LUIZ HENRIQUE BARBOSA MORMILLE

**Learning Probabilistic Relational Models:
A Novel Approach**

São Paulo

2018

LUIZ HENRIQUE BARBOSA MORMILLE

Learning Probabilistic Relational Models: A Novel Approach

Dissertação apresentada à Escola Politécnica
da Universidade de São Paulo para obtenção
do Título de Mestre em Ciências.

São Paulo

2018

LUIZ HENRIQUE BARBOSA MORMILLE

Learning Probabilistic Relational Models: A Novel Approach

Dissertação apresentada à Escola Politécnica
da Universidade de São Paulo para obtenção
do Título de Mestre em Ciências.

Área de Concentração:
Engenharia de Computação

Supervisor: Prof. Fabio Gagliardi Cozman

São Paulo

2018

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 09 de novembro de 2018

Assinatura do autor:

Assinatura do orientador: _____

Catálogo-na-publicação

Mormille, Luiz Henrique Barbosa

Learning Probabilistic Relational Models: A Novel Approach / L. H. B.

Mormille -- versão corr. -- São Paulo, 2018.

104 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1. ESTATÍSTICA E DADOS NUMÉRICOS I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II.t.

To my friend, Jimmy Lescano

Acknowledgements

Since I began this research, many positive things have occurred in my life, both personally, privately and academically. For that, I am very, very grateful to those who somehow helped me along this journey.

There are some people I would like to mention whose support was vital to get me here.

First of all, I would like to thank my mom and dad, whom I will never be able to thank enough. Your passions became my passions, and if I ever did something good, I owe it to you. I love you.

I would also like to thank Dr. Daisaku Ikeda, my master, whose ideals inspire me continually to follow a path of development linked to a grand purpose.

I would also like to express my deep gratitude to my supervisor, Professor Fabio Gagliardi Cozman, for all the advice, support, understanding and encouragement. Working with you has opened my mind in so many ways. I am deeply thankful for the opportunity.

I would also like to thank my friend and mentor, Professor Renato Vicente, and all my lab colleagues, especially Arthur Gusmão and Glauber de Bona.

I would like to express my undying gratitude to my circle of long time support: Rafael Makiyama, Mariana Akinaga, Eduardo Makiyama, Francisco Akinaga, Denis Watanabe (and Family), Cauê Barreira (and Family), Ricardo Nagano and João Mendes. You are my closest friends, from whom I am able to grab inspiration in handfulls every single day. You are the family that I chose.

I thank the many great friends at BSGI and EBS that I was able to make over the past year, especially Ricardo Miyamoto, Edjan Santos, Eduardo Shoiti Sato, Cassiano Rodrigues, Luciano Leme and Ricardo Hirata. Your support and orientation led me through great personal and private development to carry on with this research.

Finally, I would also like to thank two institutions: Serasa Experian, for providing the data for this research; and CAPES, for funding this research project.

*“We do not live in a privileged
reference frame. If we crave
some cosmic purpose, then let
us find ourselves a worthy goal.”
(Carl Sagan)*

Resumo

Embora a maioria dos métodos de aprendizado estatístico tenha sido desenvolvida para se trabalhar com dados armazenados em uma única tabela, muitas bases de dados estão armazenadas em bancos de dados relacionais. Modelos Probabilísticos Relacionais (PRM) estendem Redes Bayesianas introduzindo relações e indivíduos, tornando possível a representação de informação em uma base de dados relacional. Entretanto, aprender um PRM através de dados relacionais é uma tarefa mais complexa que aprender uma Rede Bayesiana de uma única tabela. As maiores dificuldades que se impõe enquanto se aprende um PRM são estabelecer quais são as estruturas de dependência legais, procurar por possíveis estruturas, e avalia-las. Esta tese foca em desenvolver um novo método de aprendizado de estruturas de PRM, descrever um pacote na linguagem R que suporte este método e aplica-lo a um cenário real e de grande escala, a cidade de Atibaia, no estado de São Paulo, Brasil. Esta pesquisa está baseada em uma base de dados combinando três tabelas distintas, cada uma representando uma classe no domínio de estudo. A primeira tabela contém 27 atributos de 110.816 habitantes de Atibaia, e a segunda tabela contém 9 atributos de 20.162 empresas da cidade. Por fim, a terceira tabela possui 8 atributos para 327 setores censitários (pequenas unidades territoriais que formam a cidade de Atibaia). A proposta é aplicada para aprender-se a estrutura de um PRM e seus parâmetros através desta base de dados. O modelo foi utilizado para verificar se a *classe social* de uma pessoa pode ser explicada pelo local onde ela vive, seus vizinhos e as companhias próximas. Experimentos preliminares foram conduzidos e um artigo foi publicado no *Symposium on Knowledge Discovery, Mining and Learning* (KDMiLe). O desempenho do algoritmo foi reavaliada através de extensiva experimentação, e um estudo mais amplo foi conduzido com os dados da *Serasa Experian*. Por fim, o pacote em R que suporta o método proposto foi refinado, e documentação e tutorial apropriado foram descritos.

Palavras-chave: Probabilistic Graphical Models. Bayesian Network. Multi-Relational Data Mining. Inductive Logic Programming.

Abstract

While most statistical learning methods are designed to work with data stored in a single table, many large datasets are stored in relational database systems. Probabilistic Relational Models (PRM) extend Bayesian networks by introducing relations and individuals, thus making it possible to represent information in a relational database. However, learning a PRM from relational data is a more complex task than learning a Bayesian Network from “flat” data. The main difficulties that arise while learning a PRM are establishing what are the legal dependency structures, searching for possible structures, and scoring them. This thesis focuses on the development of a novel approach to learn the structure of a PRM, describes a package in the R language to support the learning framework, and applies it to a real, large scale scenario of a city named Atibaia, in the state of São Paulo, Brazil. The research is based on a database combining three different tables, each representing one class in the domain of study. The first table contains 27 attributes from 110,816 citizens of Atibaia. The second table contains 9 attributes from 20,162 companies located in the city. And finally, the third table has 8 attributes from 327 census sectors (small territorial units that comprise the city of Atibaia). The proposed framework is applied to learn a PRM structure and parameters from the database. The model is used to verify if the Social Class of a person can be explained by the location where they live, their neighbors, and the companies nearby. Preliminary experiments have been conducted and a paper published in the *2017 Symposium on Knowledge Discovery, Mining and Learning* (KDMiLe). The algorithm performance was further evaluated by extensive experimentation, and a broader study using Serasa Experian data was conducted. Finally, the package in the R language that supports our method was refined along with proper documentation and a tutorial.

Keywords: Probabilistic Graphical Models. Bayesian Network. Multi-Relational Data Mining. Inductive Logic Programming.

List of Figures

Figure 1 – A relational decision tree	24
Figure 2 – Citizen Bayesian network $\mathcal{B}^{citizen}$ with CPDs.	26
Figure 3 – The relational schema and the relational skeleton for the Atibaia domain.	30
Figure 4 – An illustration of Vanilla-PRM’s <i>assumption 2</i> : the <i>instance dependency graph</i> G_σ is not allowed to have cycles at a class level.	38
Figure 5 – Slices of the Person and Census_sector datasets.	40
Figure 6 – A PRM structure learned using the proposed method considering a <i>fifth assumption</i>	41
Figure 7 – All census sectors in Atibaia plotted in a map.	48
Figure 8 – Visual representation of the blacklists of edges for the Person experiment.	54
Figure 9 – The dependency structure of Π_P	57
Figure 10 – The dependency structure of Π_{Ground}	59
Figure 11 – Π_P confusion matrix of the <i>Person experiment</i>	60
Figure 12 – Π_P performance measures of the <i>Person experiment</i>	61
Figure 13 – Π_{Ground} confusion matrix of the <i>Person experiment</i>	62
Figure 14 – Π_{Ground} performance measures of the <i>Person experiment</i>	63
Figure 15 – Visual representation of the blacklists of edges for the Company experiment.	65
Figure 16 – The dependency structure of Π_C	66
Figure 17 – Π_C confusion matrix of the <i>Company experiment</i>	68
Figure 18 – Π_C performance measures of the <i>Company experiment</i>	69
Figure 19 – Π_{Ground} confusion matrix of the <i>Company experiment</i>	70
Figure 20 – Π_{Ground} performance measures of the <i>Company experiment</i>	71

List of Tables

Table 1	– The Ontological Commitment (what exists in the world) and the Epistemological Commitment (Agent’s beliefs about facts) of four different logical languages.	20
Table 2	– Jaccard similarity between the set of arrows of Π_{Ground} and Π_P	64
Table 3	– The Jaccard similarity between the set of arrows of Π_{Ground} and Π_P considering only the dependencies inside the <i>leaf class</i> Person	64
Table 4	– Jaccard similarity between the set of arrows of Π_{Ground} and Π_P	72
Table 5	– The Jaccard similarity between the set of arrows of Π_{Ground} and Π_C considering only the dependencies inside the <i>leaf class</i> Company	72
Table 6	– A description of the attribute Person . <i>Gender</i>	80
Table 7	– A description of the attribute Person . <i>Age_group</i>	81
Table 8	– A description of the attribute Person . <i>Public_server</i>	81
Table 9	– A description of the attribute Person . <i>Bolsa_familia_flag</i>	81
Table 10	– A description of the attribute Person . <i>Company_owner_flag</i>	82
Table 11	– A description of the attribute Person . <i>Rural_producer_flag</i>	82
Table 12	– A description of the attribute Person . <i>Social_class</i>	83
Table 13	– A description of the attribute Person . <i>Consumption_index</i>	83
Table 14	– A description of the attribute Person . <i>Credit_risk</i>	84
Table 15	– A description of the attribute Person . <i>Presumed_income</i>	84
Table 16	– A description of the attribute Person . <i>Mosaic</i>	85
Table 17	– A description of the attribute Person . <i>Propensity_for_luxury</i>	86
Table 18	– A description of the attribute Person . <i>Propensity_for_credit_card</i>	87
Table 19	– A description of the attribute Person . <i>Propensity_for_mortgage</i>	87
Table 20	– A description of the attribute Person . <i>Propensity_for_broadband_internet</i>	88
Table 21	– A description of the attribute Person . <i>Propensity_for_personal_loan</i>	88
Table 22	– A description of the attribute Person . <i>Propensity_for_postpaid_phone</i>	88
Table 23	– A description of the attribute Company . <i>Age_band</i>	89
Table 24	– A description of the attribute Company . <i>Range_of_employees</i>	90
Table 25	– A description of the attribute Company . <i>Presumed_revenue</i>	90
Table 26	– A description of the attribute Company . <i>Operating_index</i>	90
Table 27	– A description of the attribute Company . <i>Rural_producer_flag</i>	91
Table 28	– A description of the attribute Census_sector . <i>Situation</i>	92
Table 29	– A description of the attribute Census_sector . <i>Average_house_hold</i>	92
Table 30	– A description of the attribute Census_sector . <i>Average_income</i>	93
Table 31	– A description of the attribute Census_sector . <i>Domiciles</i>	93
Table 32	– A description of the attribute Census_sector . <i>Houses</i>	94

Table 33 – A description of the attribute Census_sector . <i>Apartment</i>	94
Table 34 – A description of the attribute Census_sector . <i>Paid_domiciles</i>	95
Table 35 – A description of the attribute Census_sector . <i>Unpaid_domiciles</i>	95
Table 36 – A description of the attribute Census_sector . <i>Rented_domiciles</i>	95
Table 37 – A description of the attribute Census_sector . <i>Water_supply</i>	96
Table 38 – A description of the attribute Census_sector . <i>Sewage_network</i>	96
Table 39 – A description of the attribute Census_sector . <i>Garbage_collection</i>	97
Table 40 – A description of the attribute Census_sector . <i>Electricity</i>	97
Table 41 – A description of the attribute Census_sector . <i>Female_domiciles</i>	98
Table 42 – A description of the attribute Census_sector . <i>Male_domiciles</i>	98
Table 43 – Π_P PRM Dependencies	99
Table 44 – Π_{Ground} PRM Dependencies	101
Table 45 – Π_C PRM Dependencies	103

List of abbreviations and acronyms

PRM	Probabilistic Graphical Models
MRDM	Multi-Relational Data Mining
DAG	Directed Acyclic Graph
IBGE	Instituto Brasileiro de Geografia e Estatística
KDMiLe	Symposium on Knowledge Discovery, Mining and Learning
KPI	Key Performance Indicator

Contents

1	INTRODUCTION	15
2	BACKGROUND	17
2.1	Bayesian approach and reasoning	17
2.2	Inductive Logic Programming	19
2.2.1	Deductive and Inductive Logic	19
2.2.2	Propositional Logic	19
2.2.3	First-order Logic	20
2.2.4	Logic Programming	21
2.2.5	Inductive Logic Programming	21
2.3	Multi-Relational Data Mining	23
2.3.1	Relational Association Rules	23
2.3.2	Relational Decision Trees	24
2.3.3	Relational Distance-Based Approaches	25
2.4	Bayesian Networks	25
2.4.1	Scoring functions	27
2.4.2	Parameter Learning	27
2.4.3	Structure Learning	28
2.5	Probabilistic Relational Models	28
2.5.1	The PRM semantics	29
2.5.2	PRM structure learning	31
2.5.2.1	Legal structures	31
2.5.2.2	Evaluating different structures	32
2.5.2.3	Structure search	33
3	VANILLA-PRM: A NOVEL METHOD FOR PRM LEARNING . . .	35
3.1	Vanilla-PRM assumptions	36
3.2	Aggregations	39
3.3	Vanilla-PRM algorithm	42
4	EXPERIMENTS	44
4.1	The domain of study	44
4.1.1	The <i>Person</i> class	44
4.1.2	The <i>Company</i> class	46
4.1.3	The <i>Census Sector</i> class	47
4.2	The case study	50

4.2.1	Method and measures	50
4.2.1.1	Cross-validation	50
4.2.1.2	Measures	50
4.2.1.3	Comparison	52
4.3	The Person PRM experiment	53
4.3.1	The master tables for Atibaia domain	53
4.3.2	Person experiment: structure learning	56
4.3.3	The Person PRM structure	58
4.3.4	The Ground PRM structure	58
4.3.5	The Person PRM performance	60
4.4	The Company PRM	65
4.4.1	Company experiment: structure learning	65
4.4.2	The Company PRM structure	67
4.4.3	The Company PRM performance	68
4.5	The package	72
5	CONCLUSIONS	74
	BIBLIOGRAPHY	77
A	PERSON ATTRIBUTE DESCRIPTION	80
B	COMPANY ATTRIBUTE DESCRIPTION	89
C	CENSUS SECTOR ATTRIBUTE DESCRIPTION	92
D	PERSON PRM	99
E	GROUND PRM	101
F	COMPANY PRM	103

1 Introduction

Most large data collections are stored in relational database systems consisting of multiple tables; however, standard data mining techniques are only applicable to a single table. Multi-relational data mining (MRDM) focuses on the search for patterns across multiple tables (relations) of a database (DŽEROSKI, 2003), and many techniques have been proposed in that context. Indeed, relational models lead to a deeper understanding of the relations held within domains, and can be used for exploratory analysis, predictions and complex inferences.

Despite the success of Bayesian networks in a wide variety of real-world and research applications, they cannot be used to model domains where we might encounter several entities in different configurations (KOLLER, 1999), for they lack the concept of objects and relations.

Probabilistic Relational Models (PRM) are an extension of Bayesian Networks, introducing the concepts of objects and its properties, and the relations held between them, specifying a template for a probability distribution (GETOOR; TASKAR, 2007). Thus, PRMs offer a rich relational structure, allowing a property of an object to depend probabilistically, not only on the properties of that given object, but also on properties of other related objects (GETOOR; SAHAMI et al., 1999). It can be said that PRMs are to Bayesian networks as relational logic is to propositional logic.

However, learning a PRM from relational data is a more complex task than learning a Bayesian Network from “flat” data. There are three main difficulties that arises while learning a PRM. The first one is establishing what are the legal dependency structures for a given domain: we must avoid cycles in the structure. The second difficulty is how to score the possible legal structures. And the third challenge is to search for possible structures (FRIEDMAN et al., 1999).

Given the complications often faced while learning a PRM, a novel method for learning probabilistic relational models is proposed and applied in a real large-scale problem of a city named Atibaia, in the state of São Paulo, Brazil. The research is based on a database combining three different tables, each representing one class in the domain of study. The first table contains 27 attributes from 110,816 citizens of Atibaia and the second table contains 9 attributes from 20,162 companies located in the city, and they were provided by Serasa Experian. The third table was provided by the Brazilian Institute of Geography and Statistics (IBGE), and has 8 attributes from 327 census sectors (small territorial units that comprise the city of Atibaia). The proposed framework is applied to learn a PRM structure and parameters from the database. The model is used to verify

if the Social Class of a person can be explained by the location where they live, their neighbors, and the companies nearby. A software package in the R language was built to support our method and the results of the preliminary experiments were published in the 2017 Symposium on Knowledge Discovery, Mining and Learning (KDMiLe) (MORMILLE; COZMAN, 2017).

After the preliminary experiments, the algorithm performance was further evaluated by extensive experimentation, and a broader study using Serasa Experian data was conducted. The package in the R language that supports our method was refined along with proper documentation and a tutorial. Taken together, these contributions should be valuable to researchers and practitioners interested in dealing with large relational databases.

In Chapter 2 we present a quick summary over some fundamental concepts in Probabilistic Relational Models. Section 2.1 briefly review the history and concepts of Bayesian reasoning; Section 2.2 presents an overview on Inductive Logic Programming; Section 2.3 presents a quick review on some popular Multi-Relational Data Mining methods; Section 2.4 introduces the key aspects of Bayesian Networks; And Section 2.5 explains the most important concepts on Probabilistic Relational Models, as well as its semantics, and a typical structure learning method.

Chapter 3 introduces our contribution and explains the novel method for PRM learning. Chapter 4 reports on the experiments and results on the case study and an R package. And Chapter 5 briefly presents some conclusions.

2 Background

This chapter presents a quick summary over some fundamental concepts in Probabilistic Relational Models. Section 2.1 briefly reviews the history and concepts of Bayesian reasoning; Section 2.2 presents an overview on the topics of Deductive and Inductive Logic, Propositional Logic, First-order Logic, Logic Programming and Inductive Logic Programming; Section 2.3 presents a quick review on Multi-Relational Data Mining, Relational Decision Trees and Relational Distance-Based Methods; Section 2.4 introduces the key aspects of Bayesian Networks, scoring functions and methods for parameter and structure learning; and Section 2.5 explains the most important concepts on Probabilistic Relational Models, its semantics, and a typical structure learning method.

2.1 Bayesian approach and reasoning

Laplace stated that “probability theory is nothing but common sense reduced to calculation”. In fact, probability theory provides mathematical tools for quantifying uncertainty, in a way that it can be applied to any problem that involves uncertainty (MURPHY, 2012).

The basic idea behind probability is to treat it as a generalization of the concept of logical entailment in deductive logic. In a deductive valid argument, the premises logically entail the conclusion in a sense that, if the premises are all true, the conclusion can’t possibly be false. The truth of the premises guarantees the truth of the conclusion. If the argument is deductively invalid, then the premises do not logically entail the conclusion, which simply means that, even if the premises are all true, it is still possible for the conclusion to be false. Logical entailment in this sense is a bivalent or binary property, there are no degrees of validity or logical entailment.

However, there are cases where a logical support comes in degrees, where the strength of the logical entailment can be quantified using probability theory, defined in this terms: a measure of the degree of partial entailment, or the degree of logical support that a conclusion has, relative to a set of premises. Or it can also be defined as the degree of confirmation, that evidence confers on a hypothesis, where premises are equivalent to evidence, and conclusion is equivalent to hypothesis.

In this scenario, all probabilities are **conditional**, which means that there are no unconditional probabilities. Probability statements are only meaningful relative to some body of evidence or background assumptions.

Rudolph Carnap proposed a *confirmation function* that applies to formal languages on all cases where evidence has a logical relationship to a hypothesis (CARNAP, 1962). However, Carnap found that there are other objections to the whole program, and his system ran into technical and philosophical problems. This discussion is of deep academic interest on the fields of philosophy, logic and mathematics, concerning the foundations of probability; however, it is not the scope of this work.

Nonetheless, nowadays, most of the discussion is between proponents of frequency based approaches to probability, and proponents of subjective or Bayesian approaches to probability.

One of the main differences between the *Bayesian* approach and the *Frequentist* approach to probability, is that the former provides a framework that allows one to assign a *single case probability*.

According to the Bayesian approach, probability, in general, should be interpreted as a measure of the degree of belief that one has in the truth of a proposition ($P(X) = \text{degree of belief that } X \text{ is true}$). This conceptual framework can be applied even in cases where the frequency interpretation also works.

A theory of probability should distinguish good reasoning from bad reasoning. However, the only logically consistent way of reasoning with subjective degrees of belief is if those degrees of belief satisfy the basic mathematical rules of reasoning with probabilities. All the action on the subjective interpretation lies on the details of this argument.

However, a degree of belief is a feature of our inner mental state. Nonetheless, Frank Ramsey and Bruno de Finetti, independently found a way of assigning a number to a person's degree of belief. Their solution exploits the fact that there is a close relationship between belief and action, and in this case, between degrees of belief and one's betting behavior, when asked to choose between different bets (betting ratios or betting rates). This procedure is commonly used in decision theory.

Nonetheless, a rational betting strategy will automatically satisfy the basic mathematical rules for reasoning with probabilities. The basic premise is that no rational person will willingly agree to a bet that is guaranteed to lose them money, constituting a *sure-loss contract* (HACKING, 2001), or, as denoted by Frank Ramsey in 1927, a Dutch book argument (FREEDMAN, 2003). Therefore, a set of personal beliefs is called coherent if, and only if, that set satisfies the rules of probability theory, not being open to a sure-loss contract.

To that end, Bayes' Rule provides a model for how one ought to learn from experience and update degrees of belief in a hypothesis in light of new evidence:

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

On Bayes' Rule, the term $P(H|E)$ is the degree of belief one should have in a hypothesis H , given some evidence E (This is also called the *posterior probability*). $P(H)$ is the prior probability of the hypothesis, and it represents the degree of certainty one have before being introduced to new evidence. The term $P(E|H)$ is called the *likelihood* of the evidence E , if H is true. And $P(E)$ represents the “total probability” of the evidence E .

For a *Frequentist*, Bayes' Rule is an useful formulation of conditional probability, but its use is restricted to cases where relative frequencies can be defined. As for *Bayesians*, it is a central principle to an entire approach of rationality, and therefore, it can be used in a much wider range of applications in statistical inference theory.

2.2 Inductive Logic Programming

2.2.1 Deductive and Inductive Logic

Deductive Logic can be defined as the entailment process of one or more premises, in order to reach a *valid argument*. In a deductive valid argument, the premises logically entail the conclusion in a sense that, if the premises are all true, the conclusion is certainly truth. Rephrasing the above sentence, in a deductive valid argument, the truth of the premises guarantees the truth of the conclusion (STERNBERG; STERNBERG, 2016).

On the other hand, in *Inductive logic* the premises entailment is not conclusive. Still, it is a method of reasoning that aims to provide a measure of the degree of partial entailment, or the degree of logical support that a conclusion has, relative to a set of premises.

Validity, *strength* and *weakness* are logical properties of arguments. In Inductive reasoning, the premises are used to provide strong evidence for the truth of a conclusion (COPI; COHEN; FLAGE, 2016). That means that an *Inductive argument* is intended to be strong, rather than valid.

2.2.2 Propositional Logic

Propositional logic (often called sentential logic or statement logic) focuses on the study of *propositions* (or claims) formed by other propositions through the use of *logical connectives* (or logical operators).

Propositions are closed sentences that are *truth-bearers* (an entity that can be either true or false). Therefore, propositional logic deals with logical relationships between propositions takes as “wholes”. In propositional logic, the fundamental unit of analysis is

the whole proposition, which can be true or false. Propositional logic is interested in how the truth value of “compound propositions” depends on the truth value of the “individual propositions” that make it up.

In propositional logic, it is possible to construct compound claims out of a small number of basic logical connectives. It is possible to form *conjunctions* using “AND”, or *disjunctions* using “OR”, or *conditionals* using “IF-THEN”, or *contradictories* using “NOT”. In the end, the truth value of the “compound propositions” is a function of the true values of the “individual propositions” (KRAJICEK, 1995).

2.2.3 First-order Logic

Different communities might determine different language categories, which means that the same item can be classified in different categories, depending on the community. Also, some communities may define categories that other communities do not (WHORF, 1956).

Nonetheless, a study conducted by Wanner in 1975 indicates that a subject remembers the content of what they read better than the actual words, which implies that, somehow, they are storing this content in some categories. Not every single word is remembered, but the abstraction of those words is understood (WANNER; SHINER, 1976).

A representational language is also abstract and not literal. *First-order logic* embraces this idea and provides one way of representing the language that we know in a way a computer can interpret.

First-order logic represents an extension of propositional logic. In the latter, a proposition is expressed as *atoms*, while in the former, propositions can be expressed as *constructs*, allowing the use of predicates and quantifiers (HUGHES; CRESSWELL, 1996). Table 1 compare how broad Propositional and First-order Logic are, also comparing them with *Temporal Logic* and *Fuzzy Logic*.

Table 1 – The Ontological Commitment (what exists in the world) and the Epistemological Commitment (Agent’s beliefs about facts) of four different logical languages.

Language	Ontological Commitment	Epistemological Commitment
Propositional Logic	Facts	true, false, unknown
First-order logic	Facts, objects, relations	true, false, unknown
Temporal Logic	Facts, objects, relations, time	true, false, unknown
Fuzzy Logic	Facts with degree of truth	value within an interval

2.2.4 Logic Programming

Logic programs are a subfield of first-order logic usually concerned with deductive inference. A Logic program is formed by *clauses* (first-order rules). The conclusion of a clause is named *head* and the condition is named *body of the clause*. Both the head and body of the clause are formed by *atoms*. An atom is a *predicate* applied to one or more *arguments* (terms) (DZEROSKI, 2007), as the below example:

$$\text{employs}(Y, X) \leftarrow \text{works}(X, Y) \wedge \neg \text{entrepreneur}(X)$$

The above expression can be read as: If X (a person) works in Y (a company) and X is not an entrepreneur, then Y employs X ($\wedge$ is the logical and, and $\neg$ is the logical not). In this example, $\text{works}(X, Y) \wedge \neg \text{entrepreneur}(X)$ is the body of the clause, and $\text{employs}(Y, X)$ is the head. The predicates of this clause are *employs*, *works* and *entrepreneur*. *X* and *Y* are variables and $\text{employs}(Y, X)$, $\text{works}(X, Y)$ and $\text{entrepreneur}(X)$ are atoms.

According to Dzeroski (2007), logic programs are formed by a set of program clauses. A predicate corresponds to a relation in a database and the arguments of a predicate are the attributes of this relation. A set of program clauses with the same predicate in the head are called *predicate definition*.

2.2.5 Inductive Logic Programming

Traditional data mining techniques are usually designed to discover patterns in a single table, and are often referred to as *propositional learning* approaches, since the discovered patterns can be expressed in propositional logic (DŽEROSKI; LAVRAČ, 2001).

Nonetheless, a typical relational database stores its data in multiple tables. To apply traditional data mining techniques on a relational database, one usually has to “flatten” the data into a single table. That process is usually done via joins or aggregation. However, except for specific scenarios, that process results in the loss of useful information about the domain (WROBEL, 2001).

Situated in the intersection of logic programming and machine learning, *Inductive Logic Programming* (ILP) is concerned with inductive inference, and focus on techniques for *multi-relational data mining* (MRDM). ILP allows one to discover patterns in multiple tables and to represent previous knowledge about the domain of study in the form of a logic program. ILP can be also referred to as *first-order learning*, for the discovered patterns can be expressed with the formalism first-order logic (DZEROSKI, 2007).

After the specification of the background knowledge of the domain of study, ILP can find a hypothesis about unseen instances through the generalization of individual observations.

Inductive concept learning is the task of finding attributes and relations that can distinguish objects as belonging to a category or not. It is one of the most basic tasks in machine learning (RAEDT, 1997).

According to De Raedt (1997), in the concept learning problem, one needs to define the representation language for examples (L_e) and for the concepts (L_C), as well as the coverage relation between them $\in_C$, or, as Dzeroski denotes it in (DZEROSKI, 2007), $\text{covers}(C, e)$.

De Raedt (1997) presents three setting definitions for concept learning: *learning from entailment*; *learning from interpretations*; and *learning from satisfiability*.

Learning from entailment was introduced by (MUGGLETON, 1991), and can be formally described as: given a set of examples E , P are the positive examples, and N are the negative examples and B denotes the background knowledge. $E = P \cup N$, and the ILP task of relational rule induction is to find a hypothesis H where $\forall e \in P : B \wedge H \models e$ and $\forall e \in N : B \wedge H \not\models e$ ($\models$ indicates logical entailment) (DZEROSKI, 2007).

Learning from interpretations represents an upgrade on the boolean concept-learning in computational learning theory. *Interpretations* carry much more information while, when learning from entailment, an example can consist of a *single* fact. Conversely, when learning from interpretations, the facts that holds in the example are all known. (RAEDT; KERSTING, 2008). In learning from interpretations, the example e is a Herbrand interpretation¹ and it has to be a model of the hypothesis H , which is a clausal theory (DZEROSKI, 2007).

In Learning from satisfiability the example e and the hypothesis H taken together have to be satisfiable (H and e are both clausal theories). Then, H covers e under satisfiability, $e \in_S H$, if and only if $H \wedge e \not\models \perp$ (RAEDT, 1997). Nonetheless, the setting of learning from satisfiability is computationally complex and has rarely been used in practice (DZEROSKI, 2007).

One of the first approaches to relational learning problems was transforming it into propositional learning problems (LAVRAČ; DŽEROSKI; GROBELNIK, 1991). That can be done via joins or aggregations. Background knowledge is required to avoid losing of meaning and information after performing such operations, for the hypothesis language of propositional learning methods lack expressiveness (DŽEROSKI; MUGGLETON; RUSSELL, 1992).

One of the biggest advantages of addressing ILP problems with this approach, is that most of the existing data-mining algorithms are designed to work on propositional learning problems.

¹ Herbrand interpretations are sets of true ground facts and they describe a possible situation (RAEDT; KERSTING, 2008).

2.3 Multi-Relational Data Mining

Most large data collections are stored in relational database systems consisting of multiple tables; however, most data mining techniques are only applicable to a single table. Multi-relational data mining (MRDM) focuses on the search for patterns across multiple tables (relations) of a database (DŽEROSKI, 2003), and many techniques have been proposed in that context. Indeed, relational models lead to a deeper understanding of the relations held within domains, and can be used for exploratory analysis, predictions and complex inferences.

Propositional learning approaches, which deal with data stored in a single table, are stated in a less expressive language than relational learning approaches, that are usually expressed in first-order logic.

Many MRDM algorithms are derived from the field of inductive logic programming, and have advanced to data mining tasks, such as classification, regression and clustering. Just like in ILP problems, relations in the domain of discourse that can be intentionally defined are called background knowledge.

MRDM algorithms have much in common with traditional propositional learning algorithms, since they search for patterns in data. The main difference, however, is the representation of the data and the patterns (DŽEROSKI, 2003). Furthermore, extending a propositional learning algorithm into a relational learning algorithm is a nontrivial task (LAER; RAEDT, 2001), requiring considerable insight and creativity.

An alternative MRDM approach is to systematically create a single table from a relational database, as proposed by (KRAMER; LAVRAČ; FLACH, 2001). That approach is called *propositionalization*.

The three main MRDM approaches are: relational association rules; relational decision trees; and relational distance-based approaches.

2.3.1 Relational Association Rules

The task of relational association rules is to discover frequent patterns and association rules in a database. Given a relational database R , with a schema $\mathbf{R}$ and a instance $\mathbf{I}$, a *conjunctive query* Q over $\mathbf{R}$ consists of a *head* and a *body*. The body is a set of atomic formulas over $\mathbf{R}$ and the head is a *tuple* of variables occurring in the body (GOETHALS; BUSSCHE, 2002).

Then, Goethals (2002) defines the *support* of a conjunctive query Q , denoted f_Q , by the number of distinct tuples in the answer of Q on $\mathbf{I}$. A query is called frequent in $\mathbf{I}$ if its support exceeds a predetermined support threshold.

Figure 1 – A relational decision tree

A relational decision tree, predicting the class variable A in the target predicate $entrepreneur(P, A)$.

Finally, an *association rule* have the form $Q_1 \rightarrow Q_2$, such that $Q_2 \subseteq Q_1$. A *frequent* association rule in $\mathbf{I}$ is one where Q_2 is also frequent in $\mathbf{I}$. And an association rule is called *confident* if the f_{Q_2}/f_{Q_1} exceeds a predetermined support threshold (GOETHALS; BUSSCHE, 2002).

In an intuitive example presented by Džeroski (2007), an association rule of the form $A \rightarrow C$, where A and C are conjunctive queries, can be interpreted as “customers that buy A typically also buy C .” So, if itemsets A and C have *supports* f_A and f_C , the confidence of the association rule is defined as $c_{A \rightarrow C} = f_C/f_A$. The goal is to find all association rules $A \rightarrow C$ where f_C and $c_{A \rightarrow C}$ exceed a predetermined threshold (DZEROSKI, 2007).

2.3.2 Relational Decision Trees

In the field of data mining, the prediction task using *Decision trees* has been widely studied. Therefore, extending this approach from a propositional to a relational setting is of major importance.

The prediction task can be formalized as a two-argument predicate, where the first argument is an example ID, and the second argument is a class variable, and a set of background knowledge.

As shown in Figure 1, a relational decision tree has the same structure of its propositional form. They are binary, every internal node has a left (yes) and a right (no) branch, and the leaves correspond to predicted classes.

The major difference between the relational and propositional decision tree is in the tests that appear in the internal nodes. That means that, while in a propositional decision

tree, one can express every leaf in one single clause, in a relational decision tree, one has to express it in the form of first-order decision lists (ordered sets of clauses) (DŽEROSKI, 2003).

2.3.3 Relational Distance-Based Approaches

The first step towards learning a relational distance-based model is to extend the concept of distance measure from propositional to the first-order case. Then, the distance measure can be used with typical statistical approaches.

Propositional distance measures are computed between vectors of attributes. (DŽEROSKI, 2003) states that a distance between two example vectors can be computed as:

$$distance(x, y) = \sum_{i=1}^n difference(x_i, y_i) / n. \quad (2.2)$$

where:

$$difference(x_i, y_i) = \begin{cases} |x_i - y_i| & \text{if continuous,} \\ 0 & \text{if discrete and } x_i = y_i, \\ 1 & \text{otherwise.} \end{cases} \quad (2.3)$$

However, in a relational case, an instance cannot be described as a vector, but as set of facts about multiple relations. A relational distance measure, that extends the propositional form, was proposed by (EMDE; WETTSCHERECK, 1996), in their RIBL system.

On the RIBL system, to compute the distance between examples, first the distance between the attributes of the two objects (depth 0) is computed. Then, the distance between objects immediately related to those objects (depth 1) is computed. And then, the distance between objects related to objects at depth 1 (depth2), and so forth (EMDE; WETTSCHERECK, 1996).

Once the relational measure is computed, standard statistical methods can be adapted for prediction or clustering, such as the *nearest-neighbor* or *agglomerative clustering* (DŽEROSKI, 2003).

2.4 Bayesian Networks

Representing the joint distribution P over a set of n random variables is a non-trivial task. Even in cases where all the variables are binary, the joint distribution will require the computation of $2^n - 1$ probabilities. Not only such task can be computationally

Figure 2 – Citizen Bayesian network $\mathcal{B}^{citizen}$ with CPDs.

expensive, but the results might be hard for a human to interpret. Also, learning such distribution from data would require a fairly large dataset to accurately estimate the parameters.

Bayesian networks model independence properties to represent high-dimensional domains in a compact way, lessening the above difficulties.

A Bayesian network is a directed acyclic graph (DAG) $\mathcal{G}$ representing the dependencies between a set of random variables $\mathcal{X} = \{X_1, \dots, X_n\}$. On a Bayesian network, the nodes of the graph correspond to the random variables of the domain of discourse, while the edges of the graph indicate the direct influence of one variable over another (KOLLER; FRIEDMAN, 2009), as shown in Figure 2.

In a Bayesian network, the variables that exert direct influence over a node X_i are denoted the parents of X_i , or $pa[X_i]$. It is assumed that, any variable X_i in a Bayesian network is independent of any other variable, given the values of its parents $pa[X_i]$.

Considering the example displayed in Figure 2. A citizen *Social Class* (S) depends on his *income* (I) and on whether he is an entrepreneur or not (E). While the income of a given citizen I is also dependent on being an *entrepreneur* (E). On this example, the variable S has domain $Val(S) = \{A, B\}$, while $Val(I) = \{high, low\}$ and $Val(E) = \{True, False\}$.

The set of *local probability models* represents the dependence between a variable and its parents. In the citizen example, $P(E)$ represents the distribution in the population of people who own a business versus people that do not own a business. $P(I|E)$ and $P(S|I, E)$ are conditional probability distributions, calculated using the Bayes formula. The former specifies the distribution over the citizen's income as it depends on E , while the latter represents the distribution over the citizen's social class as much as it depends on I and E .

2.4.1 Scoring functions

Given a dataset D , containing a set of observations of the variables of a Bayesian Network, a score function can be used to ascertain the performance of the model.

Several score functions are available, either for categorical, or continuous data. For categorical data, some of the most common score functions are:

1. Log-likelihood (LL);
2. Akaike Information Criterion (AIC);
3. Bayesian Information Criterion (BIC);
4. Dirichlet posterior density based scores;
5. K2 score;

Of the above listed score functions, the Log-Likelihood (LL) and the Bayesian Information Criterion (BIC) are the most popular, for they can be computed locally and then be summed up for all subgraphs.

2.4.2 Parameter Learning

Considering a scenario where the DAG structure is given, one has to learn the conditional probability distribution for every variable X_i , given its parents $pa[X_i]$.

Given a set of observations X , if there is no missing information, the *maximum likelihood estimation* is a method based on the likelihood function $\mathcal{L}(\theta; X)$ that aims to estimate the parameters θ of an statistical model that maximize the likelihood function.

Nonetheless, since the local probability distributions of Bayesian networks are expressed by flat tables, the likelihood expression is given by a parameter denoted θ_{ijk} . It represents, given a variable x_i , the probability of observing all possible values x_{ik} given all possible configurations of its parents $pa[x_i] = w_{ij}$. In these cases, the method intuitively defines a maximum likelihood estimate (MLE) by counting on the occurrences of each family configuration in the training set:

$$\operatorname{argmax}_{\theta_{ijk}} L(B|D) = \frac{N_{ijk}}{N_{ik}}. \quad (2.4)$$

However, it is common to have examples in a dataset where not all of the variables are observed. Missing information in a dataset hinders the maximum likelihood estimation approach, since it cannot be directly computed. The *Expectation maximization* (EM) algorithm maximizes the *expected likelihood* given an incomplete dataset.

The EM algorithm has two steps. The first step consists in finding the expected Log-Likelihood, which can be done by computing the probability of observing each of the

datasets one could obtain by completing the values of missing variables. The second step is the maximization of the expected Log-Likelihood (DARWICHE, 2009). This cycle is iterated until the expected Log-Likelihood reaches a local maximum.

2.4.3 Structure Learning

There are several algorithms that can be applied to learn a Bayesian network structure. These algorithms can be classified within three different approaches: constraint based algorithms; search-and-score algorithms; and hybrid models.

Constraint based algorithms try to identify variables that are independent from each other given a set of other variables, via independence tests on the dataset. Then, the algorithm search for a structure that satisfies as many independence constraints as possible. Some popular constraint based algorithms are:

1. Grow-Shrink (GS);
2. Incremental Association Markov Blanket (IAMB);
3. Fast Incremental Association (Fast-IAMB);
4. Max-Min Parents & Children (MMPC).

Search-and-score algorithms uses a pre-determined scoring function to evaluate the fit of candidate structures. The parameters for candidate structures are learned, and they are all scored. The structure with the highest score is elected the final model (KOSKI; NOBLE, 2012). Two commonly used search-and-score algorithms are:

1. Hill Climbing (HC);
2. Tabu Search (Tabu);

Hybrid models combines both constraint-based and search-and-score algorithms. The *Max-Min Hill Climbing* (MMHC) is an example of a commonly used hybrid model for Bayesian network structure learning.

2.5 Probabilistic Relational Models

Probabilistic Relational Models (PRM) are an extension of Bayesian Networks, introducing the concepts of objects and their properties, and the relations held between them, specifying a template for a probability distribution (GETOOR; TASKAR, 2007). This section offers a review of PRMs, their semantics, the challenges in learning them, and an overview of a PRM structure learning algorithm.

A relational domain is usually represented by distinct tables in a database containing attributes and entities. In this section, the Atibaia city domain, which is the domain of

our case study, will be used as a running example. The city of Atibaia is a relatively small town in the state of São Paulo, Brazil. Suppose we are interested in information about its citizens, its companies and its census sectors (territorial units, in which the city is geographically divided), and that we have three different tables, each representing one of the three different *classes* in our domain, **Person**, **Company** and **Census_Sector**. As we indicate later, the goal of our case study is to learn a PRM structure and parameters with classes in our domain, so as to predict the social class of people in Atibaia.

2.5.1 The PRM semantics

The vocabulary of a relational model consists of a set of *classes* $X_1, \dots, X_n$, and a set of *relations* $R_1, \dots, R_m$. Every class in the domain has a set of *attributes* $\mathcal{A}(X_i)$, and every attribute $A_j \in \mathcal{A}(X_i)$ has a space of possible values $V(A_j)$, being that this vocabulary defines a *schema* for the relational model. One attribute A of a class X , is referred to as $X.A$. If $X.A$ has a value that is fully determined, such as a name or and identification number, it is labeled as a *fixed* attribute. The other attributes are *probabilistic* ones (FRIEDMAN et al., 1999).

The logical description of the domain is called *relational schema*, and it shows how different classes relate to each other, through what is called *reference slots*. The set of reference slots of X is denoted $R(X)$, while $X.\rho$ is used to refer to the reference slot ρ of X . A reference slot ρ in X can be interpreted as an attribute of X that is also a foreign key for another class. And an *instance* I of a schema is an interpretation of the relations between the classes in the domain, and it specifies, for a set of objects, a class, and a value for each of its attributes. Figure 3a shows the schema for the Atibaia domain.

A central concept in a relational model is the *relational skeleton* σ , defined as a partial specification of an instance of a schema [Friedman et al. 1999]. It specifies for a set of *objects* $O^\sigma(X_i)$, a class, the value of the fixed attributes within this objects, and the relations held between them, leaving only the probabilistic attributes unspecified. Figure 3b represents an example of a relational skeleton for our Atibaia domain.

In the domain of study, the **Person** class has, among others, the attribute *Social_class*, and the value space for **Person.Social_class** is $\{A, B, C, D\}$. Also, the **Person.Census_Sector_ID** attribute is a reference slot of the **Person** class, with *range type* **Census_Sector**. The same applies for the class **Company**. Every object in **Census_Sector** is associated with n objects in **Person**, that are the citizens who live in the sector, and m objects in **Company**, that are the companies located inside the area comprised by the census sector.

A PRM models the uncertainty over the possible values of the probabilistic attributes of the skeleton. The model itself, consists of a dependency structure $\mathcal{S}$, and the

Figure 3 – The relational schema and the relational skeleton for the Atibaia domain.

(a) The relational schema for the Atibaia domain. (b) The relational skeleton for the Atibaia domain.

conditional probability distribution θ_S associated with the dependency structure. Just like a Bayesian network, the dependency structure of a PRM is defined by associating a set of parents $pa(X.A)$ with each attribute $X.A$. However, in a PRM, an attribute $X.A$ can have as parent, either an intra-class attribute, denoted $X.B$, or an inter-class attribute, denoted as $X.\tau.B$, where τ is a *slot chain*, representing the set of objects that are τ -relatives of an object $x \in X$. When an object x relates with an object of another class through a slot chain, unless the relation is guaranteed to be singled-valued, $x.a$ depends on a set of objects $X.\tau.B$, and, when that occurs, a method for representing these complex dependencies is necessary. This is attained using an *aggregation* function.

Mainly, when more then one object of a class relates to a single object from another class through one of its attributes, the aggregation of this attribute is an appropriate tool to address this issue.

Some notable aggregation functions are the *mode*, if the attribute is categorical, the *mean*, if the attribute is continuous, the *median*, *maximum* or *minimum*, if the attribute is ordered, or the *cardinality*. By using an aggregation function γ , it is possible to return a summary of a multiset of values, allowing an attribute $X.A$ to have $\gamma(X.\tau.B)$ as a parent. (GETOOR; TASKAR, 2007).

For our Atibaia city domain, a legal PRM structure, with the required aggregations indicated can be seen in Figure 6. Note that, since objects from either class **Company** and **Person** can only be associated with one object from the **Census_sector** class, no aggregations are required when an attribute on **Census_sector** is the parent of an attribute from either **Company** or **Person**.

2.5.2 PRM structure learning

Learning a PRM from data can be quite difficult. The most imposing challenge is to learn the dependency structure. First, making sure that all the dependencies established are acyclic is a nontrivial task. The best scenario is when the skeleton σ is acyclic. However, in some cases, if there is a cycle in the skeleton σ , one needs to guarantee that cycles do not happen at an object level. Also, evaluating and scoring concurrent structures is also challenging; the usual approach is to adapt Bayesian model selection.

The dependency structure of the attributes is assumed to be independent of the skeleton σ . However, the use of an uniform prior over candidate structures does not work well, since the amount of possible structures might be very large.

And finally, even after choosing a method to assure that a structure is legal and score it, one must go over candidate models, comparing them. The number of possible structures may be quite large, and the cost of the search operations is also high, making it necessary to associate the search method with an heuristic approach, usually by associating a limited set of possible parents for each attribute.

Both (KOLLER, 1999) and (FRIEDMAN et al., 1999) address this complicated task and propose a method for learning a PRM structure from data, instead of having it given by the user. This section will describe their method.

2.5.2.1 Legal structures

First, it is of paramount importance that the chosen dependency structure $\mathcal{S}$ is indeed a coherent probability model. Koller (1999) states that this property can be satisfied by ensuring that the **skeleton σ is acyclic relative to $\mathcal{S}$** .

Given a structure $\mathcal{S}$, one could easily verify if it is indeed acyclic relative to the skeleton σ . However, the problem relies in providing a method that ensures the fulfillment of this property in an arbitrary database.

The approach proposed by Friedman (1999) is to guarantee that the dependency graph is *stratified*, and therefore, respects some order. An attribute $X.A$ directly depends on an attribute $Y.B$ if either $X = Y$ and $X.B$ is a parent of $X.A$ ($X.B \rightarrow X.A$), or $\gamma(X.\tau.B)$ is a parent of $X.A$ and the τ -relatives of X are of class Y . So, it is required that $X.A$ directly depends **only** on attributes that precede it in the order. Then, given a stratified graph, it is possible to verify if introduction of a new edge on the graph will result in a cycle ($Y.B \rightarrow X.A \rightarrow \dots \rightarrow Y.B$) (GETOOR; TASKAR, 2007).

This simple approach may ensure acyclicity, however it doesn't cover many important cases. In such cases, the cyclicity at the attribute level is resolved at an object level. Nonetheless, to resolve such cases one must rely on prior knowledge about the domain of discourse. The algorithm proposed by (FRIEDMAN et al., 1999) allows the user to

provide this prior knowledge and use it to determine the legality of certain dependency structures.

Three kinds of dependency are defined and, for each of them, a color is assigned:

Yellow edges A yellow edge $X.B \rightarrow X.A$, when $X.B$ is a parent of $X.A$.

Green edges A green edge $Y.B \rightarrow X.A$, when $\gamma(X.\tau.B)$ is a parent of $X.A$ and τ is ensured to be acyclic.

Red edges A red edge $Y.B \rightarrow X.A$, when $\gamma(X.\tau.B)$ is a parent of $X.A$ and τ is not guaranteed to be acyclic.

A dependency graph is guaranteed to be stratified if any eventual cycle within the structure contains at least one green edge and no red edges. When no prior knowledge is given to ensure acyclicity at an object level, the graph will have only yellow and red edges, therefore, it can only be stratified if, and only if, it is acyclic. The PRM shown in Figure 6 has only yellow and green arrows, therefore, it is a legal structure.

2.5.2.2 Evaluating different structures

With the space of legal structures defined, one needs to evaluate the different structures and choose the one that better fits the data. The method proposed by (FRIEDMAN et al., 1999) adapts *Bayesian model selection* to compute a posterior probability of a structure $\mathcal{S}$ given an instantiation $\mathcal{I}$. Using Bayes rule:

$$P(\mathcal{S}|\mathcal{I}, \sigma) \propto P(\mathcal{I}|\mathcal{S}, \sigma)P(\mathcal{S}|\sigma) \quad (2.5)$$

It is assumed that the choice of the structure is independent of the skeleton, therefore, $P(\mathcal{S}|\sigma) = P(\mathcal{S})$. Using a simple uniform prior over the possible dependency structures works well for Bayesian networks. However, in the context of PRM there might be cases where the number of possible structures can grow infinitely, even when only a finite number are worthy considering, given a skeleton (GETOOR; TASKAR, 2007). The solution found was to penalize long indirect slot chains, by having $\log P(\mathcal{S})$ proportional to the sum of the chains τ in $\mathcal{S}$.

The other component, $P(\mathcal{I}|\mathcal{S}, \sigma)$, is the *marginal likelihood*:

$$P(\mathcal{I}|\mathcal{S}, \sigma) = \int P(\mathcal{I}|\mathcal{S}, \theta_{\mathcal{S}}, \sigma)P(\theta_{\mathcal{S}}|\mathcal{S})d\theta_{\mathcal{S}} \quad (2.6)$$

Using the parameter-independent Dirichlet prior, the integral has a simple closed-form solution. Getoor proposes that, if $\mathcal{I}$ is a complete assignment, and $P(\theta_{\mathcal{S}}|\mathcal{S})$ is a

Dirichlet with hyperparameters $\alpha_{X.A}[v, \mathbf{u}]$ and satisfies parameter independence, $P(\mathcal{I}|\mathcal{S}, \sigma)$ is:

$$P(\mathcal{I}|\mathcal{S}, \sigma) = \prod_i \prod_{A \in \mathcal{A}(X_i)} \prod_{\mathbf{u} \in V(Pa(X_i.A))} DM(\{C_{X_i.A}[v, \mathbf{u}]\}, \{\alpha_{X_i.A}[v, \mathbf{u}]\}) \quad (2.7)$$

where:

$$DM(\{C[v], \{\alpha[v]\}\}) = \frac{\Gamma(\sum_v \alpha[v])}{\Gamma(\sum_v (\alpha[v] + C[v]))} \prod_v \frac{\Gamma(\alpha[v] + C[v])}{\Gamma(\alpha[v])}$$

and:

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt \text{ is the Gamma function.}$$

As it can be seen, the marginal likelihood, which is computed as a product of simple terms, is the dominant term in the probability of a structure, balancing its complexity with its fit to the data (FRIEDMAN et al., 1999).

2.5.2.3 Structure search

Finally, it is necessary to define a procedure to search the space of legal structures and score them. For Bayesian networks this is a NP-Hard task (CHICKERING, 1996). Since learning a PRM is a task that is at least as hard as learning a Bayesian network (for a Bayesian network is a PRM with only one class of objects and no relations) (GETOOR; TASKAR, 2007), this method does attempt to find the highest-scoring structure, and resorts to a heuristic search procedure.

A typical approach to Bayesian networks is to use *greedy local search* to modify a candidate structure, by adding, deleting or reversing an edge, and, at each iteration, score the structure and choose the one with the highest score (HECKERMAN, 1998).

The chosen algorithm was the *greedy hill-climbing search*, using the score proposed in Section 2.5.2.2, and dealing with local maxima using random restarts. Just like in a Bayesian network, the decomposability of the score has great impact in the search algorithm.

Still, a given skeleton σ may have an infinity number of possible structures, and the search procedure is still very expensive. So, adopting an heuristic approach to reduce the space of possible structures helps lessening the cost of the search step (FRIEDMAN et al., 1999).

The heuristic adopted in the method proposed by (KOLLER, 1999) and (FRIEDMAN et al., 1999) is the following: at each phase k a set of potential parents $Pot_k(X.A)$ is assigned to every attribute $X.A$. The search procedure is restricted to the space of structures where the parents of every attribute $X.A$ are in $Pot_k(X.A)$.

Adopting this procedure allows one to precompute the most expensive operations, which are the joins and aggregations required in the definition of the parents. However, the success of this approach relies on the choice of the potential parents $Pot_k(X.A)$.

The approach is iterative, it starts with an initial structure and selects the sets of $Pot_k(X.A)$ for every $X.A$ based on this initial structure. Then, the search procedure is applied to obtain a new higher-scoring structure. A new set of parents $Pot_k(X.A)$ is chosen and reiterate. It stops when no improvement is made (FRIEDMAN et al., 1999).

3 Vanilla-PRM: A novel method for PRM learning

In Section 2.4 it was shown that Bayesian networks have been used in a large variety of real-world and research problems. Nonetheless, a Bayesian network lacks the ability to model complex domains stored, for instance, in a relational database.

That limitation is given to the fact that a Bayesian network lacks the concept of objects and classes and, therefore, it cannot represent general principles about related objects within a domain, which can then, be applied to multiple contexts (GETOOR; TASKAR, 2007).

Probabilistic Relational Models (PRMs) are the combination of a frame-based logical representation and Bayesian networks probabilistic semantics. Hence, just like a regular Bayesian network, a PRM also specifies a probability distribution for a set of random variables, in a sense that an attribute $X.A$ is only directly dependent on a small set of parents $Pa(X.A)$.

The attributes of the objects in a skeleton σ are analogous to the random variables in a Bayesian network, and for a given skeleton σ , the PRM structure induces a ground Bayesian network over the attributes $x.A$ (GETOOR, 2001).

Every attribute $x.A$ of every object x is represented by a node in the graph, but once again, just like a regular Bayesian network, the CPD for $x.a$ is $P(X.A|Pa(X.A))$, implying that the parameters of the local probability models for attributes of objects in the same class are identical (GETOOR, 2001).

However, an attribute $x.A$ can be influenced by either intra-class attributes $x.B$, or inter-class attributes $x.K.B$. If K is not single-valued, then an aggregation function is required to return a summary of the set of random variables.

As described by Getoor 2001, in the same way as with Bayesian networks, the joint distribution over these assignments can be factored by taking the product, over all $x.A$, of the probability in the CPD of the specific value assigned by the instance to the attribute given the values assigned to its parents. The formal expression can be written as follows:

$$P(\mathcal{I}|\sigma, \mathcal{S}, \theta_{\mathcal{S}}) = \prod_{x \in \sigma} \prod_{A \in \mathcal{A}(x)} P(\mathcal{I}_{x.A} | \mathcal{I}_{Pa(x.A)}) = \prod_{X_i} \prod_{A \in \mathcal{A}(x)} \prod_{x \in \sigma(X_i)} P(\mathcal{I}_{x.A} | \mathcal{I}_{Pa(x.A)}). \quad (3.1)$$

Nonetheless, learning a PRM from relational data is a more complex task than learning a Bayesian Network from propositional data. On Section 2.5.2 the three main

difficulties of learning a PRM were presented: establishing what are the legal dependency structures for a given domain; score the possible legal structures; and search the space of possible structures.

In face of these three challenges presented, the goal of this research is to propose a method for searching for a PRM structure using some score typically used to learn Bayesian networks, which are easier to implement, with many open source libraries available.

Contrary to the method presented in Section 2.5.2, the novel proposed method relies on an heuristic approach that adjusts the problem to the existing Bayesian network structure learning techniques.

As noted already, the search space is too big, so the process is quite complex and costly in practice. The proposed approach relies in restricting the space of possible structures in ways that make sense for practical problems.

The golden rule in Bayesian networks and in PRMs is that an attribute cannot be an ancestor of itself, that is, the probabilistic dependencies must be acyclic. For that, two graphs are considered: the *instance dependency graph* G_σ , that encompasses the dependencies between attributes of objects in the skeleton; and the *class dependency graph* G_π , that considers potential dependencies on a class level.

In a PRM it is possible to have cycles at a class level, however, those cycles must be solved on an object level in the form of additional domain constraints. As introduced in Section 2.5.2, on such scenarios, the dependency graph is only stratified if every cycle has at least one *green edge* and no red edges, otherwise, the dependency structure won't be a legal one. On those cases, assuring that the cyclicity is resolved on an object level requires prior knowledge about the domain of study.

3.1 Vanilla-PRM assumptions

Since search space of possible structures is too big, an alternative approach to learning PRMs is proposed. Four assumptions are proposed, restricting the space of possible structures in ways that make sense for practical problems, allowing the adoption of methods typically used to learn Bayesian networks.

The first assumption of the method, that restricts the space of possible structures, is that the dependency structure $\mathcal{S}$ is not allowed to have cycles, not only at an attribute level, on the *instance dependency graph* G_σ , but at class level as well, on the *class dependency graph* G_π .

Assumption 1: the *class dependency graph* G_π is not allowed to have cycles.

Even with the possibility of solving an eventual cyclicity at an object level, allowing the *class dependency graph* G_π to have cycles could increase the space of possible structures

infinitely. Requiring the *class dependency graph* to be acyclic is the same as requiring a stratification between the attributes of distinct classes.

A second assumption is that the *instance dependency graph* G_σ is not allowed to have cycles, not only at the attribute level, but at class level as well. For instance, in a domain with three distinct classes of objects X , Y and Z , if there is at least one arrow in the *instance dependency graph* from an attribute $X.A \rightarrow Y.B$, no other attribute in class Y can parent any attribute in class X . Therefore, structures such as $X.A \rightarrow Y.B \rightarrow X.C$ and $X.A \rightarrow Y.B \rightarrow Z.C \rightarrow X.D$ are not allowed, even if it does not generate a cycle in the *instance dependency graph* at an attribute level.

Assumption 2: the *instance dependency graph* G_σ is not allowed to have cycles at a class level.

In practice, assumption 2 implies that the *instance dependency graph* will be stratified not only at an attribute level, but also in a class level. Figure 4a illustrates a scenario in the Atibaia domain where the inter-class dependencies, represented by green edges, do not result in cycles, neither at an attribute level, nor at a class level, respecting the premises of assumption 2.

Figure 4b shows a PRM whose inter-class dependencies, represented by red edges, violates assumption 2. It can be noticed that there is a cycle at a class level, even though the red edges do not induce any cycles at an attribute level.

A third assumption, is that in practical cases, when implementing a PRM over a domain, one might have a subset of *attributes of interest* for prediction, that usually belongs to a single distinguished class, that is referred to as the *leaf class*.

Assumption 3: given the attribute(s) of interest, the user must specify the *leaf class* beforehand.

The leaf class is the class that holds the attributes of interest. In a Bayesian network, a node that does not parent any other variable is called a *leaf node*. Similarly, attributes in the leaf class cannot parent any attribute from a distinct class.

Assumption 4: attributes on the leaf class are not allowed to parent attributes from distinct classes.

In short, the four assumptions presented in this Section allows one to learn a PRM with: no cycles on the *class dependency graph* G_π ; a *instance dependency graph* G_σ stratified at attribute and class levels; a leaf class, that holds attributes of interest for prediction; no arrows leaving the leaf class.

Figure 4 – An illustration of Vanilla-PRM’s *assumption 2*: the *instance dependency graph* G_σ is not allowed to have cycles at a class level.

- (a) The three inter-class dependencies (represented by the green edges) do not result in a cycle, neither at an attribute level, nor in at a class level, thus respecting assumption 2.

- (b) The three inter-class dependencies (represented by the red edges) do not result in a cycle at an attribute level. However, a cycle is formed at a class level, thus violating assumption 2.

3.2 Aggregations

In a PRM, an attribute $X.A$ can depend on either intra-class attributes of type $X.B$, or inter-class attributes of type $Y.B$. A dependency between attributes of distinct classes have form $Y.B \rightarrow X.A$. In such cases, if the slot chain is not guaranteed to be single-valued, an object $x \in X$ can be linked to a set of objects $\{y_1, \dots, y_i\} \in Y$. Hence, $X.A$ can only be parented by $Y.B$ by using an aggregation function γ .

Given this property of PRMs, we define four types of dependency between attributes of distinct classes.

$1 \rightarrow 1$: When every object $x \in X$ is linked to only one object $y \in Y$, if there is an dependency between attributes of the kind $Y.B \rightarrow X.A$, the slot chain is single-valued and the use of an aggregation function γ is not necessary.

$1 \rightarrow n$: When sets of objects $\{x_1, \dots, x_i\} \in X$ are linked to only one object $y \in Y$, if there is an dependency between attributes of the kind $Y.B \rightarrow X.A$, the slot chain is single-valued and the use of an aggregation function γ is not necessary.

$n \rightarrow 1$: When every object $x \in X$ is linked to a set of objects $\{y_1, \dots, y_i\} \in Y$, if there is an dependency between attributes of the kind $Y.B \rightarrow X.A$, the slot chain is not single-valued and the use of an aggregation function γ is necessary.

$n \rightarrow n$: When sets of objects $\{x_1, \dots, x_i\} \in X$ are linked to a set of objects $\{y_1, \dots, y_i\} \in Y$, if there is an dependency between attributes of the kind $Y.B \rightarrow X.A$, the slot chain is not single-valued and the use of an aggregation function γ is necessary.

Figure 5a shows a slice of the dataset from **Person** class. Each row on this table represents an object in that class, in this case, a citizen from the city of Atibaia. The complete dataset have 110.823 rows, one for each distinct object in the **Person** class.

One of the attributes of objects in that class is the **Person.Sector_ID**. This attribute indicates the ID of the *census sector* where a person lives. A census sector is a small territorial unit, and the city of Atibaia is formed by 327 census sectors.

Figure 5b shows a slice of the dataset from **Census_sector**. Once again, each row on this table represents an object in that class (a census sector in the city of Atibaia). This complete dataset have 327 rows, one for each distinct census sector in the city.

As it can be intuitively noticed, the attribute **Person.Sector_ID** is a *foreign key* establishing relations between objects from class **Person** and **Census_sector**. The rows in green on Figure 5 highlight four distinct objects on class **Person** that relate to the same object on class **Census_sector**.

Figure 5 – Slices of the **Person** and **Census_sector** datasets.

Person_ID	Census_sector_ID	Age_range	Gender	Social_class
28364817	220	59 to 65	Female	B
24120750	69	54 to 58	Male	B
72000641	236	59 to 65	Male	A
12290081	152	> 65	Male	D
42850987	220	59 to 65	Female	D
12732263	50	59 to 65	Female	C
33500361	116	59 to 65	Male	D
264720448	50	44 to 48	Female	D
41493446	212	54 to 58	Female	B
288758008	28	34 to 38	Male	B
19667172	57	59 to 65	Male	B
37586419	220	54 to 58	Male	D
289906653	220	39 to 43	Male	D
253115139	116	59 to 65	Male	D
28742065	47	54 to 58	Male	A
265135620	69	44 to 48	Female	D
42646058	50	59 to 65	Female	B
4081244	7	54 to 58	Female	B

(a)

Cod_setor	Avrg_HH	Avrg_income	Pct_houses	Families
220	< 3.0	>= 1250	51% - 75%	101 - 200
69	< 3.0	>= 1250	51% - 75%	101 - 200
236	< 3.0	>= 1250	76% - 100%	101 - 200
152	< 3.0	>= 1250	76% - 100%	101 - 200
50	< 3.0	>= 1250	76% - 100%	101 - 200
116	>= 3.5	500 - 750	76% - 100%	1 - 100
28	< 3.0	>= 1250	51% - 75%	101 - 200
212	< 3.0	>= 1250	76% - 100%	101 - 200
57	< 3.0	750 - 1250	51% - 75%	101 - 200
47	< 3.0	>= 1250	51% - 75%	101 - 200
7	< 3.0	>= 1250	76% - 100%	101 - 200

(b)

(a) a slice of the dataset containing objects from class **Person**; (b) a slice of the dataset containing objects from class **Census_sector**.

Given a dependency structure $\mathcal{S}$, if there is an arrow indicating a dependency from one attribute of **Census_sector** to an attribute of **Person**, then a $1 \rightarrow n$ relation is established.

That is a result of the fact that each object in **Census_sector** might be related to n distinct objects from class **Person**, with $n \geq 1$. These are the citizens that live inside the area comprised by that census sector. Nonetheless, distinct census sectors cannot be related to a same object on class **Person**, for a citizen is assigned to only one *Sector_ID*, that indicates the census sector where that person lives.

Inversely, if there is an arrow indicating a dependency from one attribute of **Person** to an attribute of **Census_sector** (eg. **Person.B** $\rightarrow$ **Census_sector.A**), then a $n \rightarrow 1$ relation is established.

On a previous version of the learning method, presented on the *Symposium on Knowledge Discovery, Mining and Learning*, KDMiLe 2017 (MORMILLE; COZMAN, 2017), a fifth assumption was made, reducing the space of possible structures even further by specifying that types $1 \rightarrow n$, $n \rightarrow 1$ and $n \rightarrow n$ dependencies could only exist if directed to the leaf class.

In practice, that meant that attributes from classes other than the leaf class, were not allowed to have as parent an inter-class attribute whose slot chain was not singled-valued.

That is to say, attributes in the leaf class could have an inter-class attribute as parent on every scenario, but attributes from other classes could only have inter-class parents when the relation is $1 \rightarrow 1$.

Figure 6 – A PRM structure learned using the proposed method considering a *fifth assumption*.

A PRM structure, learned using the proposed method still considering the *fifth assumption* presented in Section 3.2. The *yellow* arrows represent intra-class relations. The *light green* arrows represent inter-class relations where the slot chain is singled valued. And the *dark green* arrows represents the inter-class relations where an aggregation function was used (in our case, the mode). Source: (MORMILLE; COZMAN, 2017).

The first version of the method limited the space of possible structures to a group where the only aggregations performed were in the direction of the class that holds the variables of interest. The main advantage of this fifth assumption was that, given a relational skeleton σ , all the required aggregations for the restricted space of structures could be computed in advance and stored in a single table.

However, even though that assumption could greatly lessen the computational cost of the model, the option to maintain a broader space of possible structures was made, and the fifth assumption was excluded from the final version of the model.

A PRM, learned using the proposed method and still considering the fifth assumption can be seen on Figure 6 (MORMILLE; COZMAN, 2017). It is possible to observe on the resulting dependency structure over a reduced version of the Atibaia domain that the only inter-class arrows respect the restrictions imposed by the fifth assumption when **Person** is set as the leaf class, thus generating a network isomorphic to a legal PRM for the domain.

3.3 Vanilla-PRM algorithm

A PRM with only one class of objects and no relations is analogous to a Bayesian network; therefore, the Vanilla's method was designed to support domains with at least two classes of objects. In practice, every class in a domain of study is a table on a relational database. In this section, the terms *class* and *table* will be used interchangeably.

The algorithm takes all the tables in the domain and conducts a pairwise check, verifying the existence of relations between objects on distinct classes. For every relation discovered, the nature of such relation is also determined ¹.

Once all the relations and their respective natures are mapped, a data structure, called *master table* is computed for every class in the domain. The master table of a class X is denoted $Mt(X)$ and it starts as an exact copy of X 's table. Then, for every object $x_i \in X$, the attributes of its τ -relative objects, whose links are given by the value of the reference slot $X.\rho$, are aggregated using a proper aggregation function γ . The computed aggregation is returned to its respective row in the master table.

However, if the slot chain is guaranteed to be singled-valued, that is, if every object $x_i \in X$ relates to only one object of another class, the use of an aggregation function γ is unnecessary.

For instance, in the Atibaia domain, in the **Person** class master table, $Mt(Person)$, the attributes from objects in **Company** and **Census_sector** that are τ -relatives to objects in **Person** are aggregated (if necessary) and added to the master table.

Nevertheless, every object in the class **Person** is associated with one, and only one, object in the class **Census_sector**, meaning that the slot chain that links them is singled-valued and the use of an aggregation function γ is unnecessary in that particular case. The values of attributes on **Census_sector** are simply replicated on the rows where their corresponding keys appear in $Mt(Person)$.

Conversely, the slot chain that links **Company** and **Person** is not single-valued, and an aggregation function γ is required to return a summary of the attributes on **Company** to the master table. After all the aggregations are done, the assembly of $Mt(Person)$ is complete.

That is one of the main differences between Vanilla-PRM and the method presented in Section 2.5.2. In the former, all the aggregation operations are computed in advance, while in the latter, on every iteration, a new set of aggregations must be computed.

In possession of a master table for each class in a domain, the challenges of defining a proper structure scoring function, choosing a search method for possible legal structures, and comparing them via score, can be solved by traditional Bayesian network approaches,

¹ As specified in 3.2, there are four different natures of relation: $1 \rightarrow 1$; $1 \rightarrow n$; $n \rightarrow 1$; and $n \rightarrow n$.

and there are several popular algorithms, as presented in Section 2.4, that address these issues.

At this stage, the user must specify which of the classes in the domain is the *leaf class* (which holds the variables of interest). Herewith, a “class stratification” order can also be induced by the user.

Still, it is necessary to assure that the structure restrictions assumed in Section 3.1 are enforced during learning. To do so, a *black list* of edges is assembled for each class separately. For instance, the black list of arrows for class X is denoted $Bl(X)$. The black list role is to specify which edges are not allowed exist in the model, thus maintaining a coherent structure, given the method’s assumptions.²

Then, for each class $\{X_1, X_2, \dots, X_n\} \in \mathcal{D}$ a *local Bayesian network* structure, whose representation is a *directed acyclic graph* (DAG) describing a set of independencies (KOLLER; FRIEDMAN, 2009), can be learned from the classes respective master tables $\{Mt(X_1), Mt(X_2), \dots, Mt(X_n)\}$ considering the black lists of the respective classes $\{Bl(X_1), Bl(X_2), \dots, Bl(X_n)\}$. The structure learning of these local Bayesian networks can be conducted with any package of choice with the scoring function and search method of preference. A local Bayesian network learned from a master table $Mt(X)$, considering the black list of edges $Bl(X)$ is denoted $\mathcal{B}_X$.

At this stage, there is local Bayesian network $\mathcal{B}_{X_n}$ structure learned from the master table $Mt(X_n)$ for each class X_n in the domain $\mathcal{D}$. Hereupon, the structures of all the local Bayesian networks in the domain are combined, to form the final PRM structure.

The assumptions made in Section 3.1, ensured by the computed black lists of edges, guarantee that the combination of the local Bayesian networks will not result in any cycles whatsoever.

Finally, the PRM, just like a Bayesian network, is associated with a *conditional probability distribution* (CPD) or local probabilistic model. The CPD for X_i , given its parents $Pa(X_i)$ in the graph, is $P(X_i|Pa(X_i))$, and can be computed through equation 2.4. It captures the conditional probability of the random variable, given its parents in the graph (KOLLER et al., 2007).

It is understood that reducing the attention to a subset of PRMs that are sensible in such a scenario could result in the restriction of good candidates. However, a detailed implementation of the method, introduced in Chapter 4, shows that, even though this approach may not cover all possible cases, it can be useful whenever the *variables of interest* (for prediction) are in the same class of objects.

² Popular algorithms for Bayesian network learning typically allow for such lists of forbidden edges.

4 Experiments

In this chapter the Vanilla-PRM method is applied to a real large-scale problem. To do so, a package in the R language was implemented. After the details of the case study are presented, the freely available package is described.

4.1 The domain of study

The domain of the case study is a relatively small town named Atibaia, in the state of São Paulo, Brazil. According to the last estimate by the *Instituto Brasileiro de Geografia e Estatística* (IBGE), in 2017, the city population was of 139.683 people.

A city is often a very complex domain to model, for it comprises many different objects, of many distinguished classes. For this study, only three classes of objects were considered, representing the population of the city, the business located in Atibaia, and aspects of the census sectors that form the city territory.

The goal of the study is to apply the proposed Vanilla-PRM learning method to explore the relations between those classes, so as to create a model that uses their attributes to perform inference based on probability theory to draw conclusions about aspects of the population and the companies of Atibaia.

To that end, the data of the three distinct classes - namely **Person**, **Company** and **Census_sector** - was gathered from two different sources: *Serasa Experian*; and *IBGE*.

4.1.1 The *Person* class

The first class of the Atibaia domain is denoted **Person**. It is stored in a single table inside a relational database that represents the domain of study. On that table, each line represents an object of the class, and every column represents an attribute of the objects. Every object in **Person** (every line in the table) represents a citizen of Atibaia and its attributes.

The dataset for this class of objects was kindly provided by *Serasa Experian* in May 2017. It contains 110.823 objects and, by the time of its concession, comprised every citizen of Atibaia with at least 18 years of age registered at the Serasa Experian database.

All the objects in this class are fully anonymised, and no personally identifiable attribute was provided. Every object $p \in \mathbf{Person}$ is associated with 23 categorical attributes,

each indicating a distinct characteristics of that given citizen. Those 23 attributes are as follows:

- 1** - Census sector;
- 2** - Age group;
- 3** - Gender;
- 4** - Marital status;
- 5** - Public server flag;
- 6** - Education level;
- 7** - *Bolsa Família* flag;
- 8** - Company owner flag;
- 9** - Rural producer flag;
- 10** - Social class;
- 11** - Income group;
- 12** - Credit risk;
- 13** - Consumption index;
- 14** - Mosaic;
- 15** - Propensity for luxury;
- 16** - Propensity for tourism;
- 17** - Propensity for credit card;
- 18** - Propensity for e-commerce;
- 19** - Propensity for mortgage loan;
- 20** - Propensity for paid TV;
- 21** - Propensity for broadband internet;
- 22** - Propensity for personal loan;
- 23** - Propensity for postpaid phone.

A meticulous investigation of the dataset was conducted, leading to the application of data treatment and preprocessing techniques. This is practice of uttermost importance and should take place prior to applying any machine learning technique, for it can have significant impact on the generalization performance of an algorithm (KOTSIANTIS; KANELLOPOULOS; PINTELAS, 2006).

During this investigation, it was observed that all the attributes in **Person** were discrete features. For each attribute, all the existent levels were verified, as well as their frequencies. Then, the incidence of missing values for every attribute on the dataset was also verified.

Since the focus of this research was not structure and parameter learning with missing data, variables with a high incidence of non observed values were chosen to be discarded of the study. On the **Person** class dataset, that was the case of the **Person.Education_level** attribute, with only 43% of observed cases, and the **Person.Marital_status** attribute, with only 35% of observed cases.

On a further step, it was verified the incidence of collinearity among variables on the **Person** class dataset. For that reason, three other attributes were discarded of the case study: **Person.Propensity_for_tourism**; **Person.Propensity_for_e-commerce**; **Person.Propensity_for_paid_TV**.

Finally, only the complete cases on the dataset (those objects where all the attributes are observed) were considered, narrowing it down to 84.178 observations and 19 attributes (including the hashed ID). A description for each of those 19 variables can be found in Appendix A.

4.1.2 The *Company* class

The second class in the Atibaia domain is denoted **Company**. It is also stored in a single table inside a relational database representing the domain of study. Once again, each line on the table represents an object of the class, and every column represents an attribute of the objects. Every object in **Company** (every line in the table) represents a business located in Atibaia and its attributes.

Like **Person**, the dataset for **Company** and its objects was also kindly provided by *Serasa Experian* in May 2017. It contains 20.162 objects and comprises every business located in Atibaia registered at the Serasa database by the time of the concession.

All the objects in this class are fully anonymised with no identifiable attribute provided. Every object $c \in \mathbf{Company}$ is associated with 8 categorical attributes representing characteristics of the objects in **Company**. They are as follows:

- 1 - Census sector;
- 2 - Company age;
- 3 - Range of employees;
- 4 - Branch flag;
- 5 - Presumed revenue;

- 6 - Prescreening;
- 7 - Operating Index;
- 8 - Rural producer flag;

The approach adopted for the **Person** class was adopted for the **Company** class as well and, in the data pre-processing stage, a meticulous investigation of the data was conducted to understand the features and deal with missing data and multicollinearity.

Following the pattern verified in **Person**, it was observed that all the attributes in **Company** were discrete features. For each attribute, all the existing levels and their respective frequencies were observed. Then, the incidence of missing values for every attribute on the dataset was also verified.

Variables with a high incidence of missing values were chosen to be discarded from the study. On the **Company** class dataset, that was the case of the **Company.Prescreening** attribute, with only 55% of observed cases, and the **Company.Branch_flag** attribute, with only 61% of observed cases.

Unlike the situation of the **Person** class, there were no significant level of multicollinearity among variables on the **Company** class dataset. Hence, no variables were discarded for that reason.

Once again, since the focus of this research does not cover structure and parameter learning with missing data, only the complete cases (those objects where all the attributes are observed) were considered, narrowing down the dataset to 15.132 observations and 7 attributes (including the hashed ID). A description for each of those 7 variables can be found in Appendix B.

4.1.3 The *Census Sector* class

The third and final class of objects in the Atibaia domain is denoted **Census_sector**. It is stored in a single table inside a relational database that represents the domain of study. Once again, each line on the table represents an object of the class, and every column represents an attribute of the objects. Every object in **Census_sector** (every line in the table) represents a *census sector* of Atibaia and its attributes.

The dataset for this class of objects was generated from the data collected on the last demographic census realized in Brazil by IBGE on the year 2010. The demographic census is the most complete statistical operation realized by a country, when characteristics of the entire the population and of all domiciles inside the National territory are investigated ((IBGE), 2011).

Researching all households in the country, a demographic census constitutes a valuable source of knowledge about the life conditions of the population in all cities and

Figure 7 – All census sectors in Atibaia plotted in a map.

All census sectors in Atibaia plotted in a map using the software *QGIS 2.18.7 Las Palmas*.

The color of each sector represents the range of residents inside the sector. Source: (MORMILLE; COZMAN, 2017).

smaller territorial divisions, such as the *census sector*.

A census sector is the smaller territorial unit considered in the demographic census, formed by a continuous area, which is entirely contained in an urban or rural area, with a size adequate to the research operation. The set of census sectors comprise the entire National territory, assuring the full cover of the research ((IBGE), 2011).

For that reason, the files containing the demographic census data are aggregated by *census sector*. The data on the files comprehend characteristics of domiciles and the population in its totality.

There are approximately 3.000 variables from the demographic census, aggregated by *census sector*, which cover the following characteristics of the resident population: gender; age; color or race; condition at home; persons responsible for the domicile; literacy; birth registration of children with up to 10 years of age; and characteristics of private households ((IBGE), 2011).

The 2010 demographic census was realized between August 1st and October 30rd, 2010, using the territorial base that was constituted of 316.574 census sectors. The data collection method was through face-to-face interview, conducted by the census agents across all census sectors in Brazil.

The datasets of the 2010 demographic census can be found at the IBGE website (<http://www.ibge.gov.br/home/>). A map with the *census sectors* in Atibaia can be seen

in Figure 7; the color of the layer represents the number of citizens living inside a given census sector.

Nonetheless, of the 316.574 *census sectors* that constitute the National territory, 327 comprise the territory of the city of Atibaia. Therefore, only those 327 objects were selected for the study.

A big challenge was to choose, among the more than 3.000 variables, a significantly smaller amount to include in the study. Finally, a total of **16** variables were chosen and, from them, new variables were derived. The originally chosen variables were:

- 1 - Census sector;
- 2 - Situation;
- 3 - Amount of domiciles;
- 4 - Average House Hold;
- 5 - Average income;
- 6 - Domiciles of type house;
- 7 - Domiciles of type apartment;
- 8 - Fully paid domiciles;
- 9 - Domiciles being paid;
- 10 - Rented domiciles;
- 11 - Domiciles with water supply;
- 12 - Domiciles with access to the general sewage network;
- 13 - Domiciles with garbage collection;
- 14 - Domiciles with electricity;
- 15 - Domiciles without male resident;
- 16 - Domiciles without female resident;

Given the variance of the population and the shape of the census sectors, the choice to obtain a relative measure of variables 6 to 16 was made. This was made possible through the division of these variables by the total number of domiciles within the census sector (variable 3).

Still, all the above mentioned variables are continuous, therefore, it was also necessary to discretize them. After that process was conclude, **Census_sector** was comprised in a dataset with 327 objects, each representing a census sector in the city of Atibaia, and 16 variables, representing the attributes of the objects in **Census_sector**. A description of these variables can be found in Appendix C.

4.2 The case study

During this research, a package in the R language was developed to support our learning method. The package was named *Vanilla.prm* and will be described in a further section.

Given the Atibaia city domain and its three classes, **Person**, **Company** and **Census_sector**, the goal was to apply the Vanilla-PRM learning method to the domain and verify its performance.

To that end, two distinct experiments were conducted. In the first experiment, the method was applied to the domain by specifying the **Person** class as the leaf class, and the chosen target variable for prediction was **Person.Social_class**. The resulting PRM of this first experiment is referred to as Π_P .

In the second experiment, the method was applied to the domain by specifying the **Company** class as the leaf class, and the chosen target variable for prediction was **Company.Presumed_revenue**. The resulting PRM of the second experiment is referred to as Π_C .

4.2.1 Method and measures

Using a PRM to predict the category of a target variable is a task analogous to a classification problem. Therefore, to evaluate the performance of these experiments using a PRM, procedures and metrics associated with standard supervised machine learning classification problems were adopted.

4.2.1.1 Cross-validation

First, a *target variable* is defined. The target variable is the attribute that the model will attempt to predict. In the case of a PRM learned using the method proposed in Chapter 3, it is recommended to choose an attribute of the *leaf class* as the target variable.

The *k-fold* cross-validation consists of randomly splitting a dataset $\mathcal{D}$ into k mutually exclusive subsets of approximately equal size. The model is trained and tested k times. On each time $t \in \{1, 2, \dots, k\}$ the model is trained on $(\mathcal{D} - D_t)$ and tested on D_t (KOHAVI et al., 1995). Nonetheless, this method depends on whether the j distinct categories on the target variable are mutually exclusive.

4.2.1.2 Measures

The performance of the model is measured by taking all the predictions after the k -th step of the cross-validation and visualized using a *confusion matrix*. The confusion

matrix is a table where each columns represents the instances in its actual category, and the rows indicates the category predicted by the model (VOLOVICI, 2016).

From the confusion matrix, for each distinct category j on the target variable, tp_j is defined as the sum of *true positives* in the category j . These are cases in which an instance is classified as belonging to the category j , and it actually belongs to the class j .

The sum of *true negatives* in the category j is denoted tn_j . These are cases in which an instance is correctly classified as not belonging to a category j (it is classified as belonging to a distinct category).

The sum of *false positives* in the category j is denoted fp_j . I accounts for the amount of instances that were classified as belonging to category j , but actually belonged to a distinct category.

And finally, fn_j is defined as the sum of *false negatives* in the category j . It represents the cases where an instance was classified in a category other than j , but actually belonged to j .

In the subfield of Machine Learning specialised in problems with classification, a set of measures can be computed to evaluate the performance of a classification model from tp_j , tn_j , fp_j and fn_j .

A first measure of quality that can be derived is the *Sensitivity*, also denote as *Recall* or *True Positive Rate*. It states the proportion of cases in j that were correctly classified (POWERS, 2011). The Recall of a category j can be computed as:

$$recall_j = \frac{tp_j}{tp_j + fn_j}. \quad (4.1)$$

The *False Negative Rate* (FNR), or *Miss-rate*, is the proportion of cases in j that were wrongly classified. It can be obtained by:

$$FNR_j = \frac{fn_j}{tp_j + fn_j}. \quad (4.2)$$

Note that *recall* and *FNR* are complementary measures, in a way that $recall + FNR = 1$. Also, both measures can only assume values inside the interval $[0, 1]$, and a good classifier is expected to have high values for the recall measure, and low values for the FNR measure.

The *False Positive Rate* (FPR) or *Fall-out* is the proportion of cases $\notin j$ that were wrongly classified, and can be computed as:

$$FPR_j = \frac{fp_j}{tn_j + fp_j}. \quad (4.3)$$

The *True Negative Rate* (TNR), or *Specificity* is the proportion of cases $\notin j$ that were classified as not belonging to j . It can be computed as:

$$specificity_j = \frac{tn_j}{tn_j + fp_j}. \quad (4.4)$$

Note that *specificity* and *FPR* are complementary measures, in a way that *specificity* + *FPR* = 1. Once again, both measures can only assume values inside the interval $[0, 1]$, and a good classifier is expected to have high values for the specificity measure, and low values for the FPR measure.

The *Accuracy* is the fraction of correct predictions made by the model. Higher accuracy values are expected in a good classifier ($accuracy \subset [0, 1]$). It can be obtained by the following expression:

$$accuracy_j = \frac{tp_j + tn_j}{tp_j + fp_j + tn_j + fn_j}. \quad (4.5)$$

The *Precision* is the proportion of cases classified in j that are effectively in j (POWERS, 2011). $Precision \subset [0, 1]$ and values closer to 1 are expected of good classifiers. It can be computed as follows:

$$precision_j = \frac{tp_j}{tp_j + fp_j}. \quad (4.6)$$

The *F1-score* is an assertiveness measure for classifiers, consisting of the harmonic average of *precision* and *recall*. It is also known as *Proportion of Specific Agreement*, as it is applied to a specific category (POWERS, 2011). The F1-score can be computed by:

$$F1_j = \frac{2}{\frac{1}{recall_j} + \frac{1}{precision_j}}. \quad (4.7)$$

The F1-score assume values inside the interval $[0, 1]$, and, once again, good classifiers are expected to present values closer to 1 for this measure.

4.2.1.3 Comparison

To evaluate the performance of the proposed method over the domain of study, another PRM structure was learned from the Atibaia domain using the method proposed by (KOLLER, 1999) and (FRIEDMAN et al., 1999), presented in 2.5.2. This PRM is referred to as Π_{Ground} .

Then, Π_{Ground} is be used to predict the same *target variable* using *k-fold cross-validation* and all the measures presented in 4.2.1.2 are computed.

Finally, the Jaccard Similarity, a classical similarity measure, is used to compare the structures learned using Vanilla-PRM and Π_{Ground} .

The Jaccard Similarity measures the relative size of the overlap between two finite sets. In this case, the set of arrows in both PRMs. Let A be the set of arrows in Π_P learned using the Vanilla-PRM method, and G be the set of arrows in Π_{Ground} . The Jaccard Similarity $\mathcal{J}$ can be associated with the Jaccard distance $J\delta$ (KOSUB, 2016) and it is formally defined as:

$$\mathcal{J} = \frac{|A \cap G|}{|A \cup G|}. \quad (4.8)$$

4.3 The Person PRM experiment

As presented in Section 4.2, the Vanilla-PRM learning method, proposed in Chapter 3, was applied to the Atibaia domain. In this first experiment, the **Person** class was specified as the leaf class, and the chosen target variable for prediction was **Person.Social_class**.

The assumptions presented in Section 3.1 were strictly followed, and the algorithm described in Section 3.3 was applied using a package in the R language, developed during the course of this research.

4.3.1 The master tables for Atibaia domain

After the data preprocessing stage previously described in this Chapter, the algorithm took the three classes in the Atibaia domain - **Person**, **Company** and **Census_sector** - and performed a pairwise check between them, to verify whether there were relations among objects of those classes and the nature of the eventual relations.

In the Atibaia domain, the **Person.Sector_ID** and **Company.Sector_ID** attributes are also foreign keys that establish a relation between objects in **Person** and **Census_sector** and between objects in **Company** and **Census_sector**, respectively.

Those relations can be intuitively understood. First, an object in $p \in \mathbf{Person}$ is a citizen with a registered address. That address is a physical location and, by definition, it is contained inside the area of one, and only one, of the census sectors that comprise the territory of the city of Atibaia. Hence, an object $p \in \mathbf{Person}$ (citizen) can only be associated with one, and only one, object $s \in \mathbf{Census_sector}$ (census sector), which is the one where its residence is located.¹

Since multiple citizens can live inside the same census sector, but no citizen can be associated with more than one census sector, the **Person** $\rightarrow$ **Census_sector** relation

¹ Even though a person is allowed to have more than one house property, only one house is considered to be his/her residence.

Figure 8 – Visual representation of the blacklists of edges for the Person experiment.

(a) blacklist for $Mt(Person)$; (b) blacklist for $Mt(Company)$; (c) blacklist for $Mt(Census_sector)$. The green arrows indicates the dependencies between attributes of distinct classes that are allowed when learning the local Bayesian network from the master table. The red arrows indicates the dependencies between attributes of distinct classes that are not allowed when learning the local Bayesian network from the master table and, therefore, are added to the blacklist.

is a $n \rightarrow 1$ relation. That implies that, if an attribute of **Person** parents an attribute of **Census_sector**, the use of an aggregation function γ would be required. Inversely, if an attribute of **Census_sector** parents an attribute of **Person**, the use of an aggregation function γ would not be required.

Analogously, an object in $c \in \mathbf{Company}$ is a business with a registered address. Just like a person's address, that address of a business is a physical location contained inside the area of one, and only one, of the census sectors that comprise the territory of the city of Atibaia. Hence, an object $c \in \mathbf{Company}$ (business) can only be associate with one, and only one, object $s \in \mathbf{Census_sector}$ (census sector), which is the one where the business is located.

Given the fact that multiple companies can be located inside the same census sector, but no company can be associated with more then one census sector, the **Company** $\rightarrow$ **Census_sector** relation is a $n \rightarrow 1$ relation. Hence, if an attribute of **Company** parents an attribute of **Census_sector**, the use of an aggregation function γ would be required. Conversely, if an attribute of **Census_sector** parents an attribute of **Company**, the use of an aggregation function γ would not be required.

Objects in **Person** and **Company** does not have a direct relationship; however, a second degree relation can be established between objects of those classes if they share the same census sector. Accordingly, an indirect relation can be observed between citizens and business of Atibaia located inside the same census sector.

Since multiple citizens and multiple business can be located inside the same census sector, the **Person** $\rightarrow$ **Company** relation is a $n \rightarrow n$ relation. Consequently, any

dependency relation between attributes of those classes on the *instance dependency graph* G_σ would require an aggregation function γ .

With the relations and their natures mapped, the master tables for **Person**, **Company** and **Census_sector** can be computed.

The master table for **Person** is denoted $Mt(Person)$. At first, $Mt(Person)$ is an exact copy of the **Person**'s table. Then, for every object $p \in \mathbf{Person}$ (for every line in the table), a summary of every attribute of its τ -relative objects, whose links are given by the value of the reference slot **Person.Sector_ID**, is added to its respective row in the master table.

As previously stated, every object in **Person** is associated with one, and only one, object in the class **Census_sector**, meaning that the slot chain that links them is indeed singled-valued and the use of an aggregation function γ is unnecessary. Hence, the values of the attributes of a census sector $s \in \mathbf{Census_sector}$ are simply replicated on the rows where its corresponding key, **Census_sector.Sector_ID**, matches the value of **Person.Sector_ID** on $Mt(Person)$.

The slot chain that links **Person** and **Company** is not single-valued. Even though an object $p \in \mathbf{Person}$ and an object $c \in \mathbf{Company}$ can only relate to one object $s \in \mathbf{Census_sector}$, the inverse relation is not true. One object $s \in \mathbf{Census_sector}$ can (and usually is) linked with multiple objects on both **Person** and **Company**. Hence, an object $p \in \mathbf{Person}$ is related to multiple objects in **Company** (the business located in the census sector which a person lives).

Therefore, an aggregation function γ is required to return a summary of the attributes on **Company** to the master table $Mt(Person)$. Since the datasets contain only categorical features, the chosen aggregation function was the *mode*, and, after the computation of all the aggregations, its values are returned to its respective rows on $Mt(Person)$.

The master table for **Company** is denoted $Mt(Company)$ and it starts as an exact copy of the **Company**'s table. Then, for every object $c \in \mathbf{Company}$, a summary of every attribute of its τ -relative objects, whose links are given by the value of the reference slot **Company.Sector_ID**, is added to its respective row in $Mt(Company)$.

The **Company** $\rightarrow$ **Census_sector** relation is a $n \rightarrow 1$ relation, meaning that the slot chain that links them is indeed singled-valued and the use of an aggregation function γ is unnecessary. For that reason, the values of the attributes of a census sector $s \in \mathbf{Census_sector}$ are simply replicated on the rows where its corresponding key **Census_sector.Sector_ID** matches the value of **Company.Sector_ID** on $Mt(Company)$.

As previously verified, the **Person** $\rightarrow$ **Company** relation is a $n \rightarrow n$ relation. Hence, an object $c \in \mathbf{Company}$ can relate with multiple objects in **Person**. Therefore,

an aggregation function γ is required to return a summary of the attributes on **Person** to the master table $Mt(Company)$. Once again, the chosen aggregation function was the *mode*, and, after the computations, the results are returned to its respective rows on $Mt(Company)$.

Finally, the master table for **Census_sector** is denoted $Mt(Census_sector)$, and it starts as an exact copy of the **Census_sector**'s table. Then, for every object $s \in \mathbf{Census_sector}$, a summary of every attribute of its τ -relative objects, whose links are given by the value of the reference slot **Census_sector.Sector_ID**, is added to its respective row in $Mt(Census_sector)$.

The **Census_sector** $\rightarrow$ **Company** relation is a $1 \rightarrow n$ relation, meaning that the slot chain that links them is not singled-valued and the use an aggregation function γ is required.

For that reason, the values of the attributes of all business located inside a census sector $s \in \mathbf{Census_sector}$ are aggregated using the *mode* as an aggregation function, and the result is returned the corresponding row in $Mt(Census_sector)$.

The **Census_sector** $\rightarrow$ **Person** relation is also a $1 \rightarrow n$ relation. Consequently, the values of the attributes of all citizens that live inside a census sector $s \in \mathbf{Census_sector}$ are aggregated using the *mode* as an aggregation function, and the result is returned the corresponding row in $Mt(Census_sector)$.

4.3.2 Person experiment: structure learning

In possession of $Mt(Person)$, $Mt(Company)$ and $Mt(Census_sector)$, the challenges of defining a proper structure scoring function, choosing a search method for possible legal structures, and comparing them via score, could be solved by traditional Bayesian network approaches. Some of these approaches were introduced in Section 2.4.

At this stage, the **Person** class was specified as the leaf class, and the chosen target variable for prediction was the **Person.Social_class**. A “class stratification” order was also provided.

Still, it was necessary to assure that the structure restrictions assumed in Section 3.1 were enforced during learning. To do so, a black list of edges was assembled for each class separately. Figure 8 shows, for each class, what are the forbidden edges, given the leaf class. It can be observed that all the restrictions presented in 3.1 are respected.

Then, for each class in the domain, **Person**, **Company** and **Census_sector**, a local Bayesian network structure, was learned from their respective master tables - $Mt(\mathbf{Person})$, $Mt(\mathbf{Company})$ and $Mt(\mathbf{Census_sector})$ - compulsorily considering the black lists of the respective classes - $Bl(\mathbf{Person})$, $Bl(\mathbf{Company})$ and $Bl(\mathbf{Census_sector})$.

Figure 9 – The dependency structure of Π_P .

The dependency structure of Π_P . The node in red indicates the target variable, **Person.Social_class**. A yellow arrow indicates an intra-class dependency; a green arrow indicates an inter-class dependency. A total of 14 arrows on Π_P were omitted on this figure to preserve the readability. Nonetheless, all the dependencies regarding the target variable are represented.

The search algorithm selected to complete this task was the *Hill Climbing*, and the selected scoring function was the *BIC* score. The local Bayesian networks learned in this step were respectively denoted $\mathcal{B}_{Person}$, $\mathcal{B}_{Company}$ and $\mathcal{B}_{Census_sector}$.

Then, the structures of $\mathcal{B}_{Person}$, $\mathcal{B}_{Company}$ and $\mathcal{B}_{Census_sector}$ were combined to form the final PRM structure denoted Π_P . The assumptions made in Section 3.1, ensured by the computed black lists of edges, guarantees that Π_P will not have any cycles in the dependency graph.

4.3.3 The Person PRM structure

The PRM Π_P , learned using the Vanilla-PRM method can be seen in Figure 9. The leaf class **Person** is represented by the bottom blue box, and the node indicating the *target variable* **Person.Social_class** is highlighted in red.

A yellow arrow indicates an intra-class dependency, while a green arrow indicates an inter-class dependency. In total, Π_P have 70 edges representing 58 intra-class dependencies and 12 inter-class dependencies. In Figure 9 some intra-class arrows were omitted to maintain the readability of the dependency structure. Nonetheless, all the dependencies regarding the target variable were preserved.

It can be seen that all the dependencies respect the assumptions presented on Chapter 3 and that the graph is indeed acyclic.

On Π_P , the parents of the target variable, $Pa(\mathbf{Person.Social_class})$, are one intra-class attribute - **Person.Mosaic** - and two inter-class attributes - **Census_sector.Situation** and **Census_sector.Average_income**. Therefore, according to the Π_P model, the social class of a citizen of Atibaia is independent of all other variables, given the values of $Pa(\mathbf{Person.Social_class})$.

A full description of the Π_P model, listing all the dependencies and the conditional probability distribution table for the target variable, **Person.Social_class**, can be seen in Appendix D.

4.3.4 The Ground PRM structure

As introduced in Section 4.2.1.3, to evaluate the performance of Π_P , another PRM structure was learned from the Atibaia domain using the method proposed by (KOLLER, 1999) and (FRIEDMAN et al., 1999), presented in Section 2.5.2. This PRM is referred to as Π_{Ground} and can be seen in Figure 10.

A yellow arrow indicates an intra-class dependency, a light-green arrow indicates an inter-class dependency that did not require an aggregation function γ , and a dark-green indicates an inter-class dependency that required an aggregation function, in that particular

Figure 10 – The dependency structure of Π_{Ground} .

The dependency structure of Π_{Ground} . A yellow arrow indicates an intra-class dependency; a dark green arrow indicates an inter-class dependency where an aggregation function was used; and a light green arrow indicates an inter-class dependency where an aggregation function was not used.. A total of 11 arrows on Π_P were omitted on this figure to preserve the readability. Nonetheless, all the dependencies regarding the variables of interest are represented.

Figure 11 – Π_P confusion matrix of the *Person experiment*.

Π_P Target variable: **Person.Social_class**

Confusion Matrix 10-fold cross validation

		True condition			
		A	B	C	D
Predicted condition	A	1	4	0	0
	B	1897	35785	8329	5670
	C	20	1603	2576	1004
	D	12	6780	3785	16678

Confusion matrix displaying the results of a 10-*fold* cross validation predicting the **Person.Social_class** attribute with Π_P .

case, the mode. Some intra-class arrows were omitted to maintain the readability of the dependency structure. Nonetheless, the arrows that represent dependencies on variables of interest were fully preserved.

Considering the **Person** PRM experiment, it is possible to see in Figure 10 that the parents of the target variable, $Pa(\mathbf{Person.Social_class})$ are three intra-class attributes and two inter-class attributes. Therefore, according to the Π_{Ground} model, the social class of a citizen of Atibaia is independent of all other variables, given the values of $Pa(\mathbf{Person.Social_class})$.

A full description of the Π_{Ground} , listing all the dependencies and the conditional probability distribution table for the target variable, **Person.Social_class**, can be seen in Appendix E.

4.3.5 The Person PRM performance

With both Π_{Ground} and Π_P , the method and the measures describe in Section 4.2.1 were applied. The goal was to measure both models performance in predicting the *target variable* **Person.Social_class**.

First, a 10-*fold* cross validation was conducted on Π_P as described in Section 4.2.1. The **Person**'s table was splitted in 10 subsets of approximately the same size along with their respective related objects from **Company** and **Census_sector**.

The performance of the model was measured by taking all the predictions after the 10th step of the cross-validation and, as described in Section 4.2.1.2, visualized using a

Figure 12 – Π_P performance measures of the *Person experiment*.

Π_P Target variable: **Person.Social_class**

10-fold cross validation

		True condition				Precision	Accuracy	F1-score
		A	B	C	D			
Predicted condition	A	1	4	0	0	0,2000	0,9770	0,0010
	B	1897	35785	8329	5670	0,6924	0,7114	0,7467
	C	20	1603	2576	1004	0,4951	0,8248	0,2590
	D	12	6780	3785	16678	0,6119	0,7950	0,6591
Recall		0,0005	0,8101	0,1754	0,7142	Overall Accuracy 0,6541		
Miss Rate		0,9995	0,1899	0,8246	0,2858			
Specificity		1,0000	0,6023	0,9622	0,8260			
Fall-out		0,0000	0,3977	0,0378	0,1740			

All measures presented in 4.2.1.2 computed from the results of a 10-fold cross validation predicting the **Person.Social_class** attribute with Π_P .

confusion matrix, as shown in Figure 11.

From the confusion matrix, all the measures presented in Section 4.2.1.2 were computed for each existing level of **Person.Social_class**. The Π_P results can be seen in Figure 12.

The *Recall* for all four levels of **Person.Social_class** on Π_P was computed through Equation 4.1. *Recall* values closer to 1 are expected of good classifiers, but for levels *A* and *C* the results were bellow satisfactory.

The *Miss-rate* measure was also computed for all four levels of the target variable, using Equation 4.2. Unlike the *Recall*, values closer to 0 are expected. Since *Miss-rate* and *Recall* are complementary measures, it is intuitive to conclude that results for levels *A* and *C* also were bellow satisfactory.

The *Specificity* for all four levels of **Person.Social_class** was also computed through Equation 4.4. *Specificity* values closer to 1 are expected of good classifiers, and levels *A* and *C* and *D* presented good results.

Then, the *Fall-out* rate was computed for all the levels of the target variable, using Equation 4.3. Here, values closer to 0 are expected. *Fall-out* and *Specificity* are complementary measures. Hence, for levels *A* and *C* and *D* the results were quite satisfactory.

The *Precision* of Π_P was also measured for all four levels of **Person.Social_class** through Equation 4.6. *Precision* values closer to 1 are expected of good classifiers, but, once again, for levels *A* and *C* the results were bellow satisfactory.

Figure 13 – Π_{Ground} confusion matrix of the *Person experiment*.

Π_{ground} Target variable: **Person.Social_class**

Confusion Matrix 10-fold cross validation

		True condition			
		A	B	C	D
Predicted condition	A	9	11	3	2
	B	1874	36112	8357	5676
	C	30	1372	2542	1078
	D	14	6662	3783	16592

Confusion matrix displaying the results of a 10-*fold* cross validation predicting the **Person.Social_class** attribute with Π_{Ground} .

The *Accuracy* of Π_P indicates the fraction of correct predictions made by the model for all levels of **Person.Social_class**, and was computed by Equation 4.5. *Accuracy* values closer to 1 are expected of good classifiers, and Π_P presented good results for all four levels. The *Overall Accuracy* is the fraction of total correct predictions made by the model.

Finally, the *F1-score* was computed for all distinct levels of **Person.Social_class**. Values closer to 1 are expected for the *F1-score*. Once again, the model presented results bellow satisfactory for levels *A* and *C*.

The same procedure was adopted to evaluate the performance of Π_{Ground} in predicting the *target variable* **Person.Social_class**. Once more, the 10-*fold* cross validation was conduct following 4.2.1. The performance of the model was, once again, measured by taking all the predictions after the 10th step of the cross-validation and visualized using a *confusion matrix*, as shown in Figure 13.

From the confusion matrix derived from the results of the 10-*fold* cross validation predicting the **Person.Social_class** attribute with Π_{Ground} , once again, all the measures presented in Section 4.2.1.2 were computed for each existing level of **Person.Social_class**. The results for Π_{Ground} can be seen in Figure 14.

Following the procedure adopted for Π_P , the *Recall* for all four levels of **Person.Social_class** on Π_{Ground} was computed through Equation 4.1 and the results for levels *A* and *C* also were bellow satisfactory.

Figure 14 – Π_{Ground} performance measures of the *Person experiment*.

Π_{ground} Target variable: **Person.Social_class** 10-fold cross validation

		True condition				Precision	Accuracy	F1-score
		A	B	C	D			
Predicted condition	A	9	11	3	2	0,3600	0,9770	0,0092
	B	1874	36112	8357	5676	0,6942	0,7153	0,7510
	C	30	1372	2542	1078	0,5062	0,8262	0,2580
	D	14	6662	3783	16592	0,6134	0,7953	0,6584
Recall		0,0047	0,8178	0,1731	0,7106	Overall Accuracy 0,6569		
Miss Rate		0,9953	0,1822	0,8269	0,2894			
Specificity		0,9998	0,6019	0,9643	0,8279			
Fall-out		0,0002	0,3981	0,0357	0,1721			

All measures presented in 4.2.1.2 computed from the results of a 10-fold cross validation predicting the **Person.Social_class** attribute with Π_{Ground} .

The *Miss-rate* measure was also computed for all four levels of the target variable, using Equation 4.2. Once again, it is intuitive to conclude that results for levels *A* and *C* were bellow satisfactory.

Similarly to Π_P , Π_{Ground} also presented good *Specificity* for levels *A* and *C* and *D*. Hence, *Fall-out* rate for levels *A* and *C* and *D* were also satisfactory.

The *Precision* of Π_{Ground} was also measured for all four levels of **Person.Social_class** through Equation 4.6. The results were, once again, very similar to the ones obtained with Π_P .

The *Accuracy* of Π_{Ground} for all levels of **Person.Social_class** was computed through Equation 4.5. Good results were obtained for all four levels, and the *Overall Accuracy* was very similar to one obtained by Π_P .

Finally, the *F1-score* was again computed for all distinct levels of **Person.Social_class**. Values closer to 1 are expected for the *F1-score*. Once again, similarly to Π_P , Π_{Ground} presented results bellow satisfactory for levels *A* and *C*.

Also, as described in Section 4.2.1.3, the similarity between the graphs of Π_{Ground} and Π_P was measured using the *Jaccard similarity*, defined in Equation 4.8. Table 2 shows the details and the computed Jaccard similarity.

The Π_{Ground} PRM have 70 arrows indicating the dependencies between the attributes of the classes of the domain. The Π_P PRM also have 70 arrows, indicating the dependencies learned using the method proposed in Chapter 3.

The size of the subset containing the intersection between the sets of arrows of Π_{Ground} and Π_P is 60 ($|\Pi_{Ground} \cap \Pi_P| = 60$), which means that both PRMs have 60 dependencies in common.

The size of the subset containing the union between the sets of arrows of Π_{Ground} and Π_P is 80 ($|\Pi_{Ground} \cup \Pi_P| = 80$). Hence, a Jaccard similarity between the graphs of Π_{Ground} and Π_P is $\mathcal{J} = 0,75$.

Table 2 – Jaccard similarity between the set of arrows of Π_{Ground} and Π_P .

Arrows in Π_P	70
Arrows in Π_{Ground}	70
$ \Pi_{Ground} \cap \Pi_P $	60
$ \Pi_{Ground} \cup \Pi_P $	80
Jaccard Similarity	0,75

Given the nature of the method for learning PRMs proposed in this work, it is natural for the similarity between the graphs to not be high. In face of that scenario, the Jaccard similarity was computed once again using Equation 4.8, but this time considering only the dependencies inside or directed to the leaf class **Person**.

The results of this computation can be seen on Table 3. Considering only the dependencies of the attributes of the leaf class **Person** class, Π_{Ground} have 47 arrows and Π_P have 45 arrows.

Taking only the arrows inside or directed to the leaf class, the size of the subset containing the intersection between the sets of arrows of Π_{Ground} and Π_P is 42 ($|\Pi_{Ground} \cap \Pi_P| = 42$), which means that both PRMs have 42 dependencies in common.

And the size of the subset containing the union between the sets of arrows inside or directed to the leaf class of Π_{Ground} and Π_P is 50 ($|\Pi_{Ground} \cup \Pi_P| = 50$).

The Jaccard similarity between the graphs of Π_{Ground} and Π_P when only the arrows inside or directed to the **Person** class are considered is $\mathcal{J} = 0,84$.

Table 3 – The Jaccard similarity between the set of arrows of Π_{Ground} and Π_P considering only the dependencies inside the *leaf class Person*.

Arrows in Π_P	45
Arrows in Π_{Ground}	47
$ \Pi_{Ground} \cap \Pi_P $	42
$ \Pi_{Ground} \cup \Pi_P $	50
Jaccard Similarity	0,84

Figure 15 – Visual representation of the blacklists of edges for the Company experiment.

(a) blacklist for $Mt(Company)$; (b) blacklist for $Mt(Person)$; (c) blacklist for $Mt(Census_sector)$. The green arrows indicates the dependencies between attributes of distinct classes that are allowed when learning the local Bayesian network from the master table. The red arrows indicates the dependencies between attributes of distinct classes that are not allowed when learning the local Bayesian network from the master table and, therefore, are added to the blacklist.

4.4 The Company PRM

In the second experiment, the method was applied to the Atibaia domain by specifying the **Company** class as the leaf class and the chosen target variable for prediction was the **Company.Presumed_revenue**.

The first stage of the experiment is analogous to the one presented in Section 4.3, and the master tables for the three distinct class in the domain - $Mt(Company)$, $Mt(Person)$ and $Mt(Census_sector)$ - are the same as the ones computed in 4.3.1.

4.4.1 Company experiment: structure learning

Just like the experiment described in Section 4.3, the challenges of defining a proper structure scoring function, choosing a search method for possible legal structures, and comparing them via score, could be solved by traditional Bayesian network approaches once in possession of $Mt(Company)$, $Mt(Person)$ and $Mt(Census_sector)$.

At this stage, the **Company** class was specified as the leaf class, and the chosen target variable for prediction was the **Company.Presumed_revenue**. A “class stratification” order was also provided.

The structure restrictions assumed in Section 3.1 were enforced during learning by assembling a black list of edges for each class in the domain. Figure 15 shows, for each class, what are the forbidden edges, given the leaf class. It can be observed that all the restrictions presented in 3.1 are respected.

Figure 16 – The dependency structure of Π_C .

The dependency structure of Π_C . A yellow arrow indicates an intra-class dependency; a dark green arrow indicates an inter-class dependency where an aggregation function was used; and a light green arrow indicates an inter-class dependency where an aggregation function was not used. A total of 13 arrows on Π_P were omitted on this figure to preserve the readability. Nonetheless, all the dependencies regarding the target variable are represented.

Then, for each class in the domain - **Company**, **Person** and **Census_sector** - a local Bayesian network structure was learned from their respective master tables - $Mt(\mathbf{Company})$, $Mt(\mathbf{Person})$ and $Mt(\mathbf{Census_sector})$ - considering the black lists of the respective classes - $Bl(\mathbf{Company})$, $Bl(\mathbf{Person})$ and $Bl(\mathbf{Census_sector})$. The search algorithm selected to complete this experiment was, once again, the *Hill Climbing*, and the selected scoring function was the *BIC* score. The local Bayesian networks learned in this step were respectively denoted $\mathcal{B}_{Company}$, $\mathcal{B}_{Person}$ and $\mathcal{B}_{Census_sector}$.

Then, the structures of $\mathcal{B}_{Company}$, $\mathcal{B}_{Person}$ and $\mathcal{B}_{Census_sector}$ were combined to form the final PRM structure denoted Π_C . The assumptions made in Section 3.1, ensured by the computed black lists of edges, guarantees that Π_C do not have any cycles in the dependency graph.

4.4.2 The Company PRM structure

The PRM Π_C , learned using the Vanilla-PRM method, can be seen in Figure 16. The leaf class is **Company**, represented by the top-left blue box, and the node indicating the *target variable* **Company.Presumed_revenue** is highlighted in red.

A yellow arrow indicates an intra-class dependency, a light-green arrow indicates an inter-class dependency that did not require an aggregation function γ , and a dark-green indicates an inter-class dependency that required an aggregation function (mode). In Figure 16 some intra-class arrows were omitted to maintain the readability of the dependency structure. Nonetheless, all the dependencies regarding the target variable were preserved. In total, Π_C have 70 edges representing 58 intra-class dependencies and 11 inter-class dependencies.

It can be seen that all the dependencies respect the assumptions presented on Chapter 3 and that the graph is indeed acyclic.

On Π_C , the parents of the target variable, $Pa(\mathbf{Company.Presumed_revenue})$, are only two intra-class attributes. Therefore, according to the Π_C model, the presumed revenue of a company located in Atibaia is independent of all other variables, given the values of its two parents.

A full description of the Π_C model, listing all the dependencies and the conditional probability distribution table for the target variable **Company.Presumed_revenue** can be seen on Appendix F.

Comparing the structure of Π_C with the ground PRM Π_{Ground} , presented in 4.3.4, it is possible to notice the difference between the parents of the target variable. In Π_{Ground} , the parents of **Company.Presumed_revenue** are **Company.Range_of_employees** and **Company.Operating_index**.

Figure 17 – Π_C confusion matrix of the *Company* experiment.

Π_C Target variable: **Company.Presumed_revenue**

Confusion Matrix 10-fold cross validation

		True condition				
		Up to 200k	200k to 1mi	1mi to 10mi	10mi to 100mi	100mi or more
Predicted condition	Up to 200k	11072	1740	377	20	7
	200k to 1mi	1	885	94	0	0
	1mi to 10mi	87	163	429	28	11
	10mi to 100mi	0	0	0	0	0
	100mi or more	17	1	58	56	72

Confusion matrix displaying the results of a 10-*fold* cross validation predicting the **Company.Presumed_revenue** attribute with Π_C .

4.4.3 The Company PRM performance

The exact same procedure of Section 4.3.5 was adopted to this second experiment. With Π_{Ground} , presented in 4.3.4 and Π_C , the method and the measures describe in 4.2.1 were applied once again. This time, the goal was to measure both models performance in predicting the *target variable* **Company.Presumed_revenue**.

A 10-*fold* cross validation was conducted on Π_C following the procedure described in 4.2.1. The **Company**'s table was splitted in 10 subsets of approximately the same size along with their respective related objects from **Person** and **Census_sector**.

The performance of the model was measured by taking all the predictions after the 10th step of the cross-validation and, as described in Section 4.2.1.2, visualized using a *confusion matrix*, as shown in Figure 17.

From the confusion matrix, all the measures presented in Section 4.2.1.2 were computed for each existing level of **Company.Presumed_revenue**. The results can be seen in Figure 18.

The *Recall* for all five levels of **Company.Presumed_revenue** on Π_C was computed through Equation 4.1. *Recall* values closer to 1 are expected of good classifiers, but for levels *200K to 1mi*, *1mi to 10mi* and *10mi to 100mi* the results were bellow satisfactory.

Figure 18 – Π_C performance measures of the *Company* experiment.

Π_C Target variable: **Company.Presumed_revenue** 10-fold cross validation

		True condition					Precision	Accuracy	F1-score
		Up to 200k	200k to 1mi	1mi to 10mi	10mi to 100mi	100mi or more			
Predicted condition	Up to 200k	11072	1740	377	20	7	0,8378	0,8512	0,9078
	200k to 1mi	1	885	94	0	0	0,9031	0,8678	0,4696
	1mi to 10mi	87	163	429	28	11	0,5975	0,9459	0,5119
	10mi to 100mi	0	0	0	0	0	0,0000	0,9931	0,0000
	100mi or more	17	1	58	56	72	0,3529	0,9901	0,4898
Recall		0,9906	0,3173	0,4478	0,0000	0,8000	Overall Accuracy 0,8241		
Miss Rate		0,0094	0,6827	0,5522	1,0000	0,2000			
Specificity		0,4560	0,9923	0,9796	1,0000	0,9912			
Fall-out		0,5440	0,0077	0,0204	0,0000	0,0088			

All measures presented in 4.2.1.2 computed from the results of a 10-*fold* cross validation predicting the **Company.Presumed_revenue** attribute with Π_C .

The *Miss-rate* measure was also computed for all four levels of the target variable, using Equation 4.2. *Miss-rate* and *Recall* are complementary measures. Hence, the results of this measure for levels *200K to 1mi*, *1mi to 10mi* and *10mi to 100mi* also were bellow satisfactory.

The *Specificity* for all five levels of **Company.Presumed_revenue** was also computed through Equation 4.4. *Specificity* values closer to 1 are expected of good classifiers, and four of the five distinct levels presented excellent results.

Then, the *Fall-out* rate was computed for all the levels of the target variable, using Equation 4.3. *Fall-out* and *Specificity* are complementary measures. Levels *200K to 1mi*, *1mi to 10mi*, **10mi to 100mi** and **100mi or more** presented excellent results.

The *Precision* of Π_C was also measured for all five levels of **Company.Presumed_revenue** through Equation 4.6. *Precision* values closer to 1 are expected of good classifiers, but for levels *10mi to 100mi* and *100mi or more* the results were bellow satisfactory.

The *Accuracy* of Π_C indicates the fraction of correct predictions made by the model for all levels of **Company.Presumed_revenue**, and was computed by Equation 4.5. Π_C presented excellent results for all five levels. The *Overall Accuracy* was also very good.

Finally, the *F1-score* was computed for all distinct levels of **Company.Presumed_revenue**. Values closer to 1 are expected for the *F1-score*. However, only level *Up to 200k* presented satisfactory results.

Figure 19 – Π_{Ground} confusion matrix of the *Company* experiment.

Π_{ground} Target variable: **Company.Presumed_revenue**
 Confusion Matrix 10-fold cross validation

		True condition				
		Up to 200k	200k to 1mi	1mi to 10mi	10mi to 100mi	100mi or more
Predicted condition	Up to 200k	11070	2475	433	14	3
	200k to 1mi	22	157	113	0	0
	1mi to 10mi	72	157	343	33	12
	10mi to 100mi	2	0	3	0	3
	100mi or more	11	0	65	57	72

Confusion matrix displaying the results of a 10-*fold* cross validation predicting the **Company.Presumed_revenue** attribute with Π_{Ground} .

The same procedure was adopted to evaluate the performance of Π_{Ground} in predicting the *target variable* **Company.Presumed_revenue**. The 10-*fold* cross validation was conducted following Section 4.2.1.

The performance of the model was measured by taking all the predictions after the 10th step of the cross-validation and visualized using a *confusion matrix*, as shown in Figure 19.

From the confusion matrix derived from the results of the 10-*fold* cross validation predicting the **Company.Presumed_revenue** attribute with Π_{Ground} , once again, all the measures presented in Section 4.2.1.2 were computed for each existing level of **Company.Presumed_revenue**. The results for Π_{Ground} can be seen in Figure 20.

Following the procedure adopted for Π_C , the *Recall* and the *Miss-rate* for all five levels of **Company.Presumed_revenue** on Π_{Ground} were computed through Equations 4.1 and 4.2, respectively. Similarly to Π_C , the results for levels *200K to 1mi*, *1mi to 10mi* and *10mi to 100mi* were below satisfactory.

Π_{Ground} also presented excellent *Specificity* and *Fall-out* results for levels *200K to 1mi*, *1mi to 10mi*, *10mi to 100mi* and *100mi or more*.

The *Precision* of Π_{Ground} was also measured for all five levels of **Company.Presumed_revenue** through Equation 4.6. The results were, once again, very similar to the ones obtained with Π_P .

Figure 20 – Π_{Ground} performance measures of the *Company experiment*.

Π_{Ground} Target variable: **Company.Presumed_revenue** 10 - fold cross validation

		True condition					Precision	Accuracy	F1-score
		Up to 200k	200k to 1mi	1mi to 10mi	10mi to 100mi	100mi or more			
Predicted condition	Up to 200k	11070	2475	433	14	3	0,7910	0,7994	0,8795
	200k to 1mi	22	157	113	0	0	0,5377	0,8170	0,1019
	1mi to 10mi	72	157	343	33	12	0,5559	0,9413	0,4358
	10mi to 100mi	2	0	3	0	3	0,0000	0,9926	0,0000
	100mi or more	11	0	65	57	72	0,3512	0,9900	0,4881
Recall		0,9904	0,0563	0,3584	0,0000	0,8000	Overall Accuracy 0,7701		
Miss Rate		0,0096	0,9437	0,6416	1,0000	0,2000			
Specificity		0,2576	0,9890	0,9806	0,9995	0,9911			
Fall-out		0,7424	0,0110	0,0194	0,0005	0,0089			

All measures presented in 4.2.1.2 computed from the results of a 10-*fold* cross validation predicting the **Company.Presumed_revenue** attribute with Π_{Ground} .

The *Accuracy* of Π_{Ground} for all levels of **Company.Presumed_revenue** was computed through Equation 4.5. Only level *Up to 200K* presented satisfactory results.

At last, the *F1-score* was again computed for all distinct levels of **Company.Presumed_revenue**. Once again, similarly to Π_C , Π_{Ground} only presented good results for level *Up to 200K*.

The similarity between the graphs of Π_{Ground} and Π_C was measured using the *Jaccard similarity*, defined in Equation 4.8. Table 4 shows the details and the computed Jaccard similarity.

The Π_{Ground} PRM have 70 arrows indicating the dependencies between the attributes of the classes of the domain. The Π_C PRM have 69 arrows, indicating the dependencies learned using the method proposed in Chapter 3.

The size of the subset containing the intersection between the sets of arrows of Π_{Ground} and Π_C is 62 ($|\Pi_{Ground} \cap \Pi_C| = 62$), which means that both PRMs have 62 dependencies in common.

The size of the subset containing the union between the sets of arrows of Π_{Ground} and Π_C is 77 ($|\Pi_{Ground} \cup \Pi_C| = 77$). Hence, the Jaccard similarity between the graphs of Π_{Ground} and Π_C is $\mathcal{J} = 0,81$.

The Jaccard similarity was computed once again considering only the dependencies inside or directed to the leaf class **Company**. The results can be seen on Table 5.

Table 4 – Jaccard similarity between the set of arrows of Π_{Ground} and Π_P .

Arrows in Π_C	69
Arrows in Π_{Ground}	70
$ \Pi_{Ground} \cap \Pi_C $	62
$ \Pi_{Ground} \cup \Pi_C $	77
Jaccard Similarity	0,81

Considering only the dependencies of the attributes of the leaf class **Company** class, Π_{Ground} have 10 arrows and Π_P have 11 arrows.

Taking only the arrows inside or directed to the leaf class, the size of the subset containing the intersection between the sets of arrows of Π_{Ground} and Π_C is 7 ($|\Pi_{Ground} \cap \Pi_C| = 7$), which means that both PRMs have 7 dependencies in common.

And the size of the subset containing the union between the sets of arrows inside or directed to the leaf class of Π_{Ground} and Π_C is 14 ($|\Pi_{Ground} \cup \Pi_C| = 14$).

The Jaccard similarity between the graphs of Π_{Ground} and Π_C when only the arrows inside or directed to the **Company** class are considered is $\mathcal{J} = 0,50$.

Table 5 – The Jaccard similarity between the set of arrows of Π_{Ground} and Π_C considering only the dependencies inside the *leaf class* **Company**.

Arrows in Π_C	11
Arrows in Π_{Ground}	10
$ \Pi_{Ground} \cap \Pi_C $	7
$ \Pi_{Ground} \cup \Pi_C $	14
Jaccard Similarity	0,50

4.5 The package

Vanilla.prm is a package based on the previously described method. It is available in *github*, and installed as any regular R package (using *devtools*).²

For processing, the user must ensure that all *keys* and *foreign keys* are named the same way on every table on which it appears. If any variable does not fit this criteria, the R language provides tools to manipulate the data, in order to make it compliant with the requirements of the package.

Once the criteria for using the package are attended, the user should store the *column names* of all keys and foreign keys in a vector, which can be done, as an example, using the command:

² Further information about the package is available with its documentation, which can be found at <https://github.com/mormille/vanilla.prm>.

```
R> key.names <- c("key1", "key2", "key3")
```

The result will be stored in the object *key.names*, and will be used as an argument of the following two functions. The master table for the distinct classes in a domain are obtained by applying the function `relational.skeleton` on the classes previously loaded, and the keys stored on the *key.names* vector.

```
R> master_table <- relational.skeleton(class1, class2, class3,
key.names)
```

With the master tables, resultant from the `relational.skeleton` function, the structure the local Bayesian networks can be learned. And, for it to respect the assumptions presented in Section 3.1, a black list of arrows must also determined for each master table.

The function `structure.learn` automatically generates the black lists of arrows. Then, it applies a search algorithm (hill climbing as default) to find optimal structures through a scoring function, and it can be run with the following command:

```
R> dag <- structure.learn(master_table1, master_table2,
master_table3, key.names, class1, class2, class3)
```

The first master table assigned as an argument on the function determines the leaf class. The order in which the remaining master tables are assigned to the function will define the “class stratification” order.

The Bayesian information criterion (BIC) is set as the default scoring function, since it has performed better than other scores during the experiments presented in Chapter 4. Learning a Bayesian network is known to be NP-hard, with the number of possible structures growing exponentially with the number of variables, making it advantageous to use BIC (LIU; MALONE; YUAN, 2012). However, one may choose other search algorithms and scoring functions, and transmit it as an argument of the `structure.learn` function.

By running the above function, a direct acyclic graph will be returned and plotted, respecting the restrictions that will allow it to be isomorphic to a PRM. In practice, it is also possible for the user of this method, to elicit a Bayesian network structure based on specialist knowledge over the domain of study, as long as the edges in the black list are not used.

The parameters over the dependency structure can be learned by using the function `parameters.learn`. It takes the DAG learned on the previous step as a first argument, and the master tables as the following arguments. It can be run with the following command:

```
R> fit <- parameter.learn(dag, master_table1, master_table2,
master_table3)
```

After these steps, a PRM structure for a relational domain is completed, and can be used for inference on any attribute of the leaf class.

5 Conclusions

The goal of this work was to devise a method for learning probabilistic relational models, apply it in a real large-scale problem and built a software package in the R language that supports our method.

In spite of the success of Bayesian networks over the past decades on a great number of research and real-world problems, its lack of concepts of objects and relations hinders its application to model relational domains.

However, exploring the relational structure of databases allows a better understanding of the domain it represents. As shown in this work, Probabilistic Relational Models (PRM) are an extension of Bayesian Networks. PRMs allows concepts of objects and the relations held between them. Just like a Bayesian network, a PRM also specifies a template for a probability distribution.

Nonetheless, as presented in this work, learning a PRM structure from relational data is a nontrivial task, with three main difficulties arising: establishing what are the legal dependency structures for a given domain; score the possible legal structures; and search the space of possible structures.

The method for learning probabilistic relational models proposed in this work was named Vanilla-PRM. It heuristically reduces the space of possible structures, allowing the search for a PRM structure using traditional Bayesian networks scoring and search methods. The restriction of possible candidate structures was based on four different assumption, that were taken into account based on practical applications of PRMs. The assumptions are as follows:

Assumption 1: the *class dependency graph* G_π is not allowed to have cycles.

Assumption 2: the *instance dependency graph* G_σ is not allowed to have cycles at a class level.

Assumption 3: given the attribute(s) of interest, the user must specify the *leaf class* beforehand.

Assumption 4: attributes on the leaf class are not allowed to parent attributes from distinct classes.

It is understood that reducing the attention to a subset of PRMs that are sensible in such a scenario could result in the restriction of good candidate structures. However, even though this approach may not cover all possible cases, it was shown that it can be useful whenever the variables of interest (for prediction) are in the same class of objects.

Preliminary experiments were published in the *Symposium on Knowledge Discovery, Mining and Learning*, KDMiLe 2017 (MORMILLE; COZMAN, 2017), and the relational schema and skeleton for that case study are displayed in Figure 3. The attribute of interest the model predicts is the **Person.Social_class**.

In those preliminary experiments, a first version of Vanilla-PRM was adopted. It contemplated a fifth assumption, specifying that types $1 \rightarrow n$, $n \rightarrow 1$ and $n \rightarrow n$ dependencies could only exist if directed to the leaf class. While the fifth assumption further reduces the space of possible structures, it also lessens the computational cost associated with the method.¹

In the experiments presented in (MORMILLE; COZMAN, 2017), a PRM structure was learned (shown in Figure 6), and the resulting relational model was compared with a Bayesian network learned using attributes from **Person**, outperforming it, when comparing the misclassification rate of both models.

After the preliminary experiments, the algorithm was further enhanced² and its performance evaluated by extensive experimentation, and a broader study using both data provided by *Serasa Experian* and IBGE was conducted.

Two PRM structures were learned using the proposed method: The first took **Person** as the leaf class, and was denoted Π_P ; The second had **Company** as the leaf class, and was referred as Π_C .

The Π_P model was used to predict the attribute **Person.Social_class**, whose value space is $\{A, B, C, D\}$, in a 10-fold cross validation.

The performance of the Π_P was compared with a PRM learned using the method presented in Section 2.5.2, named Π_{Ground} . Seven distinct metrics were chosen to evaluate and compare the performance of both models. In the end, no significant difference in performance was noted between the models.

The second model, Π_C , was used to predict the attribute **Company.Presumed_revenue**, whose value space is $\{"Up to 200K", "200K to 1mi", "1mi to 10mi", "10mi to 100mi", "100mi or more"\}$, in a 10-fold cross validation.

The performance of the Π_C was also compared with Π_{Ground} using the same seven metrics. In this second experiment, in overall, Π_C outperformed Π_{Ground} .

All the experiments in this research were conducted using a freely available package in the R language developed during the course of this research to support the proposed method for PRM learning.

¹ In practice, the fifth assumption allows one to learn a PRM structure by computing only one *master table*, for the *leaf class*.

² The main change in the method was that the fifth assumption was discarded.

Ultimately, even with the possible restriction of good candidate structures, the proposed method has been a valuable option that reduces computational cost, reduces implementation time and produces similar results when compared to the traditional method.

However, this entire research was conducted using only the complete cases in the assembled database, therefore, a line of research that arises from this work is to further develop the proposed method to perform structure and parameter learning with missing data. Another topic that should be pursued is to perform parameter learning on the proposed method using Bayesian inference.

During the course of this research, it was attempted to learn the parameters of a PRM using *Edward*. *Edward* is a library in the *Python* programming language, built on *TensorFlow*, for purposes of probabilistic modeling, inference, and criticism. However, despite the efforts, no significant progress was made on this line of the research, and it still remains an open field for future researches.

Bibliography

- CARNAP, R. Logical foundations of probability. University of Chicago press Chicago, 1962. Citado na página 18.
- CHICKERING, D. M. Learning Bayesian networks is NP-complete. In: *Learning from data*. [S.l.]: Springer, 1996. p. 121–130. Citado na página 33.
- COPI, I.; COHEN, C.; FLAGE, D. *Essentials of Logic*. [S.l.]: Taylor & Francis, 2016. Citado na página 19.
- DARWICHE, A. *Modeling and Reasoning with Bayesian networks*. [S.l.]: Cambridge University Press, 2009. Citado na página 28.
- DŽEROSKI, S. Multi-relational data mining: an introduction. *ACM SIGKDD Explorations Newsletter*, ACM, v. 5, n. 1, p. 1–16, 2003. Citado 3 vezes nas páginas 15, 23, and 25.
- DZEROSKI, S. Inductive logic programming in a nutshell. *Introduction to Statistical Relational Learning [16]*, 2007. Citado 3 vezes nas páginas 21, 22, and 24.
- DŽEROSKI, S.; LAVRAČ, N. An introduction to inductive logic programming. In: *Relational data mining*. [S.l.]: Springer, 2001. p. 48–73. Citado na página 21.
- DŽEROSKI, S.; MUGGLETON, S.; RUSSELL, S. PAC-learnability of determinate logic programs. In: ACM. *Proceedings of the fifth annual workshop on Computational learning theory*. [S.l.], 1992. p. 128–135. Citado na página 22.
- EMDE, W.; WETTSCHERECK, D. Relational instance-based learning. In: *ICML*. [S.l.: s.n.], 1996. v. 96, p. 122–130. Citado na página 25.
- FREEDMAN, D. A. Notes on the Dutch Book Argument. *Lecture Notes, Department of Statistics, University of Berkley at Berkley*, <http://www.stat.berkeley.edu/~census/dutchdef.pdf>, 2003. Citado na página 18.
- FRIEDMAN, N. et al. Learning probabilistic relational models. In: *IJCAI*. [S.l.: s.n.], 1999. v. 99, p. 1300–1309. Citado 8 vezes nas páginas 15, 29, 31, 32, 33, 34, 52, and 58.
- GETOOR, L. Multi-relational data mining using probabilistic relational models: research summary. In: CITESEER. *In Proceedings of the First Workshop in Multi-relational Data Mining*. [S.l.], 2001. Citado na página 35.
- GETOOR, L.; SAHAMI, M. et al. Using probabilistic relational models for collaborative filtering. In: *Workshop on Web Usage Analysis and User Profiling (WEBKDD'99)*. [S.l.: s.n.], 1999. Citado na página 15.
- GETOOR, L.; TASKAR, B. *Introduction to statistical relational learning*. [S.l.]: MIT press, 2007. Citado 7 vezes nas páginas 15, 28, 30, 31, 32, 33, and 35.
- GOETHALS, B.; BUSSCHE, J. Van den. Relational association rules: getting w armer. In: *Pattern Detection and Discovery*. [S.l.]: Springer, 2002. p. 125–139. Citado 2 vezes nas páginas 23 and 24.

- HACKING, I. *An introduction to probability and inductive logic*. [S.l.]: Cambridge university press, 2001. Citado na página 18.
- HECKERMAN, D. A tutorial on learning with Bayesian networks. In: *Learning in graphical models*. [S.l.]: Springer, 1998. p. 301–354. Citado na página 33.
- HUGHES, G. E.; CRESSWELL, M. J. *A new introduction to modal logic*. [S.l.]: Psychology Press, 1996. Citado na página 20.
- (IBGE), I. B. de Geografia e E. *Base de informações do Censo Demográfico 2010: resultados do universo por setor censitário*. [S.l.]: IBGE Rio de Janeiro, 2011. Citado 2 vezes nas páginas 47 and 48.
- KOHAVI, R. et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: MONTREAL, CANADA. *Ijcai*. [S.l.], 1995. v. 14, n. 2, p. 1137–1145. Citado na página 50.
- KOLLER, D. Probabilistic relational models. In: SPRINGER. *International Conference on Inductive Logic Programming*. [S.l.], 1999. p. 3–13. Citado 5 vezes nas páginas 15, 31, 33, 52, and 58.
- KOLLER, D.; FRIEDMAN, N. *Probabilistic graphical models: principles and techniques*. [S.l.]: MIT press, 2009. Citado 2 vezes nas páginas 26 and 43.
- KOLLER, D. et al. Graphical models in a nutshell. *Introduction to statistical relational learning*, MIT Press, p. 13–55, 2007. Citado na página 43.
- KOSKI, T. J.; NOBLE, J. A review of Bayesian networks and structure learning. *Mathematica Applicanda*, v. 40, n. 1, p. 51–103, 2012. Citado na página 28.
- KOSUB, S. A note on the triangle inequality for the Jaccard distance. *arXiv preprint arXiv:1612.02696*, 2016. Citado na página 53.
- KOTSIANTIS, S.; KANELLOPOULOS, D.; PINTELAS, P. Data preprocessing for supervised learning. *International Journal of Computer Science*, Citeseer, v. 1, n. 2, p. 111–117, 2006. Citado na página 45.
- KRAJICEK, J. *Bounded arithmetic, propositional logic and complexity theory*. [S.l.]: Cambridge University Press, 1995. Citado na página 20.
- KRAMER, S.; LAVRAČ, N.; FLACH, P. Propositionalization approaches to relational data mining. In: *Relational data mining*. [S.l.]: Springer, 2001. p. 262–291. Citado na página 23.
- LAER, W. V.; RAEDT, L. D. How to upgrade propositional learners to first order logic: A case study. In: *Relational data mining*. [S.l.]: Springer, 2001. p. 235–261. Citado na página 23.
- LAVRAČ, N.; DŽEROSKI, S.; GROBELNIK, M. Learning nonrecursive definitions of relations with LINUS. In: SPRINGER. *European Working Session on Learning*. [S.l.], 1991. p. 265–281. Citado na página 22.
- LIU, Z.; MALONE, B.; YUAN, C. Empirical evaluation of scoring functions for Bayesian network model selection. *BMC bioinformatics*, BioMed Central, v. 13, n. 15, p. S14, 2012. Citado na página 73.

- MORMILLE, L.; COZMAN, F. Learning Probabilistic Relational Models: A Simplified Framework, a Case Study, and a Package. 2017. Citado 5 vezes nas páginas 16, 40, 41, 48, and 75.
- MUGGLETON, S. Inductive logic programming. *New generation computing*, Springer, v. 8, n. 4, p. 295–318, 1991. Citado na página 22.
- MURPHY, K. P. *Machine learning: a probabilistic perspective*. [S.l.]: MIT press, 2012. Citado na página 17.
- POWERS, D. M. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. Bioinfo Publications, 2011. Citado 2 vezes nas páginas 51 and 52.
- RAEDT, L. D. Logical settings for concept-learning. *Artificial Intelligence*, Elsevier, v. 95, n. 1, p. 187–201, 1997. Citado na página 22.
- RAEDT, L. D.; KERSTING, K. Probabilistic inductive logic programming. In: *Probabilistic Inductive Logic Programming*. [S.l.]: Springer, 2008. p. 1–27. Citado na página 22.
- STERNBERG, R. J.; STERNBERG, K. *Cognitive psychology*. [S.l.]: Nelson Education, 2016. Citado na página 19.
- VOLOVICI, D. Evaluation of Classification in More Than Two Classes. *International Journal of Advanced Statistics and IT&C for Economics and Life Sciences*, v. 6, n. 2, 2016. Citado na página 51.
- WANNER, E.; SHINER, S. Measuring transient memory load. *Journal of Verbal Learning and Verbal Behavior*, Elsevier, v. 15, n. 2, p. 159–167, 1976. Citado na página 20.
- WHORF, B. L. Language, Thought, and Reality: Selected Writings of Benjamin Lee Whorf (Edited by John B. Carroll.). Technology Press of MIT, 1956. Citado na página 20.
- WROBEL, S. Inductive logic programming for knowledge discovery in databases. In: *Relational data mining*. [S.l.]: Springer, 2001. p. 74–101. Citado na página 21.

A Person attribute description

This appendix describes the variables in the **Person** class.

1 - *Person_ID*: the first attribute on class **Person** is a hashed ID, unique for each object. Its role is provide an unique identifier for every citizen in the table, and can be considered a “key” attribute.

2 - *Sector_ID*: every object on **Person**’s table is associated with another fixed attribute, the ID of the census Sector where a given citizen lives. This attribute is also a *foreign key* to another class in the Atibaia domain, the **Census_sector** class.

3 - *Gender*: the first non-fixed attribute on this class indicates a person’s gender. Every object in **Person** represents a citizen of Atibaia, and citizen have a declared gender, that ca be either *female* or *male*. A description of this variable can be seen on Table 6.

Table 6 – A description of the attribute **Person.Gender**.

Class: **Person**
 Attribute: *Gender*
 Description: Person’s gender.

Levels	Absolute Frequency	Marginal Probability
Female	42.989	0,51
Male	41.189	0,49

4 - *Age*: the next attribute considered in the study was the age group of the objects in the **Person** class. This attribute have 10 distinct levels corresponding to age intervals. A description of it is shown on Table 7.

5 - *Public server*: this attribute is a flag indicating if the object is a public server. A public server is an employee of a State Company. It covers those who have an employment relationship with any government entity, as well as its respective municipalities and foundations. A description of this variable is shown on Table 8.

6 - *Bolsa Família*: this attribute is a flag indicating if a citizen is beneficiary of the *Bolsa Família* program. It consists of a direct income transfer Federal program directed to families living in poverty and extreme poverty throughout the country, so that they can overcome the situation of vulnerability and poverty. The program seeks to guarantee these families the right to food and access to education and health. Throughout Brazil, more than 13.9 million families are served by the *Bolsa Família* program. A description of this variable is shown on Table 9.

Table 7 – A description of the attribute **Person**.*Age_group*.

Class: **Person**
 Attribute: *Age_group*
 Description: Age group that a given person belongs.

Levels	Absolute Frequency	Marginal Probability
18 to 23	6.076	0,07
24 to 28	9.108	0,11
29 to 33	10.084	0,12
34 to 38	10.313	0,12
39 to 43	9.024	0,11
44 to 48	8.044	0,10
49 to 53	7.490	0,09
54 to 58	6.429	0,08
59 to 65	7.228	0,09
66 or more	10.382	0,12

Table 8 – A description of the attribute **Person**.*Public_server*.

Class: **Person**
 Attribute: *Public_server*
 Description: A flag indicating whether a given person is a public employee.

Levels	Absolute Frequency	Marginal Probability
No	82.900	0,98
Yes	1.278	0,02

Table 9 – A description of the attribute **Person**.*Bolsa_familia_flag*.

Class: **Person**
 Attribute: *Bolsa_familia_flag*
 Description: A flag indicating whether a person is a beneficiary of the program *Bolsa Família*.

Levels	Absolute Frequency	Marginal Probability
No	81.529	0,97
Yes	2.649	0,03

7 - *Company owner*: this attribute is a flag indicating if the object owns a company, or part of a company. Even though the domain of study is the city of Atibaia, residents of Atibaia that own a business outside the city limits are also flagged as company owners. A description of this variable is shown on Table 10.

Table 10 – A description of the attribute **Person**.*Company_owner_flag*.

Class: **Person**
 Attribute: *Company_owner_flag*
 Description: A flag indicating whether a person owns a company.

Levels	Absolute Frequency	Marginal Probability
No	64.124	0,76
Yes	20.054	0,24

8 - *Rural producer flag*: this attribute is a flag indicating if the a person is a rural producer. A rural producer is a person who exploits the land for economic or subsistence purposes through agriculture, livestock, forestry, sustainable extractivism, aquaculture, and non-agricultural activities, respecting the social function of the land. A rural producer does not necessarily needs to own a company for this purpose. A description of this variable is shown on Table 11.

Table 11 – A description of the attribute **Person**.*Rural_producer_flag*.

Class: **Person**
 Attribute: *Rural_producer_flag*
 Description: A flag indicating whether a person is a rural producer.

Levels	Absolute Frequency	Marginal Probability
No	83.964	0,997
Yes	214	0,003

9 - *Social class*: social classes are groups of people with similar cultural, political and economic standards. The financial factor is one of the most striking characteristics in the definition of a social class. Social classes can be interpreted by different points of view, however, the most usual definition refers to the limiting group of individuals that possesses a similar level of economic power and access to opportunities. A description of this variable is shown on Table 12.

Table 12 – A description of the attribute **Person.Social_class**.

Class: **Person**
 Attribute: *Social_class*
 Description: The social class of a given person.

Levels	Absolute Frequency	Marginal Probability
A	1.930	0,02
B	44.190	0,52
C	14.697	0,17
D	23.361	0,28

10 - Consumption index: the consumption index of a person reveals its buying and shopping habits. This attribute is inferred by Serasa Experian given a person history of use of credit for purposes related to the acquisition of goods (durable or non-durable). A description of this variable is shown on Table 13.

Table 13 – A description of the attribute **Person.Consumption_index**.

Class: **Person**
 Attribute: *Consumption_index*
 Description: How frequently a given person buy goods in instalments.

Levels	Absolute Frequency	Marginal Probability
Inactive	29.202	0,35
Low	44.484	0,53
Medium	9.808	0,12
High	684	0,01

11 - Credit risk: the credit risk is a result of the payment habits and relationship of the citizen with the credit market. Each person is scored by Serasa Experian according to an analysis of a number of factors, such as: payment of bills on time; history of negative debts; financial relationship with companies. A description of this variable is shown on Table 14.

12 - Presumed income: a calculation based on the cadastral and behavioral information available in the Serasa Experian database to presume the income range of a citizen. Its essence is the statistical modeling of data to obtain interval of income. A description of this variable is shown on Table 15.

Table 14 – A description of the attribute **Person.Credit_risk**.

Class: **Person**
 Attribute: *Credit_risk*
 Description: The risk of granting credit to a given person.

Levels	Absolute Frequency	Marginal Probability
Very low	22.318	0,27
Low	17.827	0,21
Medium	18.094	0,21
High	4.856	0,06
Very high	21.083	0,25

Table 15 – A description of the attribute **Person.Presumed_income**.

Class: **Person**
 Attribute: *Presumed_income*
 Description: The presumed monthly income group that a given person belongs (in Brazilian Reais).

Levels	Absolute Frequency	Marginal Probability
Up to 1.000	47.260	0,561
1.000 to 1.500	12.952	0,154
1.500 to 2.000	5.330	0,063
2.000 to 2.500	5.468	0,065
2.500 to 3.000	2.741	0,033
3.000 to 4.000	1.882	0,022
4.000 to 5.000	4.439	0,053
5.000 to 6.000	1.048	0,012
6.000 to 7.000	1.151	0,014
7.000 to 8.000	211	0,003
8.000 to 9.000	87	0,001
9.000 or more	1.609	0,019

13 - Mosaic: a clusterization of the Brazilian population, classifying it in 11 groups, considering financial, geographical, demographic, consumption, behavioral and lifestyle aspects. A description of this variable is shown on Table 16. The 11 Mosaic groups are describe by Serasa Experian as follows:

- A. **Brazilian elites:** successful entrepreneurs and executives that live the comforts afforded by high income: luxury automobiles, international travel, restaurants and exclusive products.

- B. **Experienced urban citizens living comfortably:** with more than 50 years, they are well established professionals or already retired. They now enjoy the comfortable standard of living earned from work. They live in urban areas, especially the coasts.
- C. **Youth urban worker:** with up to 35, they are young and in the beginning of their careers, but still seeking to increase their education, which is already superior to that of their parents. They are optimistic and alert, with access to technology and eyeing trends.

Table 16 – A description of the attribute **Person**.*Mosaic*.

Class: **Person**
 Attribute: *Mosaic*
 Description: The Mosaic group of
 a given person.

Levels	Absolute Frequency	Marginal Probability
A	4.660	0,055
B	3.577	0,042
C	9.699	0,115
D	21.587	0,256
E	11.821	0,140
F	3.384	0,040
G	10.163	0,121
H	6.447	0,077
I	6.529	0,078
J	326	0,004
K	5.985	0,071

- D. **Youth from the periphery:** the limited access to education and the infrastructure of the neighborhoods where they live make the routine of these young people more difficult. However, they have seen life improve and believe in a better future.
- E. **Established urban adults:** the Brazilian average adult with good schooling and effort. They have achieved a stable professional and financial life, although without luxuries. They are more cautious consumers.
- F. **Getting old in the 21st century:** the growing group of middle-aged seniors now enjoy better conditions due to retirement income and greater access to health services. Still, they feel nostalgia for the simpler times.
- G. **Business owners:** the dream of being their own boss came true for them. They are small and medium entrepreneurs who invested their economies and began to see the result, albeit with some instabilities.

- H. **Urban workers:** forming the mass of workers with low schooling and income, they live the advantages and disadvantages of large cities: access to consumption, information, mobility problems and high cost of living.
- I. **Residents of deprived areas of the south and southeast:** workers with low schooling and income, living in precarious areas, with difficult access to public spaces and services. In recent years, they have seen their consumer power grow, going beyond their immediate needs.
- J. **Inhabitants of precarious areas:** men and women living close to the poverty line and therefore dependent on social programs. Low income and schooling are aggravated by being in regions with restricted access to public services.
- K. **Inhabitants of rural areas:** they lived their entire lives in rural areas, and this defines a lot about them. Work for agribusiness or farming for own consumption, difficulty accessing education, other public services and love of land.

14 - *Luxury*: this attribute is a flag indicating whether a person is prone to consume luxury items. To that end, data from Serasa Experian's portfolio on preferences, behavior, demographic characteristics, and the history of the use of credit for this particular purpose are used to model people with greater potential to consume certain luxury items. A description of this variable is shown on Table 17.

Table 17 – A description of the attribute **Person**.*Propensity_for_luxury*.

Class: **Person**
 Attribute: *Propensity_for_luxury*
 Description: A flag indicating whether a person is prone to buy luxury items.

Levels	Absolute Frequency	Marginal Probability
No	79.090	0,94
Yes	5.088	0,06

15 - *Credit Card*: this attribute is a flag indicating whether a person is prone to own a credit card. To that end, data from Serasa Experian's portfolio on preferences, behavior, demographic characteristics, and the history of the use of credit for this particular purpose are used to model people with greater potential to own at least one credit card. A description of this variable is shown on Table 18.

Table 18 – A description of the attribute **Person.Propensity_for_credit_card**.

Class: **Person**
 Attribute: *Propensity_for_credit_card*
 Description: A flag indicating whether a person is prone to have credit card.

Levels	Absolute Frequency	Marginal Probability
No	64.708	0,77
Yes	19.470	0,23

16 - Mortgage Loan: this attribute is a flag indicating whether a person is prone to require a mortgage loan, either to raise funds to purchase real estate, or alternatively to raise funds for any purpose, while putting a lien on a previously owned property. To that end, data from Serasa Experian’s portfolio on preferences, behavior, demographic characteristics, and the history of the use of credit for this particular purpose are used to model people with greater potential to require a mortgage loan. A description of this variable is shown on Table 19.

Table 19 – A description of the attribute **Person.Propensity_for_mortgage**.

Class: **Person**
 Attribute: *Propensity_for_mortgage*
 Description: A flag indicating whether a person is prone to have a mortgage loan.

Levels	Absolute Frequency	Marginal Probability
No	66.926	0,80
Yes	17.252	0,20

17 - Broadband Internet: this attribute is a flag indicating whether a person is prone to consume a broadband internet service. To that end, data from Serasa Experian’s portfolio on preferences, behavior, demographic characteristics, and the history of the use of credit for this particular purpose are used to model people with greater potential consume broadband internet. A description of this variable is shown on Table 20.

18 - Personal Loan: this attribute is a flag indicating whether a person is prone to require a personal loan. To that end, data from Serasa Experian’s portfolio on preferences, behavior, demographic characteristics, and the history of the use of credit for this particular purpose are used to model people with greater potential to require a personal loan. A description of this variable is shown on Table 21.

Table 20 – A description of the attribute **Person.Propensity_for_broadband_internet**.

Class: **Person**
 Attribute: *Propensity_for_broadband_internet*
 Description: A flag indicating whether a person is prone to purchase broadband internet.

Levels	Absolute Frequency	Marginal Probability
No	35.665	0,42
Yes	48.513	0,58

Table 21 – A description of the attribute **Person.Propensity_for_personal_loan**.

Class: **Person**
 Attribute: *Propensity_for_personal_loan*
 Description: A flag indicating whether a person is prone to require a personal loan at a financial institution.

Levels	Absolute Frequency	Marginal Probability
No	53.614	0,64
Yes	30.564	0,36

19 - Postpaid Phone: this attribute is a flag indicating whether a person is prone to consume a postpaid mobile phone service. To that end, data from Serasa Experian's portfolio on preferences, behavior, demographic characteristics, and the history of the use of credit for this particular purpose are used to model people with greater potential to consume a postpaid mobile phone service. A description of this variable is shown on Table 21.

Table 22 – A description of the attribute **Person.Propensity_for_postpaid_phone**.

Class: **Person**
 Attribute: *Propensity_for_postpaid_phone*
 Description: A flag indicating whether a person is prone to purchase a postpaid phone plan.

Levels	Absolute Frequency	Marginal Probability
No	42.014	0,499
Yes	42.164	0,501

B Company attribute description

This appendix describes the variables in **Company**.

1 - *Company_ID*: the first attribute on **Company** is a hashed ID, unique for each object. Its role is provide an unique identifier for every business in the table, and can be considered a “key” attribute.

2 - *Sector_ID*: every object on that table is associated with another fixed attribute, the ID of the census sector where the given business is located. This attribute is also a foreign to the another class in our domain, the **Census_sector** class.

3 - *Age*: the first non-fixed attribute considered in this class is the age group of the objects in the **Company**. This attribute have 6 distinct levels corresponding to the age intervals. A description of this variable is shown on Table 23.

Table 23 – A description of the attribute **Company.Age_band**.

Class: **Company**
 Attribute: *Age_band*
 Description: The age group of a company,
 counting from the foundation date.

Levels	Absolute Frequency	Marginal Probability
Up to 1 year	1.505	0,10
1 to 3 years	2.807	0,19
3 to 5 years	2.126	0,14
5 to 10 years	3.481	0,23
10 to 20 years	3.354	0,22
20 years or more	1.859	0,12

4 - *Range of Employees*: this attribute indicates the number of employees employed by the company. It has 4 levels, each indicating a range of employees. A description of this variable is shown on Table 24.

5 - *Presumed Revenue*: since most of the active Brazilian companies do not publish a financial report, Serasa Experian uses its data portfolio to presume a business revenue on the last 12 months. A description of this variable is shown on Table 25.

6 - *Operating Index*: the operating index measures a company’s performance based on financial metrics and key performance indicators (KPIs), and compare it with the average performance of the Domain over a period of time. A description of this variable is shown on Table 26.

Table 24 – A description of the attribute **Company**.*Range_of_employees*.

Class: **Company**
 Attribute: *Range_of_employees*
 Description: The range of the number of employees that work on the company

Levels	Absolute Frequency	Marginal Probability
Up to 9	13.988	0,924
10 to 49	896	0,059
50 to 199	222	0,015
200 or more	26	0,002

Table 25 – A description of the attribute **Company**.*Presumed_revenue*.

Class: **Company**
 Attribute: *Presumed_revenue*
 Description: The presumed revenue of the company (in Brazilian Reais).

Levels	Absolute Frequency	Marginal Probability
Up to 200k	11.186	0,74
200k to 1mi	2.791	0,18
1mi to 10mi	960	0,06
10mi to 100mi	105	0,01
100mi or more	96	0,01

Table 26 – A description of the attribute **Company**.*Operating_index*.

Class: **Company**
 Attribute: *Operating_index*
 Description: The operating index of the company.

Levels	Absolute Frequency	Marginal Probability
Very low	932	0,06
Low	2.317	0,15
Average	3.583	0,24
High	5.998	0,40
Very high	2.302	0,15

7 - Rural Producer: this attribute is a flag indicating if the a company is a rural producer. A rural producer is a company who exploits the land for economic purposes

through agriculture, livestock, forestry, sustainable extractivism, aquaculture, and non-agricultural activities, respecting the social function of the land. A description of this variable is shown on Table 27.

Table 27 – A description of the attribute **Company**.*Rural_producer_flag*.

Class: **Company**
Attribute: *Rural_producer_flag*
Description: Flag indicating if the company produces rural products.

Levels	Absolute Frequency	Marginal Probability
No	14.131	0,93
Yes	998	0,07

C Census sector attribute description

This appendix describes the variables in **Census_sector**.

1 - Sector_ID: every object on **Census_sector** is associated with a fixed attribute, indicating the ID of the census sector. Its role is provide an unique identifier for every census sector in the table, and can be considered a “key” attribute.

2 - Situation: a census sector can be classified as urban or rural. Urban situation census sectors are those located inside the urban perimeter of a city, and the rural situation census sectors covers all areas outside these boundaries. A description of this variable is shown on Table 28.

Table 28 – A description of the attribute **Census_sector.Situation**.

Class: **Census_sector**
 Attribute: *Situation*
 Description: If the census sector is located in an urban or a rural area.

Levels	Absolute Frequency	Marginal Probability
Urban	181	0,55
Rural	146	0,45

3 - Household: the household of a domicile is the number of people living under the same roof, either composing a family or a social unit composed of those living together in the same domicile. The average household of a census sector is taken by dividing the total number of residents of the census sector, by the number of domiciles in the census sector. A description of this variable is shown on Table 29.

Table 29 – A description of the attribute **Census_sector.Average_house_hold**.

Class: **Census_sector**
 Attribute: *Average_house_hold*
 Description: The average house hold of the domiciles in the census sector

Levels	Absolute Frequency	Marginal Probability
<3,0	70	0,21
3,0 to 3,5	156	0,48
>= 3,5	101	0,31

4 - Average Income: this attribute indicates the value of the census sector average monthly nominal income (in Brazilian Reais) of persons with 10 years of age or more (income and income). A description of this variable is shown on Table 30.

Table 30 – A description of the attribute **Census_sector.Average_income**.

Class: **Census_sector**
 Attribute: *Average_income*
 Description: Average income of the residents of the census sector (in Brazilian reais).

Levels	Absolute Frequency	Marginal Probability
<500	48	0,15
500 to 750	123	0,38
750 to 1250	82	0,25
>= 1250	74	0,23

5 - Domiciles: number of domiciles inside the census sector. A domicile is a permanent home where at least one person lives in. A description of this variable is shown on Table 31.

Table 31 – A description of the attribute **Census_sector.Domiciles**.

Class: **Census_sector**
 Attribute: *Domiciles*
 Description: Number of domiciles inside the area of the census sector

Levels	Absolute Frequency	Marginal Probability
<100	148	0,45
100 to 200	112	0,35
200 to 300	59	0,18
>= 300	8	0,02

6 - Houses: percentage of domiciles inside the census sector that are classified as houses. A house is a building that serves as a single home. The value of this attribute was computed by taking the number of domiciles classified as houses in a census sector and dividing it by the total number of domiciles inside that same census sector. A description of this variable is shown on Table 32.

Table 32 – A description of the attribute **Census_sector.Houses**.

Class: **Census_sector**
 Attribute: *Houses*
 Description: Percentage of domiciles
 of type house.

Levels	Absolute Frequency	Marginal Probability
0% to 25%	8	0,02
26% to 50%	4	0,01
51% to 75%	12	0,04
76% to 100%	303	0,93

7 - Apartment: percentage of domiciles inside the census sector that are classified as apartments. An apartment is a part of a residential building that serves as a home. The value of this attribute was computed by taking the number of domiciles classified as apartments in a census sector and dividing it by the total number of domiciles inside that same census sector. A description of this variable is shown on Table 33.

Table 33 – A description of the attribute **Census_sector.Apartment**.

Class: **Census_sector**
 Attribute: *Apartment*
 Description: Percentage of domiciles
 of type apartment.

Levels	Absolute Frequency	Marginal Probability
0% to 25%	320	0,98
26% to 100%	7	0,02

8 - Paid Domiciles: percentage of domiciles inside the census sector that are already fully paid by its owners. These can be either house or apartment domiciles. The value of this attribute was computed by taking the number of fully paid domiciles in a census sector and dividing it by the total number of domiciles inside that same census sector. A description of this variable is shown on Table 34.

9 - Unpaid Domiciles: percentage of domiciles inside the census sector that are currently being paid by its owners. These can be either house or apartment domiciles. The value of this attribute was computed by taking the number of domiciles under payment in a census sector and dividing it by the total number of domiciles inside that same census sector. A description of this variable is shown on Table 35.

Table 34 – A description of the attribute **Census_sector.Paid_domiciles**.

Class: **Census_sector**
 Attribute: *Paid_domiciles*
 Description: Percentage of domiciles owned by the residents and fully paid.

Levels	Absolute Frequency	Marginal Probability
0% to 25%	22	0,07
26% to 50%	68	0,21
51% to 75%	192	0,59
76% to 100%	45	0,14

Table 35 – A description of the attribute **Census_sector.Unpaid_domiciles**.

Class: **Census_sector**
 Attribute: *Unpaid_domiciles*
 Description: Percentage of domiciles owned by the residents that are currently being paid.

Levels	Absolute Frequency	Marginal Probability
0% to 25%	322	0,985
26% to 50%	4	0,012
51% to 100%	1	0,003

10 - Rented Domiciles: percentage of domiciles inside the census sector that are currently being rented by its residents. These can be either house or apartment domiciles. The value of this attribute was computed by taking the number of rented domiciles in a census sector and dividing it by the total number of domiciles inside that same census sector. A description of this variable is shown on Table 36.

Table 36 – A description of the attribute **Census_sector.Rented_domiciles**.

Class: **Census_sector**
 Attribute: *Rented_domiciles*
 Description: Percentage of domiciles that are being rented by the residents.

Levels	Absolute Frequency	Marginal Probability
0% to 25%	224	0,69
26% to 50%	100	0,30
51% to 100%	3	0,01

11 - *Water Supply*: percentage of domiciles inside the census sector that have water supply from the general supply network. These can be either house or apartment domiciles. The value of this attribute was computed by taking the number of domiciles with water supply (from the general supply network) in a census sector and dividing it by the total number of domiciles inside that same census sector. A description of this variable is shown on Table 37.

Table 37 – A description of the attribute **Census_sector**.*Water_supply*.

Class: **Census_sector**
 Attribute: *Water_supply*
 Description: Percentage of domiciles that have water supply from the general supply network.

Levels	Absolute Frequency	Marginal Probability
0% to 50%	164	0,50
50% to 75%	5	0,02
76% to 99%	65	0,20
100%	93	0,28

12 - *Sewage network*: percentage of domiciles inside the census sector that have access to the general sewage network. These can be either house or apartment domiciles. The value of this attribute was computed by taking the number of domiciles in a census sector with access to the general sewage network and dividing it by the total number of domiciles inside that same census sector. A description of this variable is shown on Table 38.

Table 38 – A description of the attribute **Census_sector**.*Sewage_network*.

Class: **Census_sector**
 Attribute: *Sewage_network*
 Description: Percentage of domiciles that have access to the general sewage network.

Levels	Absolute Frequency	Marginal Probability
0% to 50%	218	0,67
50% to 75%	7	0,02
76% to 99%	69	0,21
100%	33	0,10

13 - *Garbage Collection*: percentage of domiciles inside the census sector that have access to the public garbage collection service. These can be either house or apartment domiciles. The value of this attribute was computed by taking the number of domiciles in a census sector with access to the public garbage collection service and dividing it by the total number of domiciles inside that same census sector. A description of this variable is shown on Table 39.

Table 39 – A description of the attribute **Census_sector**.*Garbage_collection*.

Class: **Census_sector**
 Attribute: *Garbage_collection*
 Description: Percentage of domiciles that have access to the public garbage collection service.

Levels	Absolute Frequency	Marginal Probability
0% to 50%	6	0,02
50% to 75%	14	0,04
76% to 99%	68	0,21
100%	239	0,73

14 - *Electricity*: percentage of domiciles inside the census sector that have access to electricity. These can be either house or apartment domiciles. The value of this attribute was computed by taking the number of domiciles in a census sector with access to electricity and dividing it by the total number of domiciles inside that same census sector. A description of this variable is shown on Table 40.

Table 40 – A description of the attribute **Census_sector**.*Electricity*.

Class: **Census_sector**
 Attribute: *Electricity*
 Description: Percentage of domiciles that have access to electricity.

Levels	Absolute Frequency	Marginal Probability
0% to 99%	23	0,07
100%	304	0,93

15 - *Female Domiciles*: percentage of domiciles inside the census sector that have only residents of the female gender. These can be either house or apartment domiciles. The value of this attribute was computed by taking the number of domiciles in a census sector with only female residents and dividing it by the total number of domiciles inside that same census sector. A description of this variable is shown on Table 41.

Table 41 – A description of the attribute **Census_sector.Female_domiciles**.

Class: **Census_sector**
 Attribute: *Female_domiciles*
 Description: Percentage of domiciles that have only female residents.

Levels	Absolute Frequency	Marginal Probability
0% to 10%	226	0,69
11% to 25%	93	0,28
26% to 100%	8	0,02

16 - Male Domiciles: percentage of domiciles inside the census sector that have only residents of the male gender. These can be either house or apartment domiciles. The value of this attribute was computed by taking the number of domiciles in a census sector with only male residents and dividing it by the total number of domiciles inside that same census sector. A description of this variable is shown on Table 42.

Table 42 – A description of the attribute **Census_sector.Male_domiciles**.

Class: **Census_sector**
 Attribute: *Male_domiciles*
 Description: Percentage of domiciles that have only male residents.

Levels	Absolute Frequency	Marginal Probability
0% to 10%	271	0,83
11% to 25%	52	0,16
26% to 100%	4	0,01

D Person PRM

This appendix introduces the dependencies in the Π_P PRM, learned in the *Person experiment*, introduced in Section 4.3.2. All the dependencies in Π_P are presented in Table D.

Table 43 – Π_P PRM Dependencies

The Person PRM Dependencies		
	From	To
1	Person.Mosaic	Person.Company_owner_flag
2	Person.Mosaic	Person.Presumed_income
3	Person.Mosaic	Person.Propensity_for_credit_card
4	Person.Propensity_for_broadband_internet	Person.Propensity_for_postpaid_phone
5	Person.Consumption_index	Person.Propensity_for_personal_loan
6	Person.Mosaic	Person.Social_class
7	Person.Propensity_for_postpaid_phone	Person.Age_group
8	Person.Mosaic	Person.Propensity_for_mortgage
9	Person.Propensity_for_personal_loan	Person.Propensity_for_broadband_internet
10	Person.Mosaic	Person.Propensity_for_luxury
11	Person.Mosaic	Person.Credit_risk
12	Person.Consumption_index	Person.Propensity_for_credit_card
13	Census_sector.Average_income	Person.Social_class
14	Census_sector.Average_income	Person.Mosaic
15	Person.Mosaic	Person.Age_group
16	Person.Presumed_income	Person.Propensity_for_mortgage
17	Person.Propensity_for_broadband_internet	Person.Mosaic
18	Person.Propensity_for_broadband_internet	Person.Age_group
19	Person.Propensity_for_personal_loan	Person.Propensity_for_postpaid_phone
20	Person.Propensity_for_mortgage	Person.Company_owner_flag
21	Person.Propensity_for_personal_loan	Person.Mosaic
22	Person.Propensity_for_broadband_internet	Person.Presumed_income
23	Person.Propensity_for_personal_loan	Person.Propensity_for_mortgage
24	Person.Presumed_income	Person.Gender.
25	Person.Consumption_index	Person.Propensity_for_broadband_internet
26	Person.Gender.	Person.Bolsa_familia_flag
27	Person.Mosaic	Person.Bolsa_familia_flag
28	Person.Consumption_index	Person.Credit_risk
29	Person.Propensity_for_personal_loan	Person.Propensity_for_credit_card
30	Person.Gender.	Person.Propensity_for_luxury
31	Census_sector.Average_income	Person.Propensity_for_personal_loan
32	Person.Mosaic	Person.Public_server
33	Census_sector.Situation	Person.Social_class
34	Person.Mosaic	Person.Propensity_for_postpaid_phone
35	Person.Propensity_for_personal_loan	Person.Propensity_for_luxury
36	Person.Age_group	Person.Gender.
37	Person.Propensity_for_personal_loan	Person.Age_group
38	Person.Presumed_income	Person.Company_owner_flag
39	Census_sector.Average_income	Person.Propensity_for_broadband_internet
40	Person.Company_owner_flag	Person.Propensity_for_luxury
41	Census_sector.Female_domiciles	Person.Mosaic
42	Census_sector.Average_income	Person.Consumption_index

43	Person.Credit_risk	Person.Rural_producer_flag
44	Person.Company_owner_flag	Person.Public_server
45	Person.Gender.	Person.Rural_producer_flag
46	Company.Age_band	Company.Operating_index
47	Company.Range_of_employees	Company.Presumed_revenue
48	Company.Operating_index	Company.Range_of_employees
49	Census_sector.Apartment	Company.Rural_producer_flag
50	Census_sector.Female_domiciles	Company.Age_band
51	Census_sector.Male_domiciles	Company.Rural_producer_flag
52	Company.Rural_producer_flag	Company.Presumed_revenue
53	Company.Rural_producer_flag	Company.Operating_index
54	Company.Rural_producer_flag	Company.Age_band
55	Company.Rural_producer_flag	Company.Range_of_employees
56	Census_sector.Female_domiciles	Company.Rural_producer_flag
57	Census_sector.Water_supply	Company.Rural_producer_flag
58	Census_sector.Situation	Census_sector.Water_supply
59	Census_sector.Water_supply	Census_sector.Sewage_network
60	Census_sector.Situation	Census_sector.Domiciles
61	Census_sector.Water_supply	Census_sector.Rented_domiciles
62	Census_sector.Sewage_network	Census_sector.Female_domiciles
63	Census_sector.Average_house_hold	Census_sector.Average_income
64	Census_sector.Female_domiciles	Census_sector.Average_house_hold
65	Census_sector.Situation	Census_sector.Garbage_collection
66	Census_sector.Houses	Census_sector.Apartment
67	Census_sector.Rented_domiciles	Census_sector.Paid_domiciles
68	Census_sector.Female_domiciles	Census_sector.Houses
69	Census_sector.Situation	Census_sector.Average_income
70	Census_sector.Situation	Census_sector.Paid_domiciles

E Ground PRM

This appendix introduces the dependencies in the Π_{Ground} PRM, learned in the *Company experiment*, introduced in Section 4.3.4. All the dependencies in Π_{Ground} are presented in Table E.

Table 44 – Π_{Ground} PRM Dependencies

The Ground PRM Dependencies		
	From	To
1	Census_sector.Female_percentage	Company.Age_band
2	Census_sector.Apartment_percentage	Company.Rural_producer_flag
3	Person.Mosaic	Company.Rural_producer_flag
4	Person.Propensity_for_postpaid_phone	Company.Rural_producer_flag
5	Census_sector.Situation	Person.Social_class
6	Census_sector.Average_income	Person.Social_class
7	Census_sector.Average_income	Person.Consumption_index
8	Census_sector.Average_income	Person.Mosaic
9	Census_sector.Female_percentage	Person.Mosaic
10	Census_sector.Average_income	Person.Propensity_for_broadband_internet
11	Census_sector.Average_income	Person.Propensity_for_personal_loan
12	Person.Age_band	Person.Mosaic
13	Person.Mosaic	Person.Company_owner_flag
14	Person.Mosaic	Person.Presumed_income
15	Person.Mosaic	Person.Propensity_for_credit_card
16	Person.Propensity_for_broadband_internet	Person.Propensity_for_postpaid_phone
17	Person.Consumption_index	Person.Propensity_for_personal_loan
18	Person.Mosaic	Person.Social_class
19	Person.Propensity_for_postpaid_phone	Person.Age_band
20	Person.Mosaic	Person.Propensity_for_mortgage_loan
21	Person.Propensity_for_personal_loan	Person.Propensity_for_broadband_internet
22	Person.Propensity_for_postpaid_phone	Person.Mosaic
23	Person.Mosaic	Person.Propensity_for_luxury
24	Person.Mosaic	Person.Credit_risk
25	Person.Consumption_index	Person.Propensity_for_credit_card
26	Person.Presumed_income	Person.Propensity_for_mortgage_loan
27	Person.Propensity_for_broadband_internet	Person.Age_band
28	Person.Propensity_for_personal_loan	Person.Age_band
29	Person.Propensity_for_broadband_internet	Person.Mosaic
30	Person.Propensity_for_personal_loan	Person.Propensity_for_postpaid_phone
31	Person.Propensity_for_mortgage_loan	Person.Company_owner_flag
32	Person.Propensity_for_broadband_internet	Person.Presumed_income
33	Person.Propensity_for_personal_loan	Person.Propensity_for_mortgage_loan
34	Person.Presumed_income	Person.Gender
35	Person.Consumption_index	Person.Propensity_for_broadband_internet
36	Person.Gender	Person.Bolsa_fam'lia_flag
37	Person.Mosaic	Person.Bolsa_fam'lia_flag
38	Person.Consumption_index	Person.Credit_risk
39	Person.Propensity_for_personal_loan	Person.Propensity_for_credit_card
40	Person.Gender	Person.Propensity_for_luxury
41	Person.Mosaic	Person.Public_server_flag
42	Person.Propensity_for_broadband_internet	Person.Social_class

43	Person.Propensity_for_personal_loan	Person.Propensity_for_luxury
44	Person.Age_band	Person.Gender
45	Person.Consumption_index	Person.Propensity_for_postpaid_phone
46	Person.Propensity_for_luxury	Person.Social_class
47	Person.Presumed_income	Person.Company_owner_flag
48	Person.Company_owner_flag	Person.Propensity_for_luxury
49	Person.Credit_risk	Person.Rural_producer_flag
50	Person.Company_owner_flag	Person.Public_server_flag
51	Person.Gender	Person.Rural_producer_flag
52	Census_sector.Situation	Census_sector.Water_supply_percentage
53	Census_sector.Water_supply_percentage	Census_sector.Sewage_network_percentage
54	Census_sector.Situation	Census_sector.Domiciles
55	Census_sector.Water_supply_percentage	Census_sector.Rented_percentage
56	Census_sector.Sewage_network_percentage	Census_sector.Female_percentage
57	Census_sector.Average_household	Census_sector.Average_income
58	Census_sector.Female_percentage	Census_sector.Average_household
59	Census_sector.Situation	Census_sector.Garbage_collection_percentage
60	Census_sector.House_percentage	Census_sector.Apartment_percentage
61	Census_sector.Rented_percentage	Census_sector.Paid_percentage
62	Census_sector.Female_percentage	Census_sector.House_percentage
63	Census_sector.Situation	Census_sector.Average_income
64	Census_sector.Situation	Census_sector.Paid_percentage
65	Company.Age_band	Company.Operating_index
66	Company.Range_of_employees	Company.Presumed_revenue
67	Company.Presumed_revenue	Company.Rural_producer_flag
68	Company.Operating_index	Company.Range_of_employees
69	Company.Age_band	Company.Rural_producer_flag
70	Company.Operating_index	Company.Presumed_revenue

F Company PRM

This appendix introduces the dependencies in the Π_C PRM, learned in the *Company experiment*, introduced in Section 4.4. All the dependencies in Π_C are presented in Table F.

Table 45 – Π_C PRM Dependencies

The Company PRM Dependencies		
	From	To
1	Company .Age_band	Company .Operating_index
2	Company .Range_of_employees	Company .Presumed_revenue
3	Company .Operating_index	Company .Range_of_employees
4	Census_sector .Apartment	Company .Rural_producer_flag
5	Census_sector .Female_domiciles	Company .Age_band
6	Company .Rural_producer_flag	Company .Presumed_revenue
7	Company .Rural_producer_flag	Company .Operating_index
8	Company .Rural_producer_flag	Company .Age_band
9	Person .Mosaic	Company .Rural_producer_flag
10	Company .Rural_producer_flag	Company .Range_of_employees
11	Person .Propensity_for_postpaid_phone	Company .Rural_producer_flag
12	Person .Mosaic	Person .Company_owner_flag
13	Person .Mosaic	Person .Presumed_income
14	Person .Mosaic	Person .Propensity_for_credit_card
15	Person .Propensity_for_broadband_internet	Person .Propensity_for_postpaid_phone
16	Person .Consumption_index	Person .Propensity_for_personal_loan
17	Person .Mosaic	Person .Social_class
18	Person .Propensity_for_postpaid_phone	Person .Age_group
19	Person .Mosaic	Person .Propensity_for_mortgage
20	Person .Propensity_for_personal_loan	Person .Propensity_for_broadband_internet
21	Person .Mosaic	Person .Propensity_for_luxury
22	Person .Mosaic	Person .Credit_risk
23	Person .Consumption_index	Person .Propensity_for_credit_card
24	Census_sector .Average_income	Person .Social_class
25	Census_sector .Average_income	Person .Mosaic
26	Person .Mosaic	Person .Age_group
27	Person .Presumed_income	Person .Propensity_for_mortgage
28	Person .Propensity_for_broadband_internet	Person .Mosaic
29	Person .Propensity_for_broadband_internet	Person .Age_group
30	Person .Propensity_for_personal_loan	Person .Propensity_for_postpaid_phone
31	Person .Propensity_for_mortgage	Person .Company_owner_flag
32	Person .Propensity_for_personal_loan	Person .Mosaic
33	Person .Propensity_for_broadband_internet	Person .Presumed_income
34	Person .Propensity_for_personal_loan	Person .Propensity_for_mortgage
35	Person .Presumed_income	Person .Gender.
36	Person .Consumption_index	Person .Propensity_for_broadband_internet
37	Person .Gender.	Person .Bolsa_familia_flag
38	Person .Mosaic	Person .Bolsa_familia_flag
39	Person .Consumption_index	Person .Credit_risk
40	Person .Propensity_for_personal_loan	Person .Propensity_for_credit_card
41	Person .Gender.	Person .Propensity_for_luxury
42	Census_sector .Average_income	Person .Propensity_for_personal_loan

43	Person.Mosaic	Person.Public_server
44	Census_sector.Situation	Person.Social_class
45	Person.Mosaic	Person.Propriety_for_postpaid_phone
46	Person.Propriety_for_personal_loan	Person.Propriety_for_luxury
47	Person.Age_group	Person.Gender.
48	Person.Propriety_for_personal_loan	Person.Age_group
49	Person.Presumed_income	Person.Company_owner_flag
50	Census_sector.Average_income	Person.Propriety_for_broadband_internet
51	Person.Company_owner_flag	Person.Propriety_for_luxury
52	Census_sector.Female_domiciles	Person.Mosaic
53	Census_sector.Average_income	Person.Consumption_index
54	Person.Credit_risk	Person.Rural_producer_flag
55	Person.Company_owner_flag	Person.Public_server
56	Person.Gender.	Person.Rural_producer_flag
57	Census_sector.Situation	Census_sector.Water_supply
58	Census_sector.Water_supply	Census_sector.Sewage_network
59	Census_sector.Situation	Census_sector.Domiciles
60	Census_sector.Water_supply	Census_sector.Rented_domiciles
61	Census_sector.Sewage_network	Census_sector.Female_domiciles
62	Census_sector.Average_house_hold	Census_sector.Average_income
63	Census_sector.Female_domiciles	Census_sector.Average_house_hold
64	Census_sector.Situation	Census_sector.Garbage_collection
65	Census_sector.Houses	Census_sector.Apartment
66	Census_sector.Rented_domiciles	Census_sector.Paid_domiciles
67	Census_sector.Female_domiciles	Census_sector.Houses
68	Census_sector.Situation	Census_sector.Average_income
69	Census_sector.Situation	Census_sector.Paid_domiciles