus popular movies come first.
- The set of genres the user prefers to watch W , and those genres which she prefers to avoid A are combined with the genres obtained from the inference process R , such that the set of movies include genres $(R \setminus A) \vee (W \setminus A)$. There is a maximum of two genres for each W and A in order to prevent incongruousness in the search process.

Concerning the last annotation, it must be clear that the coverage of the RS is not complete. There are genres which are unobserved in the data, and many others have little presence. This is a clear reflection of consumer habits with respect to movie going. Given the case that the input information produces a null probability distribution for the genre nodes, the models are not capable of producing a recommendation. In these cases, the output in the UI is compensated with movies corresponding only to genres $(W \setminus A)$. The

advantage of the CHYBAM is that regardless of the origin of the movie list, the CMPR will evaluate it according to the evidence, producing an ordering based on context.

For now, the recommendations presented to the user only include the best 10 evaluated movies, discarding possible relevant items that may appear lower on the ordered list. Since the current configuration of the RS does not provide the user with the opportunity to search more results from the same recommendation process, it would be advisable to evaluate this possibility in future studies.

6.2 Experiment setup

The purpose of user-centric evaluation to help develop recommenders that are not only helpful and accurate, but "also a pleasure to use" as McNee et al. [64] beautifully remark. Hence, it is not only important to evaluate system effectiveness, but also assess system usage as well as satisfaction or goodness of the recommendations. Specifically, we will test the main hypothesis of the research by measuring how attractive recommendations are as perceived by users, thus assessing in a general and simple way user satisfaction.

To find if any statistically significant differences exist between the strategy defined by the CHYBAM (see Section 5.10) and collaborative filtering techniques, an experiment where the users were exposed to two different RSs was designed to compare their experiences. By applying a sort of A/B test in a within-subjects experiment, CineScope was compared against MovieLens, with "The warrior" as recommender selection (see Section 4.2). For a more detailed description see Appendix A.2.

The procedure to which the participants were subject consists on interacting with one RS, to then answer a set of questions regarding the experience; after that, the same process must be completed with the other system. Half of the participants were exposed first to CineScope and then to MovieLens, while the other half completed the experiment in reverse order.

We measure the main variable with a rating, in the range 1 to 6, in which users mark how attractive they find the set of movies recommended by each system; in this way, a higher rating indicates users find the presented movies more appealing. Along with the main variable, some other variables were also considered in the study:

- **Attractiveness.** The perceived goodness of the recommendation. It directly answers the question *how appealing did you find the recommendations?*
- **Easiness.** The overall effort required to navigate through the RS to obtain recommendations. Comes from answering the question *how easy did you find to use the application?*

- **Promoters.** Directly responds to the question *how likely would you recommend the application?*

The feedback survey also asks for some opinions regarding the strengths, opportunities and weakness of the systems. A detailed description is available in Appendix A.2. Another couple of questions involve a qualitative rating scale requiring more active engagement of the user with her recommendations. Specifically, these questions are intended to identify how likely is the user to actually watch (some of) the movies, and how the recommendations relate to user's experience, considering if they recognize some, all or none of the movies and how well they fit her taste.

There are some questions that are specific to the experience with CineScope, and are designed to collect information regarding the perceived sensibility and appreciation of the context. First, it is considered necessary to identify if the user noticed any change in the recommendations after modifying some of the variables, and then give a qualitative measure of how dramatic the change was. Also, participants are asked to directly assess how relevant they consider to include context information into the recommendation.

As a closure process, a final survey requires the users to indicate which RS they will use if they had to choose only one, and specify the reason why. Despite this being not a definitive assessment of the relevance of one RS over another, it will help to identify which aspects of the systems are most valued by the users.

The target audience for the system is the same from which the data were collected. Hence, the population for the tests is the college community; and the sample was obtained from selected courses. A pilot study was performed to test the procedure and verify whether it was fully understood for proper execution. Also, the results obtained were used to determine the minimum sample size required to detect significant differences between the observed variables, as well as to obtain a rough estimation the expected effect size. The results of the pilot study and assumptions taken for this power analysis are shown in Table 6.1. This raw estimations are made considering exclusively the perceived attractiveness since it is the variable directly associated with the hypothesis of the research. The data was processed in R using package `Superpower` [20].

Measure	Pilot	Expected
Sample size	18	90
Mean CS	5.22	5.2
Mean ML	4.89	4.8
SD	1.15	1.1
Cor	0.12	0.1
Effect size	0.22	0.31
Power	13.95	≥ 70

Table 6.1: Pilot study results and expected measures

From this design analysis, a sample with 90 subjects will be enough to detect small effects ($d = 0.31$) for the desired variable. Even though the correlation between the two levels (CS for CineScope, ML for MovieLens) is rather small in the pilot ($cor = 0.1$), the power analysis implies that the study will be able to find significant results with 70% probability using samples of size 90 (assuming Type I error rate equals 0.05).

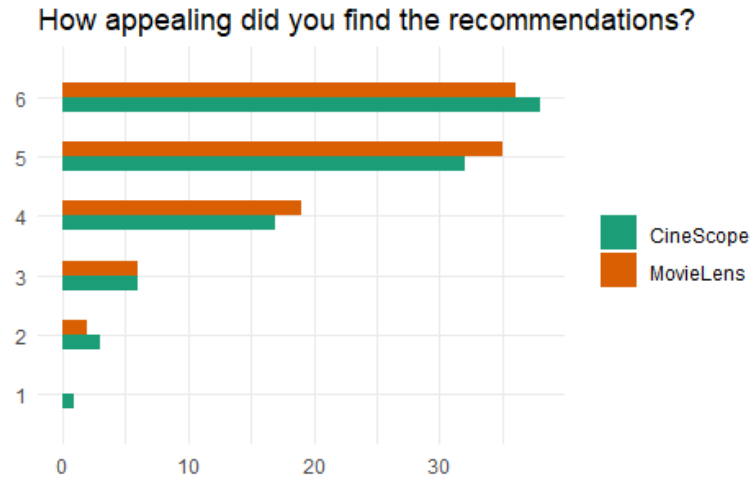
Now we can fix the sample size to determine the smallest effect that the study is well powered to detect. By using a two-tailed t -test with paired samples, it is possible to calculate the smallest effect size of interest to be $d = 0.2647$. This value will be used to test for significance when analyzing the results.

6.3 Results

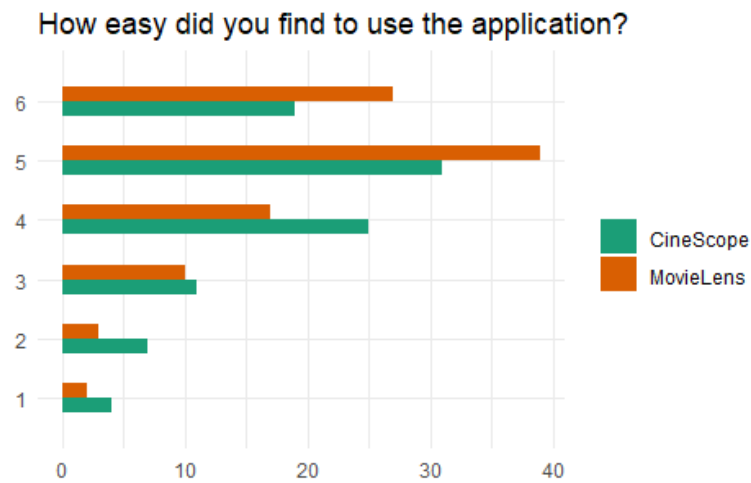
With the appropriate design considerations from Section 6.2, 195 entries were obtained from a total of 98 participants (one subject did not completed the process and only 97 entries are registered for CineScope). In total, 12 variables were measured; three quantitative, along with 9 qualitative. The first three are attractiveness, easiness, and promoters (see Section 6.2). The complete results are available at https://bit.ly/cinescope_data.

Figure 6.3 shows the results for each of these variables. In general, users evaluated both RSs very similar. By inspection, data show a slightly negative skew in the distribution of the three measures, which implies that in general the systems are rated positively. Whereas there are observable gaps between the different possible scores depending on the system evaluated, the overall behaviour reflected in the data is the very similar. Considering for instance the perceived ease of use of each for the applications in Figure 6.3b, whereas the difference between the scores is relatively large, the means only differ by 0.3 points. The largest value-to-value difference is registered in the promoters

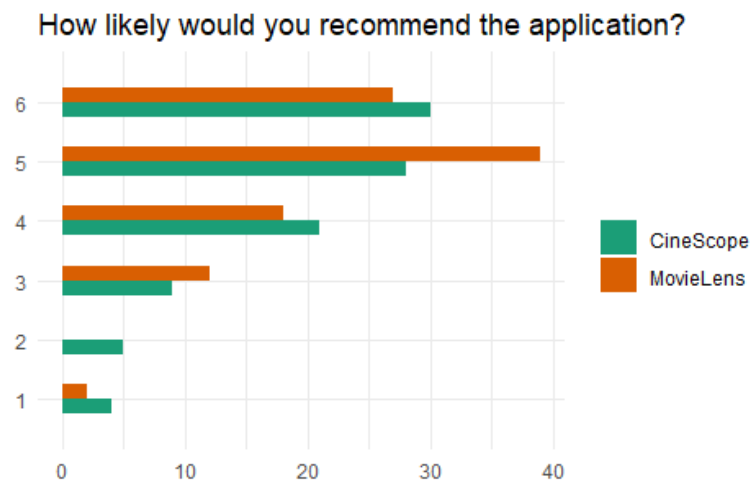
(*score* = 5), where almost one fourth of the users differ in opinion; nevertheless, the means only differ by 0.1.



(a) Attractiveness



(b) Easiness



(c) Promoters

Figure 6.3: Comparison between CineScope and MovieLens results for scalar variables

On the other hand, when comparing the self-assessed probability of watching any of the movies suggested by the recommenders, users consider almost twice as likely to watch at least some. It's interesting to discover that almost one third of the recommendations produced items already familiar to the user, and only one tenth of them did not find any interesting suggestions when trying CineScope. Figure 6.4 shows the compared results for these estimations.

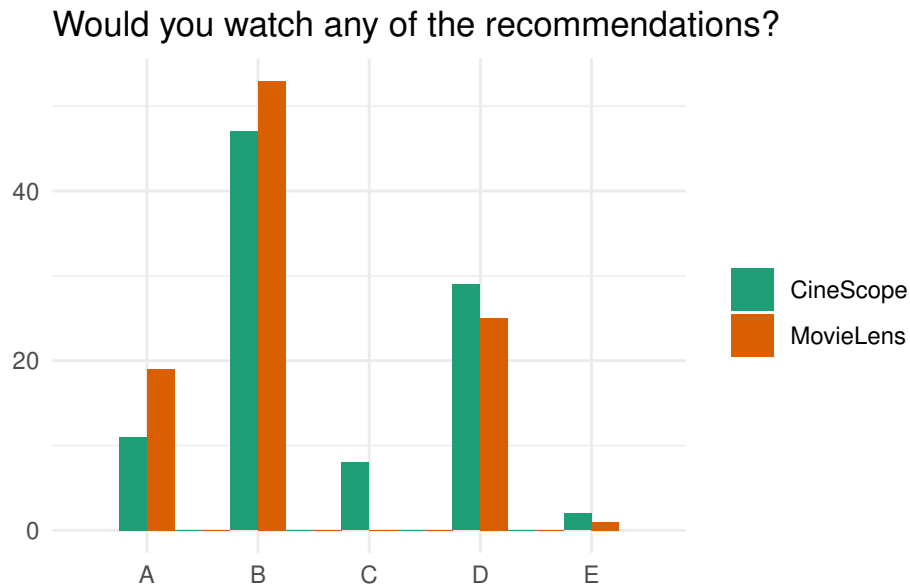


Figure 6.4: *How motivated* are users to watch at least some of the recommendations?
 LEVELS. **A:** Definitely want to watch them all, **B:** I would watch a few, **C:** I wouldn't watch any, **D:** I've watched some of them, but could watch them again, **E:** I've watched them all, and wouldn't do it again

Users were also asked to value the goodness of the recommendations by classifying the whole set according to subjective estimation of how well the recommendations match their expectations. The scale used can be understood as a modified Likert scale, which inherently implies ordering among the different values. Figure 6.5 shows how different the results were between the two RS. For evaluating MovieLens, users were presented with four different options, and in the end only three were relevant, with an absolute majority showing good appreciation for the movies recommended.

Equivalently, for the evaluation of CineScope users were given the same options as for MovieLens, but with special attention to the presence of context. The results show that more than one third of the recommendations were considered to match the expectations

for the specific situation selected by the user. More or less the same proportion of the users valued the recommendations to be at least somewhat similar to their taste, leaving only less than one fifth of the total users unsatisfied. For more specific information on this question, or any other in the survey, see Appendix A.2.

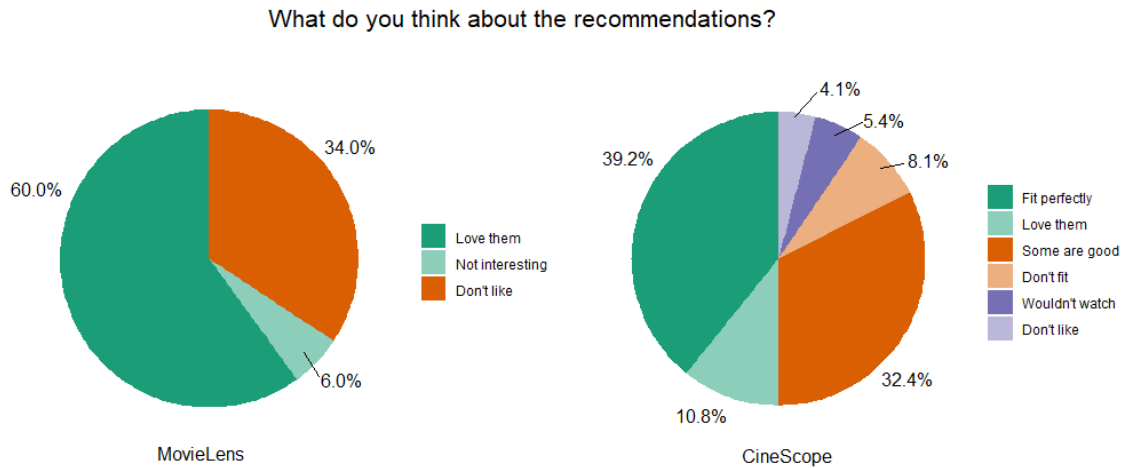


Figure 6.5: Evaluation of *goodness* of the recommendations

Now considering specifically the influence of the context in the recommendation after users were explicitly asked to test this feature in CineScope, the level of perceived change was specified in a qualitative scale. The results are mixed; nevertheless, a third of the users experienced significant change in their recommendations after the context update, and half of them noticed at least some changes. Figure 6.6 shows the summary of the responses.

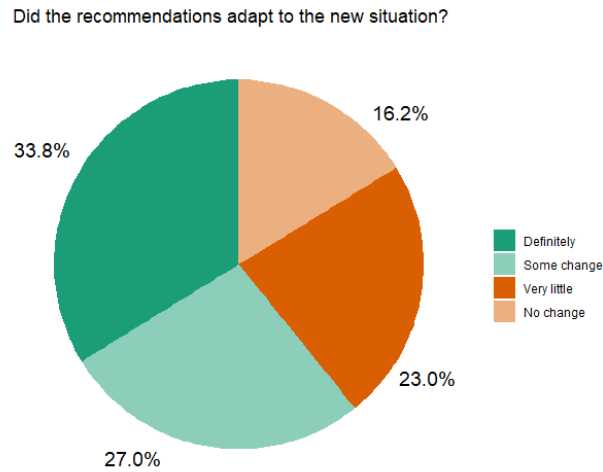


Figure 6.6: Perception of the influence of the context

Further into the context evaluation, but indirectly linked to the recommendations in CineScope, the overall relevance of a context aware system was assessed by the users. More than two thirds consider the context is rather important for providing a *good* recommendation, while only less than three percent of the users are not at least interested in the idea. Figure 6.7 presents the subjective evaluation of the relevance of the context.

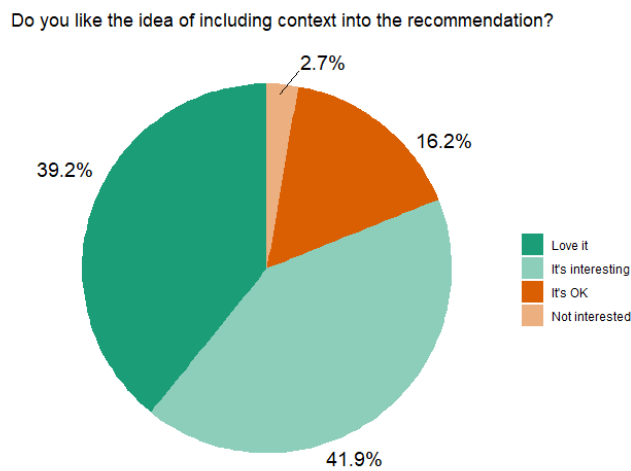


Figure 6.7: Overall perceived relevance of a CARS

Finally, users were forced into the hypothetical situation of choosing only one RS and specify the reason why. With a total of 60 answers, there is an undeniable preference for MovieLens. Table 6.2 presents a detailed breakdown of the RS selected and the main reason that motivated the choice. There are some unexpected results specifically regarding the key feature in which each RS relies its recommendations on. "Context aware recommendations" was selected two times as reason to choose MovieLens; the same happened with "Rating System", as it was appointed as the main reason to choose CineScope, again by two users. Beyond these cases, the results favor the assumptions over the RSs, and offer an interesting path to analyse the influence of one, or several aspects over the user experience.

Motivation	CineScope	MovieLens
Ease of use	2	0
UI	0	1
Variety	1	8
Personalization by genre	3	3
Context aware recommendations	9	2
New content	1	11
Familiar content	3	4
Rating system	2	9
Unspecified	1	0
Total	22	38

Table 6.2: Comparison of RS choice

6.3.1 Statistical analysis

There are several tools designed to determine the statistical significance of the observed data. Simple descriptive statistics are shown in Table 6.3. The small raw differences in the mean for the three variables reflect quite small effect sizes (Cohen's d), from which both *attractiveness* and *promoters'* effects can be neglected, $d = 0.03$ and $d = -0.12$ respectively, and the only small effect that is present in *easiness* with $d = 0.27$.

The standard deviation (SD) of the distributions are larger for the evaluation of CineScope, but not large enough to consider abnormal behaviour. Overall, the values registered are close to 1, and the largest value-to-value difference is registered in *promoters*, where the difference in SD is 0.35. By comparing the correlation between the raw measurements, the variables are not highly correlated.

The results shown in Table 6.3 are calculated using the raw observations. It is possible to also compare the data using only paired observations, where $n = 59$. When doing so, the variables are highly correlated ($\rho > 0.5$) within the same level, i.e. comparing the relation between *attractiveness* and *easiness* of CineScope results in $\rho = 0.5686$, and $\rho = 0.6280$ for MovieLens.

		CineScope	MovieLens	Effect size	ρ_{Pearson}	ρ_{Spearman}
Subjects		97	98	-	-	-
Attractiveness	Mean	4.9661	4.9322	0.0316 ± 0.3647	0.1193	0.0941
	SD	1.1592	0.9802			
Easiness	Mean	4.3051	4.6610	-0.2740 ± 0.3664	0.2284	0.2479
	SD	1.3928	1.1978			
Promoters	Mean	4.5085	4.6441	-0.1070 ± 0.3649	0.4543	0.3624
	SD	1.4309	1.0790			

Table 6.3: Summary statistics for the tests comparing the perceived attributes of the RSs

A proper analysis for the target variable, (*attractiveness*), must consider the uneven observations for the different levels, which imply different variances. Also, it is important to establish the expected smallest effect size of interest (SESOI) to determine an appropriate interval where the hypothesis of the study could be falsified.

With the proper definitions made in Section 6.2, $d = 0.2647$, $\beta = 0.3$, $\alpha = 0.05$, and the results from Table 6.3, an equivalence test is performed in order to determine if the observed effect is surprisingly small assuming there is a true effect at least as extreme as the SESOI [57]. The procedure requires two one-sided tests (TOST), one for testing if the effect is smaller than the SESOI, and the other for testing if it is larger, and is executed in R with the TOSTER package [56].

A graphical representation of the results is shown in Figure 6.8, where the 90% confidence interval (CI) is identified with a thick line, and the thin lines are the 95% CI. The equivalence test was non-significant, $t(189.85) = 1.644$, $p = 0.0509$, given equivalence bounds of -0.173 and 0.173 (on a raw scale) and $\alpha = 0.05$. Also the null hypothesis test was non-significant, $t(189.85) = -0.203$, $p = 0.839$, given $\alpha = 0.05$. Based on the equivalence test and the null-hypothesis test combined, it is possible to conclude that the observed effect is statistically not different from zero and statistically not equivalent to zero. In other words, it is not possible to conclude that the mean attractiveness of the systems are statistically equivalent, nor that the difference between

mean attractiveness is different from zero, given the expected effect $d = 0.2647$ and $\alpha = 0.5$.

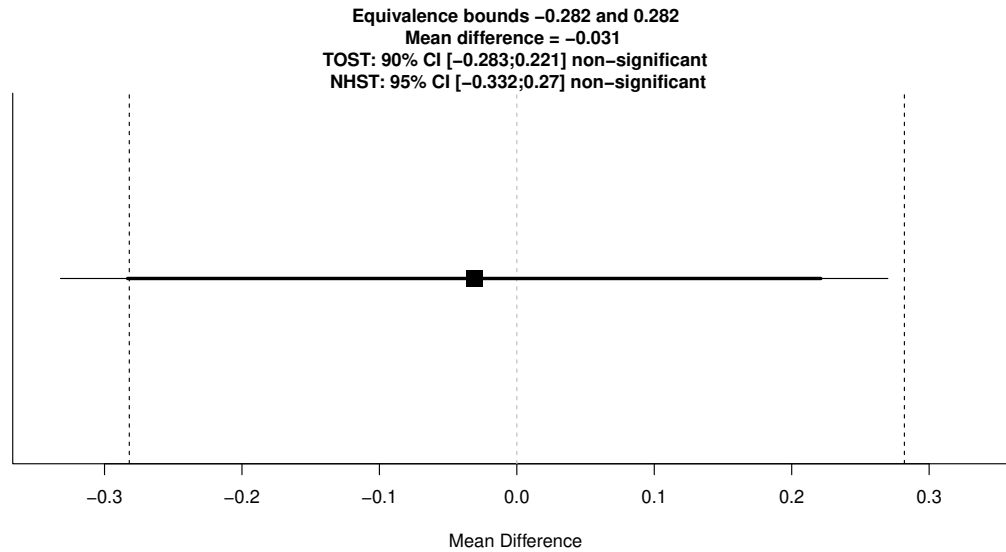


Figure 6.8: Equivalence test for the observed effect in *attractiveness*

Results are no different if the analysis is made within the reduced set of paired observations. Both the equivalence test ($t(58) = -0.975$, $p = 0.167$) and the null hypothesis test ($t(58) = 0.180$, $p = 0.858$) are non-significant given $d_z = 0.1503$, $\alpha = 0.05$ and $\beta = 0.67$ in the interval $[-0.217, 0.217]$, which only confirms the results obtained before.

6.4 Discussion

Before a comprehensive discussion of the obtained results is given, it is important to begin making clear that the tests were applied with direct personal interaction developer-user, which may have or have not influenced the responses. The clear tendency towards a positive review of both RS might be a slight indicator that there is an unmeasured effect definitely worth studying with a larger sample, and a broader scope. As a matter of fact, it is well recognized in the field that the effectiveness of a RS resides in many factors besides the recommendation algorithm [96].

Nevertheless, data show that users do not find any significant distinction in the recommendations they were provided, neither the perceived goodness nor the user

experience within the application were determinant factors to declare better one RS over the other. Despite this finding can be a little discouraging, it can be understood as a potential strength of the proposed new method.

When asked for a more personal opinion on the whole process of the recommendation with CineScope, users provide consistently satisfaction answers that matches the stated benefits of the approach made with the CHYBAM. These answers include aspects such as the easy and intuitive process, the possibility of further personalization by genre, and also the precise match of recommended movies with user's taste. More importantly, users recognize the benefit of providing a recommendation based on the specific situation that accompanies the experience of watching a movie, even before they were asked to comment on this specific subject.

The feedback provided by the users also include the aspects that should be improved for a better recommendation. The majority of these comments referred to the UI, which can definitely be enhanced with more interactive, and/or reactive tools. It is important to consider this as a priority if an improvement is to be made, since more than 50% of the success of a RS can be accounted to its design and interactive components, and only 5% to accounts for the algorithm [63].

With the version of algorithm implemented in the application, it is possible that some combinations at the input do not produce a proper recommendation, but instead display an error message to the user. This is directly related to the available information in the data with which the models were fitted, and is further examined in Section 7.2. Other factors that were mentioned as important to improve is the set of input questions, where users do not agree if it is too large or too small, and some users even ask for more personalization stages.

Concerning the set of recommended movies, users find them equally appealing coming from CineScope or from MovieLens. From the equivalence test on the paired observations, it is possible to deduce that there is a probability ($p = 0.167$) that a true effect outside of the interval $[-0.217, 0.217]$ exists, and this tests may not have sufficient statistical power to detect it [102], or were not well designed in the first place; after all, $power = 33.33\%$. The skewness in the data could also be severely influencing the results, however it would be advisable to replicate the study with a larger sample size.

Since the inclusion of the context as a key factor in the recommendation is one of the main objectives of the research, its effect was measured in three different variables. When assessing the overall goodness of the recommendations provided by CineScope, users were given the option to evaluate them as perfect for the context situation they chose, with which 39.2% of the users agreed. This outcome implies that, first of all, users are sensitive to which movies fit better which situation, and second and most important, the system is able to capture this sensitive insights. This feature is clearly one of the main advantages of the

system, and can be exploited to obtain recommendations that are both relevant and useful for the users.

In this regard, users have shown at least some interest in engaging with the recommendations, and the more than 90% of them would like to watch some (if not all) of the movies. The level of enthusiasm is proportional between the two RS, and although numbers benefit a little the recommendations presented in MovieLens, there is no generalized apathy for the movies recommended by CineScope. This sustains the idea that the recommendations on both systems are comparable to the user's eyes.

Defying the authentic satisfaction shown with respect to the content in CineScope, and the promising results described in previous paragraphs, when asked directly users prefer mostly MovieLens. This could be related to the customary design of the application, which allows to browse movies and rate them, that contrasts with the minimal design implemented for CineScope. Or could be also associated with the number of movies recommended, since CineScope only shows the top-10 fittest movies. It could be also that this type of recommendations based on context are not common, or at least not visible to the user, which rises a some kind of rejection to the system. Whichever it is, the results presented here should encourage further evaluation of the recommendation strategies implemented with the CHYBAM.

Chapter 7

Conclusion and Future Work

After the Contextual Hybrid Bayesian Model (CHYBAM) has been developed, implemented and tested according to the objective of the research, it is necessary now to make a comprehensive evaluation of the process and the results. Section 7.1 highlights the most relevant findings, and at the same time makes appropriate remarks on the strengths and opportunities of the method. In Section 7.2 a more detailed analysis is made on the conditioning factors that influenced the research. And finally, Section 7.3 explores the possibilities of continuing the work done so far.

7.1 Conclusion

The hybrid implementation of a knowledge-based system has been proven to be a relevant advance in the field of recommendation strategies. The Contextual Hybrid Bayesian Model (CHYBAM) combines the strong capability of a Bayesian network to handle uncertainty, with the benefits of environmental information to offer a fully context-based recommendation.

The knowledge provided by the experts shows to be one of the key features of the technique. It provides the solid foundation from which the complete set of models were constructed to produce recommendations. The intuitive nature of the Bayesian networks is a great advantage for including expert knowledge directly into the system. It allows to fully represent the influence of some variable over another, and the mutual dependencies that reflect the complex nature of any human task.

With respect to the performance of the CHYBAM and its implementation in CineScope, it has been demonstrated that the recommendations provided satisfy users' needs, at least as good as other well known recommenders. It is important to remark that beyond the traditional metrics and evaluation techniques, real user satisfaction has been

proven to be a relevant aspect when designing a RS [81]. Satisfaction is rather a subjective topic; however, the relevance of the recommendations was assessed directly by the users.

The context is also a key feature of the CHYBAM, and common RS do not offer explicit personalization by the environmental factors. Despite users are little, or not at all, conscious of the relevance of these factors, more than 80% consider at least somewhat interesting the idea of including context into the recommendations. This should not be disregarded, as surely will take more relevance in the near future [85, 23].

Considering the notable preference for MovieLens over CineScope shown in the tests, it would be imprecise to declare that MovieLens is a better RS just because it was selected almost twice as often over CineScope. There are several factors influencing the choice, most of which were discussed in this document, and many others that remain open for evaluation. Most of the developments made with this research are based on informed assumptions, and given that there are still no comprehensive study that could guide a line of research as the one presented here, it is reasonable to conclude the exploratory nature of this study.

Even though the process for obtaining the CHYBAM has solid foundations and was proven to be a promising source of recommendations, it is important to stress that the data from which it was originated is limited. Definitely including more observations will be necessary to enhance the model, and validate the observed results.

As a general overview of the advantages of the method, it should be stated that the approach proposed by the CHYBAM is not domain specific. It can be easily extrapolated to other domains with some knowledge engineering to obtain the basic model from the experts, and then elaborate the following steps as described in this document. Furthermore, in contrast to the traditional user model for RSs, the CHYBAM relies on a user model based on expert knowledge, which is able to update as more data are available, and independent from any other variable, such as ratings or previous interactions with the system. Hence, it does not suffer from the cold-start problem. Neither the ramp-up problem is an issue, due to the capacity of the system to consider only items' attributes relevant in the context of the recommendation. For these reasons, the CHYBAM can be applied as a solid recommendation strategy in any field, given the relevant expert knowledge.

7.2 Limitations

Regardless of some favorable results discussed in Section 6.4, it is possible to identify some aspects in the process which certainly restrict the outcome of the research. These are

mainly related to the available data, and the source from which the movies are retrieved to produce the recommendations.

Given the reduced number of data with which the models were trained, and the direct relation with the set of input questions, there exists the chance that some possible, but rather unlikely (not observed in the data set) scenarios can be specified by the user. In such situations the BNs will not have sufficient information to validate the evidence, and no recommendation will be produced.

Those considerations lead to determine the practical limitations of the proposed recommendation process. It is clear that the recommendations are restricted by the available data set; as a result, the output is a general approximation of the preferences represented in those observations. This bias is intrinsically related to the sample where the data was collected, which primarily is composed by college students with ages in the range 18 to 21 from the upper and upper-middle class in the metropolitan area of Mexico City.

Therefore, whereas the recommendation is highly context responsive, it is possible that the outcome of a specific query does not match the user's personal preferences completely. The user interface also may condition the interaction, a more appealing and intuitive design could benefit the process as a whole. Also, considering that the final set of movies is retrieved in descending order with respect to the popularity in TMDb, there is an intrinsic bias that satisfies most, but not all the users. Those looking for art-house cinema, or a more curated list of films, may be disappointed with the recommendations offered.

All these factors contribute to highlight the importance of the data set to construct a more general model. The proposed RS aims to develop a general user model by collecting the maximum number of possible situations, that allows the model to be truly based on (user) knowledge [97], and hence produce a suitable and more personalized recommendation for most, if not all, of the scenarios.

7.3 Future Work

The various advantages described along this document may serve to further study the proposed technique. There are several aspects which can benefit from a conscious revision and improved implementation. First of all, given that no statistical significance was obtained from the tests, it would be advisable to exploit the advantages of the CHYBAM, and propose a study to test with a larger sample size to clarify the assumptions made.

The overall user experience is an important aspect that deserves further

improvements. By designing a more compelling and intuitive user interface, the general sense of a reliable and robust system can be conveyed. The UI can also be enhanced by offering more information about the recommended movies, such as a short plot summary, trailer videos, and other relevant information such as cast and crew, or even information on where to watch each movie. Also, by including explanations on the recommendations made, the confidence on the system may increase.

Furthermore, if the system is applied in a fully working environment, a mobile app for example, would allow to automatically collect most of the context variables required in the inference process. In that scenario, the information that needs to be explicitly specified by the user would decrease, and the overall usability may improve. As a combined effort, it would be advisable to restrict the recommendations to fully satisfy the requirements of the user with respect to the *place* selected; for instance, given that the user prefers to go to the movie theater, displaying only movies that are currently available at the local cinemas would increase both confidence and usability.

In this regard, a medium to large scale implementation would require a larger set of observations to train the model with, that comes from a more varied audience. Additionally, a broader system evaluation will need to include a benchmark comparison against literature state-of-the-art recommendation systems that allows to find the main differences between the various approaches, both in the technical implementation and as perceived by the user. This type of evaluation would allow a more extensive understanding of the implications of a recommendation system as a whole, that will serve to take CineScope out of the college environment.

Another consideration to be made is the level of personalization. The current implementation in CineScope relies on the user specification of the desired and undesired genres to offer real sense of individual adaptation. This stage in the recommendation process can be enhanced by including more filtering options based on the attributes of the movie (plot, cast, crew, etc.). The best procedure would retrieve all this information automatically based on a user profile, which can be easily implemented via the API offered by TMDb. With this process, it would be even possible to include those attributes as part of the recommendation variables, and not only as a post-filtering method.

It would be also important to improve the coverage of the recommender. This would imply that not only the most popular movies are recommended, but also offer the possibility to include more art-house films, or some other relevant films for the user. The limitations in this regard are clear, there are more than half a million movies registered in TMDb, more than one can possible watch in a life time. The resources needed to evaluate all of those movies for each query would be enormous, and the whole process, pointless. Instead, a more clever solution would be to add certain filters; and if at any point a successful implementation of users' profile is available, the combination of this feature

would be more natural. With the current implementation of CineScope, it would be possible to improve the coverage by allowing the user to explore more movies than just the top-N from the same recommendation query, or even update the list of recommendations by evaluating the score assigned to the complete set (see Figure 6.2-(3)).

The possibilities of implementing a BN with the complete set of variables were discarded in benefit of a hybrid implementation. Nonetheless, the models obtained with such characteristics could offer a comprehensive representation of the relationships between all the variables describing the user, the context and the movie attributes. The advances in this line of research could yield good results opening the possibility for other recommendation techniques, and even hybrid implementations with other approaches, such as content-based or social recommendations.

All of these improvements combined would not only increase the performance of the system, but would help increase the adoption of the system in a real life scenario.

Appendix A

Questionnaires

A.1 Questionnaire for initial data collection

Loose translation of the original questionnaire in Spanish.

For all the rating scales, 1 is the lowest and 5 is the highest score.

Link to the original form: http://bit.ly/mas_peliculas

1. Movie title
2. Where did you choose to watch the movie?
 - Commercial movie theater
 - Non-commercial movie theater
 - Streaming service
 - Physical copy
 - Other
3. Is the first time you watch movies this way?
 - Yes/No
4. What made you choose this option?
 - It's my favorite place
 - It was my only choice
 - I usually watch movies here
5. Is this the first time you watch the movie?
 - Yes/No
6. Why did you decided to watch this movie?
 - Hobby/Distractation

- I wanted to watch this movie specifically
 - A friend or relative recommended it
 - It is a special exhibition
 - It was the only one available
 - No special reason
 - Other
7. Where did you watch the movie? (city)
8. Where did you buy the tickets? (Specific for Commercial movie theater)
- Box office
 - Automatic box office
 - Web page
 - Mobile app
9. How would you rate your experience for buying tickets? (1 to 5) (Specific for Commercial movie theater)
10. Movie theater company (Specific for Commercial movie theater)
- Cinopolis
 - Cinemex
 - Other
11. What made you choose this format? (Specific for Streaming service)
- The movie was not been exhibited
 - Schedule
 - Only format available
 - Convenience
12. Where did you watch the movie? (Specific for Streaming service)
- At home
 - At a friend's/relative's house
 - Transportation
 - Other
13. What do you like the most of this format? (Specific for Streaming service)
- Variety
 - Portability
 - No restrictions
14. Choose your streaming service: (Specific for Streaming service)
15. Did you have to pay to watch the movie? (Specific for Streaming service)
- Yes
 - It was included with my subscription
 - Free access
 - No

16. Why did you choose this format? (Specific for Non-commercial movie theater)
- It is the only way to watch the movie
 - Low entrance cost
 - I'm a regular at the exhibition place
 - Special screening
17. How did you find out about the screening? (Specific for Non-commercial movie theater)
- Somebody invited me
 - I follow the schedule
 - It was part of a festival or similar
 - I was at the screening place
18. What do you consider before watching a movie? (multiple choice)
- Plot
 - Cast
 - Director
 - Studio
 - Reviews
 - Recommendations
 - Awards and nominations
 - Genre
 - Special effects
 - Runtime
19. In what language did you watch the movie?
- Original without subs
 - Original with subs
 - Dubbed
20. How relevant is to choose the language? (1 to 5)
21. Rate the movie (1 to 5)
22. What did you enjoy the most about the movie? (multiple choice)
- Plot
 - Screenplay
 - Narrative
 - Ending
 - Characters
 - Cast
 - Acting
 - Directing
 - Runtime
 - Special effects
 - Makeup and Hairstyling
 - Original Soundtrack

- Originality
- Editing

23. On average, how many movies do you watch?

- More than one a day
- 1 every day
- 2 or 3 by week
- 1 by week
- 2 or 3 by month
- 1 by month
- 2 or 3 by year
- 1 or less by year

24. When did you watch the movie?

- Premier day
- Today
- Yesterday
- The day before yesterday
- This week
- Last week
- In the previous two weeks
- This month
- Last month
- More than a month ago

25. How important was to watch the movie on that date? (1 to 5)

26. At what time did you watch the movie?

- Before non
- After noon
- Evening
- Late at night

27. How much did you like that time? (1 to 5)

28. With whom did you watch the movie with?

- Alone
- Significant other
- Friends
- Family

29. How much did you enjoy the companion? (1 to 5)

30. Who chose the movie?

- Me
- Companion
- Everyone

31. Did you consume any snacks?

- Yes/No
32. What do you enjoy the most about the experience?
- Time of day
 - Day of week
 - Day off
 - Season of the year
 - Place
 - Weather
 - Companion
 - Snacks
 - Available time
33. How likely is that you will repeat the same experience? (1 to 5)
34. Where do you buy the snacks? (Only if consumed snacks)
- Where I watched the movie
 - From home
 - Local store
35. How important is to have snacks for the movie? (1 to 5) (Only if consumed snacks)
36. How much time did you have available to watch the movie?
- More than plenty
 - Enough to watch this movie
 - Enough to watch any movie
37. Which type of day did you watch the movie?
- Weekday
 - Weekend
 - Holiday
38. How did the available time motivate you to watch the movie? (1 to 5)
39. Did you have to travel to watch the movie? To the local cinema for example
- Yes/No
40. How much time did you have to travel? (only if had to travel)
- Less than 10 minutes
 - Between 10 and 20 minutes
 - Between 20 and 30 minutes
 - More than 30 minutes
41. How did it influence your motivation to watch the movie? (1 to 5) (only if had to travel)
42. Do you usually travel to watch the movie?

- Yes/No

43. What transportation mean did you use? (only if had to travel)

- By foot
- Bicycle
- Car
- Public transportation

44. How much did it influenced your motivation to watch the movie? (1 to 5) (only if had to travel)

45. Age

46. Education level

47. Current job

48. Gender

A.2 User evaluation

A.2.1 Experimental procedure

The procedure design to test the hypothesis is outlined in Table A.1. The users directly interacted with flyers containing the same information to guide them during the process. Those are available at https://bit.ly/Flyers_test.

CineScope	MovieLens
<ol style="list-style-type: none"> 1. Go to http://bit.ly/app_cinescope 2. Introduce your unique user name 3. Answer the required set of questions, and then click on "Obtain recommendations" 4. Check out the recommended movies 5. Modify your answers and verify that the movies adapt to the new situation 6. When you are satisfied, click on "Send responses" 7. Answer the feedback survey available at http://bit.ly/opinion_cs 	<ol style="list-style-type: none"> 1. Go to http://movielens.org/ 2. Create a new account by clicking on "Sign up now". Use your unique user name. 3. Follow the instructions that guide you through the system. <ol style="list-style-type: none"> (a) First, select your favorite categories from the options by assigning a total of three points (b) Start rating movies by selecting the stars you consider appropriate. Either good or bad ratings will affect your recommendations 4. Whenever you have rated at least 15 movies, be sure you have selected "The warrior" as the recommender 5. Explore and personalize your recommendations to adjust them for popularity. 6. Answer the feedback survey available at http://bit.ly/opinion_ml
<p>After completing both stages proceed to the final feedback survey available at http://bit.ly/opinion_final</p>	

Table A.1: Experimental process defined for the study

A.2.2 Questionnaire for evaluating CineScope

Loose translation of the original questionnaire in Spanish.

For all the rating scales, 1 is the lowest and 6 is the highest score.

Link to the original form: http://bit.ly/opinion_cs

1. How easy was to obtain recommendations? (1 to 6)

2. How appealing do you find your recommendations? (1 to 6)
3. What did you like the most?
4. What could definitely improve?
5. What must change to improve your experience?
6. Did the recommendations adapted to new situations after you modified your answers?
 - Yes/No
7. How well did the recommendations adapt?
 - Definitely adapted to the new situation
 - Not completely, just some changed
 - Very little
 - The same movies, but different order
8. How likely is that you watch any of the recommended movies?
 - Definitely want to watch them all
 - I want to watch some of them
 - I have seen some of them, but could watch them again
 - I have seen all of them, and I don't want to watch them again
 - I don't want to watch any
9. Complete the phrase: "All the movies recommended by Cinescope..."
 - ... are movies I would definitely watch, or have watched, and I love them
 - ... are movies I would definitely watch, or have watched, and fit perfectly with that context situation
 - ... are movies I would watch, or have watched, but I don't like all of them
 - ... are movies I would watch, or have watched, and but I wouldn't choose them for that context situation
 - ... are movies that I don't want to watch because I don't know them, or are not appealing to me
 - ... are movies I would never watch because I don't like them
10. How important do you consider to include the context into the recommendations?
 - I love it, in this way I can see recommendations for specific situations
 - It is interesting, I allows me explore better options
 - It's ok, but maybe it is not necessary
 - Not interested, I always know which movie I want to watch
11. How likely is that you recommend CineScope to a friend or relative? (1 to 6)

A.2.3 Questionnaire for evaluating MovieLens

Loose translation of the original questionnaire in Spanish.

For all the rating scales, 1 is the lowest and 6 is the highest score.

Link to the original form: http://bit.ly/opinion_ml

1. How easy was to obtain recommendations? (1 to 6)
2. How appealing do you find your recommendations? (1 to 6)
3. What did you like the most?
4. What could definitely improve?
5. What must change to improve your experience?
6. How likely is that you watch any of the recommended movies?
 - Definitely want to watch them all
 - I want to watch some of them
 - I have seen some of them, but could watch them again
 - I have seen all of them, and I don't want to watch them again
 - I don't want to watch any
7. Complete the phrase: "All the movies recommended by Cinescope..."
 - ... are movies I would definitely watch, or have watched, and I love them
 - ... are movies I would watch, or have watched, but I don't like all of them
 - ... are movies that I don't want to watch because I don't know them, or are not appealing to me
 - ... are movies I would never watch because I don't like them
8. How likely is that you recommend CineScope to a friend or relative? (1 to 6)

A.2.4 Questionnaire for comparing CineScope and MovieLens

Loose translation of the original questionnaire in Spanish.

For all the rating scales, 1 is the lowest and 6 is the highest score.

Link to the original form: http://bit.ly/opinion_final

1. If you could only choose one app, which one would it be? (randomized order)
 - MovieLens
 - CineScope
2. What made you choose? (randomized order)
 - I like to choose genres or group of movies
 - I like the variety among the recommended items
 - I like it recommended movies I like
 - I like to receive specific recommendations for the specific situation I choose
 - I like that it recommends me new movies, or movies I haven't watched yet
 - I like that I can rate movies

Bibliography

- [1] ADOMAVICIUS, G., AND KWON, Y. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Transactions on Knowledge and Data Engineering* 24, 5 (2012), 896–911.
- [2] ADOMAVICIUS, G., MOBASHER, B., RICCI, F., AND TUZHILIN, A. Context-aware recommender systems. *Ai Magazine* 32, 3 (2011), 67–80.
- [3] AGUILAR, J., VALDIVIEZO-DÍAZ, P., AND RIOFRIO, G. A general framework for intelligent recommender systems. *Applied Computing and Informatics* 13, 2 (2017), 147 – 160.
- [4] AKAIKE, H. This week’s citation classic. *Current Contents Engineering, Technology, and Applied Sciences* 12, 51 (1981), 42.
- [5] ALEGRE, U., AUGUSTO, J. C., AND CLARK, T. Engineering context-aware systems and applications: A survey. *Journal of Systems and Software* 117 (2016), 55–83.
- [6] AYYAZ, S., QAMAR, U., AND NAWAZ, R. Hcf-crs: A hybrid content based fuzzy conformal recommender system for providing recommendations with confidence. *PloS one* 13, 10 (2018), e0204849.
- [7] BALABANOVIC, M. An adaptive web page recommendation service. In *Proceedings of the First International Conference on Autonomous Agents (Agents’97)* (New York, May–August, 1997), L. W. Johnson and B. H. Roth, Eds., ACM Press, pp. 378–385.
- [8] BALTRUNAS, L., LUDWIG, B., PEER, S., AND RICCI, F. Context relevance assessment and exploitation in mobile recommender systems. *Personal and Ubiquitous Computing* 16, 5 (2012), 507–526.

- [9] BASU, C., HIRSH, H., AND COHEN, W. W. Recommendation as classification: Using social and content-based information in recommendation. In *AAAI/IAAI* (1998), pp. 714–720. hallo miranda-note!
- [10] BILLSUS, D., AND PAZZANI, M. J. User Modeling for Adaptive News Access. *User Modeling and User-Adapted Interaction 10*, 2-3 (2000), 147–180.
- [11] BOBADILLA, J., ORTEGA, F., HERNANDO, A., AND GUTIÉRREZ, A. Recommender systems survey. *Knowledge-Based Systems 46* (2013), 109 – 132.
- [12] BONHARD, P., HARRIES, C., MCCARTHY, J., AND SASSE, M. A. Accounting for taste: Using profile similarity to improve recommender systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2006), CHI '06, Association for Computing Machinery, p. 1057–1066.
- [13] BONHARD, P., AND SASSE, M. A. 'Knowing me, knowing you' – Using profiles and social networking to improve recommender systems. *BT Technology Journal 24*, 3 (July 2006), 84–98.
- [14] BOUCKAERT, R. R. *Bayesian belief networks: from construction to inference*. PhD thesis, Utrecht University, 1995.
- [15] BREESE, J. S., HECKERMAN, D., AND KADIE, C. Empirical analysis of predictive algorithms for collaborative filtering, 2013.
- [16] BULANDER, R., DECKER, M., SCHIEFER, G., AND KOLMEL, B. Comparison of different approaches for mobile advertising. In *Second IEEE International Workshop on Mobile Commerce and Services* (08 2005), vol. 2005, pp. 174–182.
- [17] BURKE, R. Hybrid recommender systems: Survey and experiments. *User Modelling and User-Adapted Interaction 12*, 4 (2002), 331–370.
- [18] BURKE, R. D., HAMMOND, K. J., AND YOUND, B. C. The findme approach to assisted browsing. *IEEE Expert 12*, 4 (1997), 32–40.
- [19] CAI, R., ZHANG, C., WANG, C., ZHANG, L., AND MA, W.-Y. Musicsense: contextual music recommendation using emotional allocation modeling. In *Proceedings of the 15th ACM international conference on Multimedia* (01 2007), pp. 553–556.
- [20] CALDWELL, A., LAKENS, D., DEBRUINE, L., AND LOVE, J. Superpower: Simulation-based power analysis for factorial designs (v0.0.3).

- [21] CAMI, B. R., HASSANPOUR, H., AND MASHAYEKHI, H. User preferences modeling using dirichlet process mixture model for a content-based recommender system. *Knowledge-Based Systems 163* (2019), 644 – 655.
- [22] CHANG, S., HARPER, F. M., AND TERVEEN, L. Using groups of items for preference elicitation in recommender systems. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing* (New York, NY, USA, 2015), CSCW '15, Association for Computing Machinery, p. 1258–1269.
- [23] CHATZIDIMITRIS, T., GAVALAS, D., KASAPAKIS, V., KONSTANTOPOULOS, C., KYPRIADIS, D., PANTZIOU, G., AND ZAROLIAGIS, C. A location history-aware recommender system for smart retail environments. *Personal and Ubiquitous Computing* (2020).
- [24] CHENG, H.-F., WANG, R., ZHANG, Z., O'CONNELL, F., GRAY, T., HARPER, F. M., AND ZHU, H. Explaining decision-making algorithms through ui: Strategies to help non-expert stakeholders. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019), pp. 1–12.
- [25] CLAYPOOL, M., GOKHALE, A., MIRANDA, T., MURNIKOV, P., NETES, D., AND SARTIN, M. Combining content-based and collaborative filters in an online newspaper. In *In Proceedings of ACM SIGIR Workshop on Recommender Systems* (1999).
- [26] CONDLIFF, M. K., LEWIS, D. D., AND MADIGAN, D. Bayesian mixed-effects models for recommender systems. In *In ACM SIGIR '99 Workshop on Recommender Systems: Algorithms and Evaluation* (1999).
- [27] COOPER, G. F. The computational complexity of probabilistic inference using bayesian belief networks. *Artificial Intelligence 42*, 2 (1990), 393 – 405.
- [28] COSLEY, D., LAM, S. K., ALBERT, I., KONSTAN, J. A., AND RIEDL, J. Is seeing believing? how recommender system interfaces affect users' opinions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2003), CHI '03, Association for Computing Machinery, p. 585–592.
- [29] DEY, A. K. Understanding and using context. *Personal Ubiquitous Comput.* 5, 1 (jan 2001), 4–7.

- [30] DOURISH, P. What we talk about when we talk about context. *Personal Ubiquitous Comput.* 8, 1 (Feb. 2004), 19–30.
- [31] EKSTRAND, M. D., AND KONSTAN, J. A. Lecture notes in recommender systems: Evaluation and metrics, February 2020.
- [32] FRIEDMAN, N., NACHMAN, I., AND PE'ER, D. Learning bayesian network structure from massive datasets: The "sparse candidate" algorithm. In *UAI (1999)*, K. B. Laskey and H. Prade, Eds., Morgan Kaufmann, pp. 206–215.
- [33] GONZALEZ, A. *Context Model for Personalized Recommendations -CMPR*. PhD thesis, Tecnologico de Monterrey, 2018.
- [34] HE, J., AND CHU, W. W. A social network-based recommender system (snrs). In *Data Mining for Social Network Data*, N. Memon, J. J. Xu, D. L. Hicks, and H. Chen, Eds., vol. 12 of *Annals of Information Systems*. Springer, 2010, pp. 47–74.
- [35] HECKERMAN, D., GEIGER, D., AND CHICKERING, D. M. Learning bayesian networks: The combination of knowledge and statistical data. *CoRR abs/1302.6815* (1995).
- [36] HERLOCKER, J. L., KONSTAN, J. A., TERVEEN, L. G., AND RIEDL, J. T. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* 22, 1 (Jan. 2004), 5–53.
- [37] HERSKOVITS, E. H., AND COOPER, G. F. Kutato: An entropy-driven system for construction of probabilistic expert systems from databases, 1992.
- [38] HUANG, Y., AND BIAN, L. A bayesian network and analytic hierarchy process based personalized recommendations for tourist attractions over the internet. *Expert Syst. Appl.* 36, 1 (2009), 933–943.
- [39] IMCINE. Anuario estadístico de cine mexicano 2018. PDF, 1 2019.
- [40] JIN, X., ZHOU, Y., AND MOBASHER, B. Task-oriented web user modeling for recommendation. In *International Conference on User Modeling (07 2005)*, pp. 109–118.
- [41] KAMINSKAS, M. Matching information content with music. In *Proceedings of the third ACM conference on Recommender systems (01 2009)*, pp. 405–408.

- [42] KARATZOGLOU, A., AMATRIAIN, X., BALTRUNAS, L., AND OLIVER, N. Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems* (01 2010), pp. 79–86.
- [43] KARUMUR, R. P., NGUYEN, T. T., AND KONSTAN, J. A. Personality, user preferences and behavior in recommender systems. *Inf. Syst. Frontiers* 20, 6 (2018), 1241–1265.
- [44] KJAERULFF, U. B., AND MADSEN, A. L. Bayesian networks and influence diagrams. *Springer Science+ Business Media* 200 (2008), 114.
- [45] KLUVER, D., EKSTRAND, M. D., AND KONSTAN, J. A. Rating-based collaborative filtering: algorithms and evaluation. In *Social Information Access*. Springer, 2018, pp. 344–390.
- [46] KNIJNENBURG, B., REIJMER, N., AND WILLEMSSEN, M. Each to his own: how different users call for different interaction methods in recommender systems. In *RecSys '11 Proceedings of the fifth ACM International Conference on Recommender Systems, 23-27 October 2011, Chicago, Il., USA* (United States, 2011), B. Mobasher and R. Burke, Eds., Association for Computing Machinery, Inc, pp. 141–148.
- [47] KNIJNENBURG, B., AND WILLEMSSEN, M. *Evaluating Recommender Systems with User Experiments*. Springer, 01 2015, pp. 309–352.
- [48] KNIJNENBURG, B., WILLEMSSEN, M., AND KOBASA, A. A pragmatic procedure to support the user-centric evaluation of recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems* (10 2011), pp. 321–324.
- [49] KNIJNENBURG, B. P. Conducting user experiments in recommender systems. In *Proceedings of the sixth ACM conference on Recommender systems* (2012), pp. 3–4.
- [50] KNIJNENBURG, B. P., WILLEMSSEN, M. C., GANTNER, Z., SONCU, H., AND NEWELL, C. Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction* 22, 4 (2012), 441–504.
- [51] KOLLER, D. Lecture notes in probabilistic graphical models: Representation, March 2020.

- [52] KOLLER, D., AND FRIEDMAN, N. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.
- [53] KONSTAN, J. A., AND RIEDL, J. Recommender systems: from algorithms to user experience. *User Modeling and User-Adapted Interaction* 22, 1 (2012), 101–123.
- [54] KOREN, Y. The bellkor solution to the netflix grand prize. Netflix.
- [55] KOSIR, A., ODI, A., KUNAVR, M., TKALČIČ, M., AND TASIC, J. Database for contextual personalization. *ENGLISH EDITION* 78 (01 2011), 270–274.
- [56] LAKENS, D. Toster: Two one-sided tests (tost) equivalence testing (v0.3.4).
- [57] LAKENS, D., SCHEEL, A. M., AND ISAGER, P. M. Equivalence testing for psychological research: A tutorial. *Advances in Methods and Practices in Psychological Science* 1, 2 (2018), 259–269.
- [58] LARRAÑAGA, P., POZA, M., YURRAMENDI, Y., MURGA, R. H., AND KUIJPERS, C. M. H. Structure learning of bayesian networks by genetic algorithms: A performance analysis of control parameters. *IEEE transactions on pattern analysis and machine intelligence* 18, 9 (1996), 912–926.
- [59] LENZ, M. Use my viewing habits to sort suggestions, not just predicted ratings.
- [60] LESKIN, P. Google just revealed youtube’s ad revenue, 14 years after acquiring it, and the video site brought in 15 billion last year. *Business Insider* (2020).
- [61] LU, J., WU, D. S., MAO, M. S., WANG, W., AND ZHANG, G. Q. Recommender system application developments: A survey. *Decision Support Systems* 74 (2015), 12–32.
- [62] MARGARITIS, D. Learning bayesian network model structure from data. Tech. rep., Carnegie-Mellon Univ Pittsburgh Pa School of Computer Science, 2003.
- [63] MARTIN, F. J. Recsys’09 industrial keynote: Top 10 lessons learned developing deploying and operating real-world recommender systems. In *Proceedings of the Third ACM Conference on Recommender Systems* (New York, NY, USA, 2009), RecSys ’09, Association for Computing Machinery, p. 1–2.
- [64] MCNEE, S. M., RIEDL, J., AND KONSTAN, J. A. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI’06 extended abstracts on Human factors in computing systems* (2006), pp. 1097–1101.

- [65] MOONEY, R. J., AND ROY, L. Content-based book recommending using learning for text categorization. In *Proceedings of the ACM SIGIR Workshop Recommender Systems: Algorithms and Evaluation* (1999).
- [66] NAGARAJAN, R., SCUTARI, M., AND LÈBRE, S. Bayesian networks in r. *Springer 122* (2013), 125–127.
- [67] NEAPOLITAN, R. E., ET AL. *Learning bayesian networks*, vol. 38. Pearson Prentice Hall Upper Saddle River, NJ, 2004.
- [68] OKU, K., NAKAJIMA, S., MIYAZAKI, J., AND UEMURA, S. Context-aware svm for context-dependent information recommendation. In *7th International Conference on Mobile Data Management (MDM'06)* (06 2006), IEEE, pp. 109 – 109.
- [69] ONO, C., KUROKAWA, M., MOTOMURA, Y., AND ASOH, H. A context-aware movie preference model using a bayesian network for recommendation and promotion. In *User Modeling* (2007), C. Conati, K. F. McCoy, and G. Paliouras, Eds., vol. 4511 of *Lecture Notes in Computer Science*, Springer, pp. 247–257.
- [70] OZOK, A. A., FAN, Q., AND NORCIO, A. F. Design guidelines for effective recommender system interfaces based on a usability criteria conceptual model: results from a college student population. *Behaviour & Information Technology* 29, 1 (2010), 57–83.
- [71] PALMISANO, C., TUZHILIN, A., AND GORGOGNONE, M. Using context to improve predictive modeling of customers in personalization applications. *Knowledge and Data Engineering, IEEE Transactions on* 20 (12 2008), 1535–1549.
- [72] PARK, H.-S., YOO, J.-O., AND CHO, S.-B. A context-aware music recommendation system using fuzzy bayesian networks with utility theory. In *FSKD* (2006), L. Wang, L. Jiao, G. Shi, X. Li, and J. Liu, Eds., vol. 4223 of *Lecture Notes in Computer Science*, Springer, pp. 970–979.
- [73] PARSONS, S. Bayesian artificial intelligence by kevin b. korb and ann e. nicholson, chapman and hall, 369 pp., isbn 1-58488-387-1. *Knowledge Eng. Review* 19, 3 (2004), 275–276.
- [74] PAZZANI, M. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review* 13, 5–6 (1999), 393–408.

- [75] PEARL, J. Chapter 2 - bayesian inference. In *Probabilistic Reasoning in Intelligent Systems*, J. Pearl, Ed. Morgan Kaufmann, San Francisco (CA), 1988, pp. 29 – 75.
- [76] PEARL, J. Chapter 3 - markov and bayesian networks: Two graphical representations of probabilistic knowledge. In *Probabilistic Reasoning in Intelligent Systems*, J. Pearl, Ed. Morgan Kaufmann, San Francisco (CA), 1988, pp. 77 – 141.
- [77] PEARL, J. Chapter 8 - learning structure from data. In *Probabilistic Reasoning in Intelligent Systems*, J. Pearl, Ed. Morgan Kaufmann, San Francisco (CA), 1988, pp. 381 – 414.
- [78] PEARL, J., GEIGER, D., AND VERMA, T. Conditional independence and its representations. *Kybernetika* 25 (1989), 33–44.
- [79] PESSEMIER, T. D., DERYCKERE, T., AND MARTENS, L. Extending the bayesian classifier to a context-aware recommender system for mobile devices. In *2010 Fifth International Conference on Internet and Web Applications and Services* (2010), pp. 242–247.
- [80] PHUKSENG, T., AND SODSEE, S. Recommender system based on expert and item category. *Engineering Journal* 22, 2 (2018), 157–168.
- [81] PU, P., CHEN, L., AND HU, R. A user-centric evaluation framework for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems* (2011), pp. 157–164.
- [82] PU, P., CHEN, L., AND HU, R. Evaluating recommender systems from the user’s perspective: survey of the state of the art. *User Modeling and User-Adapted Interaction* 22, 4 (2012), 317–355.
- [83] RAMÍREZ-NORIEGA, A., JUÁREZ-RAMÍREZ, R., AND MARTÍNEZ-RAMÍREZ, Y. Evaluation module based on bayesian networks to intelligent tutoring systems. *Int. J. Inf. Manag.* 37, 1 (2017), 1488–1498.
- [84] RICCI, F., ROKACH, L., AND SHAPIRA, B. *Introduction to Recommender Systems Handbook*. Springer US, Boston, MA, 2011, pp. 1–35.
- [85] RICHA, BEDI, P., THAMPI, EL-ALFY, MITRA, AND TRAJKOVIC. Parallel proactive cross domain context aware recommender system. *Journal of Intelligent & Fuzzy Systems* 34, 3 (2018), 1521 – 1533.

- [86] SAE-UENG, S., PINYAPONG, S., OGINO, A., AND KATO, T. Personalized shopping assistance service at ubiquitous shop space. In *22nd International Conference on Advanced Information Networking and Applications-Workshops (aina workshops 2008)* (04 2008), pp. 838 – 843.
- [87] SARWAR, B., KARYPIS, G., KONSTAN, J., AND RIEDL, J. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web* (2001), pp. 285–295.
- [88] SARWAR, B. M., KONSTAN, J. A., BORCHERS, A., HERLOCKER, J., MILLER, B., AND RIEDL, J. Using filtering agents to improve prediction quality in the grouplens research collaborative filtering system. In *Proceedings of the 1998 ACM conference on Computer supported cooperative work* (1998), ACM Press, pp. 345–354.
- [89] SCHWAB, I., KOBZA, A., AND KOYCHEV, I. Learning user interests through positive examples using content analysis and collaborative filtering. Draft.
- [90] SCOTT, D. W. On optimal and data-based histograms. *Biometrika* 66, 3 (12 1979), 605–610.
- [91] SCUTARI, M., AND NAGARAJAN, R. On identifying significant edges in graphical models of molecular networks, 2011.
- [92] SCUTARI, M., AND NESS, R. bnlearn: Bayesian network structure learning, parameter learning and inference (v4.4.1).
- [93] SEDLÁČEK, J. On local properties of finite graphs. In *Graph Theory* (Berlin, Heidelberg, 1983), M. Borowiecki, J. W. Kennedy, and M. M. Sysło, Eds., Springer Berlin Heidelberg, pp. 242–247.
- [94] SHANI, G., AND GUNAWARDANA, A. Evaluating recommendation systems. In *Recommender systems handbook*. Springer, 2011, pp. 257–297.
- [95] SMITH, B., AND COTTER, P. A personalized tv listings service for the digital tv age. *Knowledge-Based Systems* 13 (2000), 53–59.
- [96] SWEARING, K., AND SINHA, R. Interaction design for recommender systems. In *Designing Interactive Systems* (2002).
- [97] TOWLE, B., AND QUINN, C. Knowledge based recommender systems using explicit user models. In *Papers from the AAAI Workshop, AAAI Technical Report WS-00-04* (2000), Menlo Park, CA: AAAI Press, pp. 74–77.

- [98] TRAN, T., AND COHEN, R. Hybrid recommender systems for electronic commerce. In *In Knowledge-Based Electronic Markets, Papers from the AAAI Workshop, AAAI Technical Report WS-00-04* (2000), AAAI Press, pp. 78–83.
- [99] TSAMARDINOS, I., ALIFERIS, C. F., STATNIKOV, A. R., AND STATNIKOV, E. Algorithms for large scale markov blanket discovery. In *FLAIRS conference* (2003), vol. 2, pp. 376–380.
- [100] TSAMARDINOS, I., BROWN, L. E., AND ALIFERIS, C. F. The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning* 65, 1 (Oct 2006), 31–78.
- [101] VERMA, T., AND PEARL, J. *Equivalence and synthesis of causal models*. UCLA, Computer Science Department, 1991.
- [102] WALKER, E., AND NOWACKI, A. S. Understanding equivalence and noninferiority testing. *Journal of general internal medicine* 26, 2 (Feb 2011), 192–196. 20857339[pmid].
- [103] WANG, Y., AND VASSILEVA, J. Bayesian network-based trust model. In *Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003)* (11 2003), pp. 372– 378.
- [104] WASFI, A. M. Collecting user access patterns for building user profiles and collaborative filtering. In *Proceedings of the International Conference on Intelligent User Interfaces* (1999), pp. 57–64.
- [105] WINSTON CHANG AND JOE CHENG AND, JJ ALLAIRE AND, YIHUI XIE AND JONATHAN MCPHERSON. shiny: Web application framework for r (v.1.4.0).
- [106] YAO, Y., AND HARPER, F. Judging similarity: a user-centric study of related item recommendations. In *Proceedings of the 12th ACM Conference on Recommender Systems* (09 2018), pp. 288–296.
- [107] YARAMAKALA, S., AND MARGARITIS, D. Speculative markov blanket discovery for optimal feature selection. In *Fifth IEEE International Conference on Data Mining (ICDM'05)* (2005), pp. 4 pp.–.
- [108] YOO, K.-H., AND GRETZEL, U. *Creating More Credible and Persuasive Recommender Systems: The Influence of Source Characteristics on Recommender System Evaluations*. Springer, 12 2010, pp. 455–477.

- [109] YUAN, Q., CONG, G., ZHAO, K., MA, Z., AND SUN, A. Who, where, when, and what: A nonparametric bayesian approach to context-aware recommendation and search for twitter users. *ACM Transactions on Information Systems* 33 (02 2015), 1–33.
- [110] ZHAO, Q., HARPER, F. M., ADOMAVICIUS, G., AND KONSTAN, J. A. Explicit or implicit feedback? engagement or satisfaction? a field experiment on machine-learning-based recommender systems. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing* (2018), pp. 1331–1340.
- [111] ZHAO, Q., WILLEMSSEN, M. C., ADOMAVICIUS, G., HARPER, F. M., AND KONSTAN, J. A. Interpreting user inaction in recommender systems. In *Proceedings of the 12th ACM Conference on Recommender Systems* (New York, NY, USA, 2018), RecSys '18, Association for Computing Machinery, p. 40–48.
- [112] ÇANO, E., AND MORISIO, M. Hybrid recommender systems: A systematic literature review, 2019.

Curriculum Vitae

Nora Hernández is a maker and education enthusiast. Her passion for learning can be traced back to her college years when she spent countless hours developing new and exciting projects involving music, robots and a bit of science.

Her maker spirit shined when she earned her Bachelor's degree in Mechatronics Engineering from Tecnológico de Monterrey in 2016. She gained experience in the steel industry while working for one of the biggest firms in the continent, where she worked as a production intern and then became a quality specialist.

Nora was accepted in the Computer Science master's program in 2018 with the Academic Excellence Scholarship from Tecnológico de Monterrey, while also being recipient of the National Graduate Studies Scholarship from the Mexican's National Commission for Science and Technology. She specialised in probabilistic inference models to build recommendation systems using Bayesian networks, and was amazed to discover how much she enjoyed statistics and probability theory. So much so that now her current interests include both, alongside cognitive science and education. Her dream job would combine all of the above, and the possibility to share the experience of learning with the entire world.

Nora enjoys drinking tea with her fellows while addressing academic and epistemological issues. Her hobbies also include watching, talking and writing about films, jazz music, and dreams with becoming an skilled trumpeter and percussionist.

This document was typed in using $\text{\LaTeX} 2_{\epsilon}$ ^a by Nora Patricia Hernández López.

^aThe style file `phdThesisFormat.sty` used to set up this thesis was prepared by the Center of Intelligent Systems of the Instituto Tecnológico y de Estudios Superiores de Monterrey, Monterrey Campus