Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Computación

Comparative Study of Methods for the Inference of Socioeconomic Status in a Communications Graph

Tesis presentada para optar al título de
Licenciado en Ciencias de la Computación

Martín Fixman

Director: Carlos Sarraute

Codirector: Esteban Feuerstein

Buenos Aires, 2018

Abstract

Obtaining and processing demographical and sociological data have been some of the most important processes for understanding population-wide phenomena since at least 17th century [Fri06], and finding simple and intuitive ways of visualizing them has a big impact in our ways of understanding the data [Min44, Sno55]. Common ways of obtaining useful qualitative data on socioeconomic stratification usually involved archival research or social surveys [Bul77], and rely on statistical methods.

Telecommunication operators ("telcos") have access to a wealth of information about their users' communications and habits [Huu03], but the ability to store and process that data has taken large strides in the last few years thanks to new and more powerful computers and data mining techniques. The same can be said for sociological and economic information owned by banks and credit cards, and the relation between these two data sources.

Large scale data mining of data from the telecommunications industry is a relatively new area that's been so far mostly used for internal applications [HAK+02], but the gigantic wealth of real-time sociological data has been of interest for academic purposes related to sociology. This thesis builds on methods used by Óskarsdottir et al. [ÓBV+16] and Singh et al. [SFLP13], along with a large dataset of information for a certain telco and a large bank to find that the income distribution of the users follows closely (but not exactly) the income distribution of the whole population.

We have observed a strong homophily between the incomes of contacts in the telco, which along with the uneven distribution of wealth in the population is leveraged to create a methodology, grounded in Bayesian statistics, to infer socioeconomic level of a large subset of users in the network without banking information which is very accurate at AUC = 0.746. The Bayesian method is later compared to several other methods based on supervised machine learning to prove that, even though it uses less input information, it is a better predictor of social features in this particular kind of network.

Resumen

Obtener y procesar datos demográficos y sociológicos fueron uno de los procesos más importantes para entender fenómenos que afectan a toda la población desde por lo menos el Siglo XVII [Fri06], y encontrar formas simples e intuitivas de visualizarlos tiene un gran impacto en nuestra manera de entender los datos [Min44, Sno55]. Formas comunes de obtener datos cuantitativos de estratificación económica usualmente involucran investigación de archivos o encuestas sociales [Bul77], y dependen de métodos estadísticos.

Las operadoras de telecomunicaciones ("telcos") tienen acceso a una gran cantidad de información sobre las comunicaciones y hábitos de sus usuarios [Huu03], pero la habilidad de guardar y procesar esos datos ha dado grandes pasos en los últimos años gracias a nuevas y más poderosas computadoras y técnicas de minería de datos. Lo mismo puede decirse sobre la información sociológica y económica contenida por bancos y tarjetas de crédito, y por la relación entre estas dos fuentes de datos.

La minería de datos de telcos a gran escala es un área relativamente nueva que se usa principalmente para aplicaciones internas [HAK+02], pero la gran cantidad de información sociológica es de gran interés para temas académicos relacionados a la sociología. Esta tesis se basa en métodos usaros por Óskarsdottir et al. [ÓBV+16] y Singh et al. [SFLP13], además de una fuente de información de una telco y de un banco grande para encontrar que la distribución de ingresos de los usuarios sigue de manera cercana (pero no exacta) la distribución de ingresos de la población en general.

Hay una fuerte homofilia entre los ingresos de contactos en la telco, que se usa junto con la distribución desigual de dinero en la población para crear una metodología, basada en estadística bayesiana, para inferir el nivel socioeconómico de un gran subconjunto de usuarios en la red sin información bancaria con AUC = 0.746. El método bayesiano es luego comparado con otros métodos basados en aprendizaje automático supervisado para probar que, aunque toma menos información de entrada, es un mejor predictor de características sociales en este tipo particular de red.

To my teachers.

Table of Contents

Chapter 1

Introduction

1.1 Motivation of the Thesis

In recent years, we have witnessed an exponential growth in the capacity to gather, store and manipulate massive amounts of data across a broad spectrum of disciplines: in astrophysics our capacity to gather and analyze massive datasets from astronomical observations has significantly transformed our capacity to model the dynamics of our cosmos; in sociology our capacity to track and study traits from individuals within a population of millions is allowing us to create social models at multiple scales, tracking individual and collective behavior both in space and time, with a granularity not even imagined twenty years ago.

In particular, mobile phone datasets provide a very rich view into the social interactions and the physical movements of large segments of a population. The voice calls and text messages exchanged between people, together with the call locations (recorded through cell tower usages), allow us to construct a rich social graph which can give us interesting insights on the users' social fabric, detailing not only particular social relationships and traits, but also regular patterns of behavior both in space and time, such as their daily and weekly mobility patterns [GHB08, PSS13, SLPA15].

Demographic factors play an important role in the constitution and preservation of social links. In particular concerning their age, individuals have a tendency to establish links with others of similar age. This phenomenon is called age homophily [MSLC01], and has been verified in mobile phone communications graph [BE10, SBB14] as well as the Facebook graph [UKBM11].

Economic factors are also believed to have a determining role in both the social network's structure and dynamics. However, there are still very few large-scale quantitative analyses on the interplay between economic status of individuals and their social network. In [LFAH$^+$16], the authors analyze the correlations between mobile phone data and banking transaction information, revealing the existence of social

stratification. They also show the presence of socioeconomic homophily among the networks participants using users' income, purchasing power and debt as indicators. The authors of [LMS+17] studied the correlation between the position of a node in a mobile phone communications graph and its socio-economic status. They showed that the position and topological attributes in the graph can be used to generate inferences of the users' financial status. In particular the study [LMS+17] shows the value of the Collective Influence [MM15] as a topological attribute for the prediction of individual financial status.

1.2 Summary of our Approach

In this work, we leverage the socioeconomic homophily present in the cellular phone network to generate inferences of socioeconomic status in the communication graph. To this aim we will use the following data sources: (i) the Call Detail Records (CDRs) from the operator allow us to construct a social graph and to establish social affinities among users; (ii) banking reported income for a subset of their clients obtained from a large bank data source. We then construct an inferential algorithm that allows us to predict the socioeconomic status of users close to those for which we have banking information. To our knowledge, this is the first time both mobile phone and banking information has been integrated in this way to make inferences based on a social telecommunication graph. Part of this work was published in [FBB+16].

The work done on this thesis is based the hypotheses that there is a significant level of homophily between a person's socioeconomic level and the one from its contacts (Section 5.1), and that using this correlation we can infer the first from the second (Section 5.2). At the same time, this thesis presents several "conventional" Machine Learning algorithms (Chapter 7) along with an inference algorithm based in Bayesian Inference which works thanks to the correlation hypothesis (Chapter 5). This extra information should give this algorithm better results than the conventional ones.

Multiple strategies can be used to generate network features based on the CDRs. For instance, in [ÓBV+16] the authors evaluate different collective inference methods applied to the churn prediction problem. Furthermore, the work [ÓBV+17] studies the impact of the social graph definition on the performance of the prediction methods. This motives the second part of the thesis, where we perform a comparative study of

methods to generate network features for the nodes in the communication graph, and evaluate their impact on the inference of the income. We also compare the effectiveness of machine learning methods such as Logistic Regression and Random Forest on the different feature sets.

1.3 Summary of Results

The final results are presented in Chapters 6 and 7. This section presents a short summary.

- When using the Bayesian Algorithm, the *Area Under the Curve* (which is the target metric user in this thesis) is maximized when the socioeconomic comparison is done by number of contacts (Table 6.3) to a value of 0.746.
- Of the common machine learning methods used, the ones which use the labels of the neighbouring users to make a prediction have a better result of the ones which don't. However, even in this case the results are worse than in the Bayesian Algorithm (Section 7.6).

1.4 Organisation of the Thesis

The remainder of the thesis is separated into 7 chapters.

Chapter 2 provides an introduction to the theoretical ideas used in the thesis: the concept of homophily in social networks, introductions to concepts in Bayesian probability and machine learning, and some techniques used to define the level of homophily in the dataset.

Chapter 3 reviews some of the related work on correlations in social-economic networks and on relation between socioeconomic status and mobile phone use that was used as a base for this thesis.

Chapter 4 reviews the telecommunications and bank sources used in this study, how they work together, and also some extra insights about the data that can be found after merging both datasets.

Chapter 5 presents the *Bayesian Algorithm*, used as the main inference algorithm in this thesis. In the first part, it presents a theoretical justification of its correctness using the dataset. Later, it formally presents the algorithm and its possible variations.

Chapter 6 contains a high level description of the testing environment and of the evaluation method of the *Bayesian Algorithm*. It later finds optimal hyperparameters and, in Table 6.3, presents the final results of the algorithm.

Chapter 7 Presents other algorithms of differing complexity that work in the same dataset as the previous one, including ones based in common *Machine Learning* with novel feature extraction methods. The final results are presented in Section 7.6.

Chapter 8 Presents the conclusions of the work, along with some possible work to be done in the future using this same dataset.

Chapter 2

Theoretical Building Blocks

2.1 Social Homophily

> "People love those who are like themselves."
>
> *Rhetoric*
> *Aristotle*

Similarity breeds connection [MSLC01]. People have several visible characteristics, such as age, gender, and socioeconomic status, for which contact between people with similar properties occurs at a higher rate than between dissimilar people.

There are two overall types of homophily that can be distinguished in groups [PFL54]: *status homophily*, in which similarity is based on status, and *value homophily*, which is based on values, attitudes, and beliefs. Status homophily, a part of which is the main study of this thesis, includes the major sociodemographic dimensions that stratify society — ascribed characteristics like race, ethnicity, sex, or age, and acquired characteristics like religion, education, occupation, and behaviour patterns.

2.1.1 Age Homophily

One of the most common homophily patterns in human relations is related to the people's ages [ea11][MSLC01]. This result is expected because of the many societal reasons that explain the homophily: schools tend to group people according to age into the same classrooms, work opportunities tend to be clustered into age groups, which affects work environments and neighbourhood composition, and people have a strong tendency to confide in someone of one's own age.

This correlation has a waterfall effect. Since this kind of homophily is present early into people's life, the produced connections are closer, longer lived, have a larger number of exchanges, and tend to be more personal than other kinds of connections.

There's an interesting exception to this homophily: there is a significant number of connections between parents and their younger children [SBB14]. This exception is addressed later in this paper.

2.1.2 Gender Homophily

McPherson et al. also noted an important degree of homophily between members of the same gender [MSLC01]. In particular, ever since school age children learn that gender is a permanent personal characteristic, homophily can be observed in play patterns and friend groups.

By the time people are adults, people's friendship networks are relatively gender-integrated. However, when controlling for kinship networks and not counting close family members, there is a considerable level of gender homophily [Mar88]. However, this level is still lower than the one for race, education, age, and many other social dimensions.

Gender homophily is lower among the young and the highly educated[Mar87]. One of the main reasons for this is that most environments where people make their networks, such as work establishments and voluntary organizations, are highly sex segregated. Therefore, it is not surprising that the networks formed in these settings display a significant amount of baseline homophily on gender.

2.2 Spearman's Coefficient

Spearman's Rank Correlation Coefficient (also known as Spearman's rho) is a non-parametric measure of rank correlation which measures how well the relationship between two variables can be described using a monotonic function [Mye03]. Unlike Pearson's Correlation Coefficient, which measures lineal relationship between variables, Spearman's Coefficient uses the *rank* of the variables in its calculations; therefore is measures its monotonicity.

For a sample of size n with scores X_i and Y_i, the Spearman Coefficient r_s is defined as in Equation (2.2.1).

$$r_s = \rho_{\mathrm{rank}(X)\,\mathrm{rank}(Y)} = \frac{\mathrm{cov}(\mathrm{rank}(x), \mathrm{rank}(y))}{\sigma_{\mathrm{rank}(X)}\sigma_{\mathrm{rank}(Y)}} \tag{2.2.1}$$

Where $\rho_{a,b}$ denotes the *Pearson Correlation* between the variables a and b. This value will be closer to 1 when the variables are directly monotonic, closer to -1 when they are inversely monotonic, and closer to 0 when there is no tendency for either variable to increase or decrease when the other increases.

2.3 Bayesian Inference

> "Given the number of times in which an unknown event has happened and failed: required the chance that the probability of its happening in a single trial lies somewhere between any two degrees of probability that can be named."
>
> *An Essay towards solving a Problem in the Doctrine of Changes [Bay63]*
> *Thomas Bayes*

This work uses a Bayesian approach to statistics instead of the usual Frequentist approach. In the Frequentist point of view, parameters are fixed and unknown: hypotheses are either true or false, and they cannot be described with a probability. In the Bayesian approach, anything unknown is described with a probability distribution since uncertainty must be described by probability [Mac03].

2.3.1 Bayes Theorem

The base of *Bayesian Inference* is *Bayes' Theorem*, presented in Equation (2.3.1), which describes the probability of an event base on prior knowledge of conditions that may be related to it [GCSR03].

$$P(H \mid E) = \frac{P(E \mid H) \cdot P(H)}{P(E)} \tag{2.3.1}$$

Each one of the terms in Equation (2.3.1) has a different definition and interpretation.

- $P(H \mid E)$, the **Posterior Probability** is the conditional probability that is assigned after the relevant evidence is taken into account.
- $P(H)$, the **Prior Probability**, expresses the assumptions made on the problem before the experiments. While these assumptions will be subjective, the same thing can be said about the other probabilities in this model.
- $P(E \mid H)$, the **Likelihood**, is the degree of belief in E given that H is true. In most real world problems, this tends to be easier to define than the *Prior*.
- $P(E)$, the **Marginal Likelihood**, as the likelihood function where some parameter variables were marginalized. It is used as a normalizing constant to that the *Posterior Probability* integrates to 1, thus making it a valid probability. Since it is constant on the perspective of H, it is usually ignored when taking proportionality, as in Equation (2.3.3).

It can be proven in a simple way by using basic theorems of the probability, as seen in Equation (2.3.2).

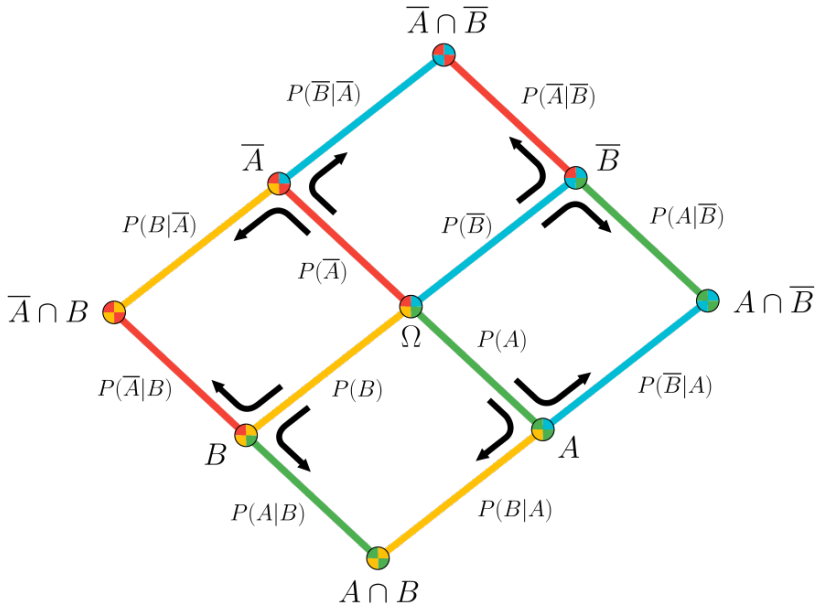
$$\begin{aligned} P(H \cap E) &= P(H \mid E)\, P(E) \\ &= P(E \mid H)\, P(H) \\ P(H \mid E)\, P(E) &= P(E \mid H)\, P(H) \\ P(H \mid E) &= \frac{P(E \mid H)\, P(H)}{P(E)} \end{aligned} \tag{2.3.2}$$

Most of the equations presented in this section deal with continuous probabilities, which by definition must integrate to 1 [Kol56]. Therefore, the theorem is usually used in the version presented in Equation (2.3.3), which defines the proportionality of the *Posterior*.

$$P(H \mid E) \propto P(E \mid H) \cdot P(H) \tag{2.3.3}$$

2.3.2 Conjugate Priors

For a single problem there may be many different possible *Prior Probabilities*, which can be defined depending on the approach taken on defining the model to represent



$$P(A|B)\cdot P(B) = P(A \cap B) = P(B|A)\cdot P(A)$$

Figure 2.1: Graphical visualization of *Bayes Theorem* between two probabilities A and B by the superposition of two decision trees starting in hypothesis space Ω.

different measures of knowledge and certainty about the data*. In particular, if the prior is less informative then the posterior is more likely to be determined by the data.

A simple way to choose a correct prior is using a *Conjugate Prior*. A distribution $P(H)$ is *Conjugate* to $P(H \mid E)$ if multiplying the two distributions together and normalizing the results in another distribution has the same form as $P(H)$.

The *Conjugate Prior* has some philosophical significance in the context of *Bayesian Estimator*. In the practical case, the *Prior Probability* contains more or less information compared to the *Posterior Probability* depending on the amount of data seen. In particular, if the experiment has seen little data, a single datapoint can influence your beliefs significantly. On the other hand, if the experiment has a lot of data, then one single extra datapoint shouldn't influence them as much [GCSR03].

2.4 The Beta Distribution

The *Beta Distribution* is a family of continuous probability distributions defined in the interval $[0, 1]$ which is parametrized by two shape parameters, α and β.

*An extreme case is the *Jeffreys Prior*, used to express total ignorance about the data [Jef46].

The distribution can be used to model the behaviour of *Random Variables* limited to intervals of a finite length. It is often used as a statistical function to model unknown data from a known sample, such as allele frequencies in population genetics [BN95], Malaysian sunshine data [SOWZ99], and heterogeneity in the probability of HIV transmission [WHP89].

In the context of *Bayesian Inference*, the *Beta Distribution* is the *Conjugate Prior* of the *Binomial Distribution*, which allows us to describe initial knowledge concerning probability of success of a single bi-variate distribution. In layman terms, this allows us to know what is the distribution of the continuous p parameter of a binomial distribution for which we have α positive and β negative samples.

2.4.1 Probability Density Function

Given a variable $0 \leq x \leq 1$, which represents the unknown probability of having a *Positive Sample* from the distribution, and the shape parameters $\alpha > 0$ and $\beta > 0$, the *Probability Density Function* of the beta distribution can be described as in Equation (2.4.1), where κ represents some constant.

$$\begin{aligned} f(x;\alpha,\beta) &= \kappa \cdot x^{\alpha-1}(1-x)^{\beta-1} \\ &= \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\int_0^1 u^{\alpha-1}(1-u)^{\beta-1}du} \\ &= \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \cdot x^{\alpha-1}(1-x)^{\beta-1} \\ &= \frac{1}{B(\alpha,\beta)} \cdot x^{\alpha-1}(1-x)^{\beta-1} \end{aligned} \tag{2.4.1}$$

$$B(\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \tag{2.4.2}$$

Equation (2.4.2), describes the *Beta Function*, which is related to the *Gamma Function* and describes a similar pattern [Art64].

Regarding this thesis, the *Beta Distribution* will be used to model a real life problem in Chapter 5. In this problem, both $\alpha \in \mathbb{N}$ and $\beta \in \mathbb{N}$, so the *Beta Function* can be simplified using the identity $(x-1)! = \Gamma(x)$ as shown in Equation (2.4.3).

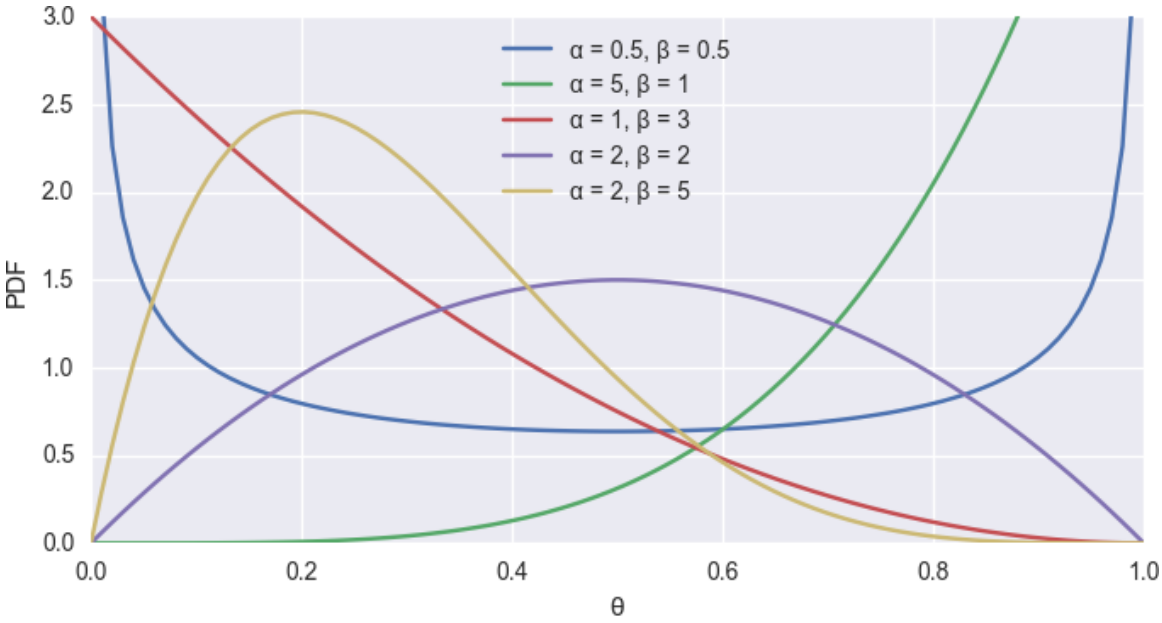


Figure 2.2: Beta distribution with different parameters

$$B(\alpha, \beta) = \frac{(\alpha + \beta - 1)!}{(\alpha - 1)! \cdot (\beta - 1)!} \tag{2.4.3}$$

Additionally, the *Beta Function* can be generalized into the *Incomplete Beta Function* for some parameter x as in Equation (2.4.4). This function is, confusingly, also represented with the Greek letter B; to ease comprehension this thesis will refer to it as $\mathrm{B}_{\mathrm{inc}}$.

$$\mathrm{B}_{\mathrm{inc}}(x; \alpha, \beta) = \int_0^x t^{\alpha-1}(1-t)^{\beta-1} dt \tag{2.4.4}$$

As we get more data from the sampling, the *Beta distribution* turns more concentrated towards the actual θ and its shapes resembles more a normal curve, as can be seen in Figure 2.2. This represents the increased certainty which comes from the acquired knowledge of the problem.

2.4.2 Cumulative Distribution Function

The *Cumulative Distribution Function* of the *Beta Distribution* is defined in Equation (2.4.6).

$$
\begin{aligned}
X &\sim \text{Beta}\,(\alpha, \beta) \\
F\,(x; \alpha, \beta) &= P\,(X \leq x)
\end{aligned}
\tag{2.4.5}
$$

$$
\begin{aligned}
F\,(x; \alpha, \beta) &= \int_0^x f\,(t; \alpha, \beta)\,dt \\
&= \int_0^x \frac{1}{B\,(\alpha, \beta)} t^{\alpha-1}(1-t)^{\beta-1} dt \\
&= \frac{1}{B\,(\alpha, \beta)} \cdot \int_0^x t^{\alpha-1}(1-t)^{\beta-1} dt \\
&= \frac{\text{B}_{\text{inc}}\,(x; \alpha, \beta)}{B\,(\alpha, \beta)}
\end{aligned}
\tag{2.4.6}
$$

F is also known as the *Regularized Incomplete Beta Function*, represented as $I_x(\alpha, \beta)$. This function is related to the *Cumulative Distribution Function* of the *Binomial Distribution*, as shown in Equation (2.4.7).

$$
\begin{aligned}
X &\sim \text{Binom}\,(n, p) \\
P\,(X \leq k) &= I_{1-p}\,(n-k, k+1)
\end{aligned}
\tag{2.4.7}
$$

2.4.3 Inverse Cumulative Distribution Function

The problems solved in this thesis require the use of the *Inverse Cumulative Distribution Function* (also known as the *Quantile Function* or the *Percent-Point Function*) of the *Beta Distribution*, which returns a value such x that meets the expression in Equation (2.4.5) is equal to some value p. It can also be expressed as in Equation (2.4.8).

$$
Q\,(p) = \inf\,\{x \in \mathbb{R} \mid p \leq F\,(x)\} \tag{2.4.8}
$$

Like with the *Cumulative Distribution Function*, there is no closed form formula for expressing its inverse [Kip13]. However, there are fast and accurate ways of computing it using either *Interval Halving* or *Newton's Method*, such as the `incbi` implementation in the *Cephes* library [Mos10] which is use in this thesis via a wrapper from `sklearn`, as explained in Section 6.1.

2.4.4 The Beta-Binomial Model

In the *Beta-Binomial Model* compromises a family of discrete probability distributions similar to the *Binomial Distribution*, with the important difference that, instead of each trial having a constant probability of success, that probability is random and follows the *Beta Distribution* [Sch96].

Given a binary experiment which is run n times, and the probability of success of any of those experiment is some constant θ, the *Probability Distribution* of the amount of successes k can be modelled with a *Binomial Distribution*, as shown in Equation (2.4.9).

$$\begin{aligned} k \mid n, \theta &\sim \text{Bin}\left(\theta, n\right) \\ P\left(k = x \mid n, \theta\right) &= \binom{n}{k} \cdot \theta^k (1-\theta)^{n-k} \end{aligned} \tag{2.4.9}$$

θ is a random continuous probability distribution, which is defined using the *Beta Distribution* in Equation (2.4.10).

$$\begin{aligned} \theta \mid \alpha, \beta &\sim B\left(\alpha, \beta\right) \\ P\left(\theta \mid \alpha, \beta\right) &= \frac{1}{B\left(\alpha, \beta\right)} \cdot \theta^{\alpha-1} (1-\theta)^{\beta-1} \end{aligned} \tag{2.4.10}$$

Once the binary experiment is run, the model has additional information which may change the distribution of θ. This can be modelled as a *Posterior Distribution* using *Bayes Theorem* [NP14].

$$\begin{aligned} P\left(\theta \mid n, k, \alpha, \beta\right) &= \frac{P\left(k \mid n, \theta\right) P\left(\theta \mid n, \alpha, \beta\right)}{P\left(k \mid n, \alpha, \beta\right)} \\ &\propto P\left(k \mid n, \theta\right) P\left(\theta \mid n, \alpha, \beta\right) \\ &= P\left(k \mid n, \theta\right) P\left(\theta \mid \alpha, \beta\right) \\ P\left(\theta \mid n, k, \alpha, \beta\right) &= \binom{n}{k} \theta^k (1-\theta)^{n-k} \cdot \frac{1}{B\left(\alpha, \beta\right)} \theta^{\alpha-1} (1-\theta)^{\beta-1} \\ &\propto \theta^k (1-\theta)^{n-k} \cdot \theta^{\alpha-1} (1-\theta)^{\beta-1} \\ &= \theta^{k+\alpha-1} (1-\theta)^{n-k+\beta-1} \end{aligned} \tag{2.4.11}$$

This is exactly the same function as the one in Equation (2.4.10). That is, the *Posterior Distribution* of this model is also a *Beta Distribution*.

$$\theta \mid k, n, \alpha, \beta \sim \text{Beta}\left(\alpha + k, \beta + n - k\right) \tag{2.4.12}$$

This way it is possible to see that the *Beta Distribution* has the properties of a *Conjugate Prior Distribution* seen in Section 2.3.2 to the *Binomial Likelihood*. This makes it extremely desirable for *Bayesian Analysis*, and for this reason it is used as the main model of this thesis. This is seen in more detail Section 5.2.5.

2.5 Machine Learning Validation Metrics

In the following subsections we present the outline of several supervised machine learning algorithms which are used to compare the Bayesian method to a more realistic baseline. First, we'll present several ways to validate the different algorithms when applied to the data. In the following section, we'll present many of the algorithms used for comparison.

Given a set of features Z, all of which belong to members of a population which belong to a certain category, and a random subset of those features $X \subseteq Z$ whose category y is known, the models should be trained with X and y in order to correctly predict the values corresponding to all the features in Z. Since those values are unknown validation of the output is impossible; therefore, we validate the model using the known values in X and y.

There are many metrics that can be used to measure the performance of a classifier or a predictor [Pow07]; different fields have different preferences due to different goals. In this section, we present many metrics to evaluate different results that are commonly used in the area of mobile phone data analysis [ÓBV+16].

2.5.1 Classification of individual results

Once we define out classifier g and run it against a matrix of features, we get a predicted result y_{pred} which, when compared to the actual result $\text{y}_{\text{true}} = y$, can be classified as the one in Table 2.1.

Additionally, this table can be easily seen in a graphical way in Figure 2.3.

		Predicted Condition	
	Total Population	Condition Positive	Condition Negative
True Condition	Condition Positive	**True Positive**	**False Negative** (Type II error)
	Condition Negative	**False Positive** (Type I Error)	**True Negative**

Table 2.1: Confusion Table, showing different classifications of an individual prediction. True and False Positives (TP/FP) refer to the number of predicted positives that were correct/incorrect, and similarly for True and False Negatives (TN/FN).

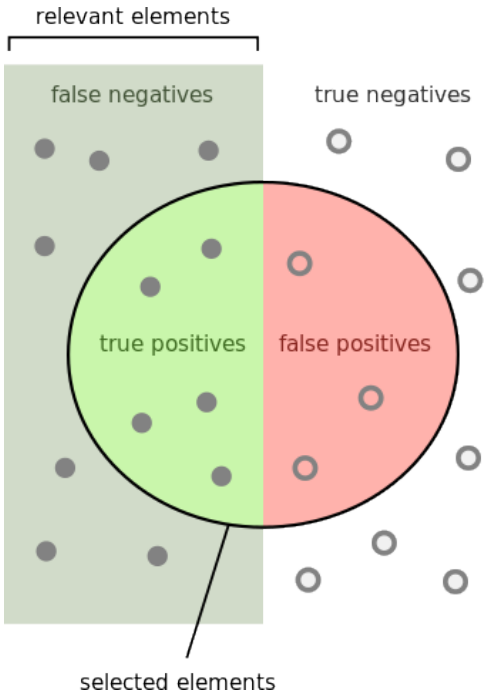


Figure 2.3: Visual explanation of *Precision* and *Recall*

2.5.2 Precision and Recall

Precision denotes the proportion of predicted positive cases that are correctly real positive. Trying to maximize this would allow us to adjust a particular predictor so that the majority of the predicted cases are actually positive. Conversely, *recall* is the proportion of real positive cases that are correctly predicted positive, and maximizing it would allow us to adjust a predictor so that the majority of positive cases are predicted.

$$\begin{aligned} \text{Precision} = \text{TPA} &= \frac{\text{TP}}{\text{TP} + \text{FP}} \\ \text{Recall} = \text{TPR} &= \frac{\text{TP}}{\text{TP} + \text{FN}} \end{aligned} \tag{2.5.1}$$

These two measures and their combinations focus only on the positive examples and predictions, although between them they capture some information about the rates and kind of errors made [Pow07]. While the *recall* has been shown to have a major weight in working with machine translation [FM07], they aren't particularly useful to use alone since they don't take into account many factors of the prediction [Pow07].

2.5.3 Inverse Precision and Inverse Recall

As a corollary of the previous metrics, we can add metrics that measure the proportion of real negative cases that are correctly predicted negative, referred as the *Inverse Recall*, and the proportion of predicted negatives that are real negatives, referred as the *Inverse Precision*[Pow07]. We can see that these are equivalent to finding the *Precision* and *Recall* of the negative category.

$$\begin{aligned} \text{Inverse Precision} = \text{TNR} &= \frac{\text{TN}}{\text{FP} + \text{TN}} \\ \text{Inverse Recall} = \text{TNA} &= \frac{\text{TN}}{\text{FN} + \text{TN}} \end{aligned} \tag{2.5.2}$$

2.5.4 Accuracy

The *accuracy*, commonly referred in the context of binary classifiers as **Rand Accuracy**[Pow15], is used as a statistical measure of how well a binary classification

test identifies or excludes a condition. Unlike the *precision*, it takes into account the negatives, and it is expressible [Pow07] both as a weighted average of *precision* and inverse *precision* or *recall* and *inverse recall*.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{N} \tag{2.5.3}$$

This can be more simply expressed using the weighted average of either the *Precision* and *Inverse Precision* or the *Recall* and the *Inverse Recall*.

$$\begin{aligned} \text{Accuracy} &= (\text{TP} + \text{TN}) \cdot \text{TPR} + (\text{FP} + \text{TN}) \cdot \text{TNR} \\ &= (\text{TP} + \text{FP}) \cdot \text{TPA} + (\text{FN} + \text{TN}) \cdot \text{TNA} \end{aligned} \tag{2.5.4}$$

2.5.5 ROC Curve

A *Receiver Operating Characterising* graph is a technique for visualizing, organizing, and selecting classifiers based on their performance [Faw05]. The curve is created by plotting the *True Positive Rate* against the *False Positive Rate* at various threshold settings.

This allows to compare different classifiers before having to select a particular threshold value for them. In particular, a random classifier will score near the positive diagonal ($\text{FPR} = \text{TPR}$), while a perfect classifier will score in the top left hand corner ($\text{FPR} = 1, \text{TPR} = 0$) and a worst case classifier will score in the bottom right hand corner*[Pow07].

2.5.6 Area Under the Curve

The *ROC Curve* allows us to compare classifiers and choose the one which is closer to optimal in some sense. While there are many possible parametrizations, the most common is to minimize the *Area Under the Curve*, which is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one [Faw05]. This can be formulated as shown in Equation (2.5.5).

*Note that, for any binary classifier, it is trivial to transpose the entire ROC curve (or a part of it) to the other part of the diagonal; therefore the worst "realistic" case is the random one

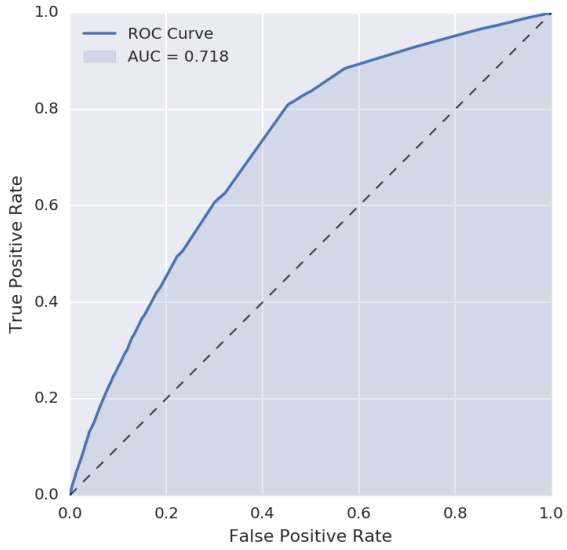


Figure 2.4: A *ROC Curve*, where the *Area Under the Curve* is marked. This particular graph comes from data used in early experiments to finding the socioeconomic index of a person, which is explained with more detail in Chapter 5.

$$\begin{aligned} \mathrm{AUC} &= P\left(X_1 > X_0\right) \\ &= \int_0^1 \mathrm{TPR}\left(t\right) \mathrm{FPR}'\left(t\right) dt \end{aligned} \tag{2.5.5}$$

2.5.7 F-measure

The *F-measure* is another measure of a tests accuracy. It considers both the *Precision* and the *Recall* of the test to compute the score. It can be considered the weighted average of both values for some weight β, where F_β reaches the best score 1 when both precision and recall are 1.

$$\begin{aligned} F_\beta &= \left(1+\beta^2\right) \cdot \frac{\mathrm{TPA} \cdot \mathrm{TPR}}{\left(\beta^2 \cdot \mathrm{TPA}\right) + \mathrm{TPR}} \\ &= \frac{\left(1+\beta^2\right) \cdot \mathrm{TP}}{\left(1+\beta^2\right) \cdot \mathrm{TP} + \beta^2 \cdot \mathrm{FN} + \mathrm{FP}} \end{aligned} \tag{2.5.6}$$

The most commonly used *F-measure*, F_1, measures the *Precision* and *Recall* is that harmonic mean of the *Precision* and *Recall*. In particular, for an *F-measure* with $\beta > 1$ weights Recall higher than Precision, while with $\beta < 1$ weights Precision higher than Recall.

2.6 Supervised Machine Learning Models

This section presents several *Supervised Machine Learning* models that are used in the paper.

Models are separated into two different groups depending on how they describe the input and output variables.

Continuous Models take a matrix $X \in \mathbb{R}^{n \times f}$ and a vector $y \in \mathbb{R}^n$ of the same height where each element represents the features corresponding to some user and its real value, respectively. Its accuracy is measured according to some function of the result of the regression $\mathrm{y}_{\mathrm{pred}}$ and $\mathrm{y}_{\mathrm{true}} = y$.

Categorical Models take a matrix $X \in \mathbb{R}^{n \times f}$ and a vector $y \in \mathbb{S}^n$, for some set of *Categories* $\mathbb{S}$, and predicts the correct category each of the users in y belongs to. Its accuracy is measured according to several metrics which are explained in Section 2.5.

2.6.1 Linear Regression

A *Linear Regression* is an approach for modeling the relationship between the matrix X and the real vector y. While it is possible to predict many variables in what's known as the *Multivariate Linear Regression* [MBK79], this thesis will focus in the single-dimensional variant.

The regression assumes that the relationship between the sum of some linear combination of the elements of X and y is itself linear. This combination is represented by the variable β, and the relationship is represented through an unobserved random vector ε as the *Error Term*.

The model takes the form of Equation (2.6.1).

$$y = X\beta + \varepsilon \tag{2.6.1}$$

where

$$X = \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1f} \\ 1 & x_{21} & \cdots & x_{2f} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{nf} \end{pmatrix}$$
$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \ \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_f \end{pmatrix}, \ \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix} \tag{2.6.2}$$

β is a vector with size $f + 1$, where β_0 is called the *Constant Term*. The statistical estimation and inference focuses in this variable, as two different models of *Linear Regression* will give different results where this parameter is different [Yan09].

ε_i is called the *Error Term* or *Noise*. The variable captures the factors which influence the vector y other than the matrix X and the *Constant Term* β. The relationship between ε and those variables and knowing whether they are correlated is important in the formulation of an *Linear Regression* model [Yan09].

This model makes several assumptions about the data.

Weak Exogenety implies than the variables in X can be treated as fixed values, rather than random variables.

Linearity implies that the mean of the response variable is a linear combination of the parameters.

Homoscedasticity implies that different response variables have the same *Variance* in their errors. While this is almost never true in practice, since variables tend to vary over a large scale, the data is usually *Standardised* so that this is true.

Independence of errors implies that the errors in response variables are uncorrelated with each other, even if they are statistically dependent.

Lack of multicollinearity implies that the matrix X must have full column rank; there can't be two perfectly correlated input variables. In this case there won't be a unique solution for the vector β.

There are many ways of effectively calculating the optimal *Lineal Regression* for a particular pair $\langle X, y\rangle$. One of them is using the **Least Squares Estimation**, which can be solved through the *Least Squares Principle*.

$$\hat{\beta} = \text{argmin}_{\beta} \left[(y - X\beta)^T (y - X\beta) \right] \tag{2.6.3}$$

Where $\hat{\beta}^T = (b_0, b_1, \ldots, b_{k-1})$, a k-dimensional vector of the estimations of the regression coefficients.

Assuming $\left(X^T X\right)$ is a non-singular matrix, the *Least Squares Estimation* of β can for the model in Equation (2.6.1) can be found from Equation (2.6.4) [Yan09].

$$\hat{\beta} = \left(X^T X\right)^{-1} X^T y \tag{2.6.4}$$

Additionally, Equation (2.6.4) presents an unbiased estimator of β [Yan09].

2.6.2 Logistic Regression

The *Logistic Regression*, also referred to as the *Logit Model* [Fre09] is a regression model similar to the *Linear Regression*, with the particularity that the *Dependent Variable* y is categorical. While it is possible to calculate the variable for sets of categories of several finite sizes (when it is referred to as a *Multinomial Logistic Regression* [Gre11]) or even for ordered sets of categories (an *Ordinal Logistic Regression* [McC80]), this thesis will focus on the case where $y \in \{0, 1\}^n$, that is, each variable in y is binary.

The model uses the results of a *Linear Regression* on the data $\langle X, y\rangle$, where coefficients that solve Equation (2.6.1) are found. The result of that regression is a real value, which is normalized using a function $f : \mathbb{R} \to [0, 1]$, whose result is considered the probability that the result of some element is 1.

A commonly used function for f is the *Logistic Function* $\sigma : \mathbb{R} \to [0, 1]$, defined in Equation (2.6.5) and plot in Figure 2.5.

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}} \tag{2.6.5}$$

This probability is used for estimating the value of y, namely as defined in Equation (2.6.6).

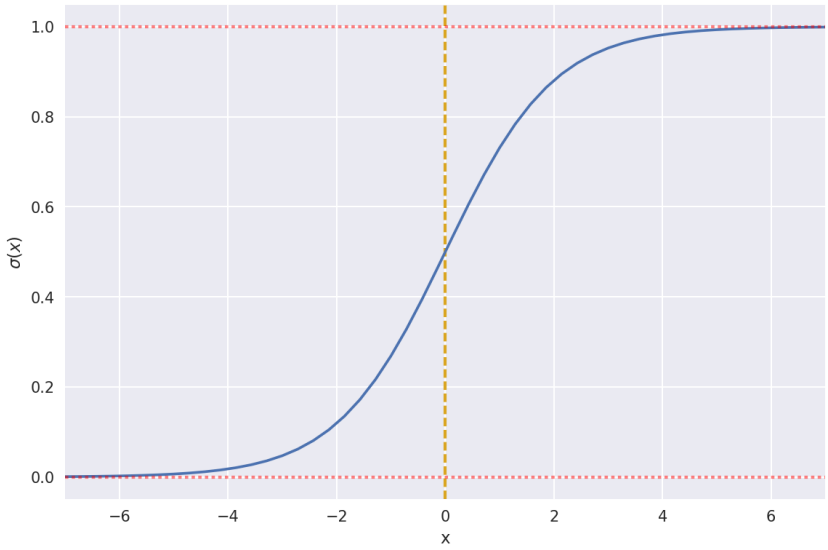


Figure 2.5: The standard logistic sigmoid function $y = \sigma(x)$

$$\begin{aligned} P\left(y_i = 1 \mid X\right) &= \sigma\left(X_i\beta\right) \\ P\left(y_i = 0 \mid X\right) &= 1 - \sigma\left(X_i\beta\right) \end{aligned} \tag{2.6.6}$$

The inverse of the *Logistic Function*, referred to as the *Logit Function*, gives the logarithm of the *Odds* of a certain element belonging to some category.

$$\text{logit}\left(x\right) = \log\left(\frac{\sigma\left(x\right)}{1 - \sigma\left(x\right)}\right) \tag{2.6.7}$$

The logit model says that the vector y is independent given the matrix X [Fre09], and can be considered as inverse of the probability defined in Equation (2.6.6).

$$\text{logit}\, P\left(y_i = 1 \mid X\right) = X_i\beta \tag{2.6.8}$$

The regression is usually calculated using *Maximum Likelihood Estimation* [FCH+08] and, unlike the estimation of the *Linear Regression* presented in Section 2.6.1, it is not possible to find a closed form expression of the coefficients that maximize the value of the likelihood function. Instead, the iterative *Newton's Method* is used.

The exact formulas used to ensure fast and correct conversed used in this thesis are the ones present in `LIBLINEAR`, which is presented in [HZL].

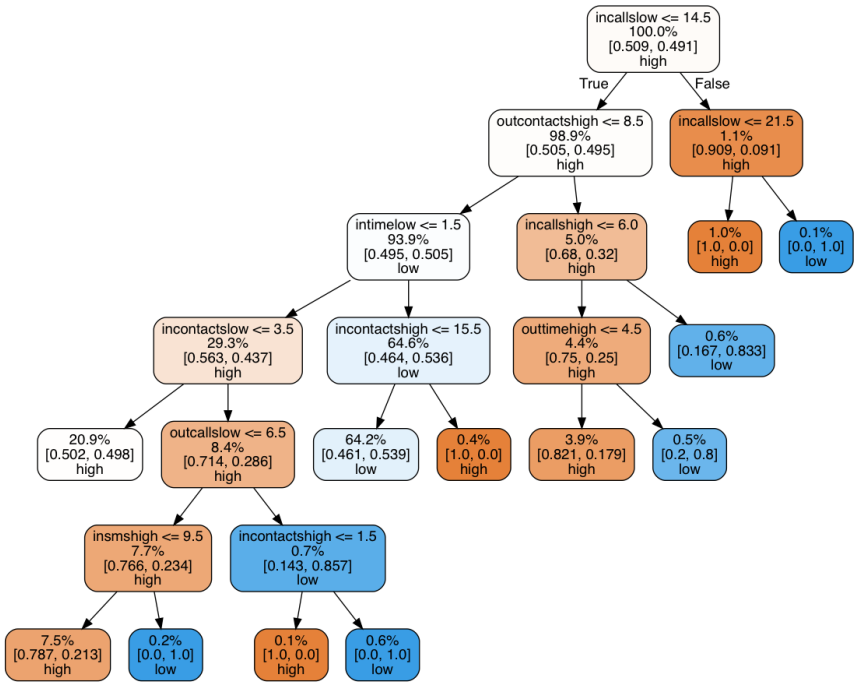


Figure 2.6: A decision tree. This is a simplified version of one of the decision trees used in the *Random Forest* predictors in Section 7.4.2.

2.6.3 Decision Trees

A *Decision Tree* is a decision support tool that uses a tree of decisions and their possible consequences, including change event outcomes. They are commonly used in decision analysis to help identify the most optimal strategy to reach a goal.

The tree themselves contain a root at the top, and each non-final node (including the root) contains a binary test of a certain variable; each one of these nodes contains exactly two edges connecting it to the following level, one which is followed in the case the condition is true and another for the case when the condition is false.

In the context of *Machine Learning*, it can be used as a supervised predictive model which matches properties about the list of features X to the likeliest category y [MR08]. In the binary case, each of the final nodes contains the output label which, according to the input data, is the most probable given the result of the tests corresponding to the previous nodes to this one along with the probability of this label being true.

There are many algorithms that can generate a tree that's optimized for many different metrics. A common one is the *Classification and Regression Trees (CART)* method, introduced in [Bre93], which builds the tree starting from the root and, for

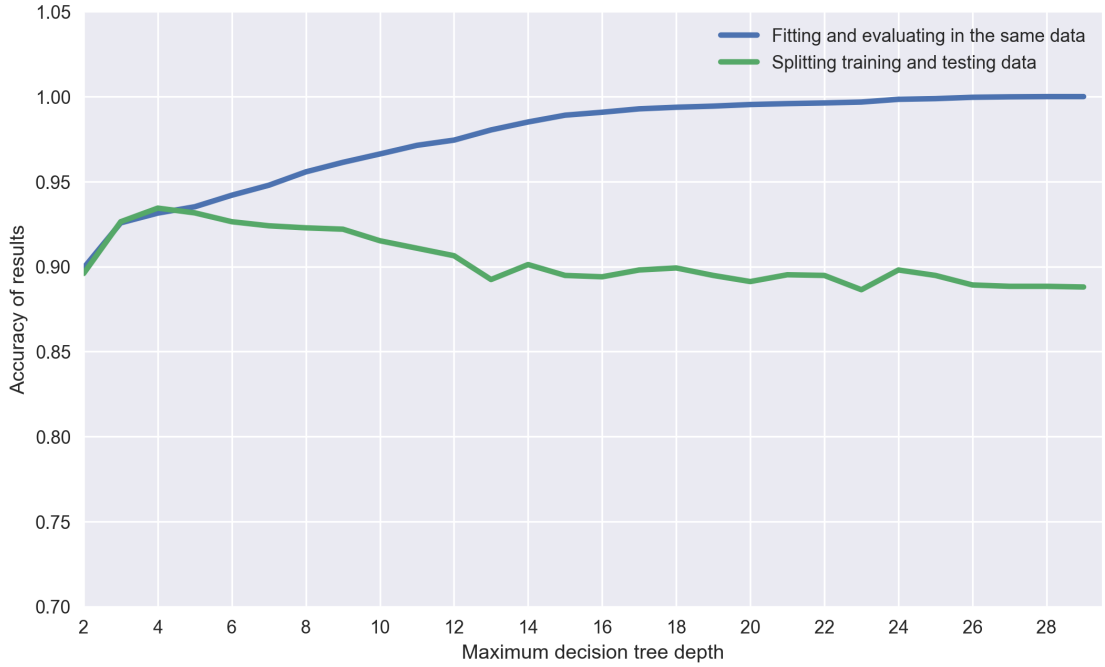


Figure 2.7: Overfitting in decision trees. When training and testing on the same data, making a classifier more complex makes the accuracy of the prediction increasingly higher until it is pretty much perfect. This result is wrong since the classifier is *Overfitting* its tree to the training data. When testing with different data, it is possible to see how the prediction actually becomes worse. This graph was simulated by training decision trees with data from the `sklearn.datasets.make_classification` from the sklearn library[PVG+11].

each non-leaf node, the variable that generates the best split according to some metric is selected [Loh11]. This is applied recursively until *CART* detects that no further gains can be made, or some pre-set stopping rules are met.

Decision trees are prone to overfitting [MR08]. Since a single tree can make an arbitrarily good classification of a set of features despite their properties (unlike models like linear regression, where the data has to be linearly separable), it can lose the capability to generalize for instances not presented in training. This occurs when the tree has too many nodes relative to the amount of training data; Figure 2.7 illustrates the overfitting process.

A common mechanism to prevent overfitting in *Decision Trees* is *Pruning*, which reduces the size of decision tree by cutting sections that provide relatively little improvement of the result on the training data.

2.6.4 Random Forest

Random Forests are an ensemble learning method commonly used for classification tasks. While they use ideas that are similar to the ones used in Section 2.6.3 to build *Decision Trees*, their design prevents them from having problems with overfitting in regards to the training set[HTF03].

A trained *Random Forest* consists of a set of different *Decision Trees* on the same feature space. However, these trees should have different biases to compensate for the bias of a single one, since generally using multiple distinct classifiers lowers this value[HHS94]. To achieve these two objectives, each *Decision Tree* is trained on a different subset of the initial feature set[Ho95]. Each of these trees classifies the training data with 100% of accuracy within its feature subspace, yet it generalizes the classification in a different way. Since there are 2^m possible subspaces for m features, there are many choices in practice.

There are many ways to select which features to use to build each tree. A simple one is *Bagging* (**B**ootstrap **agg**regat**ing**)[Bre96], which separates the training set X into m subsets $X'_{1\ldots m}$ uniformly and with replacement. Later, the m models are fitted using the bootstrap samples, and their output is combined by average the output.

Other methods include *Random Split Selection*, which uses randomization by computing the k (where $k = 20$ in the original paper) top candidate splits, and choosing one of them at random[Die00]. The best results are seen when using *Adaboost* (for which its authors won the Gödel Prize), a method of *Boosting*, where the features in each tree are selected because of the misclassifications in previous classifiers[FS$^+$96].

Given the set of features $X \in \mathbb{R}^{n \times f}$ and the set of labels $y \in \{0,1\}^n$, each iteration of *Adaboost* creates a new decision tree depending on the *Error* of the previous classifiers. Given a set of already build trees $k_1 \ldots k_l$ which output a classification $k_i(x_i) \in \{0,1\}$ for each item, we can define the current random forest as a linear combination of the previous classifiers C_l. With these values, we can find the weighted *Error* E.

$$\begin{aligned} C_l(x_i) &= \sum_{i=1}^{l} \alpha_i k_i(x_i) \\ C_{l+1}(x_i) &= C_l(x_i) + \alpha_m k_m(x_i) \\ E &= \sum_{i=1}^{n} e^{-y_i C_m(x_i)} \end{aligned} \tag{2.6.9}$$

These values can be later used to improve the current random forest C_{l+1} that minimizes this error.

Along with *Logistic Regression*, *Random Forests* are one of the learning methods used in the practical parts of this thesis. In Chapter 7 it is used to classify the resulting inputs of many feature extraction methods with the *Bayesian Algorithm* introduced in Chapter 5. As Section 7.6.1 shows, while the latter algorithm is better suited for our data, using *Random Forest* on a good set of features performs consistently better than other generic *Machine Learning* algorithms.

Chapter 3

Related Work

This thesis adds new data and experiments to the fast-growing area of *Mobile Phone Social Network Analysis*.

Earlier works in the general area of *Social Network Analysis* and *Socioeconomic Indices* and their relation to demographic features were drawn from sparse sociological studies [KC01] and surveys analyzing a single nation [Dea97]. However, the advent of massive clusters of real-world data along with computers big enough to process it completely changed the landscape of human data analysis, both for industry purposes and for academia.

This chapter will discuss several scientific papers in this area which were relevant for the research done in this one.

3.1 Correlations of Consumption Patterns in Social-Economic Networks

Léo et. al. present correlations between purchasing patterns and socioeconomic position of users from a dataset similar to the one used in this thesis [LKSF16]. In particular, the authors have access to a database of credit card purchases for a set of users, with information about the amount of money spent and the general category (MCC) to which the purchase belongs, and also to a cellphone communications graph which allows them to infer the relationship between any two people.

The first of two interesting studies this thesis makes is to categorize the population depending on their total spending, and find out the spending level of each user category on one of several aggregated purchase groups. It makes it easy to see the difference in spending for lower income and higher income people: the former group tends to spend comparatively more money in entertainment and retail stores, while the latter group spends more money in hotels and vehicles.

The second study presented in this paper relates to the correlation between people who buy from each of these groups to find categories which are commonly purchased

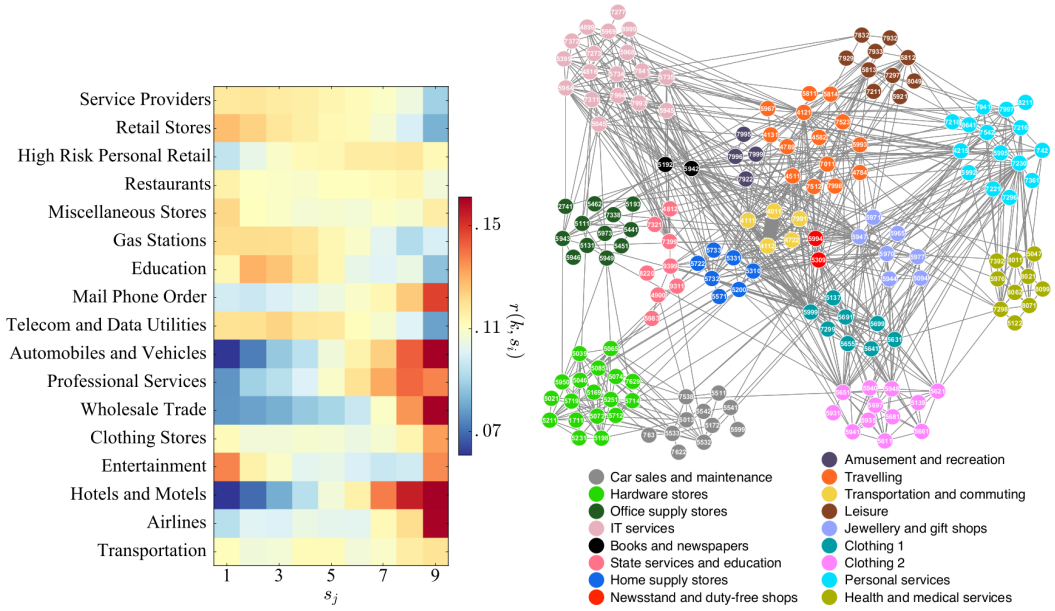


Figure 3.1: Data from the study of categories of purchases. The heatmap on the left side is the adjacency matrix of G_ρ, shown with logarithmically scaled colors. The graph on the right represents $G_\rho^>$, a weighted subgraph of G_ρ which shows only significant correlations where $\rho(c_i, c_j) > 1.5$.

together. Some groups, like *Transportation*, *IT*, or *Personal Services* play a central role and are connected to many other communities, while some others like *Car Sales and Maintenance* and *Hardware Stores* and pairwise connected.

The correlation between two categories is presented in Equation (3.1.1), where $r(c_i, u)$ quantifies the fraction of money spent on a category c_i by a user u.

$$\rho\left(c_i, c_j\right) = \frac{n\left(\sum_u r(c_i, u) r(c_j, u)\right)}{\left(\sum_u r(c_i, u)\right)\left(\sum_u r(c_j, u)\right)} \tag{3.1.1}$$

If $\rho(c_i, c_j) > 1$, then categories c_i and c_j are positively correlated; if $\rho(c_i, c_j) < 1$, then the categories are negatively correlated. Using this data, the authors build the weighted correlation traph $G_\rho = (V_\rho, E_\rho, \rho)$, where links $(c_i, c_j) \in E_\rho$ are weighted by the $\rho(c_i, c_j)$ correlation values. This graph can be seen intuitively in Figure 3.1.

This thesis uses similar methods to find a correlation between the socioeconomic index of a user and the ones of his contacts. In addition, there's merit in the observation that just as wealthier people spend more money in entertainment, they also are more active users of their telephones.

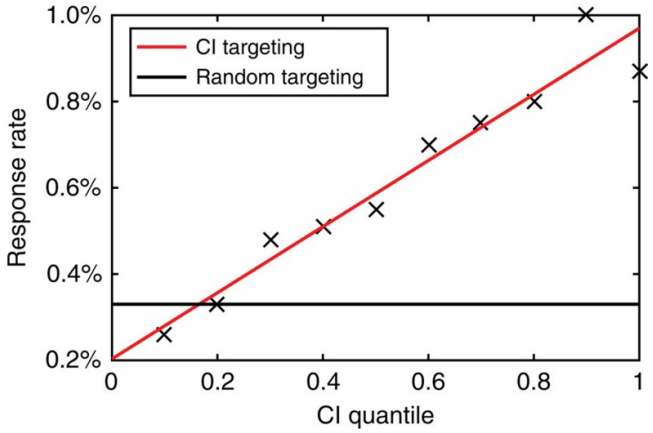


Figure 3.2: Response rate versus *Collective Influence*, a measure of the variance between links.

3.2 Inferring Personal Economic Status from Social Network Location

Luo et. al. show that an individual's location is highly correlated with its socioeconomic status [LMS+17]. In addition, the paper also finds the interesting observation that some social network patterns mimic the economic inequality patterns, and that there is a significant ($R^2 = 0.96$) correlation between the diversity of an individual's links and their financial status: the wealthiest 1-percenters have higher diversity in mobile contacts and are centrally located, surrounded by other highly connected people (network hubs). On the other hand, the poorest individuals have low contact diversity and are weakly connected to fewer hubs.

The results were validated by performing a social marketing campaign for the acquisition of new credit card clients by sending message for individuals that were predicted to be affluent. Compared to a control group, the users with the most covariance between their links (that is, with the highest link diversity) would more probably request the offered product, which was ideal for affluent users. These results can be seen in Figure 3.2.

Additionally, to prove that the results were not dependent on the validation campaign, the authors produced an *Analysis of Covariance*[WA78] on all the features they had access to test the variance caused by network metrics and other factors. This resulted in the conclusion that the correlation between collective influence is positive and significant in all groups of geographical communities, across genders, and among all age groups older than 24 years. Such robust network effects imply that network

metrics are a potential indicator for financial status.

Unlike this thesis, this paper is completely observational and doesn't provide a direct inference method for socioeconomic status. However, both its strict methodology and its prediction of *Collective Influence* were useful for completing many parts of this work, specially since the dataset used by Luo et al. has many similarities with the one used in this.

3.3 Socioeconomic Status and Mobile Phone Use

Blumenstock and Eagle combines data from direct demographic surveys with *Call Details Records* obtained from a phone company to get demographical data about cellphone users in Rwanda [BE10].

The paper combines data about the overall demographic composition of Rwanda with the demographic composition of a representative sample of mobile phone users, along with voluntary survey results and the call history of the survey residents.

Two interesting tests made to measure the socioeconomic status of the respondents, which is particularly hard in a country where a significant percentage most people's income derives from informal channels.

- Asking the respondents directly some of the demographic questions previously used in a nation-wide survey from the Rwandan government. This resulted a stark difference in socioeconomic level between the general population and the cellphone-owning people in the survey.

- Using this same government survey to compute total expenditures by aggregating expenditures across some subcategories as explained in [DZ02], and then fit the model to the data.

With this data it was possible to characterize economic stratification and inequality within the population of mobile phone users. Additionally, using the CDRs, it was possible to characterize graph properties for rich and poor users, in addition to other demographic indicators such as gender. In particular, while the mobile phone population is in general wealthier than the general population of Rwanda, there's still considerable inequality within the group of mobile phone users.

The analysis on users' CDRs is very similar to the one used in this thesis. In particular, the stratification of the nodes in the graph into "rich" and "poor", along with other demographic patterns, to find previously unknown data is the same one that was used here.

3.4 Understanding Individual Human Mobility Patterns

Gonzalez et. al. explore the statistical properties of a population's mobility patterns by using a mobile phone dataset similar to the one used in this thesis [GHB08]. In it, the authors find that the distribution of displacements of users over time can be approximated by a truncated Lévy flight[Man82].

$$P(\Delta r) = (\Delta r + \Delta r_0)^{-\beta} \exp(-\Delta r/\kappa) \tag{3.4.1}$$

Where $\beta = 1.75 \pm 0.15$, $\Delta r_0 = 1.5$km, and cutoff values $\kappa|_{D_1} = 400$km and $\kappa|_{D_2} = 80$km.

The author also proposes 3 distinct hypothesis for this behaviour: Ⓐ each individual follows a Lévy trajectory, Ⓑ the distribution captures heterogeneity between individuals' movement patterns, or Ⓒ some heterogeneity coexists with Lévy patterns.

To distinguish between these hypotheses, the author approximated the *Radius of Gyration* r_g for each user, interpreted as the characteristic distance travelled by some user at some time. While the ensemble of Lévy agents had a significant level of heterogeneity, the author suggests that this is because of the big range of mobility patterns in individuals, thus ruling out hypothesis Ⓐ.

Conversely, the data also shows that users with small r_g travel mostly over small distances, whereas those with large radius display a combination of small and large jump sizes. After rescaling all the distributions with this value, the author shows that the data collapsed into a single curve. Therefore, travel patterns of individual user may be approximated by a Lévy flight up to a distance characterized by r_g; with this definition we can see that large displacements are statistically absent. This indicates that the jump size distribution of $P(\Delta r)$ is the convolution between the statistics of individual trajectories $P(\Delta r_g \mid |r_g)$ and the population heterogeneity $P(r_g)$, which is consistent with hypothesis Ⓒ.

This study demonstrates that the individual trajectories are characterized by the same r_g-independent probability distribution, and this suggests that the statistical elements of individual categories are indistinguishable after rescaling. Therefore, this has the basic ingredients of realistic agent-based models. Given the known correlations between spatial proximity and social links, this could help quantify the role of space in network development and evolution, and improve the general understanding of diffusion processes.

3.5 Link-based Classification

Lu et. al. propose a statistical framework for modeling distributions between linked documents (such as relational databases or URLs in websites) to use in machine learning [LG03]. This is a hard problem, since naïvely applying traditional statistical inference procedures, that assume that links are independent, can lead to inappropriate conclusions[Jen99].

The study proposes using several features for these links, while later using a logistic regression model for each of the features. Since the original problem required multilabeling classification, the study calculates the probability of each label given the features in a one-against-others model and picked the one for the highest posterior probability.

The authors also propose an iterative algorithm to compute the category of each link depending on the labels of their neighbour (which change on each step), which reports an improvement in classification accuracy.

Some of the algorithms proposed in Chapter 7 are inspired by this paper. In particular, the methodology of having two independent sets of feature extraction and prediction methods is the same one used in this thesis.

3.6 Socioeconomic Correlations in Communications Networks

Léo et. al. use a similar dataset to the one used in this thesis that collects communication and bank information about individuals, and shows that consumption patterns are correlated with identified socioeconomic classes leading to social stratification in a similar way to the paper discussed in Section 3.1 [LFAH$^+$16]. In addition, the paper

introduces a correlation between merchant categories.

Given the set of users' purchases, the authors separate it into 17 categories, and measure the fractional distribution of spending for each one as the amount of money each user spent into each category over the total money spent.

As expected, people in lower socioeconomic classes spend more in groups associated with essential needs, such as retail stores, gas stations, or service providers, while people in higher socioeconomic classes spend the majority of the money on jewelry, automobiles, and professional services.,

3.7 A Comparative Study of Social Network Classifiers

Óskarsdottir et. al. present several methods to address machine learning classification in social networks and other graph-related structures [ÓBV+16]. The methods are separated into several categories.

1. *Relational Classifiers*, which infer labels for each node based on the strength of the links to other nodes and the labels to those nodes.

2. *Collective Inference* methods, which infer labels for the nodes of the network while taking into account how the inferred labels affect each other.

The paper uses telco data from many separate sources for testing, and uses a single logistic regression model to predict the correct label from the given features. Despite testing it with several metrics, there are some feature generating methods that are better than the other ones in every single one.

The way this paper presented several features, and the methods it used for testing, were a great inspiration for the work in this thesis.

Chapter 4

Experimental Data Sources

4.1 Mobile Phone Data Source

4.1.1 Dataset Description

The data used in this study consist of a multiset P of composed of voice calls, and another multiset S composed of text messages from a telecommunication company (*telco*) for a 3 month period. These two sets are referred as the *Call Detail Records*, or CDRs.

Every call $p \in P$ contains the phone numbers of the caller and callee $\langle p_o, p_d \rangle$, which are anonymized using a cryptographic hash function for privacy reasons, the starting time p_t, and the call duration p_s. The same datum, except for the call duration, can be found for each element $s \in S$.

Additionally, the latitude and longitude of the antenna used for each call and SMS $\langle p_y, p_x \rangle$ are given for certain users V'. Subsets $P' \subseteq P$ and $S' \subseteq S$ contain those calls.

Given that our collections P and S of CDRs are coming from one telephone company, we are able to reconstruct all communication links between clients of this company N, as well as communications between the clients and other users. However, we have no information on communications where neither users are clients of our telco company, and therefore users not in N don't have complete call information.

The *Communications Graph* $G = \langle V, E \rangle$ is composed of he set of nodes $V = P_o \cup P_d \cup S_o \cup S_d$, and the set of directed edges E, where each element $e \in E$ is composed of an origin and destination $\langle e_o, e_d \rangle$, the total amount of calls between these two users e_c, the total time of all calls e_t, and the amount of SMS e_s. Unlike the multisets P and S, there is at most one element per every pair $\langle e_o, e_d \rangle$ (although there may be two distinct elements with those values flipped).

The set E can be formally constructed with the instructions of the Equation (4.1.1).

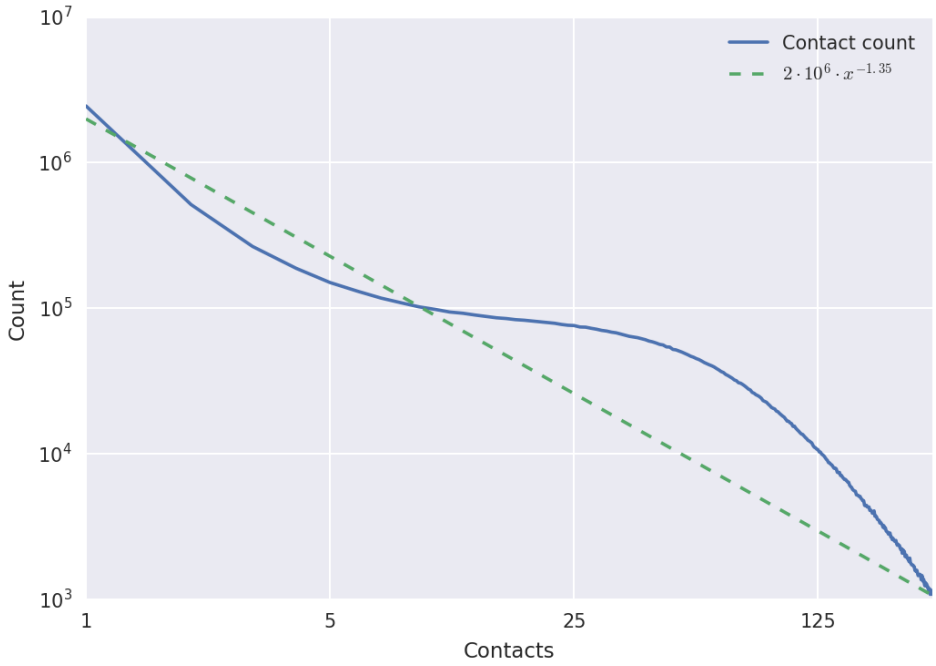


Figure 4.1: Distribution of amount of total contacts per user.

$$
\begin{gathered}
(\forall e \in E) \\
\begin{pmatrix}
e_c = |\{p \in P \mid \langle p_o, p_d \rangle = \langle e_o, e_d \rangle\}| \\
e_s = |\{s \in S \mid \langle s_o, s_d \rangle = \langle e_o, e_d \rangle\}| \\
e_t = \sum_{\substack{p \in P \\ \langle p_o, p_d \rangle = \langle e_o, e_d \rangle}} p_t
\end{pmatrix}
\end{gathered}
\tag{4.1.1}
$$

For simplicity sake, this paper will also refer to the elements of these three sets as calls_e, sms_e, and time_e respectively.

4.1.2 Magnitudes and Distributions

As a corollary, G_N can be defined as the graph $\langle N, E_N \rangle$, where E_N contains only calls between users of the telco, and G' as $\langle V', E' \rangle$, where E' contains the calls from and to users whose calls are located. This form of graph can be seen in Figures 4.1 and 4.2.

Both the amount of calls and the amount of contacts per user are distributed roughly in a negative exponential distribution. Since the *Average Call Time* isn't anywhere

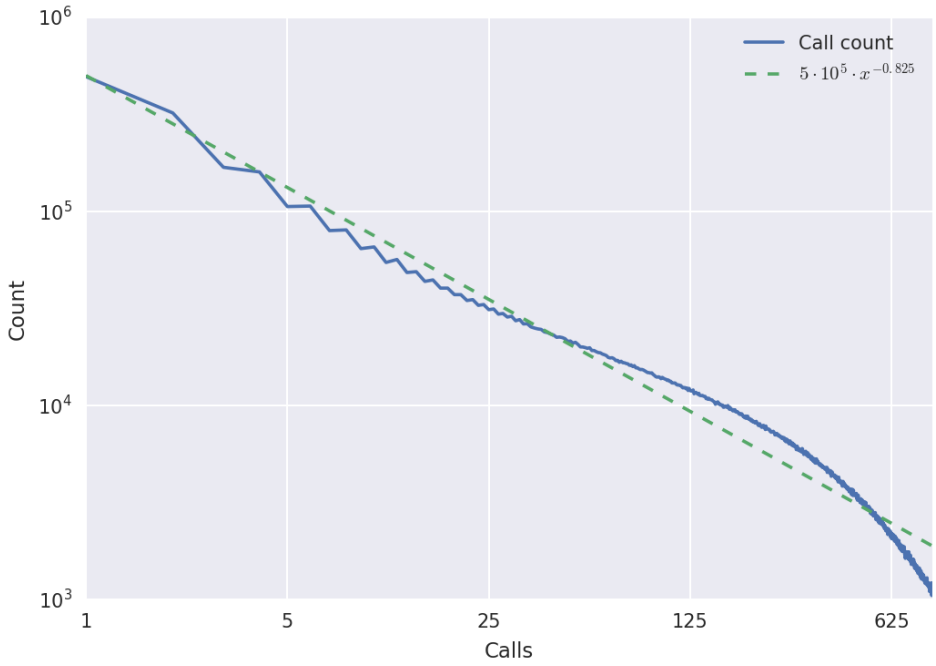


Figure 4.2: Distribution of amount of total calls per user.

near 0, the pattern of *Call Durations* follows a *Gamma Distribution* $\Gamma(3.8, -8.4)$, where Γ corresponds to the standard version of the distribution with location and shape parameters.

4.2 Banking Information

For this study we also obtained the set B of account balanced of over 10 million clients of a certain bank for a period of 6 months, which finishes at the same date as the period used in Section 4.1. This dataset is represented by the set B, and each client $b \in B$ contains the phone number b_p, anonymized with the same hash as the datasets in the previous section, along with the reported income of this person in over 6 months $b_{s_0}, \ldots, b_{s_5}$. We average these 6 values to obtain b_s, the estimate of each users' monthly income.

The bank also provided us demographic information for a subset of its clients $A \subseteq B$. For each user $a \in A$, we are given the age a_a and the gender a_g of the user, which allows us to observe differences in the income distribution according to the age. The distribution is shown in Figure 4.4.

Our data source requested the income data to be agnostic of currency. Since this

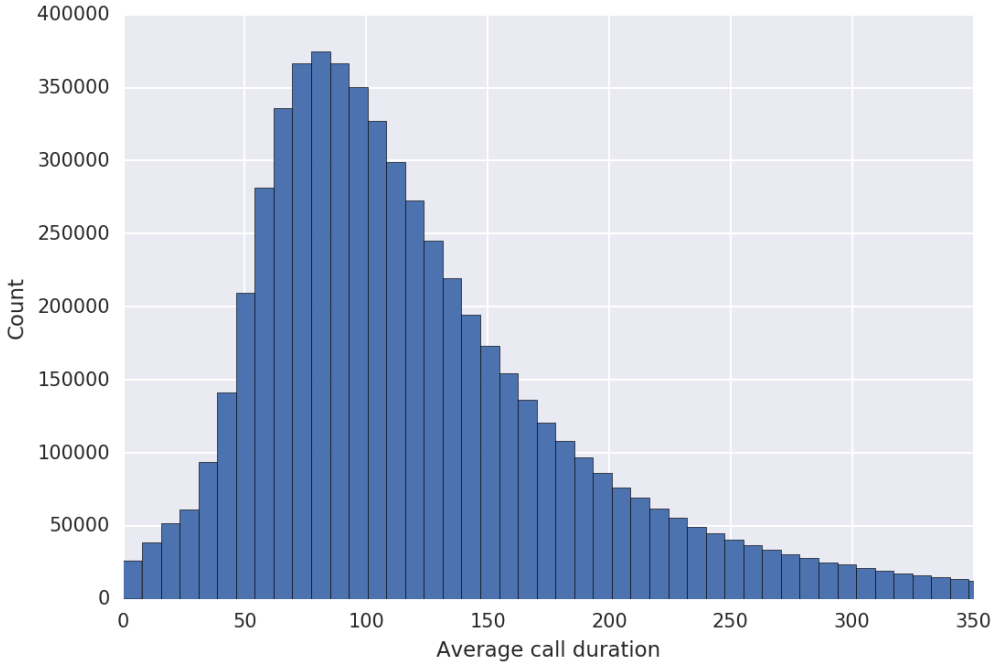


Figure 4.3: Distribution of the durations of calls.

study could apply to any country with a similar level of bancarization, we'll use the $ symbol to refer to a generic currency.

The demographic data can be easily combined with the income data to show income by age, as figured in Figure 4.5. The data shows how the median income increases with age up to the age of retirement, at around 60–65 years, and later it rapidly decreases.

In another line of work, homophily with respect to age has been observed and used to generate inferences [BBMS14].

The income distribution, as shown in Figure 4.6 presents a similar distribution to the values in the dataset presented in Section 4.1.2.

4.3 Matching of Bank and Telco Information

Since the phone numbers in each call in the list of users V are anonymized with the same hash function as the phone number in the bank data in the set B, the users can be matched to their unique phone to augment the *Social Graph* G, where the elements in the set $S = V \cap B$ contain banking information.

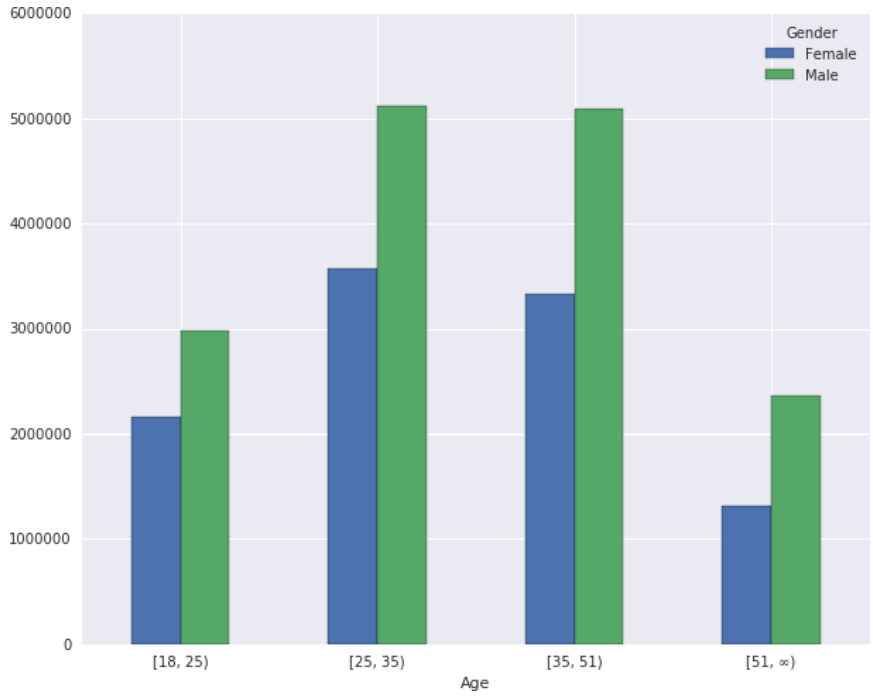


Figure 4.4: Amount of users in B by gender and age.

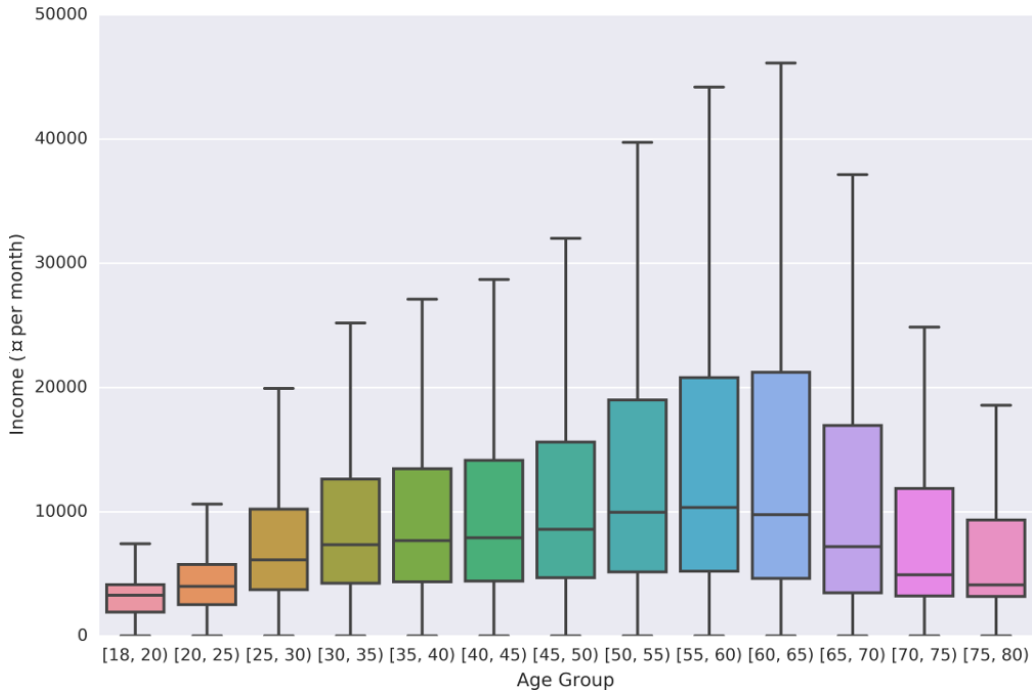


Figure 4.5: Distribution of income a_s as a factor of age a_a. This is consistent with real-world data from the country where the data for this thesis comes from.

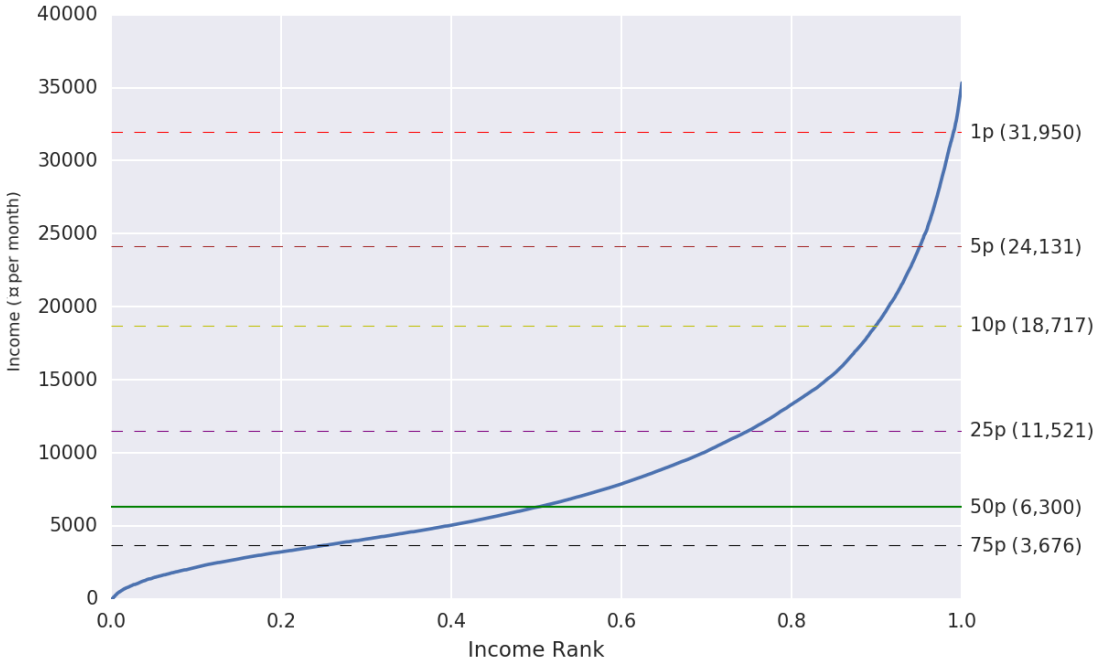


Figure 4.6: Distribution of incomes of users, represented by the set B, with different percentiles marked. This plot helps appreciate the unequal distribution of income, since the 50th percentile has a comparatively low income, while the differential between higher percentile of incomes is each time higher.

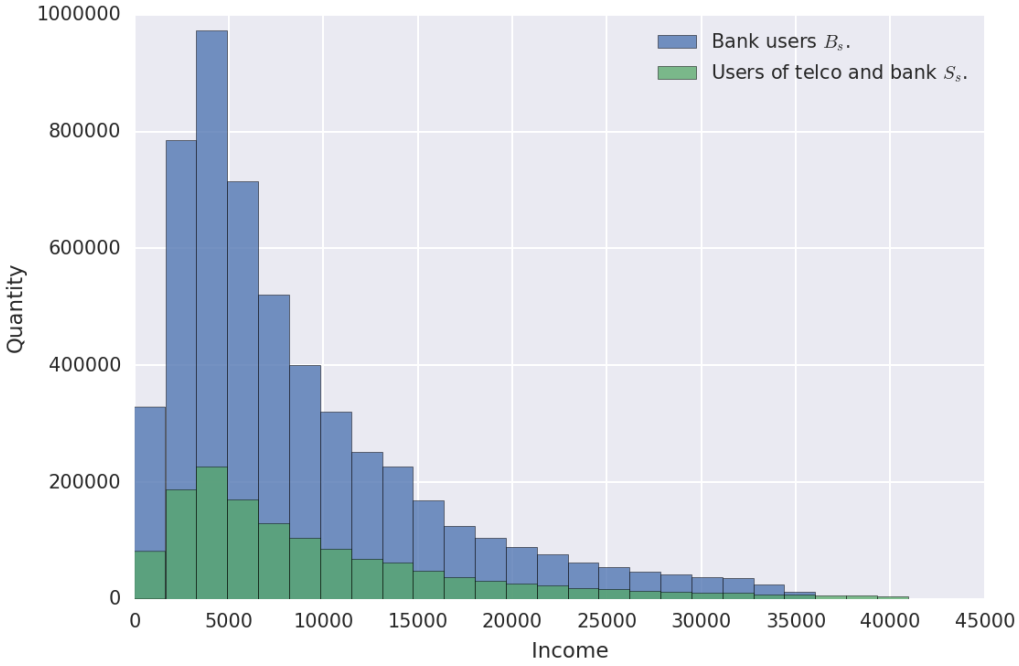


Figure 4.7: Distribution of incomes for users of both the bank and telco, represented by the set S.

$$\begin{gathered} G = \langle V, E \rangle \\ (\forall e \in E) \\ e_o = b_p \implies e_{\text{so}} = b_s \\ e_d = b_p \implies e_{\text{sd}} = b_s \end{gathered} \tag{4.3.1}$$

The distribution of bank users in S is similar to the one in B, as shown in Figure 4.7. This demonstrates that the users of this particular telco have mostly the same socioeconomic patterns as the clients of the bank in general.

4.4 Outlier Filtering

The dataset contains information about bank and telco users, some of which may not directly correspond to a human user, or may not have useful information for our research.

Most of the telco users in the first case are already filtered by the intersection between the bank and the telco data. However, to make sure the users are relevant enough for this study, we only keep the users which have a minimum amount of

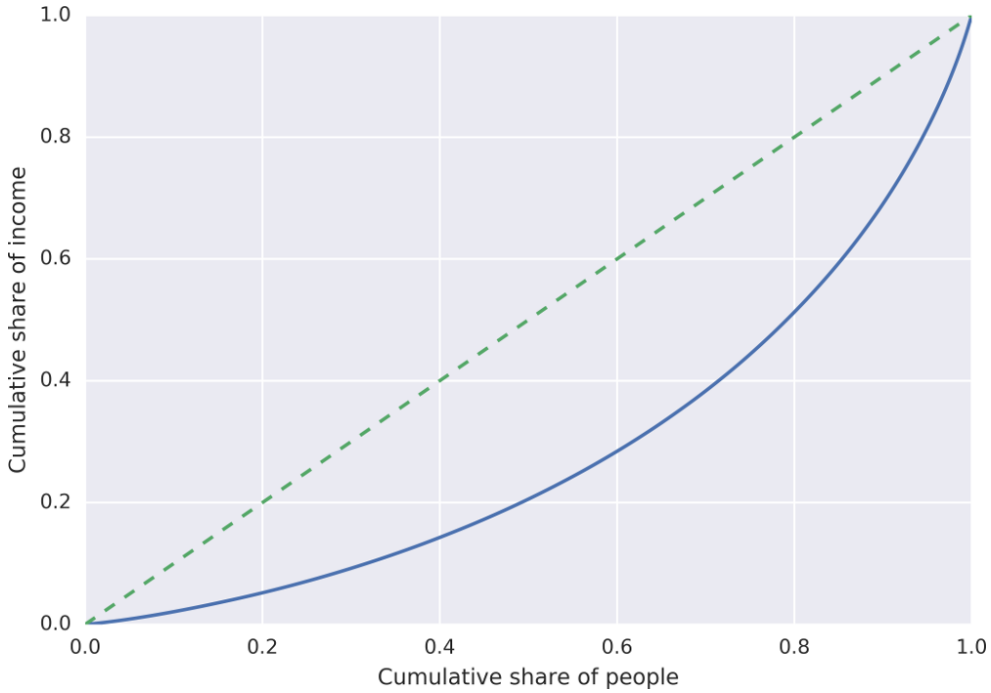


Figure 4.8: Lorenz curve representing the distribution of income of bank clients.

information, defined in the following items.

- A monthly income of at least $1000.
- A monthly income in the 99th percentile (i.e. we filter users with a monthly income in the top 1%).

4.5 Unequal Distribution of Income

We provide here some observations of the distribution of income of the bank clients. These observations correspond to the filtered dataset, obtained after applying the filters of the previous section.

Figure 4.8 shows the Lorenz curve, graphical representation of the distribution of income [Sat87]. The curve plots the cumulative share of clients, sorted by income, to the fraction of the total income of the population.

From the Lorenz curve, we can compute the Gini coefficient as the area that lies between the line of perfect equality and the Lorenz curve over the total area under the line of equality. The data presents a coefficient of Gini = 0.45.

According to the World Bank [Ban16], the Gini coefficient for the population of the country where the data used on this thesis belongs was 0.481 in 2012. Our result is consistent with this information, since the income inequality is expected to be lower when accounting only to bank clients than within the whole population of the country.

Analyzing Figure 4.6, we can observe that the top 10% of the people accumulate 33% of the total income, while the top 20% accumulate 50.5% and the top 30% accumulate 63.1%; the rest of the income is distributed among the remaining 70% of the population.

Chapter 5

The Bayesian Method

5.1 Income Homophily

The main contribution of this work is the estimation of the income of the telco users for which we lack banking data, but have bank clients in their neighborhood of the network graph. To show the feasibility of this task, we first show the existence of a strong income homophily in the telco graph.

Using the values of $G = \langle V, E \rangle$, the *Social Graph* defined in Chapter 4 that contains communication data and bank data of a set of users, Figure 5.1 is defined so that the color in each point $\langle X, Y \rangle$ is the amount of pairs $\{\langle g_o, g_d \rangle \in G \mid g_{o_s} \in X \wedge g_{d_s} \in Y\}$. A simple glance reveals a significant correlation between X and Y.

Given the broad non-Gaussian distribution of the income's values, we choose to use a rank-based measure of correlation which is robust to outliers. Namely, the *Spearman's rank correlation*, as defined in Section 2.2, is computed to test the statistical dependence of sets of incomes of callers and callees. Applying this formula to the data gives a correlation coefficient of $r_s = 0.474$.

The result was compared with a randomized null hypothesis, where links between users are selected randomly disregarding income data, obtaining a p-value of $p < 10^{-6}$. These values for r_s and p show a strong indication of income homophily among users in our communication graph. This observation is consistent with the results reported in other investigation of similar data, namely [LFAH$^+$16].

We can take advantage of this homophily to propagate income information to the rest of our graph G, where the income of the complete set of users is unknown.

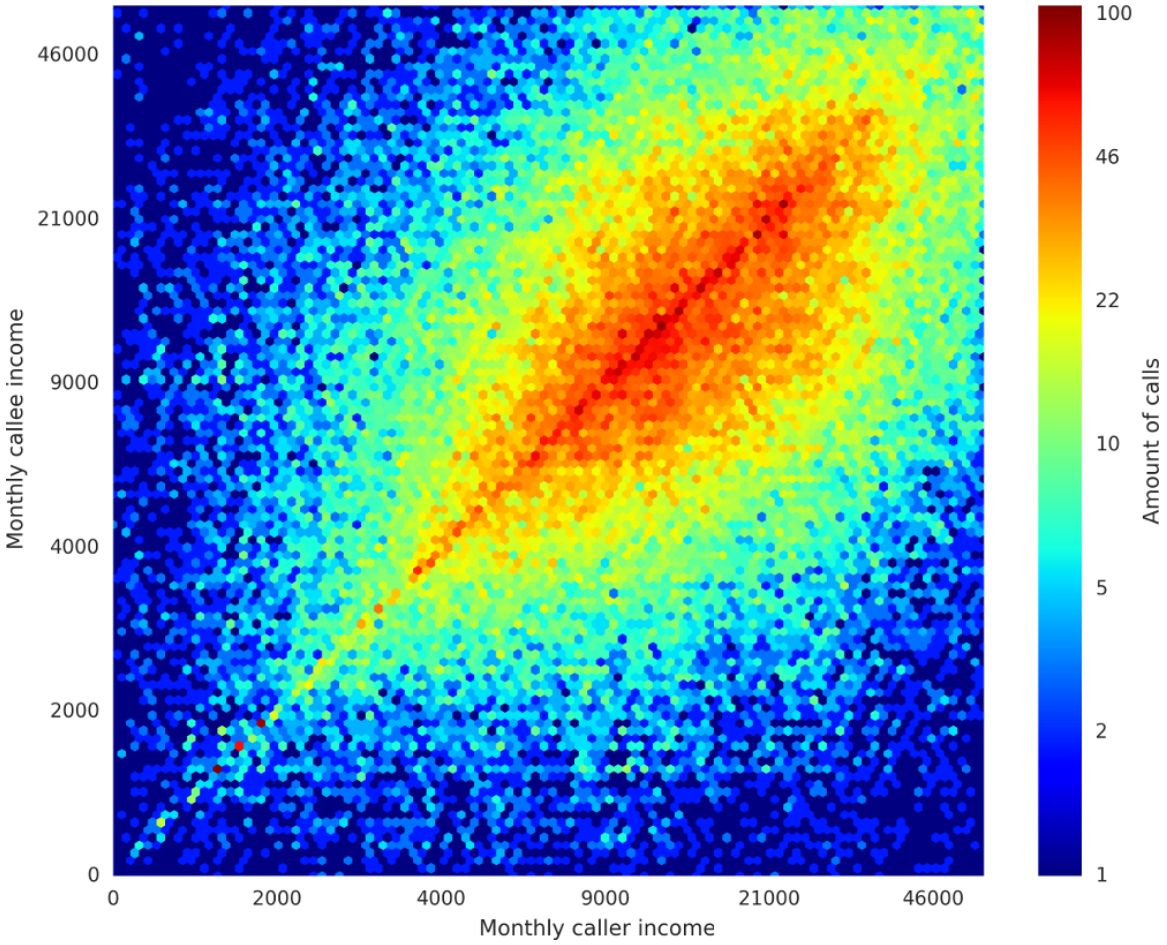


Figure 5.1: Heatmap showing the number of calls between users, according to their monthly income. There is a higher probability that the callee and the caller have similar income levels.

5.2 Prediction Algorithm

5.2.1 Discrimination by Wealth

The main objective of this research is to identify users with higher income. To make the task simpler and more efficient, the actual income values as described in Section 4.2 are forfeited and instead the customers are separated into distinct groups: H_1, containing customers whose income is lower or equal than the median, and H_2, containing users whose income is higher than this value. From now on, these will be referred as *Low Income* and *High Income* users respectively.

The median value in the dataset, after accounting for outliers as explained in Section 4.4, is of exactly **$6300**[*]. This way, we can define the two groups as in Equation (5.2.1).

$$\begin{aligned} H_1 \cup H_2 &= S \\ (\forall h \in H_1)\, h_s &\leq 6300 \\ (\forall h \in H_2)\, h_s &> 6300 \end{aligned} \tag{5.2.1}$$

5.2.2 Feature Accumulation

Given the *Social Graph* $G = \langle V, E \rangle$, as defined in Chapter 4, contains information about the *Calls*, *SMS*, and *Total Time* of each link between two users. These can be accumulated for each user, along with its *Degree*, to produce the *User Data*, which is used for another evaluation in Section 7.3.1.

Another possible way to accumulate these values is discriminating them by the category to which the other endpoint belong. That is, for every edge feature F and for the *Degree*, it is possible to define two features F_{low} and F_{high} that only accumulate features from edges whose other endpoint is a user with *Low Income* or *High Income*, respectively. This approach is formalized in Equations (5.2.2) to (5.2.5).

[*]The input data, including the source country, is anonymized. The symbol $ refers to a certain world currency and not necessarily the American Dollar.

$$\text{calls}_v^{\text{low}} = \sum_{\substack{e\in E\\ e_d=v\\ e_o\in H_1}} e_c + \sum_{\substack{e\in E\\ e_o=v\\ e_d\in H_1}} e_c \qquad \text{calls}_v^{\text{high}} = \sum_{\substack{e\in E\\ e_d=v\\ e_o\in H_2}} e_c + \sum_{\substack{e\in E\\ e_o=v\\ e_d\in H_2}} e_c \tag{5.2.2}$$

$$\text{time}_v^{\text{low}} = \sum_{\substack{e\in E\\ e_d=v\\ e_o\in H_1}} e_t + \sum_{\substack{e\in E\\ e_o=v\\ e_d\in H_1}} e_t \qquad \text{time}_v^{\text{high}} = \sum_{\substack{e\in E\\ e_d=v\\ e_o\in H_2}} e_t + \sum_{\substack{e\in E\\ e_o=v\\ e_d\in H_2}} e_t \tag{5.2.3}$$

$$\text{sms}_v^{\text{low}} = \sum_{\substack{e\in E\\ e_d=v\\ e_o\in H_1}} e_s + \sum_{\substack{e\in E\\ e_o=v\\ e_d\in H_1}} e_s \qquad \text{sms}_v^{\text{high}} = \sum_{\substack{e\in E\\ e_d=v\\ e_o\in H_2}} e_s + \sum_{\substack{e\in E\\ e_o=v\\ e_d\in H_2}} e_s \tag{5.2.4}$$

$$\begin{aligned} \text{contacts}_v^{\text{low}} &= \left|\{e \in E \mid e_o = v \wedge e_d \in H_1\} \cup \{e \in E \mid e_d = v \wedge e_o \in H_1\}\right| \\ \text{contacts}_v^{\text{high}} &= \left|\{e \in E \mid e_o = v \wedge e_d \in H_2\} \cup \{e \in E \mid e_d = v \wedge e_o \in H_2\}\right| \end{aligned} \tag{5.2.5}$$

5.2.3 Uncertainty

One important thing to note is that the only nodes accumulated in Equations (5.2.2) to (5.2.5) for some node $v \in V$ are the neighboring ones that belong to S. Equation (5.2.6) indicates how to build the subset $D \subseteq S$ of bank users who also have a neighbor in the *Social Graph* G which is also part of the bank.

$$\begin{aligned} E^D &= \{e \in E \mid e_o \in S \wedge e_d \in S\} \\ D &= E_o^D \cup E_d^D \end{aligned} \tag{5.2.6}$$

D is defined as a subset of S, instead of one of V, because we cannot analyze the performance of a predictor on telco users who aren't part of the bank.

There is an additional level of uncertainty: while the data contains information about calls, and the inference assumes that callee and caller know each other, there is simply no information about the socioeconomic level of acquaintances who don't share phone calls. However, the more calls a user makes, and the more people it calls, the more certain we can be that the algorithm in this section is correct.

5.2.4 Modelling Users — Frequentist Approach

The final objective of this thesis is to detect *High Income* users in a dataset by just knowing their calls using the hypothesis, described in Section 5.1, that a user will mostly call people from his same income category. For this, some calculations will be

made for some property ϖ^* on the low and high income users, which is one of the values defined in Equation (5.2.7). Part of the performance in Chapter 6 will imply finding the ϖ which maximizes some score.

$$\varpi \in \{\text{calls}, \text{time}, \text{sms}, \text{contacts}\} \tag{5.2.7}$$

As it was discussed in Section 5.2.3, even having that data it is impossible to have complete information about a user's relationships. However, it is possible to create a *Model* where we can predict the rate of *High Income* to *Low Income* contacts a user has, and with that data the Probability p_v that this user $v \in V \setminus B$ is a *High Income* user.

The naïve way of solving this problem would be to assign this probability using only the current data, as in Equation (5.2.8).

$$p_v = P\left(v \in H_2\right) = \frac{\varpi_v^{\text{high}}}{\varpi_v^{\text{high}} + \varpi_v^{\text{low}}} \tag{5.2.8}$$

This method fails to account for the *Certainty* that a user belongs to some category, which can be exemplified in scenarios where the probability of a user with only 1 *High Income* contact being *High Income* is slightly higher than one for another user with 100 *High Income* contacts and 1 *Low Income* contact.

5.2.5 Modelling Users — Bayesian Approach

This section uses a variant of the *Beta-Binomial Model* presented in Section 2.4.4.

For each user $v \in V \setminus B$ we define a *Beta Distribution* Beta_v which can define the probability of p_v falling between two numbers. This approach is formalized in Equation (5.2.9).

$$\mathcal{B}_v \sim \text{Beta}_v\left(\varpi_v^{\text{high}} + 1, \varpi_v^{\text{low}} + 1\right) \tag{5.2.9}$$

The previous equation allows us to define a formula for the probability of each users p_v, as shown in Equation (5.2.10).

*ϖ is a variant of the Greek letter π, which represents a **P**roperty.

$$P\left(\mathcal{B}_v \leq x\right) = \frac{1}{B\left(\varpi_v^{\text{high}} + 1, \varpi_v^{\text{low}} + 1\right)} \cdot \int_0^x t^{\varpi_v^{\text{high}}} (1-t)^{\varpi_v^{\text{low}}} dt \tag{5.2.10}$$

Here it is also possible to assign the probability depending on the *Mean* or the *Mode* of the distribution. However, that would also fail to account for the *Certainty* of the numbers, which is the disadvantage discussed in the method defined in Section 5.2.4.

Instead, given the definition of an arbitrary $\Theta \in [0, 1]$, it is possible to use the formula presented in Section 2.4.3 to define a suitable p_v, as shown in Equation (5.2.11).

$$\begin{aligned} p_v &= Q\left(\Theta\right) \\ &= \inf\left\{x \in [0,1] \mid \Theta \leq F(x)\right\} \end{aligned} \tag{5.2.11}$$

This probability represents the *Posterior Probability* of p_v given the data. To adjust the prediction, the method compares the value for each user with the rest and assumes that, if $p_v > p_u$ for two users v and u, then v has a higher probability of being wealthy.

Since we don't have any prior information about a good value of Θ, we initially take a value from *Jeffrey's Prior*. This distribution, which was defined by Harold Jeffreys in 1946 [Jef46], is a completely uninformative prior distribution for a parameter that makes it possible to parametrize prior ignorance about values in a Bernoulli model.

The *Probability Density Function* of the prior for a parameter γ is presented in Equation (5.2.12). It is interesting to note that this is the same PDF than in the one for a *Beta Distribution* with parameters $\alpha = 0.5, \beta = 0.5$.

$$P\left(\Theta = \gamma\right) \propto \frac{1}{\sqrt{\gamma\left(1-\gamma\right)}} \tag{5.2.12}$$

5.2.6 Categorizing Users

In the previous section the probability of belonging of being a *High Income* user p_v was defined for every $v \in V \setminus B$. While this value alone doesn't give any information about whether user v belongs which category, it does tell that his category will be higher or equal than users with a lower probability. This approach is formalized in Equation (5.2.13).

$$
\begin{gathered}
(\forall u, w \in V \setminus B) \\
\begin{pmatrix}
p_u > p_w \wedge u \in H_1 \implies w \in H_1 \\
p_u < p_w \wedge u \in H_2 \implies w \in H_2 \\
p_u = p_w \implies (u \in H_1 \iff w \in H_1)
\end{pmatrix}
\end{gathered}
\tag{5.2.13}
$$

Thanks to this approach we can define some limit, τ, and categorize each user $v \in V \setminus B$ as either category depending on the magnitude of p_v compared to τ, as in Equation (5.2.14).

$$
\begin{aligned}
p_v \leq \tau &\implies v \in H_1 \\
p_v > \tau &\implies v \in H_2
\end{aligned}
\tag{5.2.14}
$$

τ represents the *Threshold* on which this algorithm separates the *Low Income* and *High Income* users. Since its value changes many metrics on the model that are related to the final labels, it can be set to arbitrarily increase or decrease *Precision*, *Recall*, or any combination of those.

5.3 Performance Evaluation

It is easy to evaluate the performance by calculating p_v for all $v \in B$, which allows us to know whether each value is a *True Positive*, a *False Positive*, a *False Negative*, or a *True Negative*, and collecting any of the metrics described in Section 2.5.

One advantage of this method is that it is possible to evaluate the method by calculating the *Area Under the Curve* without having to specify a particular τ, since this follows the necessary pattern defined in Section 2.5.6. This way it is possible to select the best ϖ without having to select the respective τ.

Once this methods defines which $\varpi \in \{\text{calls}, \text{time}, \text{sms}, \text{contacts}\}$ it will use, it chooses Θ as to maximize the *Area Under the Curve* of the model. While calculating the *Inverse Cumulative Function* for the *Beta Distribution* is particularly slow, it is possible to infer less datapoints by using a *Statistical Prior*, as defined in Section 5.2.5. Another probability (still unexplored) is to figure out that the *Area Under the Curve* for a particular ϖ is monotonic increasing until the optimal Θ, and later is monotonic

decreasing, as shown in the later Figure 6.1. If this hypothesis were true, it would be possible to use *Ternary Search* to find the best value of Θ.

After finding a ϖ and Θ, it is possible to find the point p_v at which the *Beta Distribution* $\mathcal{B}_v$ of each user $v \in V$ integrates to Θ. Having this data we can find the τ that maximizes the *Accuracy* of the model by comparing the predicted categories to the data in the *Testing Set*.

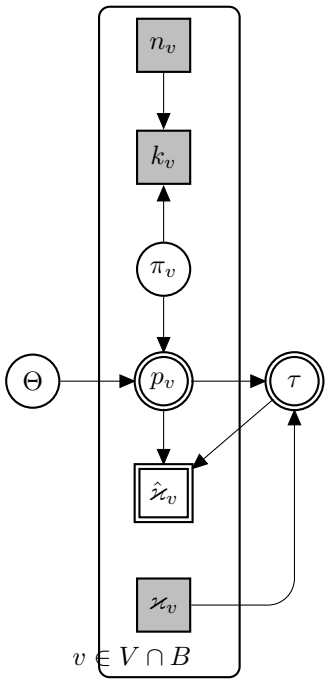
5.4 Bayesian Graphical Model

An alternative way of seeing this problem is through a *Bayesian Graphical Model*, which is explained in more detail in [LW14].

In the model, shown in Figure 5.2, doesn't directly define a *Beta Distribution* $\mathcal{B}_v \sim \text{Beta}(\varpi_v^{\text{high}} + 1, \varpi_v^{\text{low}} + 1)$ for every user v. Instead, the problem is defined as the *Binomial Distribution* in Equation (5.4.1) for some probability π_v.

$$\varpi_v^{\text{high}} \sim \text{Binom}\left(\varpi_v^{\text{low}} + \varpi_v^{\text{high}}, \pi_v\right) \tag{5.4.1}$$

For simplicity, other two new categorical variables are defined for this model for each user v: $\varkappa_v$*, which is 1 if and only if v is part of the high category of income, and $\hat{\varkappa}_v$, which is 1 if and only if v is predicted to be of the high category of income. Additionally, $n_v = \varpi_v^{\text{low}} + \varpi_v^{\text{high}}$ and $k_v = \varpi_v^{\text{high}}$.



$$k_v \sim \text{Binom}\left(n_v, \pi_v\right)$$

$$\Theta \sim \text{Beta}\left(0.5, 0.5\right)$$
$$\pi_v \sim \text{Beta}\left(1, 1\right)$$

$$p_v = Q_{\pi_v}\left(\Theta\right)$$
$$\tau = \underset{t \in [0,1]}{\text{argmax}} \left|\left\{v \in V \cap B \mid p_v \geq t \iff k_v\right\}\right|$$

$$\hat{\varkappa}_v = \begin{cases} 0 \text{ if } p_v < \tau \\ 1 \text{ if } p_v \geq \tau \end{cases}$$

Figure 5.2: A Bayesian Graphical Model representing an interesting alternative approach to the problem, using a *Binomial Distribution* instead of a *Beta Distribution*. If this model was applied with a program such as `Jags`, the results would be the same.

*$\varkappa$ is a variant of the Greek letter κ, which represents a **C**ategory

Chapter 6

Evaluating Performance of Bayesian Prediction Algorithm

In this chapter, we test the *Prediction Algorithm* presented in Chapter 5 with the data present on the graph.

6.1 Experimental Environment

All the tests are being performed in a single `Linux 3.16` server with 16 `Intel Xeon D-1540` cores with 2GHz of power, and 128GByte of RAM.

The programming environment consists of `Python 2.7.9`, which along with the libraries `numpy 1.12.1`, `scipy 0.18.1`, `pandas 0.19.2`, and `scikit-learn 0.18`[PVG+11] were used for creating both the *Bayesian Algorithm*, all the *Machine Learning* methods described in Chapter 7, and the many programs created for feature engineering.

The experimentation was achieved using the previous libraries, using the amazing `Jupyter 5.1.0` notebooks as an environment and `matplotlib 1.5.2` and `seaborn 0.7.1` to create the graphs used in this thesis.

6.2 Data Partitioning

6.2.1 Train Test Split

As with many other classification problems, the *Bayesian Algorithm* is prone to overfitting [Mit97]. In this particular case, since the information presented in Section 5.1 shows that users tend to communicated with users of the same socioeconomic level, by running the algorithm in the complete data and using the same users as part of the features and of the labels, we would erroneously be having more data per user than we would have when modelling the problem.

Since the input data used in this experiment comes from B, the banking information of the users in the telco, we can avoid most of the effects of overfitting separating the

data into a *Training Set* and a *Testing Set*, where the data in B is separated into two disjoint groups as shown in Equation (6.2.1).

$$\begin{aligned} B_{\text{train}} \subseteq B \quad |B_{\text{train}}| &= 0.8 \cdot |B| \\ B_{\text{test}} \subseteq B \quad |B_{\text{test}}| &= 0.2 \cdot |B| \\ B_{\text{train}} \cap B_{\text{test}} &= \varnothing \end{aligned} \tag{6.2.1}$$

6.2.2 Erasing Uninformative Data

The *Social Graph* $G = \langle V, E \rangle$, which contains information about the communication networks of the users in this dataset is extremely sparse. Because of the property and that $|B| \ll |V|$, the vast majority of users don't have any kind of contact with users of the bank. For this reason it is useless to evaluate the performance of the algorithm using all the nodes, and therefore the *Testing Set* used in this thesis will instead focus on the bank users that have at least one contact with another bank user. This approach is formalized in Equation (6.2.2).

$$\begin{aligned} \hat{E} &= \{e \in E \mid e_o \in B_{\text{train}} \vee e_d \in B_{\text{train}}\} \\ \hat{B}_{\text{test}} &= B_{\text{test}} \cap \left(\hat{E}_o \cup \hat{E}_d\right) \end{aligned} \tag{6.2.2}$$

The accuracy of this approach depends of the value of ϖ, which is one of calls, time, sms, or contacts and defines which property of the users that's used as a feature for the classification. If $\varpi =$ contacts, then this approach works perfectly. However, it is possible that for other values of ϖ there won't be any information available in $\hat{B}_{\text{test}}$ in the case of users who either didn't receive any call from a bank user or didn't receive any message.

Equations (6.2.3) and (6.2.4) formalize new variables to use for informative data in those cases.

$$\begin{aligned} \hat{E}^{\text{calls}} &= \{e \in E \mid e_c > 0 \wedge (e_o \in B_{\text{train}} \vee e_d \in B_{\text{train}})\} \\ \hat{B}^{\text{calls}}_{\text{test}} &= B_{\text{test}} \cap \left(\hat{E}^{\text{calls}}_o \cup \hat{E}^{\text{calls}}_d\right) \end{aligned} \tag{6.2.3}$$

$$\begin{aligned} \hat{E}^{\text{sms}} &= \{e \in E \mid e_s > 0 \wedge (e_o \in B_{\text{train}} \vee e_d \in B_{\text{train}})\} \\ \hat{B}^{\text{sms}}_{\text{test}} &= B_{\text{test}} \cap \left(\hat{E}^{\text{sms}}_o \cup \hat{E}^{\text{sms}}_d\right) \end{aligned} \tag{6.2.4}$$

6.2.3 Rebalancing Labels

Since the testing data B_{test} was a random subsample of a balanced set (see Sections 5.2.1 and 6.2.1), it was also balanced itself. However, since *High Income* users tend to communicate more often than *Low Income* ones, $\hat{B}_{\text{test}}$ is unbalanced and has a significant bias for high-income users.

Since the income categories tend to be balanced in the real world, this isn't wanted. However, since it is not necessary to use the entire *Testing Set* for testing the algorithm, a simple way would be to create a new, balanced, and final testing set, $\Upsilon \subseteq \hat{B}_{\text{test}}$ containing all users from $\hat{B}_{\text{test}}$ *Low Income*, along with a random sample of the same size with *High Income*.

$$\begin{aligned} \Upsilon^{\text{low}} &= \hat{B}_{\text{test}} \cap H_1 \\ \Upsilon^{\text{high}} &\subseteq \hat{B}_{\text{test}} \cap H_2 \\ \left|\Upsilon^{\text{low}}\right| &= \left|\Upsilon^{\text{high}}\right| \\ \Upsilon &= \Upsilon^{\text{low}} \cup \Upsilon^{\text{high}} \end{aligned} \tag{6.2.5}$$

Υ will be the only *Testing Set* used from now on, while B_{train} will be used as training set.

Additionally, the sets Υ^{calls} and Υ^{sms} refer to similar sets which are taken from users from the *Testing Set* that had at least one call or sent at least one SMS, respectively, to another user in the *Training Set*.

6.2.4 The Inner and Outer Graph

Most of the rest of the thesis will use the subgraphs Υ, later called the *Inner Graph*, containing only users with at least one labeled neighbour, and B_{test}, the *Outer Graph*, containing all users. For simplicity sake, from now on the *Outer Graph* B_{test} will be referred to as Ω, and the unbalanced outer graph (which is not used after this section) as $\hat{\Omega}$.

As shown in Section 6.2.3, the labels have roughly the same amount of high income and low income users.

The *Outer Graph* Ω is not used in this section, as the *Bayesian Algorithm* requires some information about the *Socioeconomic Level* of the neighbours of the users used

Set	Total Size	High Income	Low Income	Ratio
B	5,402,959	2,702,628	2,700,331	—
Ω	1,080,592	540,526	540,066	—
$\hat{\Omega}$	53,691	35,215	18,476	1.000
Υ	36,952	18,476	18,476	1.000
Υ^{calls}	30,715	15,653	15,062	0.831
Υ^{sms}	11,909	6046	5863	0.322

Table 6.1: Amount of users in the *Testing Set* after trimming it several times to prevent overfitting while keeping the labels balanced

in the prediction. However, it is used as one of the main inputs for generating features in the later Chapter 7.

6.2.5 Set Magnitudes

While the new set Υ contains significantly less users than the original set B, it still has a sufficient amount of people to make a prediction. Table 6.1 shows the number of users that remain after every trim used in this Subsection, along with the ratio of users which we would be able to assign an *Income Category* using these datasets assuming the real data is equally distributed from the *Test Data*.

6.3 Optimizing Θ

In Section 5.2.5 we define the variable Θ, which is defined as the quantile used to define the *Posterior Probability* p_v of a user v being part of the higher income category. Choosing a good value of Θ is an essential step in creating a correct algorithm since it is the most important constant of Equation (5.2.11), one of the crucial parts for finding the category of an user.

As explained in Section 5.2.5, it is convenient to use *Jeffrey's Prior* as the prior distribution for this variable. However, knowing how it will affect the prediction of the category of each v for every $\varpi \in \{\text{calls}, \text{time}, \text{sms}, \text{contacts}\}$ will allow us to get a good posterior value.

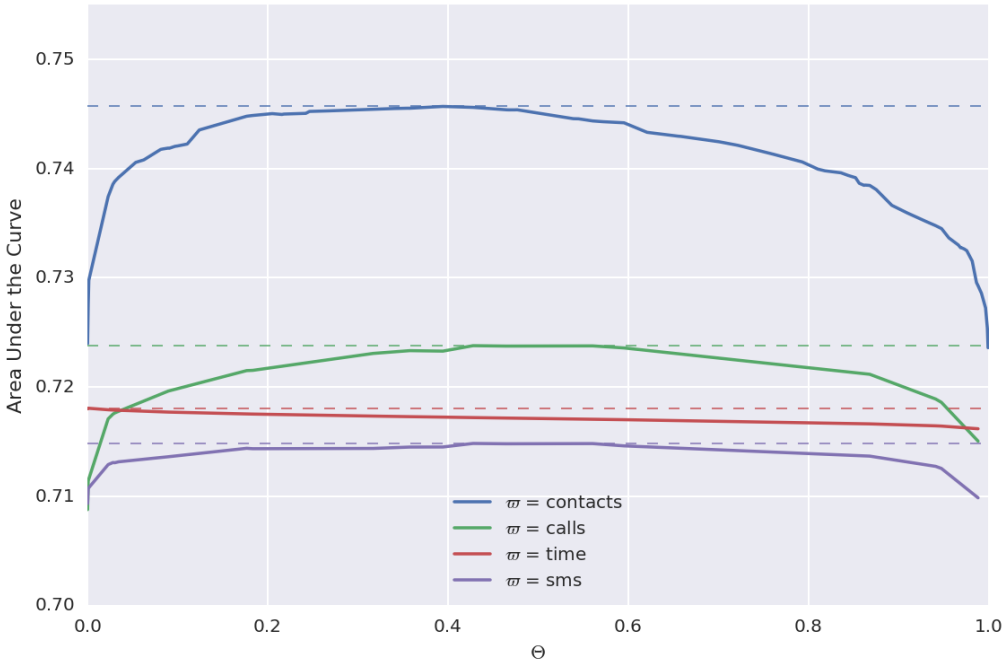


Figure 6.1: The *Area Under the Curve* for different Θ and every possible ϖ. This is the preliminary version of the analysis seen in Section 6.4

.

ϖ	Optimal Θ	AUC
contacts	0.394	0.746
calls	0.428	0.724
time	0.001	0.718
sms	0.428	0.715

Table 6.2: Optimal Θ for each ϖ

The main hypotheses tested in this part of the theses were two assumptions.

1. The optimal value of Θ will be the same for any ϖ.

2. Overlooking extreme values, the value of Θ won't improve or deteriorate the prediction.

As Table 6.2 and Figure 6.1 show, both hypotheses are false.

The *Area Under the Curve* is a good way to analyze the performance of the algorithm with given hyperparameters since it provides a good equilibrium between *Precision* and *Recall*. Additionally, in Section 5.2.6, value τ is defined to set the limit

between the users categorized between *High Income* and *Low Income*. Since this value is independent from the *Area Under the Curve*, it is not necessary to define it in this part of the analysis.

In this analysis we can see that there are different optimal Θ for every ϖ, contradicting the first hypothesis although the best value seems to be the same for $\varpi = \text{calls}$ and $\varpi = \text{sms}$. The reason for this equality remains a mystery.

Additionally, there is a significant difference between the values of Θ on all input types, contradicting the second one, except for $\varpi = \text{time}$. This is probably caused because the calling time is a lot more varied than with the other statistics, as is shown in Section 6.4.2.

6.4 Algorithm Performance on All Users

The *Bayesian Algorithm* will be ran for every $\varpi \in \{\text{contacts}, \text{calls}, \text{time}, \text{sms}\}$. For every possible configuration, we present 3 plots for the optimal Θ.

- A **histogram** presenting the distribution of the p_v values which result from applying Equation (5.2.11) presented in Section 5.2.5 to each distinct *Beta Distribution*.
  - Some interesting pairs $\langle \alpha, \beta \rangle$ which correspond to particularly high bars in the histogram are marked.
- A **Receiver Operating Characteristic Curve**, showing the trade-off of *False Positive Rate* to *True Positive Rate* when selecting every possible τ. The *Area Under the Curve* is marked, as this is the metric that is being maximized when selecting the correct ϖ.
- An **Accuracy Curve**, which shows the *Accuracy* of the predictor by its *False Positive Rate*.

We also use the *Accuracy Curve* to define the value of τ, the limit between the users defined in the *High Income* and the *Low Income* categories, in order to to maximize accuracy in this method.

Many metrics, previously described in Section 2.5 will be used to measure the *Bayesian Algorithm* for different ϖ. The results are later shown in Table 6.3.

6.4.1 Inferring by Calls

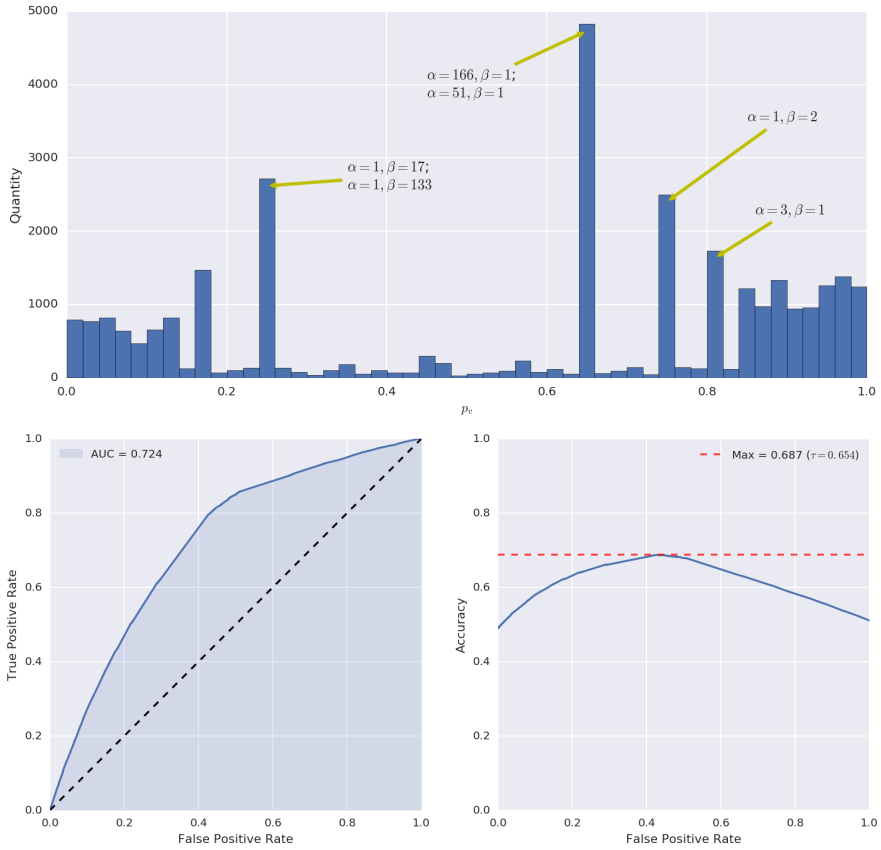


Figure 6.2: Results of the *Bayesian Algorithm* for call data.

Figure 6.2 contains data about the predictor when ϖ = calls and the data is analyzed using Υ^{calls} as *Testing Set*. The *Inverse Cumulative Distribution Function* contains a few peaks for users with a similar amount of calls.

After analyzing the data, we find that the *Area Under the Curve* using this method is of 0.724, which is significantly higher than all the naïve and *Machine Learning* methods that will be presented in the later Chapter 7.

Setting $\tau = 0.654$ maximizes the accuracy at Accuracy = 0.687. Additionally, that value of τ results in Precision = 0.653, Recall = 0.815, $F_1 = 0.725$, and $F_4 = 0.804$.

6.4.2 Inferring by Time

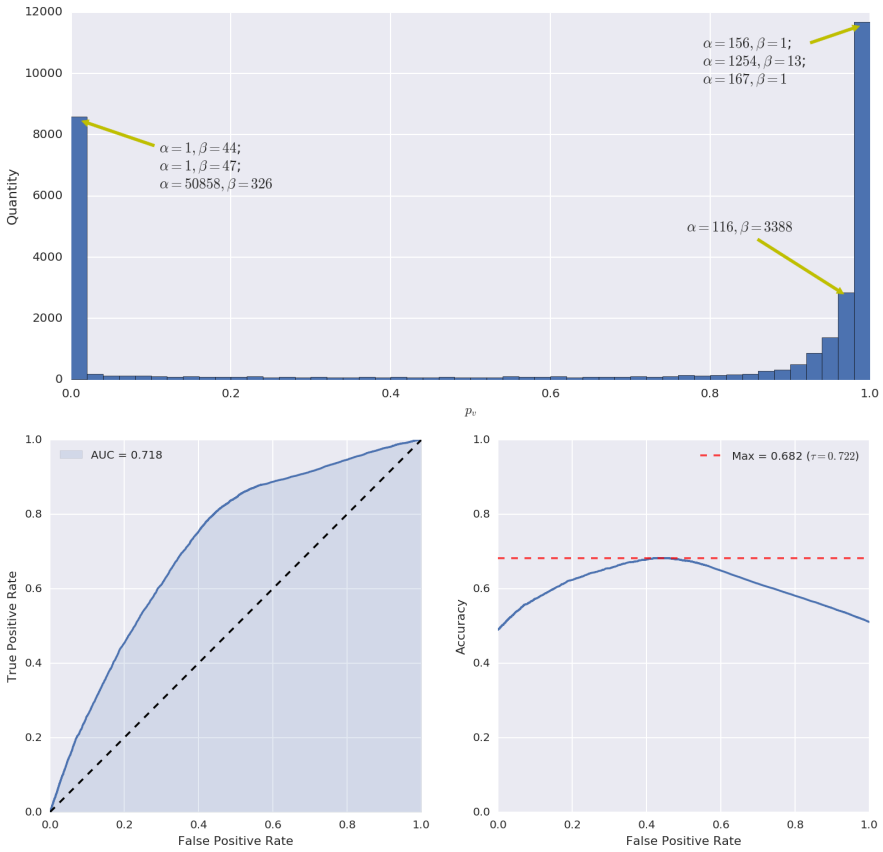


Figure 6.3: Results of the *Bayesian Algorithm* for time data.

Figure 6.3 contains data about the predictor when ϖ = time and the data is analyzed using Υ^{calls} as *Testing Set*, there are two big clusters of data at the edges; this is explained because the majority of users spend most of their time talking to either *High Income* or *Low Income* users.

The *Area Under the Curve* of this inference mechanism is AUC = 0.718, which is lower than the one for the calls in Section 6.4.1. The *Accuracy Curve* is unsurprisingly similar to that one, and even the *Accuracy* at $\tau = 0.722$ is the close. This is probably a result of the fact that there is an obvious correlation between total talking time and total calls.

This τ also results in a predictor where Accuracy = 0.682, Precision = 0.649, Recall = 0.819, $F_1 = 0.724$, and $F_4 = 0.807$.

6.4.3 Inferring by SMS

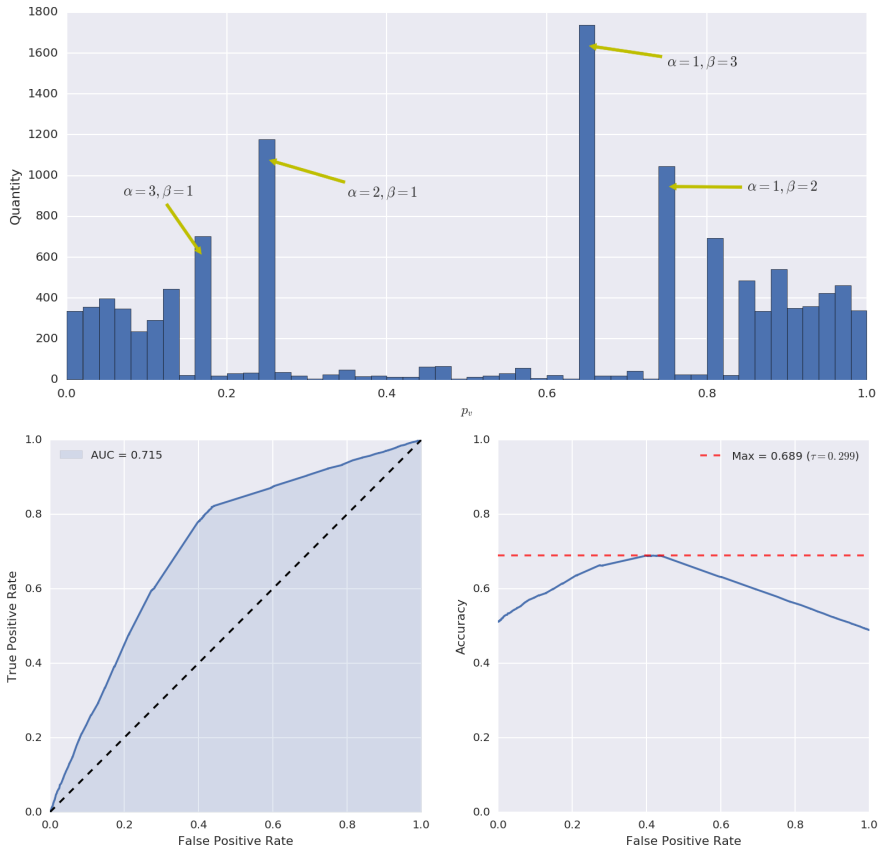


Figure 6.4: Results of the *Bayesian Algorithm* for SMS data.

Figure 6.4 shows the distributions when $\varpi = \text{sms}$. Since the total amount of SMS is much lower than the amount of calls, the peaks of the result of the *Inverse Cumulative Functions* of the *Beta Distribution* applied on Υ^{sms} that happen with the majority of users that have few of both are located closer to the center than in Sections 6.4.1 and 6.4.2. This makes some interesting cases if $\varpi = \text{sms}$ is chosen, since the distribution is different than in the other cases.

In particular, this gives an AUC = 0.715, which is lower than both in the case of *Calls* and *Time*. Interestingly, the maximum *Accuracy* at $\tau = 0.299$ is slightly higher than both of the other cases; this is probably a side-effect of the fact that $|\Upsilon^{\text{sms}}| < \left|\Upsilon^{\text{calls}}\right|$.

Additionally, Precision = 0.696, Recall = 0.186, $F_1 = 0.293$, and $F_4 = 0.194$.

6.4.4 Inferring by Contacts

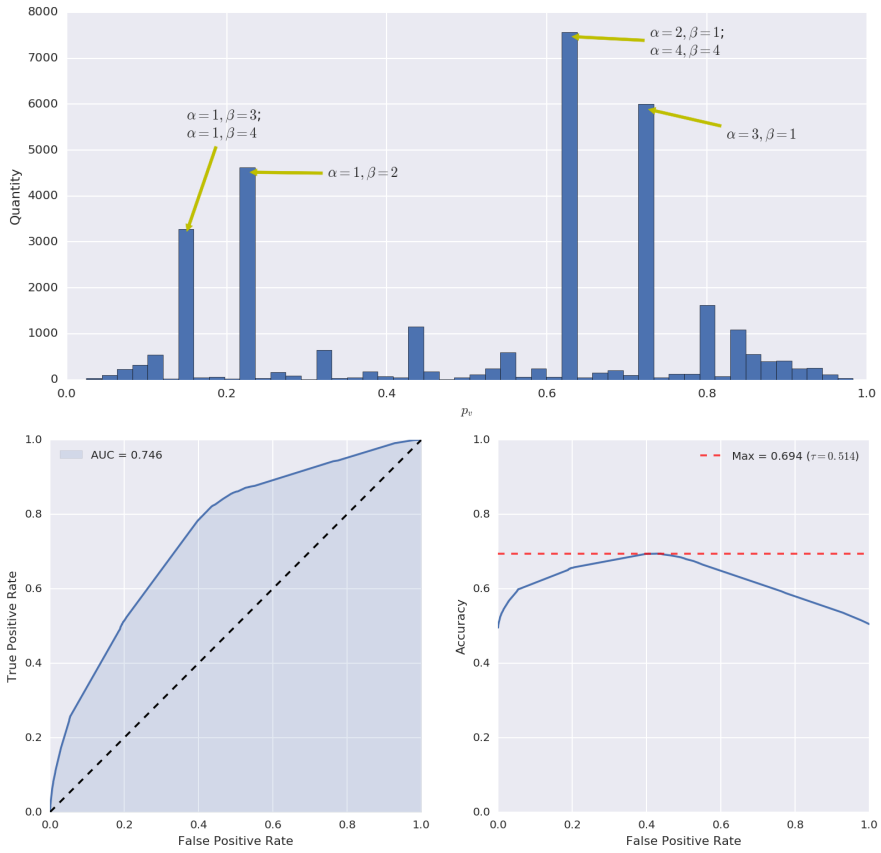


Figure 6.5: Results of the *Bayesian Algorithm* for degree data.

Figure 6.5 shows the distributions when $\varpi =$ contacts, where it is possible to get a pattern similar to the one shown in Section 6.4.3 when $\varpi =$ sms, where the majority of users have relatively few contacts and the peaks in the histogram. Additionally, since the total amount of contacts is exponentially distributed (as shown in Figure 4.2), and people with *High Income* tend to have more contacts in general, there peaks are clustered in areas with low p_v (where the majority of calls are made to *Low Income* users), near the middle (where the calls are mostly equally distributed), but not at high p_v; this last section would belong to the few users with many calls to *High Income* users.

Using this method it is possible to find that AUC $= 0.746$, which is higher than all the other methods presented in Section 6.4. Additionally, when selecting $\tau = 0.514$, Accuracy $= 0.694$ which is higher than the maximum *Accuracy* in all other methods.

ϖ	Θ	τ	Acc.	Prec.	Rec.	AUC	F_1	F_4
calls	0.428	0.654	0.686	0.654	0.816	0.724	0.726	0.804
time	0.001	0.722	0.681	0.652	0.806	0.718	0.721	0.795
sms	0.428	0.299	0.688	0.648	0.789	0.715	0.712	0.779
contacts	0.394	0.514	0.693	0.665	0.792	0.746	0.723	0.783

Table 6.3: Metrics for the Bayesian algorithm using every user in Υ

These metrics, combined with the fact that Υ contains every user in the *Testing Set*, result in the fact that $\varpi =$ contacts is unambiguously the best way to classify the data for the algorithm. Additionally, Precision $= 0.556$, Recall $= 0.792$, $F_1 = 0.723$, and $F_4 = 0.783$.

6.4.5 Final Results

Table 6.3 presents every metric discussed in Section 2.5 for the optimal Θ and τ for every ϖ.

In particular, the results show that using *Contacts* as the predictor for the *Bayesian Algorithm* results in a significantly higher *Area Under the Curve* and a higher *Accuracy*.

6.5 Algorithm Performance of Users with at least 3 Contacts

The algorithm tends to be a better predictor of the *Socioeconomic Level* for users with high amount of information on the graph G, namely that the amount of users in their neighborhood that also belong to B is big.

In this section, we run the *Bayesian Algorithm* for the subset of the users presented in Equation (6.5.1), which restrict the users in the *Testing Set* to only those who have at least 3 contacts. This would allow us to have better metrics in the dataset, at the expense of a much smaller Universe of users for which the algorithm could be applied. Additionally, Table 6.4 shows the sizes of the *Testing Sets* used in a manner similar to Table 6.1.

$$\begin{gathered} I = \left\{ v \in \Upsilon \mid \text{contacts}_v^{high} + \text{contacts}_v^{\text{low}} > 3 \right\} \\ I^{\text{calls}} = \Upsilon^{\text{calls}} \cap I \end{gathered} \tag{6.5.1}$$

Set	Total Size	High Income	Low Income	Ratio
I	7932	4637	3295	0.258
I^{calls}	7910	4627	3283	0.214

Table 6.4: Amount of users in the *Testing Set* after trimming it several to only have users with at least 3 contacts.

6.5.1 Inferring by Calls on Users with at least 3 Contacts

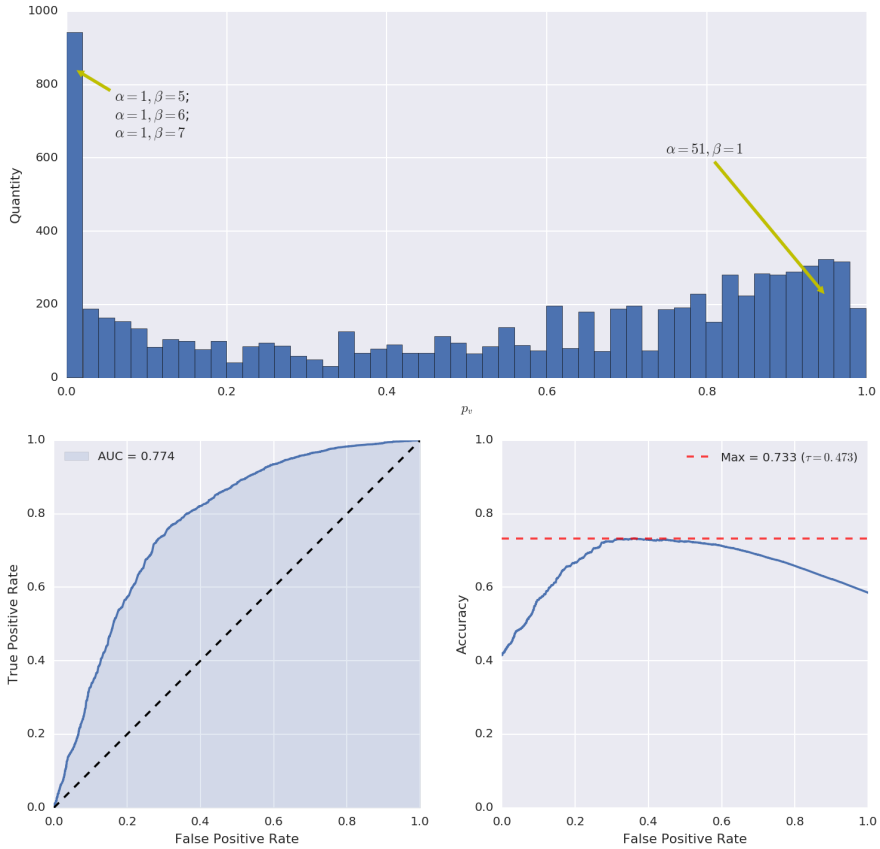


Figure 6.6: Results of the *Bayesian Algorithm* for call data by only counting users with at least 3 contacts.

Figure 6.6 uses $I^{\text{calls}} \subseteq \Upsilon^{\text{calls}}$ and $\varpi = \text{calls}$ to create a predictor where both the *Area Under the Curve* and the *Accuracy* are higher than in all predictors that use every possible user. However, this data predicts less users and is strictly worse than the one presented in the following section.

6.5.2 Inferring by Degree on Users with at least 3 Contacts

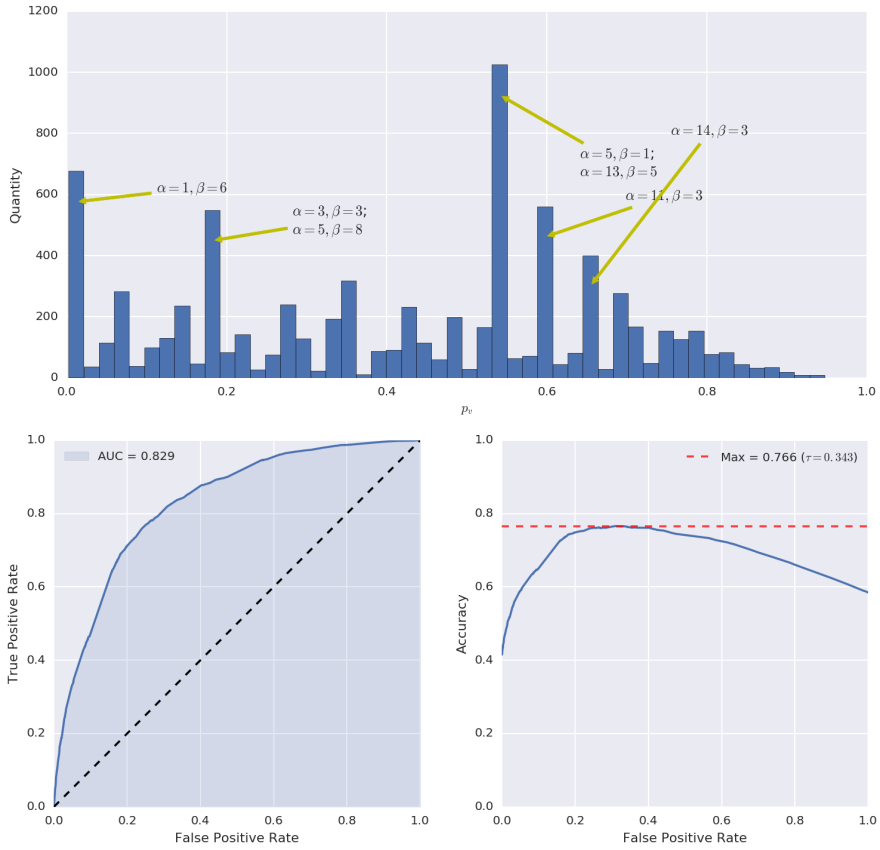


Figure 6.7: Results of the *Bayesian Algorithm* for contacts data by only counting users with at least 3 contacts.

The results in Figure 6.7 show the best possible predictor for a reasonable subset of the users. Both the *Area Under the Curve* and the *Accuracy* are higher than for all other predictors, with AUC $= 0.829$ and Accuracy $= 0.766$.

All of those values are considerably better than any other predictor in this page at the cost of using only a small subset of the possible users. This is interesting as it is exactly the same subset as the one used in [FBB+16], and it scores significantly higher in all the metrics.

Chapter 7

Comparison with Other Inference Methods

In this chapter, we compare the results of the Bayesian method presented in Chapter 5 with common machine learning methods using several feature extraction methods in the graph used for this study. Like in the previous chapter, the idea is to know whether each user v belongs to a *Low Category* of income, H_1, or to a *High Category*, H_2, by knowing the distribution of calls and text messages to his contacts.

These classifier creates a proper baseline for other comparisons. These will be used in the *Social Graph* $G = \langle V, E \rangle$, that contains data about the communications network for every user, along with the banking data B separated into training and testing sets B_{train} and B_{test}, and will be used to provide either information about the users.

Unlike the method presented in the previous chapter, we can optionally forego the necessity for having at least one neighbour with known socioeconomic index for each user. For this reason, we experiment with these methods with two possible inputs, both previously defined in Section 6.2.4.

- The *Outer Graph* Ω, with information about every node in the graph G.
- The *Inner Graph* Υ, which only contains information about the subset of nodes which have at least one neighbour with known *Socioeconomic Index*

At the end of this chapter, in Section 7.6, we present a group of tables that compare the results of these methods with the *Bayesian Algorithm* described in Chapter 5 along several metrics.

7.1 Random Selection

One of the simplest methods for solving many classification problems is using random selection. This classifier simply chooses randomly to which strata each user belongs.

$$\begin{aligned} P\left(v \in H_1\right) &= 0.5 \\ P\left(v \in H_2\right) &= 0.5 \end{aligned} \tag{7.1.1}$$

This produces a good baseline for comparing other inference methods.

7.2 Majority Voting

The method of *Majority Voting* is a basic but powerful way of inferring which category a user belongs to. It simply chooses the category of each user $v \in V$ as the most common category within its contacts.

In case of a tie (which happens often when using the *Outer Graph*, as many users don't have any neighbour with a known category), the category is chosen randomly. This approach can be formalized as in Equation (7.2.1). Here, we use the values $\text{contacts}_v^{\text{low}}$ and $\text{contacts}_v^{\text{high}}$ defined in Chapter 5: the amount of contacts user v has of either low or high socioeconomic index, respectively.

$$P\left(v \in H_1\right) = \begin{cases} 0 & \text{if } \text{contacts}_v^{\text{low}} < \text{contacts}_v^{\text{high}} \\ 0.5 & \text{if } \text{contacts}_v^{\text{low}} = \text{contacts}_v^{\text{high}} \\ 1 & \text{if } \text{contacts}_v^{\text{low}} > \text{contacts}_v^{\text{high}} \end{cases} \tag{7.2.1}$$

7.3 Methods Based in Machine Learning

The following methods use commonly used supervised Machine Learning algorithms, described in Section 2.6, used with many similar *Feature Extraction* methods on the *Social Graph G*.

The initial feature extraction method, referred to as *User Data*, is described in Section 7.3.1 and marked as Nbr_0, consists of accumulating the total information about the links neighboring some user $v \in V$. Later, as shown in Section 7.3.3, it is possible to accumulate links from more levels of the *Ego Network* to create feature sets that be used to build a better prediction of the data, marked Nbr_n for some level n.

An extra method for feature extraction is presented in Section 7.3.2, which in addition to doing the extraction of other methods it accumulates separately edges

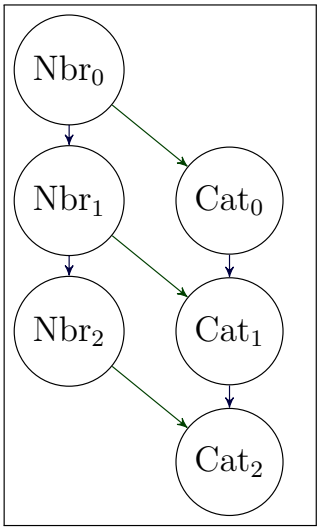


Figure 7.1: Relationships between the *Feature Extraction* methods of Section 7.3. blue edges represent an increase in *Ego Network* size, a process which is describe in Section 7.3.3, while green edges represent adding label information, which is described in Section 7.3.2

Level	Features
Nbr_0	8
Nbr_1	16
Nbr_2	24
Cat_0	24
Cat_1	48
Cat_2	72

Table 7.1: Number of features used for testing each Machine Learning model on each level

that go to *High Income* and *Low Income* users. These methodologies are marked with Cat_n for some level n.

The features of each method are merged with the ones from the immediate predecessor methods, as shown in the graph in Figure 7.1, while the amount of features in each level is described in Table 7.1.

7.3.1 User Data — Level Nbr_0

The features on the *Social Graph* $G = \langle V, E \rangle$, which contains communications data between all the users, can be accumulated for all users in a manner similar to the one

in Section 5.2.2, presented again in Equation (7.3.1).

$$\begin{aligned}
\text{calls}_v^{\text{low}} &= \sum_{\substack{e\in E\\ e_d=v\\ e_o\in H_1}} e_c + \sum_{\substack{e\in E\\ e_o=v\\ e_d\in H_1}} e_c \quad \text{calls}_v^{\text{high}} = \sum_{\substack{e\in E\\ e_d=v\\ e_o\in H_2}} e_c + \sum_{\substack{e\in E\\ e_o=v\\ e_d\in H_2}} e_c \\
\text{time}_v^{\text{low}} &= \sum_{\substack{e\in E\\ e_d=v\\ e_o\in H_1}} e_t + \sum_{\substack{e\in E\\ e_o=v\\ e_d\in H_1}} e_t \quad \text{time}_v^{\text{high}} = \sum_{\substack{e\in E\\ e_d=v\\ e_o\in H_2}} e_t + \sum_{\substack{e\in E\\ e_o=v\\ e_d\in H_2}} e_t \\
\text{sms}_v^{\text{low}} &= \sum_{\substack{e\in E\\ e_d=v\\ e_o\in H_1}} e_s + \sum_{\substack{e\in E\\ e_o=v\\ e_d\in H_1}} e_s \quad \text{sms}_v^{\text{high}} = \sum_{\substack{e\in E\\ e_d=v\\ e_o\in H_2}} e_s + \sum_{\substack{e\in E\\ e_o=v\\ e_d\in H_2}} e_s \\
\text{contacts}_v^{\text{low}} &= |\{e\in E \mid e_o = v \wedge e_d \in H_1\} \cup \{e\in E \mid e_d = v \wedge e_o \in H_1\}| \\
\text{contacts}_v^{\text{high}} &= |\{e\in E \mid e_o = v \wedge e_d \in H_2\} \cup \{e\in E \mid e_d = v \wedge e_o \in H_2\}|
\end{aligned} \tag{7.3.1}$$

However, and unlike in the experiments in Chapter 5, this time we aren't constrained either by having only users which have contacts with other users with known *Income Category*, nor with having to have only two features for each category which was necessary since the solution used the *Beta Distribution*.

This way, it is possible to accumulate features for each user $v \in V$ as in Equation (7.3.2).

$$\begin{aligned}
\text{incalls}_v &= \sum_{\substack{e\in E\\ e_d=v}} \text{calls}_e \quad \text{outcalls}_v = \sum_{\substack{e\in E\\ e_o=v}} \text{calls}_e \\
\text{outtime}_v &= \sum_{\substack{e\in E\\ e_d=v}} \text{time}_e \quad \text{outtime}_v = \sum_{\substack{e\in E\\ e_o=v}} \text{time}_e \\
\text{insms}_v &= \sum_{\substack{e\in E\\ e_d=v}} \text{sms}_e \qquad \text{insms}_v = \sum_{\substack{e\in E\\ e_o=v}} \text{sms}_e \\
\text{incontacts}_v &= |\{e\in E \mid e_d = v\}| \\
\text{outcontacts}_v &= |\{e\in E \mid e_o = v\}|
\end{aligned} \tag{7.3.2}$$

These features will be referred as the *User Data* of user v. There features contain information about the *Neighborhood* of v, also referred to as the *Ego Network of Distance 1*. This definition is used later in this chapter, and the nodes belonging to the *Neighborhood* of v can be formally defined as in Equation (7.3.3).

$$\text{Neigh}(v) = \{e_o \mid e\in E \wedge e_d = v\} \cup \{e_d \mid e\in E \wedge e_o = v\} \tag{7.3.3}$$

7.3.2 Categorical User Data — Level Cat_0

Given the *Inner Graph* Υ, containing the subset of users where at least one neighbour has bank information, a possible feature set consists of separating the data of the neighborhood of each user $v \in \Upsilon$ into two disjoint groups, L_v and K_v, which contain the neighbors of v in the *Low Income* and *High Income* categories of income respectively*.

$$\begin{aligned} L_v &= H_1 \cap \mathrm{Neigh}\,(v) \\ K_v &= H_2 \cap \mathrm{Neigh}\,(v) \end{aligned} \tag{7.3.4}$$

Having defined these groups it is possible to define a set of features similar to the one in Section 7.3.1, where each feature is separated by the category of the neighbor. Equation (7.3.5) represents an intuitive way to define the names of the new features.

$$\left\{ \begin{matrix} \text{in} \\ \text{out} \end{matrix} \right\} \times \left\{ \begin{matrix} \text{calls} \\ \text{time} \\ \text{sms} \\ \text{contacts} \end{matrix} \right\} \times \left\{ \begin{matrix} \text{low} \\ \text{high} \end{matrix} \right\} \tag{7.3.5}$$

Equations (7.3.6) and (7.3.7) contain the way to calculate those features. Since the number of individual features is high and the formulas are similar and repetitive, the variable ζ is defined for all the types of properties.

$$\begin{gathered} \text{For } \zeta \text{ being any of } \{\text{calls}, \text{time}, \text{sms}\}, \\ \underline{\text{in}\zeta\text{low}}_v = \sum_{\substack{e \in E \\ e_d \in L_v \\ e_o = v}} \zeta_e \quad \underline{\text{in}\zeta\text{high}}_v = \sum_{\substack{e \in E \\ e_d \in K_v \\ e_o = v}} \zeta_e \\ \underline{\text{out}\zeta\text{low}}_v = \sum_{\substack{e \in E \\ e_o \in L_v \\ e_d = v}} \zeta_e \quad \underline{\text{out}\zeta\text{high}}_v = \sum_{\substack{e \in E \\ e_o \in K_v \\ e_d = v}} \zeta_e \end{gathered} \tag{7.3.6}$$

$$\begin{aligned} \text{incontactslow}_v &= |\{e \in E \mid e_d = v \wedge e_o \in L_v\}| \\ \text{incontactshigh}_v &= |\{e \in E \mid e_d = v \wedge e_o \in K_v\}| \\ \text{outcontactslow}_v &= |\{e \in E \mid e_o = v \wedge e_d \in L_v\}| \\ \text{outcontactshigh}_v &= |\{e \in E \mid e_o = v \wedge e_d \in K_v\}| \end{aligned} \tag{7.3.7}$$

*Note that, since not all users have banking information, there may be nodes in the neighborhood of v which don't belong to either L_v or K_v.

Unlike the features in Section 7.3.1, and like the method presented in Chapter 5, the features in this section will be be different when testing between the nodes of Ω (the *Outer Graph*) and Υ (the *Inner Graph*).

7.3.3 Higher Order User Data — Level $\mathrm{Nbr}_{n>0}$

The features described in Section 7.3.1 correspond to the information about calls and SMS from an user $v \in \Upsilon$ towards all of its neighbors, which is described as the *Ego Network of Distance 1*. However, there's no reason why this information can't be extended to other nodes at a higher distance from v.

If the distance between two nodes is defined using the intuitive definition presented in Equation (7.3.8), it is possible to define the *User Data of Order n* for any number $n \in \mathbb{N}$ as the accumulation of calls and SMS where one endpoint is on the border of the *Ego Network of Order n* and the other one isn't. The *Ego Network of Order n* or a certain node v is the sub-graph composed of the node v and all the nodes which are at at most distance n of v.

$$d(a,b) = \begin{cases} 0 & \text{if } a = b \\ 1 + \min_{v \in \mathrm{Nbr}(b)} d(a,v) & \text{otherwise} \end{cases} \tag{7.3.8}$$

This method creates a set of *Higher Order Features* similar to the ones in Equation (7.3.2), with the exception that the values of the edges summed for each node are defined on the distance instead of on the definition of the E itself. Indeed, for the *Social Graph* $G = \langle V, E \rangle$ we can define features with formulas similar to the ones in Equation (7.3.9).

$$\mathrm{incalls}_v^n = \sum_{\substack{e \in E \\ d(e_o,v)=n \\ d(e_d,v)=n+1}} \mathrm{calls}_e \quad \mathrm{outcalls}_v^n = \sum_{\substack{e \in E \\ d(e_d,v)=n \\ d(e_o,v)=n+1}} \mathrm{calls}_e \tag{7.3.9}$$

This definition can be seen more intuitively in the graph in Figure 7.2.

Every set Nbr_n contains features from that level and all the previous ones as presented formally in Equation (7.3.10). This implies that, if the prediction algorithms are smart enough, the result of using the features from Nbr_{n+1} should be at least as good as using the ones from Nbr_n.

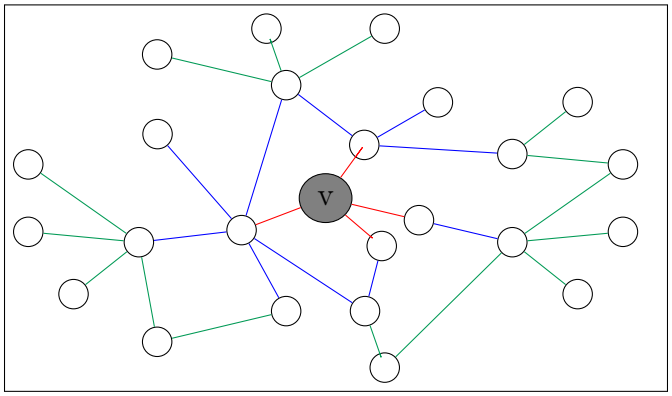


Figure 7.2: Example of the edges present in the calculation of the *Higher Order User Data* for a certain node v. Red edges represent edges whose features are accumulated in the *User Data of Order 0*, blue edges represent those of *Order 1*, and green those of *Order 2*.

$$(\forall n \in \mathbb{N})\, \mathrm{Nbr}_n \subset \mathrm{Nbr}_{n+1} \tag{7.3.10}$$

In one of the initial experiments in this thesis we added more features related to the $\{\mathrm{in}, \mathrm{out}\}$ group to indicate how many "in" or "out" edges must be traversed from v. However, this added an exponentially large new amount of features without much significant data, and thus was left out from the following experiments.

7.3.4 Higher Order Categorical User Data — Level $\mathrm{Cat}_{n>0}$

It is possible to combine the ideas in Sections 7.3.2 and 7.3.3 to create a group of sets of features containing the data from the border of the *Ego Network or Order n*, separated by two disjoint groups depending in the category of the node outside the ego network, as in Equation (7.3.11), with E defined as in Chapter 4 as the set of edges where e_o is the user that originated a call and e_d the user in the destination.

$$\begin{aligned}
\mathrm{incallslow}_v^n &= \sum_{\substack{e\in E \\ e_d\in H_1 \\ d(e_o,v)=n \\ d(e_d,v)=n+1}} \mathrm{calls}_e \qquad \mathrm{incallshigh}_v^n = \sum_{\substack{e\in E \\ e_d\in H_2 \\ d(e_o,v)=n \\ d(e_d,v)=n+1}} \mathrm{calls}_e \\
\mathrm{outcallslow}_v^n &= \sum_{\substack{e\in E \\ e_o\in H_1 \\ d(e_d,v)=n \\ d(e_o,v)=n+1}} \mathrm{calls}_e \qquad \mathrm{outcallshigh}_v^n = \sum_{\substack{e\in E \\ e_o\in H_2 \\ d(e_d,v)=n \\ d(e_o,v)=n+1}} \mathrm{calls}_e
\end{aligned} \tag{7.3.11}$$

Using these features we can have a more complete understanding of the data surrounding each node v. In Section 7.6, where we present the final results, we prove that the best predictor between all the methods explored in this chapter is this same method defining when using a *Random Forest* to predict the final labels, both when using the inner graph (and using only the ego network when $n = 1$) and when using the entire outer graph (and using the ego network with $n = 2$).

The amount of features grows exponentially in each step (see Table 7.1), and so does the time spent in each computation. Since the best results are found with metrics when looking at the *Ego Network* of level 1 of the inner graph (that is, the information contained in the edges of the neighbours of the neighbours of each node), which can be explained by the fact that the noise of the information about users far away from the noise has more impact than the useful information, no tests were attempted in cases where $n \geq 3$.

7.4 Machine Learning Methods

All the feature sets described in the previous sections are individually going to be used as the input objects of the *Training Data* and either the *Outer Graph* Ω or the *Inner Graph* Υ will be used as the output with several *Supervised Machine Learning* methods. The result of these methods will be measured using many metrics of the results, previously described in Section 2.5, and compared against other methods (including the Bayesian Algorithm of Chapter 5) in Section 7.6.

To prevent the problem of *Overfitting* the results are generated by using *Cross-Validated* estimates of the data using *K-Folds* with $K = 5$. This way, each quintile of the data is predicted using only data from the other four.

The two methods used in this thesis, *Logistic Regression* and *Random Forest* tend to have different variance in the results [TMVS16], therefore different sources of errors may end up decreasing the models accuracy differently. This is a good for our model, since it means that a single source of errors has less probability of affecting either predictor.

These *Machine Learning* methods used in this section contain many hyperparameters which may affect the result, and calculating the optimal value for them isn't trivial. For this reason this experiment includes a *Grid Search* of the data against

Data	Level	Log. Regression	Random Forest		
		$\log_{10}(C)$	Criterion	Features	W/Replacement
Υ	Nbr_0	-2	Entropy	$\sqrt{f}$	True
	Nbr_1	-2	Entropy	f	True
	Nbr_2	-2	Entropy	f	True
	Cat_0	-3	Entropy	$\log_2 f$	True
	Cat_1	0	Entropy	f	True
	Cat_2	-1	Entropy	f	True
Ω	Nbr_0	0	Gini	$\log_2 f$	True
	Nbr_1	1	Entropy	$\log_2 f$	True
	Nbr_2	0	Entropy	$\sqrt{f}$	True
	Cat_0	-2	Entropy	$\log_2 f$	True
	Cat_1	2	Gini	$\sqrt{f}$	True
	Cat_2	2	Entropy	$\log_2 f$	True

Table 7.2: Best hyperparameters for each group of features in each model used for predicting the result.

all possible hyperparameters. The resulting hyperparameters of the *Grid Search* are presented in Table 7.2.

7.4.1 Logistic Regression

The method of *Logistic Regression*, which is described with more detail in Section 2.6.2, consists of doing some regression analysis with the data after applying some *Logistic Function* to normalize the data.

The most important hyperparameter of this data is the *Regularization Factor* C, which specifies the regularization of the input. As shown in Equation (7.4.1), this value is searched in exponential increments.

$$C \in \left\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}\right\} \tag{7.4.1}$$

Since the different data aren't linearly distributed, the input is *Regularized* before using this method so that, when running the model, each column has its mean in $\mu = 0$ and its variance $\sigma^2 = 1$. This makes the model more robust [Mit97].

7.4.2 Random Forest

The method of *Random Forest*, which is described with more detain in Section 2.6.4, consists of constructing a multitude of *Decision Trees* and outputting the class that is the *mode* of the classes.

There are several hyperparameters used in this method, namely the `Criterion` to measure the quality of a split (`gini` uses Gini impurity, while `entropy` uses information gain), the function used to calculate the `Sample Size` given the amount of features f, and whether the samples are taken with or without `Replacement`. This is formalized in Equation (7.4.2).

Another possible hyperparameter would be the number of trees. However, as it is been shown in [Bre01], *Random Forests* converge quickly for a high amount of trees. Since the objective of this section isn't to optimize by time, the value will be set to the sufficiently high `n_estimators` $= 50$.

$$\begin{aligned} \text{Criterion} &\in \{\text{Gini}, \text{Entropy}\} \\ \text{Features} &\in \left\{ f, \sqrt{f}, \log_2 f \right\} \\ \text{Replacement} &\in \{\text{True}, \text{False}\} \end{aligned} \tag{7.4.2}$$

7.5 Validation Metrics

There are several validation metrics used for each method.

Accuracy (Acc.) as described in Section 2.5.4, which measures the general performance of this method.

Precision (Prec.) as described in Section 2.5.2, which measures the performance regarding the positive instances found by this method.

Recall (Rec.) as described in Section 2.5.2, which measures the performance regarding the positive instances in the dataset.

Area Under the Curve (AUC) as described in Section 2.5.6, which measures the general performance disregarding which threshold is used.

F_1 Score (F_1) as described in Section 2.5.7 which is generalized score balancing Precision and Recall.

F_4 Score (F_4) as described in Section 2.5.7, which gives more weight to the Recall. This is usually wanted since the ultimate practical objective of this study is to find wealthier people, even if the result has low Precision.

Fit Time (t_{fit}) is the time it takes to fit a model. It is particularly high in ensemble methods such as *Random Forest*.

Predict Time ($t_{predict}$) can be used to break ties between similar models.

7.6 Results

7.6.1 Inner Graph

Table 7.3 shows various metrics which result from applying the methods described in this section to the datasets presented in this thesis as the *Inner Graph* Υ, which contains the subset of people where at least one neighbour contains socioeconomic data.

By comparing only the methods based in *Machine Learning*, we can reach a conclusion consistent with the one reached by Muchlinski et al. [MSHK16], where the methods based in *Random Forest* tend to perform better in real-world scenarios than the ones based in *Logistic Regression*.

The model based in *Random Forest* has better results in all metrics when comparing the datasets of Nbr_0 and Nbr_1, that is, when making the *Ego Network* of users where data is collected one degree bigger. Interestingly, not only these results aren't repeated when comparing the following two levels (Nbr_1 and Nbr_2), but most metrics, including *Area Under the Curve*, show a slight decrease. While this means that an extra degree of data adds no useful information instead of just noise, using this data it is possible to see the fallibility of common *Random Forest* fitting methods, as this regression shouldn't be possible since $Nbr_2 \supset Nbr_1$ and the noise features could have been ignored by the hypothetical perfectly-trained classifier.

The results are worse in all metrics of the Nbr datasets when predicting using a *Logistic Regression* method. One possible cause of this is that the *Random Forest*

Model	Level	Acc.	Prec.	Rec.	AUC	F_1	F_4	t_{fit}	t_{pred}
Random		0.499	0.499	0.500	0.499	0.500	0.500	—	0.005 s
Majority		0.681	0.640	0.826	0.681	0.721	0.712	—	0.059 s
Bayesian		0.693	0.665	0.792	**0.746**	**0.723**	**0.783**	—	33.155 s
LR	Nbr_0	0.536	0.531	0.625	0.536	0.574	0.619	0.145 s	0.002 s
	Nbr_1	0.535	0.525	0.730	0.535	0.611	0.714	0.141 s	0.011 s
	Nbr_2	0.568	0.578	0.525	0.569	0.550	0.528	0.119 s	0.003 s
	Cat_0	0.686	0.655	0.785	0.686	0.714	0.776	0.167 s	0.005 s
	Cat_1	0.693	0.665	0.780	0.693	0.718	0.772	1.588 s	0.011 s
	Cat_2	0.693	0.670	0.764	0.692	0.714	0.758	0.956 s	0.009 s
RF	Nbr_0	0.548	0.548	0.550	0.548	0.549	0.550	5.986 s	0.588 s
	Nbr_1	0.582	0.583	0.577	0.582	0.580	0.577	56.548 s	0.483 s
	Nbr_2	0.576	0.577	0.580	0.576	0.579	0.580	50.197 s	0.253 s
	Cat_0	0.671	0.665	0.690	0.671	0.677	0.688	6.346 s	0.539 s
	Cat_1	**0.714**	0.713	0.716	0.714	0.714	0.716	96.005 s	0.460 s
	Cat_2	0.709	0.710	0.711	0.709	0.711	0.711	81.528 s	0.242 s

Table 7.3: Resulting metrics of different methods used in Chapters 6 and 7 tested on the *Inner Graph* Υ, which contains only nodes which have at least one neighbour with socioeconomic information. **Bolded** items represent the highest value for each metric.

classifier is more versatile to odd cases and non-linear data when compared to the other classifier [Sac]. The fact that there is almost a null increase in the metrics when going down the graph seems to ratify this decision.

By far, the greatest increase in results for the methods based in *Machine Learning* presented in Chapter 7 is adding labels related to the accumulated features separated by the neighbours' socioeconomic index, as in the levels Cat_0 to Cat_2. This shows that, in problems related to real data in graphs, there are better results when using more informative features taken from the same subset of data than taking data from a bigger *Ego Network*.

When using categorical methods, predicting with features extracted from the set Cat_1 resulted in better values for almost every metric than using features extracted from the set Cat_2. As in the previous point about comparing Nbr_1 and Nbr_2, getting features about the links of users further away than 2 degrees adds more noise than useful information to the dataset. Indeed, this is the reason why the experimentation stopped at this degree instead of going one level further in the *Ego Network*.

Model	Level	Acc.	Prec.	Rec.	AUC	F_1	F_4	t_{fit}	t_{pred}
Random		0.499	0.499	0.500	0.499	0.500	0.500	—	0.005 s
Majority		0.565	0.747	0.197	0.565	0.312	0.206	—	0.204 s
LR	Nbr_0	0.534	0.586	0.234	0.534	0.335	0.243	0.937 s	0.016 s
	Nbr_1	0.547	0.617	0.250	0.547	0.356	0.260	1.347 s	0.035 s
	Nbr_2	0.563	0.586	0.430	0.563	0.496	0.437	1.055 s	0.023 s
	Cat_0	0.565	0.746	0.198	0.565	0.313	0.207	1.871 s	0.041 s
	Cat_1	0.577	0.727	0.247	0.577	0.368	0.257	9.816 s	0.077 s
	Cat_2	0.589	0.636	0.415	0.589	0.503	0.424	9.456 s	0.065 s
RF	Nbr_0	0.543	0.544	0.529	0.543	0.536	0.530	25.789 s	4.878 s
	Nbr_1	0.578	0.585	0.537	0.578	0.560	0.540	102.961 s	5.608 s
	Nbr_2	0.583	0.590	0.541	0.583	0.564	0.543	70.447 s	3.148 s
	Cat_0	0.568	0.573	0.536	0.568	0.554	0.538	32.981 s	5.371 s
	Cat_1	0.613	0.634	0.533	0.613	0.579	0.538	44.911 s	6.002 s
	Cat_2	**0.614**	0.635	0.534	**0.614**	**0.580**	**0.539**	50.589 s	3.484 s

Table 7.4: Resulting metrics of different methods used in Chapter 7 tested in the *Outer Graph* Ω, which includes all nodes. **Bolded** items represent the highest value for each metric.

The most remarkable part of these results is that, even in the best case scenario, **the results for any of the methods based in *Machine Learning* presented in Chapter 7 are worse than the ones from the *Bayesian Algorithm* presented in Chapter 5**. This is remarkable since the latter method uses a much lower amount of features per node (2 vs 48 in the case of Cat_1) and takes a shorter amount of time predicting data than the best of the former methods takes to train a classifier on it.

Further conclusions related to this last point are discussed in Chapter 8.

7.6.2 Outer Graph

Table 7.4 shows various metrics which result from applying the metrics described in this section to the datasets presented in this thesis as the *Outer Graph* Ω, which contains the entire *Social Graph* G, including the majority of nodes for which no socioeconomic data is known about their neighbours. This makes it impossible to run the *Bayesian Algorithm* presented in Chapter 5, as it uses this data directly as features.

The relative results between different *Machine Learning* and feature extraction

methods are similar to the ones presented for the *Inner Graph* presented in Section 7.6.1. However, since there is a significant amount of new users without much significant information the absolute results are considerably worse.

Like in Section 7.6.1, adding an single level to the size of the original *Ego Network* improved the values of all metrics, since more useful data was used to generate the features. This is truth both when using only general general features (which are present for all users in Ω) as seen in the difference between Nbr_0 and Nbr_1, and when using socioeconomic data in the features (where most users have all metrics equal to 0, as there is no information) as in between Cat_0 and Cat_1.

Interestingly, unlike the results in Table 7.3, all metrics keep improving when making the *Ego Network* another level deeper (Nbr_2 and Cat_2). The reason for this is a simple matter of biases between the *Inner Graph* and the *Outer Graph*: users present in the latter graph ($\Omega \setminus \Upsilon$) tend to have less contacts in general than the rest. This makes the first few levels of the *Ego Network* contain less useful data to use as *Machine Learning* features, while the relatively noisy features generated from the latter levels actually improves these.

Like in the previous table, *Machine Learning* methods that use *Random Forest* have a considerable advantage over all metrics compared to the ones using *Logistic Regression*, using features related to the *Socioeconomic Level* of a users' neighbour improves the results. However, due to the amount of users without any significative data about their neighbours, the difference is small compared to the one in Table 7.3.

Chapter 8

Conclusions

8.1 General Objectives

This thesis presented several methods of manipulating communication and socioeconomic data to produce useful insights about its users. The combination of different datasets present an unique opportunity to study group behaviour and to experiment with gathering data that isn't directly present as an input.

The first data source used in this study is the multiset of mobile phone *Call Detail Records* P (for phone calls) and S (for SMS), which is presented in Section 4.1.1. These datasets contain all the communication done by the users of a certain telco, and having them allowed us to create a *Social Graph* which allows us to find many insights about its members.

The second data source is the set with *Bank Information* about the users B, which is presented in Section 4.2. While the values in this set aren't perfectly correlated to the socioeconomic index of the users, its data is a good enough proxy for the object of this study. For this reason, separated the users into 2 distinct groups depending on which side of the median of \$6300* their monthly salary fell into.

- $\mathbf{H_1}$ the set of *Low Income* users, with an income less or equal to \$6300* a month.
- $\mathbf{H_2}$ the set of *High Income* users, with income greater than \$6300* a month.

Getting the intersection of both datasets it is possible to create the *Social Graph* G, with the same users of the set of *Call Detail Records*, but where a subset of users also contain banking information from the set B, as shown in Section 4.3. This set, after being cleaned of outliers in Section 4.4, is the main input of the methods used in the thesis.

*As explained in Chapter 4, the symbol \$ refers to a certain currency and is not necessarily the American dollar.

8.2 Similar Studies

This thesis is not the first socioeconomic study done in this dataset.

Section 3.1 presents a summary of the work done by Léo, Karsai, et al. to correlate different consumption patterns with the socioeconomic level of these users. In it, the authors present a model of *Homophily* of calls between users of the same socioeconomic level, and proves that many different properties of the *Social Graph* follow these correlations.

Section 3.2 does a study similar to the previous one, where the author finds a correlation between users in the same socioeconomic level in social network patterns, mainly in the diversity of users in their links. The study also presents several ways of preventing bias in these comparisons.

8.3 The Bayesian Algorithm

Chapter 5, the most important part of this thesis, presented the general method used for predicting the socioeconomic level of the users given information about the neighbours. The model uses a Bayesian approach to deal with the uncertainty that comes from not having complete information about the users in the dataset by defining, for every user in the dataset, a *Beta Distribution* with parameters equal to the accumulated information of the edges leaving to high and low income neighbours.

The testing for this method is done in Chapter 6, separating the data into two distinct training and testing sets and ignoring some users in order to get an income distribution that's closer to the one in real life and not being subject to the bias that wealthier users tend to have more contacts. We refer to this graph as the *Inner Graph*, annotated with the Greek letter Υ.

We use two hyperparameters to define which model we use. The first one is ϖ, originally presented in Section 5.2.4, which shows which property of the users is being accumulated and used as the parameters of the *Beta Distribution*.

$$\varpi \in \{\text{calls}, \text{time}, \text{sms}, \text{contacts}\} \tag{8.3.1}$$

The second hyperparameter is Θ, presented in Section 5.2.5, which is used for defining which quantile of the cumulative probability function is used to decide the

Accuracy	Precision	Recall	AUC	F_1	F_4	time
0.693	0.665	0.792	0.746	0.723	0.783	33.155 s

Table 8.1: Resulting metrics of applying the *Bayesian Algorithm* to the *Inner Graph*

probability of each user to belong to a socioeconomic category. This value is important since it defines the way in which the model's uncertainty plays a role in the prediction. Since each user has a certain *Beta Distribution* depending on the properties coming from high and low income users, instead of a single value, the distribution for a user with many neighbours with known socioeconomic index will different from one with fewer, even if they have the same ratio of high to low income contacts.

Following experimentation in Sections 6.3 and 6.4, the values for these hyperparameters that maximized *Area Under the Curve* were found. There are presented in Equation (8.3.2).

$$\begin{aligned} \varpi &= \text{contacts} \\ \Theta &= 0.394 \end{aligned} \tag{8.3.2}$$

This method has a third hyperparameter, τ, whose value doesn't affect the *Area Under the Curve* but defines at which probability of being of high income a user is inferred to be of that category. This hyperparameter was optimized to maximize *Accuracy*, and in Section 6.4.4 the optimal value, shown in Equation (8.3.3) was found.

$$\tau = 0.514 \tag{8.3.3}$$

The resulting metrics for this approach are shown in Table 8.1.

As shown in Chapter 7, these metrics are extremely good, specially for a model that only takes 2 parameters per user and is faster than several more conventional ones.

8.4 Comparison with different algorithms

The result of applying this data to the *Bayesian Algorithm* was compared with different, more conventional methods is presented in Chapter 7.

These additional methods can be grouped into 3 groups.

1. **Trivial Algorithms**, used for having a base of comparison.
    - Random Selection, which chooses a category randomly.
    - Majority Voting, which chooses the category for which the majority of the users' neighbours belong.
2. **Ego Network** based algorithms, which use accumulated data about the edges of a users' *Ego Network* of a certain level.
    - $\text{Nbr}_{0,\dots,2}$, where the subscript denotes the size of the *Ego Network* used.
3. **Categorical Ego Network** based algorithms, which besides of using the total values in the previous point it also does separate accumulations of each property of the edges in the *Ego Network* depending on the socioeconomic group of the node at the other end.
    - $\text{Cat}_{0,\dots,2}$, where the subscript denotes the size of the *Ego Network* used.

The algorithms based in the *Ego Network* use many more features per user than the *Bayesian Algorithm*, as shown in Table 7.1 and compared to the 2 features user in the latter. Later, these features are fed to a *Machine Learning* method, either *Random Forest* or a *Logistic Regression*, doing an *Grid Search* with *K-folds* using $K = 5$.

The result of these experiments, and its comparison with the *Bayesian Algorithm*, are shown in Table 7.3. There are several conclusions that can be found from these results.

- In problems with highly nonlinear input data, like this thesis, methods based in *Random Forest* perform better than methods based in *Logistic Regression* since the feature space can be divided into more complex patterns.
- Socioeconomic data on the edges, like the amount of wealthy people someone calls, is a much better predictor of socioeconomic level than more general metrics, like the total amount of people called, even when there is less data present of this kind.
- Adding more levels to the *Ego Network* of each user that's being used for engineering features adds useful features to the prediction and can make the final metrics better.

- However, the precious point also adds an amount of noise. By the time the features contain data about the edges adjacent to the neighbours of the neighbours of the user, the metrics decreased.

- While this doesn't always follow, the *Area Under the Curve* is a good measurement of the results of each algorithm, since it mostly raises at the same rate as other metrics.

Additionally, a similar experimentation has been done in the *Outer Graph* Ω, whose training and testing data contains all nodes in the graph, including those without any neighbour with known socioeconomic data. Its results can be seem in Table 7.4.

These results are understandably worse than in the previous table. As an interesting result, the lower levels of the *Machine Learning* algorithms now contain less useful information and more noise. For this reason, the signal-to-noise ratio of the featuresets from a bigger *Ego Network* is higher, and unlike in the previous case the *Area Under the Curve* of both Cat_2 and Nbr_2 are higher than that value for Cat_1 and Nbr_1.

8.5 Comparison between the Bayesian Algorithm and the methods based in Machine Learning

> "Plurality is not to be posited without necessity"
>
> *Occam's Razor*

The *Bayesian Algorithm* has a better performance than every single algorithm based in common *Machine Learning* algorithms and novel feature extraction methods. This is remarkable since it uses only 2 features per node, compared to the 48 used by Cat_1, the version with the highest resulting metrics, or the rest of the methods whose amount of features are shown in Table 7.1.

A good way to explain this result is using *Occam's Razor* that explains that, among competing hypotheses, the one with the fewest assumptions should be selected.

In the *Machine Learning* methods we use the hypothesis that the features are correlated with the results, and that all of them have a high signal-to-noise ratio. While this is true in a general sense, as Section 5.1 and Figure 5.1 show, it is not a

perfect correlation and not even using a highly nonlinear method like *Random Forest* could improve on a model with less assumptions.

However, for the *Bayesian Method*, after setting the hyperparameters ϖ, Θ, and τ, our only hypotheses is that the more calls, SMS, calling time, or contacts a user has with high income, the higher is the *Certainty* that this user is of this category, and vice-versa.

This is a good general hypothesis, and, as shown in this thesis, strong enough to handle a very good socioeconomic level predictor.

Symbol Glossary

This thesis uses a large amount of symbols for different properties and data. To simplify its comprehension, the following table contains information about some of the most commonly used ones.

Symbol	Definition	Section
P	Multiset of all voice calls in the graph.	4.1.1
S	Multiset of all SMS in the graph.	4.1.1
V	Set of users in the telco network.	4.1.1
E	Set of edges between users in the telco network, along with their communication properties.	4.1.1
G	Social graph: a tuple containing both V and E.	4.1.1
B	Set with banking data for all users.	4.2
$H_{\{1,2\}}$	Set of lower and upper income users, respectively.	5.2.1
ϖ	Selected property of users for the Bayesian distribution.	5.2.4
$\mathcal{B}_v$	*Beta distribution* defined for a user v to predict its wealth.	5.2.5
Θ	Quantile used to define the *Posterior Probability* of a user's category.	5.2.5
τ	Limit for the *Posterior Probability* of high and low income users.	5.2.6
B_{train}	*Training Set* used for B.	6.2.1
B_{test}	*Testing Set* used for B.	6.2.1
$\hat{B}_{\text{test}}$	Contacts in the *Testing Set* with some contact with socioeconomic data.	6.2.2
Υ	The *Inner Graph*. Subset of $\hat{B}_{\text{test}}$ with the same amount of low and high income users.	6.2.3
Ω	The *Outer Graph*. Same as B_{test}.	6.2.4
Nbr_n	Accumulated properties of the user in the *Ego Network of Order* n of the users.	7.3
Cat_n	Union of the properties of Nbr_n with the same properties partitioned by income group.	7.3

Bibliography

[Art64] Emil Artin. *The Gamma Function*, pages 18–19. Hamburg University, 1964.

[Ban16] World Bank. World bank open data, 2016. [Online, accessed 14-July-2016]. URL: `http://data.worldbank.org/`.

[Bay63] Thomas Bayes. An essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, f. r. s. communicated by mr. price, in a letter to john canton, a. m. f. r. s. *Philosophical Transactions of the Royal Society of London*, 53(0):370–418, jan 1763. `doi:10.1098/rstl.1763.0053`.

[BBMS14] Jorge Brea, Javier Burroni, Minnoni Martin, and Carlos Sarraute. Harnessing mobile phone social network topology to infer users demographic attributes. In *ACM SIGKDD*. ACM, 2014.

[BE10] Joshua Blumenstock and Nathan Eagle. Mobile divides: gender, socioeconomic status, and mobile phone use in Rwanda. In *Proceedings of the 4th ACM/IEEE International Conference on Information and Communication Technologies and Development*, page 6. ACM, 2010.

[BN95] David J. Balding and Richard A. Nichols. A method for quantifying differentiation between populations at multi-allelic loci and its implications for investigating identity and paternity. *Genetica*, 96(1):3–12, 1995. `doi:10.1007/BF01441146`.

[Bre93] Leo Breiman. *Classification and regression trees*. Chapman & Hall, New York, 1993.

[Bre96] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[Bre01] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[Bul77] M. Bulmer. *Sociological Research Methods*. Macmillan, 1977.

[Dea97] Angus Deaton. The analysis of household surveys: a microeconometric approach to development policy. *World Bank*, 1997.

[Die00] Thomas G Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning*, 40(2):139–157, 2000.

[DZ02] Angus Deaton and Salman Zaidi. Guidelines for constructing consumption aggregates for welfare analysis. *The World Bank*, 2002.

[ea11] Johan Ugander et al. The anatomy of the facebook social graph. 2011.

[Faw05] Tom Fawcett. An introduction to roc analysis. *Data Mining: Contepts and Techniques*, 2005. doi:10.1016/j.patrec.2005.10.010.

[FBB+16] Martin Fixman, Ariel Berenstein, Jorge Brea, Martin Minnoni, Matias Travizano, and Carlos Sarraute. A Bayesian approach to income inference in a communication network. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 579–582. IEEE, Aug 2016. doi:10.1109/asonam.2016.7752294.

[FCH+08] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: a library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[FM07] Alexander Fraser and Daniel Marcu. Measuring word alignment quality for statistical machine translation. *Comput. Linguist.*, 33(3):293–303, September 2007. doi:10.1162/coli.2007.33.3.293.

[Fre09] David A. Freedman. *Statistical Models: Theory and Practice*. Cambridge University Press, 2009.

[Fri06] Michael Friendly. A brief history of data visualization. In C. Chen, W. Härdle, and A Unwin, editors, *Handbook of Computational Statistics: Data Visualization*, volume III. Springer-Verlag, Heidelberg, 2006. (In press).

[FS+96] Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. In *Icml*, volume 96, pages 148–156. Bari, Italy, 1996.

[GCSR03] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis, Second Edition (Chapman & Hall/CRC Texts in Statistical Science)*. Chapman and Hall/CRC, 2003.

[GHB08] Marta C Gonzalez, Cesar A Hidalgo, and Albert-Laszlo Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, 2008.

[Gre11] William H. Greene. *Econometric Analysis*. Prentice Hall, 7 edition, February 2011.

[HAK+02] Jiawei Han, Russ B Altman, Vipin Kumar, Heikki Mannila, and Daryl Pregibon. Emerging scientific applications in data mining. *Communications of the ACM*, 45(8):54–58, 2002.

[HHS94] Tin Kam Ho, Jonathan J. Hull, and Sargur N. Srihari. Decision combination in multiple classifier systems. *IEEE transactions on pattern analysis and machine intelligence*, 16(1):66–75, 1994.

[Ho95] Tin Kam Ho. Random decision forests. In *Document analysis and recognition, 1995., proceedings of the third international conference on*, volume 1, pages 278–282. IEEE, 1995.

[HTF03] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction (Springer Series in Statistics)*. Springer, 2003.

[Huu03] A.A. Huurdeman. *The Worldwide History of Telecommunications*. A Wiley-interscience publication. Wiley, 2003.

[HZL] Chih-Yang Hsia, Ya Zhu, and Chih-Jen Lin. A study on trust region update rules in newton methods for large-scale linear classification. Technical report, Technical report, Department of Computer Science and Information Engineering, National Taiwan University, 2017.

[Jef46] Harold Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 186(1007):453–461, 1946. doi:10.1098/rspa.1946.0056.

[Jen99] David D Jensen. Statistical challenges to inductive inference in linked data. In *AISTATS*, 1999.

[KC01] Elizabeth G. Katz and Maria Cecilia Correia. *The economics of gender in Mexico: Work, family, state, and market*. Africa Region Human Developments. World Bank Publications, 2001.

[Kip13] David M. Kipping. Parametrizing the exoplanet eccentricity distribution with the beta distribution. *Monthly Notices of the Royal Astronomical Society: Letters*, 434(1):L51, 2013. doi:10.1093/mnrasl/slt075.

[Kol56] A. N. Kolmogrov. *Foundations of the Theory of Probability*. Martino Fine Books, 1956.

[LFAH+16] Yannick Leo, Eric Fleury, J. Ignacio Alvarez-Hamelin, Carlos Sarraute, and Márton Karsai. Socioeconomic correlations and stratification in social-communication networks. *Journal of The Royal Society Interface*, 13(125), 2016. doi:10.1098/rsif.2016.0598.

[LG03] Qing Lu and Lise Getoor. Link-based classification. In *ICML*, volume 3, pages 496–503, 2003.

[LKSF16] Yannick Leo, Marton Karsai, Carlos Sarraute, and Eric Fleury. Correlations of consumption patterns in social-economic networks. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 493–500. IEEE, Aug 2016. doi:10.1109/asonam.2016.7752280.

[LMS+17] Shaojun Luo, Flaviano Morone, Carlos Sarraute, Matias Travizano, and Hernan A. Makse. Inferring personal economic status from social network location. *Nature Communications*, 8(15227), May 2017. doi:10.1038/ncomms15227.

[Loh11] Wei-Yin Loh. Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):14–23, 2011.

[LW14] Michael D. Lee and Eric-Jan Wagenmakers. *Bayesian Cognitive Modeling: A Practical Course*. Cambridge University Press, 4 2014.

[Mac03] David MacKay. *Information Theory, Inference, and Learning Algorithms*. 2003.

[Man82] Benoit Mandelbrot. *The fractal geometry of nature*. W.H. Freeman, San Francisco, 1982.

[Mar87] Peter V. Marsden. Core discussion networks of americans. *American Sociological Review*, 52(1):122–131, 1987.

[Mar88] Peter V. Marsden. Homogeneity in confiding relations. *Social Networks*, 10(1):57–76, Mar 1988. doi:10.1016/0378-8733(88)90010-x.

[MBK79] K. V. Mardia, J. M. Bibby, and J. T. Kent. *Multivariate Analysis*. Academic Press, London, UK, 1979.

[McC80] Peter McCullagh. Regression models for ordinal data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 42(2):109–142, 1980.

[Min44] Charles Joseph Minard. *Tableaux graphiques et cartes figuratives*. Autogr. Regnier et Dourdet, Bibliothèque numérique patrimoniale des ponts et chaussées, 1844.

[Mit97] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

[MM15] Flaviano Morone and Hernán A Makse. Influence maximization in complex networks through optimal percolation. *Nature*, 524(7563):65–68, 2015.

[Mos10] Stephen L. Moshier. Cephes mathematical library, 2010. doi:10.5281/zenodo.8475.

[MR08] Oded Z Maimon and Lior Rokach. *Data Mining with Decision Trees: Theory and Applications (Series in Machine Perception and Artifical Intelligence)*. World Scientific Publishing Company, 2008.

[MSHK16] David Muchlinski, David Siroky, Jingrui He, and Matthew Kocher. Comparing random forest with logistic regression for predicting class-imbalanced civil war onset data. *Political Analysis*, 24(1):87–103, 2016. doi:10.1093/pan/mpv024.

[MSLC01] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, pages 415–444, 2001.

[Mye03] Jarome L. Myers. *Research Design and Statistical Analysis*. 2003.

[NP14] Dan Navarro and Amy Perfors. An introduction to the beta-binomial model. 2014.

[ÓBV+16] María Óskarsdóttir, Cristián Bravo, Wouter Verbeke, Carlos Sarraute, Bart Baesens, and Jan Vanthienen. A comparative study of social network classifiers for predicting churn in the telecommunication industry. In *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on*, pages 1151–1158. IEEE, 2016.

[ÓBV+17] María Óskarsdóttir, Cristian Bravo, Wouter Verbeke, Carlos Sarraute, Bart Baesens, and Jan Vanthienen. Social network analytics for churn prediction in telco: Model building, evaluation and network architecture. *Expert Systems with Applications*, 2017.

[PFL54] Robert K. Merton Paul F. Lazarsfeld. Friendship as a social process: A substantive and methodological analysis. 1954.

[Pow07] David M W Powers. Evaluation: From precision, recall and f-factor to roc, informedness, markedness and correlation. Technical report, School of Informatics and Engineering, Flinders University of South Australia, 2007.

[Pow15] David M. W. Powers. What the f-measure doesn't measure: Features, flaws, fallacies and fixes. *CoRR*, abs/1503.06410, 2015.

[PSS13] Nicolas Ponieman, Alejo Salles, and Carlos Sarraute. Human mobility and predictability enriched by social phenomena information. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 1331–1336. ACM, 2013.

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[Sac] Lalit Sachan. Logistic regression vs decision trees vs svm. http://www.edvancer.in/logistic-regression-vs-decision-trees-vs-svm-part2/.

[Sat87] S. E. Satchell. Source and subgroup decomposition inequalities for the lorenz curve. *International Economic Review*, 28(2):323–329, 1987. doi:10.2307/2526727.

[SBB14] Carlos Sarraute, Pablo Blanc, and Javier Burroni. A study of age and gender seen through mobile phone usage patterns in Mexico. In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 836–843. IEEE, 2014.

[Sch96] Mark J. Schervish. *Theory of Statistics*. Springer Series in Statistics, 1996.

[SFLP13] Vivek K Singh, Laura Freeman, Bruno Lepri, and Alex Sandy Pentland. Predicting spending behavior using socio-mobile features. In *Social Computing (SocialCom), 2013 International Conference on*, pages 174–179. IEEE, 2013.

[SLPA15] Carlos Sarraute, Carolina Lang, Nicolas B Ponieman, and Sebastian Anapolsky. The city pulse of Buenos Aires. In *Workshop Big Data & Environment*, 2015.

[Sno55] John Snow. *On the Mode of Communication of Cholera*. John Churchill, The Bavarian State Library, 1855.

[SOWZ99] M.Yusof Sulaiman, W.M Hlaing Oo, Mahdi Abd Wahab, and Azmi Zakaria. Application of beta distribution model to malaysian sunshine data. *Renewable Energy*, 18(4):573 – 579, 1999. doi:10.1016/S0960-1481(99)00002-6.

[TMVS16] Jo-Anne Ting, Franziska Meier, Sethu Vijayakumar, and Stefan Schaal. *Locally Weighted Regression for Control*, pages 1–14. Springer US, Boston, MA, 2016.

[UKBM11] Johan Ugander, Brian Karrer, Lars Backstrom, and Cameron Marlow. The anatomy of the Facebook social graph. *Structure*, 5:6, 2011.

[WA78] Albert R Wildt and Olli T Ahtola. Analysis of covariance. quantitative applications in the social sciences series# 12, 1978.

[WHP89] James A. Wiley, Stephen J. Herschkorn, and Nancy S. Padian. Heterogeneity in the probability of hiv transmission per sexual contact: The case of male-to-female transmission in penile—vaginal intercourse. *Statistics in Medicine*, 8(1):93–102, 1989. doi:10.1002/sim.4780080110.

[Yan09] X. Yan. *Linear Regression Analysis: Theory and Computing*. World Scientific Publishing Company Pte Limited, 2009.