UNIVERSIDADE FEDERAL DE MINAS GERAIS
Instituto de Ciências Exatas
Programa de Pós-Graduação em Ciência da Computação

Bruno Demattos Nogueira

On the relation of privacy and fairness through the lenses of quantitative information flow

Belo Horizonte
2023

Bruno Demattos Nogueira

On the relation of privacy and fairness through the lenses of quantitative information flow

Final Version

Thesis presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Mário Sérgio Ferreira Alvim Júnior

Belo Horizonte
2023

Nogueira, Bruno Demattos.

N778o On the relation of privacy and fairness through the lenses of quantitative information flow [recurso eletrônico] / Bruno Demattos Nogueira - 2023.

1 recurso online (111 f. il., color.) : pdf.

Orientador: Mário Sérgio Ferreira Alvim Júnior
Dissertação (Mestrado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciências da Computação.
Referências: f. 91-96

1. Computação – Teses. 2. Aprendizado do computador – Teses. 3. Teoria da informação - Teses. 4. Direito à Privacidade - Teses. 5. Proteção de dados – Teses. I. Alvim Júnior, Mário Sérgio Ferreira. II. Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Computação. III.Título.

CDU 519.6*82(043)

Ficha catalográfica elaborada pela bibliotecária Irénquer Vismeg Lucas Cruz
CRB 6/819 - Universidade Federal de Minas Gerais - ICEx

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

On the relation of privacy and fairness through the lenses of quantitative information flow

BRUNO DEMATTOS NOGUEIRA

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores: (a)

Prof. Mário Sérgio Ferreira Alvim Júnior - Orientador
Departamento de Ciência da Computação - UFMG

Profa. Natasha Fernandes
School of Computing - Macquarie University

Profa. Catuscia Palamidessi
Informatics Laboratory of l'École Polytechnique - INRIA and LIX, École Polytechnique

Belo Horizonte, 30 de novembro de 2023.

Resumo

Ao desenvolver um sistema de aprendizado de máquina, existem duas preocupações além do desempenho do algoritmo. O primeiro é se o sistema é justo, isto é, se ele trata indivíduos de grupos distintos da mesma maneira, os classificando de forma similar. O segundo é se o sistema é privado, isto é, se ele não revela informações privadas de indivíduos que fazem parte do conjunto de treino quando a saída é exibida a um observador. Inicialmente, essas duas preocupações foram consideradas independentemente, mas recentemente, a conexão entre os dois tem atraído cada vez mais atenção na comunidade de aprendizado de máquina. Nesse trabalho, nós exibiremos uma expansão do arcabouço do fluxo de informação quantitativo para descrever de maneira completa todas as situações que podem ocorrer em termos de privacidade e justiça. Além disso, modelaremos essas duas quantidades como duais. Depois, modelaremos quatro métricas de justiça já existentes usando nosso arcabouço. Por fim, descreveremos experimentos que mostram como nosso modelo se comporta em cenários com dados reais, o testando com diferentes bases de dados e algoritmos.

Palavras-chave: Teoria da Informação, Aprendizado de Máquina, Justiça, Privacidade

Abstract

When developing a machine learning (ML) system, there are two common concerns besides the algorithm's performance. The first one is whether the system is fair, that is, if it treats individuals from different groups similarly, giving them similar classifications. The second is whether the system is private, that is, if it does not reveal private information about individuals on the training set when the output is shown to an observer. Initially, they were considered separately, but recently, the connection between these two concerns has gathered increased attention in the ML community. In this work, we will show an expansion of the quantitative information flow framework to fully describe which situations can happen in terms of fairness and privacy and model them as duals. After that, we model four different existing fairness notions using our framework. Ultimately, we describe experiments showing how our model behaves in real-world scenarios, testing it with different datasets and ML algorithms.

Keywords: Information Theory, Machine Learning, Fairness, Privacy

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Machine learning (ML) classifiers can be used to make decisions about an individual. Banks, for example, use these classifiers to predict whether a person will default if they receive a loan. These classifiers are not explicitly programmed. They are created by algorithms that receive data from previous observations and detect patterns that are then used to classify future individuals. Generally, the more data used to train the algorithms, the better the predictions. This fact incentivizes data scientists to use all their data, including possibly sensitive features like gender, race, and medical conditions.

In some scenarios, the use of those characteristics causes two problems. The first one is fairness. If a minority group often receives worse classifications than a majority, the algorithm is said to be unfair. The second issue is privacy. If an adversary can observe the prediction given to an individual and infer the value of those sensitive features, the algorithm has a privacy problem.

For years, academia has been focused on these concepts as separate phenomena. The study of fairness has focused on how to measure it, build algorithms that avoid it, and change the data to mitigate biases. Simultaneously, most of the study in privacy has been on building algorithms that ensure it and on transforming the training data to protect the identity of members. However, privacy and fairness have a clear relationship. If the classifier is unfair, then knowing the value of the sensitive attribute gives information about the classification. If it is not private, the predicted class influences the sensitive feature. Recently, there has been some development in the relationship between the two concepts. Some studies have shown the link between statistical notions of fairness and a measure of privacy called differential privacy, but there is still work to be done.

One tool that has yet to be thoroughly explored for this use is the quantitative information flow (QIF) framework. It was initially designed to measure how information flows through a computer system when executed, mainly in applications concerning security. However, it is possible to model both fairness and privacy using it.

The main goal of this master's thesis is to characterize the relationship between privacy and fairness completely. To do this, we need to formally define an expansion of the QIF framework, model existing notions of fairness with QIF, create a new pure QIF model to capture fairness, and perform experiments to show how they perform on real-world data.

1.1 Contributions

There are five main contributions in this work.

- A new notation for an existing expansion of the QIF framework to deal with information flow in two directions.
- A complete characterization of all combinations of flow that can happen concerning average and maximum Bayes flow and capacity, in both the additive and multiplicative cases.
- A modeling of existing fairness metrics using QIF, specifically statistical parity, equal opportunity, equalized odds, and conditional statistical parity.
- Expansion of a QIF model that simultaneously captures fairness and privacy in ML.
- Experiments that show the feasibility of all these points in real-world data. They have to confirm that the measures are non-trivial and how different factors (dataset, algorithm, sensitive feature) change the information flows.

1.2 Outline of the thesis

We begin with a background covering QIF, ML, and fairness in Chapter 2. After that, Chapter 3 has a brief literature review summarizes the related research. Then, Chapter 4 reformulates an expansion of the QIF framework to include two flow directions and a model where these directions represent the privacy and fairness of an ML classifier.

The main contributions follow these parts. We show all theoretical bounds that govern the values of information flow that can happen simultaneously in several situations in Chapter 5. After that, Chapter 6 models several existing notions of fairness using QIF. Finally, experiments with four different ML algorithms and four different datasets show how our new metrics behave in Chapter 7.

Chapter 2

Background

This chapter provides the necessary background information for all the new concepts that will be introduced in this thesis. In the first section, we will describe the quantitative information flow framework. After that, we will briefly review what is a machine learning task, some algorithms that can perform them and some other related concepts. Finally, we study what fairness is in machine learning. We present four metrics and divide them into two groups.

2.1 Quantitative information flow

2.1.1 What is QIF

When dealing with security threats, it is not enough to know if some secret has been leaked or not. Consider, for example, someone trying to log into somebody else's account by guessing a password. Even if the snooper guesses incorrectly, they will discover that a possible password is incorrect, so information flows when the system outputs the message “incorrect password”. Therefore, there is a need to create a framework that measures information flow quantitatively.

Besides that, not all bits are equal [4]. Imagine a situation where a hacker is trying to find out the bank's password of several people. The password corresponding to Bill Gate's account is certainly worth more than the password of the author of this thesis, even though the number of bits of both passwords is roughly the same. Furthermore, there are occasions where the same system is run twice in different contexts, so there is a need to know how the security of such a system changes when the value of the secrets changes as well.

Quantitative Information Flow (QIF) is a framework that can handle both demands. It can measure precisely the amount of information that flows through a system, considering the value of such information.

It has been used to model different scenarios, like multi-party computation and flow of information due to side channels. Now, we will measure the privacy of sensitive machine learning systems and their fairness.

2.1.2 Main concepts surrounding QIF

Because QIF was created with a focus on security and privacy, the first thing we need to model are secrets. We suppose that an adversary will observe a system running and wants to know the value of a secret. A probability distribution models this secret to show the uncertainty of the adversary in their belief.

Definition 1 (Prior distribution)**.** *A* prior distribution π *is a distribution on a set* $\mathcal{X}$ *of possible values and represents the adversary's knowledge of the secret before the system is run.*

We will use $\mathbb{D}\mathcal{X}$ to denote the set of all distributions over the set $\mathcal{X}$.

With this knowledge, there are several decisions that the adversary can take. This is captured by gain functions that model the adversary's actions and possible rewards.

Definition 2 (Gain function)**.** *Given a set* $\mathcal{X}$ *of possible values for a secret and a set* $\mathcal{W}$ *of possible actions, the* gain function *is a function of type* $g : \mathcal{W} \times \mathcal{X} \to \mathbb{R}$*.* $g(w, x)$ *represents the gain of the adversary when he takes action* $w \in \mathcal{W}$ *and the secret is* $x \in \mathcal{X}$*.*

The set of all possible gain functions is $\mathbb{G}$ *and the set of all non-negative gain functions is* $\mathbb{G}^+$*.*

In a scenario with a given prior distribution and gain function, it is possible to compute the expected gain of the adversary. This is the g vulnerability of the prior with respect to the gain function. It measures how unsafe a situation is.

Definition 3 (Prior g-vulnerability)**.** *The* prior g-vulnerability *of a prior distribution* π *given a certain gain function* g *measures the expected gain of an optimal adversary. It is defined as*

$$V_g(\pi) = \max_{w \in \mathcal{W}} \sum_{x \in \mathcal{X}} \pi_x g(w, x).$$

One of the most straightforward gain functions is the identity gain function. It represents a scenario where the adversary wants to guess exactly the secret, but they only have one chance. The corresponding V_g is called Bayes vulnerability and measures the probability of an optimal adversary guessing the secret correctly.

Definition 4 (Bayes vulnerability). *Denoted by V_{id},* Bayes vulnerability *is the vulnerability with respect to the gain function where $\mathcal{W} = \mathcal{X}$ and $g(w, x) = 1$ if $w = x$ and 0 otherwise.*

From the definition, we have that

$$V_{id}(\pi) = \max_{x \in \mathcal{X}} \pi_x.$$

The channel abstraction is used to model a scenario where a computational system uses the secret as an input and returns an output.

Definition 5 (Channel). *Let $\mathcal{X}$ be a set of inputs (secrets) and $\mathcal{Y}$ a set of outputs. A* channel *of type $C : \mathcal{X} \to \mathbb{D}\mathcal{Y}$ receives an input from $\mathcal{X}$ and produces an output from $\mathcal{Y}$ according to a pre-defined distribution for each $\mathcal{X}$. $C_{x,y}$ is the probability of outputting y when the secret is x.*

When a channel is run with the input coming from a prior distribution, a joint is defined. The joint establishes the probability of each pair of input and output occurring.

Definition 6 (Joint). *In a scenario where a channel C receives a secret from a prior π as a distribution, a* joint *is defined. It is denoted by $\pi \triangleright C$. The value of an entry is*

$$(\pi \triangleright C)_{x,y} = \pi_x C_{x,y}.$$

Running a system using the input drawn from a prior π also defines a hyper-distribution, often called a hyper for brevity. A hyper is a distribution on distributions. That is, it defines the probability that each distribution can happen. The first probability is the probability of outputs, each of which defines a probability on the secrets.

Definition 7 (Hyper distribution). *A* hyper-distribution *Δ of a set $\mathcal{X}$ is a distribution on distributions of $\mathcal{X}$, i.e. it has type $\mathbb{D}\mathbb{D}\mathcal{X}$, abbreviated to $\mathbb{D}^2\mathcal{X}$.*

Each distribution in $\mathbb{D}\mathcal{X}$ is called an inner distribution*, and the distribution of the inners is the* outer distribution.

With the joint, it is possible to compute the adversary's expected gain after observing a channel's output. This is computed by taking the weighted average of the expected vulnerabilities with respect to all possible outputs.

Definition 8 (Posterior g-vulnerability). *The* posterior g-vulnerability *of a prior $\pi : \mathbb{D}\mathcal{X}$ and a channel $C : \mathcal{X} \to \mathbb{D}\mathcal{Y}$ with respect to a gain function $g : \mathcal{W} \times \mathcal{X} \to \mathbb{R}$ represents the* *expected gain of an adversary with knowledge of C after observing the output of a channel that received a secret from π. It is denoted*

$$V_g(\pi, C) = \sum_{y \in \mathcal{Y}} \max_{w \in \mathcal{W}} \sum_{x \in \mathcal{X}} \pi_x C_{x,y} g(w, x).$$

The posterior is the expected g-vulnerability after the output is observed, given that the adversary has complete knowledge of the channel. In some cases, the average case is not the one that matters the most because some extremely risky situations may only happen sometimes. Thus, another possible measure is the worst-case scenario. The scenario where the adversary is expected to gain the most is captured by the maximum posterior g vulnerability.

Definition 9 (Maximum posterior g-vulnerability)**.** *The* maximum posterior g-vulnerability *of a channel $C : \mathcal{X} \to \mathbb{D}\mathcal{Y}$ and a prior $\pi : \mathbb{D}\mathcal{X}$ with respect to a prior $g : \mathcal{W} \times \mathcal{X} \to \mathbb{R}$ represents the expected gain of the adversary in a worst-case scenario, that is, the situation where the output $y \in \mathcal{Y}$ reveals the most. It is written as*

$$V_g^{\max}(\pi, C) = \max_{y \in \mathcal{Y}} \max_{w \in \mathcal{W}} \sum_{x \in \mathcal{X}} \frac{\pi_x C_{x,y} g(w, x)}{\sum_{x \in \mathcal{X}} \pi_x C_{x,y}}$$

.

The information flow of a system measures how much information the adversary has gained after the system is executed. There are two possible ways to measure this: the ratio or the difference between the adversary's posterior and prior gain. We present both of them; the first is called multiplicative flow, and the second is additive flow.

Definition 10 (Multiplicative information flow)**.** *The* multiplicative information flow *of a channel C, a prior π with respect to a gain function g is the ratio of the posterior and the prior g-vulnerabilities. It is defined by*

$$\mathcal{L}_g^{\times}(\pi, C) = \frac{V_g(\pi, C)}{V_g(\pi)}.$$

Definition 11 (Additive information flow)**.** *The* additive information flow *of a channel C, a prior π with respect to a gain function g is the difference between the posterior and the prior g-vulnerabilities. Is it defined by*

$$\mathcal{L}_g^{+}(\pi, C) = V_g(\pi, C) - V_g(\pi).$$

In some situations, the g-function may change, and we are worried about what happens to the flow in this case. We can define capacity as how much the g-vulnerability can change when the g-function is altered, but the prior distribution and joint stay the same. There are other notions of capacity in the literature, but in this work, we will focus only on the capacity with respect to the gain function.

Just like with information flow, we can measure capacity in two ways: multiplicative and additive.

Definition 12 (Multiplicative capacity)**.** *The* multiplicative capacity *of a prior π and a channel C is*

$$\mathcal{ML}^{\times}_{\mathbb{G}^+}(\pi, C) = \max_{g \in \mathbb{G}^+} \frac{V_g(\pi, C)}{V(\pi)}.$$

Definition 13 (Additive capacity)**.** *The* additive capacity *of a prior π and a channel C is*

$$\mathcal{ML}^{+}_{\mathbb{G}^{\updownarrow}}(\pi, C) = \max_{g \in \mathbb{G}^{\updownarrow}} V_g(\pi, C) - V(\pi),$$

where $\mathbb{G}^{\updownarrow}$ is the set of all 1-bounded gain functions.

There are formulas that can compute both these quantities.

Theorem 1. *[3] The multiplicative capacity of a prior π and a channel C can be obtained by computing*

$$\mathcal{ML}^{\times}_{\mathbb{G}^+}(\pi, C) = \sum_{y \in \mathcal{Y}} \max_{x \in \lceil \pi \rceil} C_{x,y},$$

where $\lceil \pi \rceil$ denotes the support (the non-zero values) of π.

The additive capacity of a prior π and a channel C can be obtained by computing

$$\mathcal{ML}^{+}_{\mathbb{G}^{\updownarrow}}(\pi, C) = 1 - \sum_{y \in \mathcal{Y}} \min_{x \in \lceil \pi \rceil} C_{x,y}.$$

2.2 Machine learning

Some tasks, such as playing chess or deciding if an applicant is suitable for a given job, are very hard for a human to code a computer program that can deal with all possible scenarios.

To deal with these problems, machine learning (ML) was created. The idea is to create a general-purpose algorithm to learn from the data for a specific scenario and build a classifier. Formally, Mitchell et al. [50] defines an ML program as "A machine learning program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E".

This section will give the necessary background on machine learning. We begin by defining it. This is done by dividing ML into these three parts: the task T, the performance measure P, and the experience E. After that, we will present some algorithms that were used in the thesis as examples.

2.2.1 What is machine learning

2.2.1.1 The task

A task in machine learning corresponds to how a model should process an example. An example consists of a *feature vector* $\mathbf{x} \in \mathbb{R}^n$, where n is the number of dimensions, and each x_i represents the value of feature i. A feature is one characteristic of an example. It can be a categorical feature, representing that the individual belongs to a group, such as race. Or it can be a numerical feature, described by a number, such as height. The set of all possible feature vectors is $\mathcal{X}$.

In this thesis, we will only deal with classification problems. In this scenario, each example belongs to one of k possible classes $\{1, 2, \cdots, k\}$, and the goal is for the ML model to specify which class is the class of the example provided as input. The set of all possible classifications is $\mathcal{Y}$.

An example of a classification problem is determining what object is on an image. The feature vector $\mathbf{x} \in \mathbb{R}^n$ represents the brightness of each pixel on the image of size n, and the classes can represent different classifications, such as {cat, dog, car, ...}.

In this thesis, we will focus on binary classification tasks. In this scenario, there are only two possible classifications for each individual: positive and negative.

2.2.1.2 The performance measure

When training a model, it is necessary to have an objective and qualitative way to measure how good (or bad) the algorithm is to optimize it. This is the performance measure.

Some of the relevant measures are

- *Accuracy*: the probability that the model will determine correctly the class of an example [9].

- *Precision*: the probability of an example being positive given that the classification given by the model was positive.

- *False positive rate (recall)*: the probability of an example being classified as positive given that it is positive.

- *F-score*: the harmonic mean between precision and recall [59].

To get a precise estimate on any metrics, it is not recommended to use the data used to train the algorithm to estimate it. Instead, a *test set* is used to measure these quantities. It consists of a set of examples and labels with no intersection with the training data and is a significant sample of the population being considered [31].

2.2.1.3 The experience

The experience is usually the *dataset*. It consists of a set of examples. In a classification problem, each example comprises a vector feature and the target classification.

2.2.2 Specific algorithms

In the experiments chapter, we used four common ML algorithms. In this section, we explain how each of them works.

2.2.2.1 Naive Bayes

The *naive Bayes algorithm* is a classical algorithm in statistics and machine learning [33].

Consider that each example is drawn from a joint probability distribution on the set $\mathcal{X} \times \mathcal{Y}$. The goal of a classifier when it receives $\mathbf{x}$ as input is to choose one class $\hat{y} \in \mathcal{Y}$ such that

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} P(y|\mathbf{x}).$$

Using Bayes rule, we can write

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} \frac{P(y)P(\mathbf{x}|y)}{P(\mathbf{x})}.$$

Because the probabilities $P(\mathbf{x})$ are the same for every class $y \in \mathcal{Y}$, we do not need to consider it. The prediction then becomes

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} P(y)P(\mathbf{x}|y).$$

Now, we need to find a way to compute this term for every class.

The probability $P(y)$ is estimated as the fraction of examples in the dataset that belong to the class y. This is called the frequentist approach.

Estimating $P(\mathbf{x}|y)$ is more complicated. The first assumption we will make is that all the features are independent. This is why the algorithm is called naive. Thus, we can write

$$P(\mathbf{x}|y) = \prod_{i \in \{1,\cdots,n\}} P(x_i|y).$$

There are two ways to compute $P(x_i|y)$. If the i-th feature is discrete, then the probability is estimated as the fraction of entries with this feature as x_i and are on class y. If the i-th feature is continuous, then the probability is estimated as

$$P(x_i|y) = \frac{1}{\sigma_{i,y}\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x_i - \mu_{i,y}}{\sigma_{i,y}}\right)\right),$$

where $\mu_{i,y}$ is the mean of the i-th feature when the label is y in the training dataset, and $\sigma_{i,y}$ is the standard deviation. In other words, the probability is estimated as being a normal variable [55].

As we said at the beginning, the class that is predicted is the one that has the highest probability according to this model.

2.2.2.2 Logistic regression

We will only explain the *logistic regression* algorithm applied to the binary classification case.

The goal is to create a function that predicts the probability that an example $\mathbf{x}$ is on the positive class. The first step is to create a function $f : \mathcal{X} \to \mathbb{R}$ that will produce a real number for every example. It is an affine function of all parameters. It is written as

$$f(\mathbf{x}) = \beta_0 + \sum_{i \in \{1,\cdots,n\}} x_i \beta_i.$$

The second step is to apply this function to the sigmoid function to get

$$\hat{P}(+|x) = \frac{1}{1 + \exp(-f(x))},$$

and then we can interpret the result as being the probability that it is on the positive class, according to the model. The logistic function always produces a real number between 0 and 1, so using it as a probability estimate makes sense.

Training a logistic regression model is finding a good choice of β_i for all i. We refer to the vector of all β_i as $\boldsymbol{\beta}$. To do this, we first need a metric of how good is a choice of $\boldsymbol{\beta}$. We will use mean squared error as a loss function. It is defined as

$$\text{MSE} = \sum_{i \in \{1, \cdots, m\}} \left(I(i) - \hat{P}(\mathbf{x}_i) \right)^2 ,$$

where m is the size of the dataset, $\mathbf{x}_i$ is the feature vector of the i-th entry on the dataset and $I(i)$ is equal to one if the i-th entry is positive and zero otherwise, and $\hat{P}(\mathbf{x}_i)$ is the prediction for the i-th example.

Using a black-box optimizer, like gradient descent or Newton's method [31], it is possible to find good values for $\boldsymbol{\beta}$.

2.2.2.3 Random forest

To understand the random forest method, it is necessary to explain what is a decision tree.

Decision tree. A *decision tree* is a flowchart structure that outputs a classification. It is represented by a tree with one root. The decision process starts with the root as the current node. At every node, a test is made about one feature of the example being considered. With the test result, the current node decides which child becomes the current node. This process is repeated recursively in the sub-trees of the decision tree until a leaf becomes the current node. Each leaf has a constant output that can be the class the classifier predicts or the probability of the model being in each class.

Random forest. A *random forest* is a collection of decision trees. Each decision tree only has access to a subset of the features that is randomly selected [38]. In the training phase, each decision tree is trained separately. Each tree is trained separately when making a classification, and the predictions are aggregated later. If the predictions are in the form of a class, the random forest predicts the most voted class. If the predictions are probabilities, they are averaged to form a general prediction.

2.2.2.4 Gradient boosting

The idea of gradient boosting is to have simple predictors that we will denote by $h(\mathbf{x})$ that produce classifications. These predictors can assume different forms. For instance, it can predict that $\mathbf{x}$ belongs to the positive class if and only if $x_i \geq \alpha$ for a given i and α.

These predictors are going to be used in M steps. At every step, we begin with an incomplete classifier F_m. To improve F_m, the algorithm adds a new simple classifier such that $F_{m+1}(\mathbf{x}) = F_m(\mathbf{x}) + h_m(\mathbf{x})$.

The idea is to fit the new classifier $h_m(\mathbf{x})$ to the residual $y - F_m(\mathbf{x})$, the prediction error. By doing this multiple times, we can get a good classifier.

2.2.3 Observations

In this subsection, we will present some observations on notation and some other useful concepts that are not related to QIF or ML.

2.2.3.1 Nomenclature

In this work, the word *classifier* will refer to a program that gets a feature vector and outputs a classification. The word algorithm refers to a method where some data is used to produce a classifier. So, for instance, we can use the random forest algorithm to create a set of decision trees called a classifier. This classifier receives feature vectors as inputs and produces classifications.

The word *model* can refer to a machine learning algorithm in some works. Here, the word model will refer to scenarios where QIF will describe a machine learning system.

2.2.3.2 Pearson correlation

In the experiments, we are going to show that two variables are closely related. To measured this relation, we are going to use the Pearson correlation [55].

We first need the covariance between two random variables to define the Pearson correlation.

Definition 14 (Covariance)**.** *The* covariance *of two random variables X and Y is equal to the expected value of the product of their deviations from the mean. That is*

$$cov(X, Y) = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)],$$

where μ_A is the expected value of the random variable A.

The Pearson correlation or Pearson correlation coefficient measures the correlation between two random variables. It is equal to the covariance of both variables divided by the product of both standard deviations.

Definition 15 (Pearson correlation)**.** *The* Pearson correlation *of two random variables X and Y is equal to*

$$\rho_{XY} = \frac{cov(X, Y)}{\sigma_X \sigma_Y},$$

where σ_A is the standard deviation of the random variable A.

When dealing with samples, there is an approximation for the Pearson correlation, usually written as r_{XY}.

Theorem 2. *The Pearson correlation of a sample can be computed by*

$$r_{XY} = \frac{\sum_{i \in \{1,\ldots,n\}} (x_i - \bar{x})(y_i - \bar{Y})}{\sqrt{\sum_{i \in \{1,\ldots,n\}} (x_i - \hat{X})^2} \sqrt{\sum_{i \in \{1,\ldots,n\}} (y_i - \hat{Y})^2}},$$

where n is the number of samples of both variables, and $\hat{A}$ is the sample mean of the variable A.

2.3 Fairness in machine learning

In this section, we will enumerate four different fairness metrics and group them into two different types of metrics. However, first, we need to show some notation.

2.3.1 Fairness notation

In this work, we will only deal with binary classification problems, so there are only two possible classifications: positive and negative. We will also suppose that the favorable classification is the positive one. So, in a scenario where we want to predict if a loan should be accepted, the acceptance of such a loan is the positive class. In a situation where we predict if a person is going to commit a crime, we will predict if they are not going to commit crimes, such that the positive classification predicts that no crime will be committed. The positive classification is denoted by $+$, and the negative is $-$.

There are two classifications we may refer to. The first one is the correct one that is present in the dataset. This random variable is denoted by Y. The second one is the classification that is predicted by a classifier. $\hat{Y}$ denotes this random variable.

When we deal with fairness, there are usually two groups. The first is the unprotected group, sometimes called the majority. We will denote it using s_0 and refer to it as unprotected. The other group is the protected group; we denote it by s_1.

2.3.2 Fairness measures

2.3.2.1 Statistical parity

The first and most simple measure is statistical parity. It measures the difference of the probabilities of getting a positive classification for both groups.

Definition 16 (Statistical parity [22, 57, 68])**.** *The statistical parity α of a classifier is the absolute value of the difference between the probabilities of getting a positive classification for the unprotected and protected group:*

$$\alpha = |P(+|s_0) - P(+|s_1)|.$$

It is also called mean difference, equal acceptance rate, or benchmarking.

2.3.2.2 Equal opportunity

The second measure is a version of statistical parity where we only consider individuals with a positive classification. Ideally, we would know which individuals deserve a positive classification. In this thesis, we approximate it using the label from the dataset.

Definition 17 (Equal opportunity [34])**.** *The equal opportunity α of a classifier is the absolute value of the difference between the probabilities of getting a positive classification for the unprotected and protected groups given that the classification is positive in the dataset:*

$$\alpha = |P(\hat{Y} = +|Y = +, s_0) - P(\hat{Y} = +|Y = +, s_1)|.$$

2.3.2.3 Equalized odds

Equalized odds is an expansion of equal opportunity. The intuition behind it is that the probability of getting a correct classification must be independent of the group.

Definition 18 (Equalized odds [35, 7, 64])**.** *Equalized odds is satisfied if the probability of getting a correct classification is the same for both groups in the scenario of positive and negative classifications. That is:*

$$P(\hat{Y} = +|Y = i, s_0) = P(\hat{Y} = +|Y = i, s_1), \forall i \in \{+, -\}.$$

It is also called conditional procedure accuracy equality and disparate mistreatment.

2.3.2.4 Conditional statistical parity

The idea of conditional statistical parity is to have statistical parity given that some relevant features are equal.

Definition 19 (Conditional statistical parity [17])**.** *Given a set L of relevant features, conditional statistical parity is satisfied when the two groups have the same probability of* *getting a correct classification given that they have the same value of L. That is:*

$$P(+|L=\ell, s_0) = P(+|L=\ell, s_1), \forall \ell \in L.$$

2.3.3 Types of fairness measures

Now, we will distinguish these metrics into two groups. They were introduced in [29] and show two different world views.

2.3.3.1 We are all equal

The first notion of fairness is *we are all equal (WAE)*. The underlying principle is that all individuals have the same probability of satisfying specific criteria, regardless of their group.

Statistical parity falls under this group of fairness measures because it states that the probability of the protected and unprotected groups getting a positive classification must be the same.

2.3.3.2 What you see is what you get (WYSIWYG)

The second notion of fairness is *what you see is what you get (WYSIWYG)*. The idea is that individuals with similar characteristics must receive similar treatments. So, according to this notion, a classifier can be considered acceptable if it treats the protected and unprotected groups differently overall.

Equal opportunity is in this group because given that the dataset considers that two individuals should get a positive classification, then a classifier must give them a positive classification with the same probability. The same is true for equalized odds and conditional statistical parity.

Chapter 3

Literature review

This master's thesis will discuss how the quantitative information flow framework can be used to measure fairness in machine learning systems. Thus, it is necessary to present a literature review discussing fairness in a broader context and how it can be applied to computer science, followed by different definitions and measures. Afterwards, possible unfairness causes will be listed, and how some algorithms try to mitigate each. Finally, there will be a review of privacy in machine learning and how the QIF framework models different applications.

3.1 Fairness

This section will present related works in fairness. We begin by listing some notions of fairness outside of computer science. Then, we analyze different studies in computer science.

3.1.1 Fairness outside of Computer Science

The discussion about concepts such as justice and fairness has been present in Western Society as long as it exists. The book Theory of Justice by Plato is one of the first books on the subject and states that justice is the most important trait of a person [8]. More recently, political philosophers like John Rawls have stated that justice happens when all individuals have fair equality of opportunity [54].

These notions of fairness and justice have inevitably been used when writing legislation. The Universal Declaration of Human Rights, a document made by the United Nations to state the rights and freedoms of all human beings, declares that "Everyone is entitled to all the rights and freedoms set forth in this Declaration, without distinction of any kind, such as race, colour, sex, language, religion, political or other opinion, national or social origin, property, birth or other status" [5].

In Brazil, two relevant pieces of legislation on the subject are the Constitution and federal law 7.716/1989. The fifth article of the constitution guarantees that men and women have the same rights and duties and that no one can lose a right based on religion. Law 7.716/1989 establishes that no individual can suffer discrimination based on race, ethnicity, religion, or nationality.

These rules were all created at a time when only humans made decisions. However, in recent years, computers running machine learning algorithms have started to work in situations such as hiring individuals, granting loans, and university applications. So, the notions of fairness and justice have to be reassessed in a scenario where not only people make calls. This involves defining what justice is and how to measure it, as well as techniques that follow these definitions.

3.1.2 Fairness in Computer Science

In the last few years, Machine Learning algorithms have been used extensively in academia and industry. This resulted in more diverse applications, like predicting if a person applying for a loan would default. Because of the delicate nature of some applications, part of the attention from academics and general society turned to how fair those methods are, creating the area of fairness in machine learning [11], [12].

The field has several smaller divisions. The first works were in fairness on simple classification tasks. Naturally, it evolved to monitor fairness in increasingly complicated scenarios, e.g., fairness in Reinforcement Learning and how it affects the impact of each variable on a model [41], [45].

An important area tries to define what causes unfairness. There are a few different ways to make a machine learning model unfair. The first one arises when there is bias in data, and the model mimics the data. [11] showed that models trained using a sexist corpus also ended up being sexist. One other reason that can make an ML system unfair is that when training to minimize average error, it makes sense for the model to fit the majority. That leads to a different distribution of errors in the minority populations [16]. One last way to create a system that is not fair is that when joining different components that are all independently fair, their composition may be unfair [23]. A feedback loop can happen when the output of a classification system is used to enforce a policy in real life that can lead to data that is even more biased [47].

Academics also started creating ways to overcome fairness problems. [65] introduced fair representation learning, where the data is progressively transformed in a way that it stays relevant for the classification task, but the correlation with the sensitive variable is erased, making it harder for redlining (phenomena where the classifier learns to distinguish the protected and unprotected groups without this information being explicit in the input data) to happen. [25] showed a possible way to break a feedback loop in a machine learning pipeline, using different data sources to minimize the bias. There are works concerning fairness outside of classification problems, such as ranking [15] and reinforcement learning [21].

Another field of study is measuring unfairness. Individual notions of fairness try to guarantee that if two individuals have similar characteristics, they will probably receive the same treatment[22]. The problem with this approach is that it is extremely hard to create measures that can capture what similar characteristics mean, so they are not widely used [43].

A second technique to measure unfairness is to consider statistical properties. If all groups have similar rates of positive and negative classifications, then the algorithm is considered fair [14]. It is widely used, and many measures were created that follow this reasoning [49]. All of the metrics shown in the background fall under this category.

3.2 Privacy in Machine Learning

This thesis concerns the relationship between fairness and privacy in machine learning algorithms.

Thus, reviewing the main concepts of privacy in artificial intelligence is necessary. This section is divided into two parts: the threats of an ML system and how to deal with those threats.

3.2.1 Threats to privacy in Machine Learning systems

Attacks on ML systems can be partitioned into attacks on the party that holds the data and attacks on the party with the trained classifier. Both these attacks aim to discover the training set's characteristics, but they achieve this in different ways. Because this thesis is focused on the machine learning aspect of privacy, we will not explain how attacks on the party with the data happen. We will focus on attacks on the ML classifier.

There are two main groups of attacks: black-box attack and white-box.

In *black-box attacks* of a classifier, the attacker has access only to predictions of the ML classifier. Every time an input is given to the classifier, the attacker can view the input and the probability of it being in each possible class. It is very similar to what happens with "Machine Learning as a service", for example, Amazon ML[1] and Google Prediction API[2]. With only this information, the attacker wants to determine whether the input given is a member of the training dataset. There are three main ways to do this.

The first one is to compute several statistics, for example, the entropy of the probability distribution predicted and its greatest value, and claim that the entry is a member of the training set if these values exceed certain thresholds. The idea is that in entries that were used to train the classifier, the classifier is more confident in its answer, thus, these values are more extreme. This approach is already implemented in popular libraries of Machine Learning [46]. A second, and only slightly different, path is to feed these statistics into another ML algorithm. It can be trained to detect whether the values are from a member of the original dataset.

The third kind of black-box attack is slightly different. It creates a lot of ML classifiers that mimic the original one. These are called shadow models. These are fed data similar to the data fed to the classifier being attacked, and then they are combined to get a better prediction if a new entry was or was not used in the training. This method was tested in ML on-demand services (without any prior knowledge about the algorithms that were internally used) with sensitive data (location data, medical and purchase history), and the result was accurate [56]. These three approaches depend on the quality of the data used to feed the black-box attacks, thus, they depend on the amount of knowledge the attacker has.

In a *white-box attack*, the attacker knows the architecture used and the value of each parameter after training. An example of a white-box attack is an attack on a deep neural network [44]. When training this kind of algorithm, the gradient descent algorithm performs steps where the partial derivative of each neuron in the network is minimized. Thus, when an entry is fed to the net and most derivatives are close to zero, this is evidence that this input was already used to train the net [58]. Attacks on other algorithms have also been developed [39].

A slight variation of this attack starts with a random image and changes it in an iterative way to make the gradients get close to zero. Doing this with multiple random images allows the attacker to reconstruct a large part of the original dataset [28]. This works not only in applications concerning images but also in cases where algorithms train

[1] https://aws.amazon.com/machine-learning

[2] https://cloud.google.com/prediction

on natural language [67].

3.2.2 Privacy-aware Machine Learning

These attacks show that traditional algorithms can not be naively trusted to guarantee the privacy of their data. Thus, some other techniques were developed to ensure that Machine Learning systems are private. In this subsection, some general ideas that guide the construction of these algorithms will be listed.

There are four main ways to increase privacy in Machine Learning.

The first two ideas are based on each data owner keeping their data on their servers and doing the machine learning procedure in a distributed way.

In *federated learning*, each node (machine with stored data that wants to train a classifier) performs a step of optimization of some parameters of a classifier locally using only their data. After each optimization round, all nodes send the updated parameters to a central server. This server performs the averages of these values and returns them to each node that repeats the process. It continues until a convergence metric is satisfied [48]. The main advantage of this technique is that when using more machines, it is possible to significantly improve the speed of the computation [40]. From the point of view of privacy, no data is explicitly shared, increasing the data owners' privacy. Nevertheless, this method is not foolproof, and sharing the updates can be a privacy risk as well [30].

Another security problem related to Federated Learning is that all nodes must trust the central server to update, but the node can be an adversary or fail. To avoid this issue, *Decentralized Learning* was created. There is no central machine in this method, and the network can change with time [37]. Each data owner can choose with whom they will share updates, and there is not one node that every machine needs to trust. This technique can also be adapted so it is more resistant to byzantine faults, that is, nodes that can fail or design attacks to hinder the progress of others or even try to get information about the private datasets [63].

The other idea is to use *data encryption*. In data encryption, the computations needed to train the algorithm are made with encrypted data, not raw values. It is possible to do it using a technique called homomorphic encryption [26], where all the computations are done without revealing the actual value of the variables being used. Another option is to use garbled circuits [10]. They allow two parties that do not trust each other to compute a function of values that are distributed between them so that neither can find out the values of other secrets. Another way is to use a Trusted Execution Environment (TEE). These are modules in some modern processors that allow computation in encrypted data.

Thus, part of the training or the prediction phase can be made in those modules to not use the raw unprotected data. Different algorithms can be adapted to work on TEEs in decentralized environments [51] [20].

The final idea is *adding noise* to the original data so that ML algorithms can still be used, but the data can not serve malicious purposes. The main technique in this branch is called Differential Privacy (DP)[24].

Definition 20 (Differential Privacy)**.** *A mechanism $\mathcal{M}$ is ϵ-differentially private if, given two datasets d, d' that are only different in one line, the probability of every output S changes at most e^{ϵ} when the datasets change. That is:*

$$P(\mathcal{M}(d) = S) \leq e^{\epsilon} P(\mathcal{M}(d') = S).$$

The idea behind the concept is that taking part or not in a poll or census does not change significantly the results, so there is no risk in responding. DP was originally conceived so that statisticians could disclose queries about census data without worrying that information about particular individuals could be revealed. However, because DP mechanisms give mathematical certainties without making any assumptions about attackers, this technique has been growing in popularity and is used in many contexts [36] [19] [66] [61]. In Machine Learning, many algorithms have been tweaked so that noise is added so that the training algorithm becomes ϵ-differentially private. [1] [32].

These methods add noise when the parameters of the classifier are being set. Another way to add noise to the Machine Learning pipeline is to add it to the data that will then be fed to the system. This is usually called local differential privacy, because the most secure way to do it is when it is done locally, by the data producer. This happens because even if the data is leaked in any part of the process after it is produced, it is already secure. One example of such a technique is RAPPOR, a technology used by Google to add one extra layer of security and anonymization to diagnose data created by their browser [27].

3.3 Relation of privacy and fairness

As mentioned in the previous sections, there are two very active fields of study in ML: privacy and fairness. So, it is natural that both were studied at the same time. And this indeed has happened.

Some results show how to adapt existing algorithms so that they satisfy definitions of both fairness and privacy [62].

However, there are other interesting results. [2] showed a trade-off between certain notions of privacy and fairness, and it is impossible to create an algorithm that can always satisfy both conditions. Similarly, [18] showed that some definitions of fairness and privacy cannot be satisfied simultaneously. However, by relaxing the fairness definition, it is possible to build an algorithm that is private and approximately fair simultaneously.

Using the QIF framework, [60] created a new modeling that unifies fairness and privacy. It is interesting because it shows that the two properties are dual of one another. This modeling will be the main focus of this thesis.

Chapter 4

Extending QIF with direct and reverse flows

As we mentioned in the literature review, QIF was initially proposed to model situations where security is a concern. There is always a variable, called a secret, that is not known by an observer, and another variable is revealed to them. The goal is to measure how much information flows from the first variable through the observation of the second.

However, there are some cases where there is some information flow, but there may not be a hidden variable. There are also scenarios where there are two agents, and each knows only one variable. To model these scenarios, we will need to expand the QIF framework.

In this chapter, we will discuss the concepts of reverse and direct flow. Chapter 2 has already defined the traditional notion of flow, but we will revisit it and change the nomenclature slightly to reverse flow. Direct flow is going to be defined formally. After that, we will discuss the differences between both of them.

The reverse and direct flow concepts were originally presented in [60]. We will revisit them here to present some new terminology and notation.

4.1 Reverse flow

In the background section, we defined flow of information as the knowledge we gain about the input of a channel when the output is observed. Since the observation of output revealed something about the input, we call this reverse flow of information. We need to add this definition because there is another flow of information (knowledge gained about the output when the input is observed) that will be important to distinguish.

So, to further differentiate the QIF concepts we have defined in Chapter 2, we will present new names for old concepts in table 4.1

We will also present some new names for old concepts to make the text more precise. They are shown in table 4.1.

Table 4.1: Redefining the names of already presented concepts.

Original name	**New name**
Prior distribution	Reverse prior distribution
Prior vulnerability	Reverse prior vulnerability
Posterior vulnerability	Reverse posterior vulnerability
Flow	Reverse flow
Capacity	Reverse capacity

Sometimes, the original names will be used, but only when it is clear from context what they mean.

4.2 Direct flow

As we mentioned, QIF has only been used to measure the flow of information when the output is observed. However, this does not capture all possible scenarios. capture all possible scenarios, we need the concept of direct flow that will measure the flow of information when the input is observed and we want to infer information about the output. Nevertheless, before presenting it, we need a few extra concepts.

The first definition we are going to introduce is the direct prior distribution. While the reverse prior distribution is the distribution of inputs from the set $\mathcal{X}$, the direct prior represents the distribution of the outputs drawn from the set $\mathcal{Y}$.

Definition 21 (Direct prior distribution)**.** *The direct prior distribution of a prior $\pi : \mathbb{D}\mathcal{X}$ and a channel $C : \mathcal{X} \to \mathbb{D}\mathcal{Y}$ is denoted by $\rho^{\pi,C} : \mathbb{D}\mathcal{Y}$ and represents the distribution of the outputs . It is obtained by marginalizing the joint on $\mathcal{Y}$: $\rho^{\pi,C}_y = \sum_{x\in\mathcal{X}}(\pi \triangleright C)_{x,y}$ for all $y{\in}\mathcal{Y}$.*

We then can use this to compute the direct prior vulnerability. The idea behind the concept is the expected gain of an observer who does not know what the input is and takes an action that rewards them according to the output. It is precisely the same idea as the reverse prior vulnerability, but the secret is in the set $\mathcal{Y}$, not the set $\mathcal{X}$.

Definition 22 (Direct prior vulnerability)**.** *Given a set of actions $\mathcal{W}$ and a gain function $g : \mathcal{W}\times\mathcal{X} \to \mathbb{R}$, the direct prior vulnerability of a prior distribution $\pi : \mathbb{D}\mathcal{X}$ and a channel $C : \mathcal{X} \to \mathbb{D}\mathcal{Y}$ is*

$$V_g(\rho^{\pi,C}) = \max_{w\in\mathcal{W}} \sum_{y\in\mathcal{Y}} \rho^{\pi,C}_y g(w, y).$$

Just like we expanded the definition of prior vulnerability, it is possible to expand the definition of posterior vulnerability. The direct posterior vulnerability of a prior and a channel is the expected gain of an observer who knows the input and takes an action that rewards them according to the output.

Definition 23 (Direct posterior vulnerability)**.** *Given a set of actions $\mathcal{W}$ and a gain function $g : \mathcal{W} \times \mathcal{X} \to \mathbb{R}$, the direct posterior vulnerability of a prior distribution $\pi : \mathbb{D}\mathcal{X}$ and a channel $C : \mathcal{X} \to \mathbb{D}\mathcal{Y}$ is*

$$\overrightarrow{V_g}(\pi, C) = \sum_{x \in \mathcal{X}} \max_{w \in \mathcal{W}} \sum_{y \in \mathcal{Y}} \pi_x C_{x,y} g(w, y).$$

We can extend this definition to the case where we care about the worst-case scenario, not the average case.

Definition 24 (Direct maximum posterior vulnerability)**.** *Given a set of actions $\mathcal{W}$ and a gain function g, the direct maximum posterior vulnerability of a prior distribution π and a channel C is*

$$\overrightarrow{V_g}^{\max}(\pi, C) = \max_{x \in \mathcal{X}} \max_{w \in \mathcal{W}} \sum_{y \in \mathcal{Y}} \pi_x C_{x,y} g(w, y).$$

With these two concepts, we can extend the definition of a third concept: flow. The direct flow of a prior and a channel with respect to a gain function is how much the vulnerability increases after the system is run, considering only direct vulnerabilities.

Definition 25 (Direct flow)**.** *The multiplicative direct flow is the ratio between the posterior vulnerability and the prior vulnerability of a prior π, a channel C and a gain function g. It is denoted by*

$$\overrightarrow{\mathcal{L}_g^{\times}}(\pi, C) = \frac{\overrightarrow{V_g}(\pi, C)}{V_g(\rho^{\pi,C})}.$$

The additive direct flow is the difference between posterior vulnerability and the prior vulnerability of a prior π, a channel C and a gain function g. It is denoted by

$$\overrightarrow{\mathcal{L}_g^{+}}(\pi, C) = \overrightarrow{V_g}(\pi, C) - V_g(\rho^{\pi,C}).$$

Again, it makes sense to extend this definition to a scenario where the vulnerability that matters is the vulnerability of the maximum-case scenario. So, we define a metric that is the flow concerning the maximum posterior flow.

Definition 26 (Direct maximum flow)**.** *The direct maximum multiplicative flow is the ratio between the posterior vulnerability and the prior vulnerability of a prior π, a channel C and a gain function g. It is denoted by*

$$\overrightarrow{\mathcal{L}_g^{\times,\max}}(\pi, C) = \frac{\overrightarrow{V_g}^{\max}(\pi, C)}{V_g(\rho^{\pi,C})}.$$

The maximum additive reverse flow is the difference between the posterior vulnerability and the prior vulnerability of a prior π, a channel C and a gain function g. It is denoted by

$$\overrightarrow{\mathcal{L}_g^{+,\max}}(\pi, C) = \overrightarrow{V_g}^{\max}(\pi, C) - V_g(\rho^{\pi,C}).$$

In some scenarios, the gain function of interest may not be already defined. So, it is interesting to study the worst-case scenario, that is, of all gain functions, which one represents the largest flow. This is only defined for the average case flow. It is analogous to the original definition of capacity.

Definition 27 (Direct capacity)**.** *The direct multiplicative capacity of a prior π and a channel C is given by*

$$\overrightarrow{\mathcal{ML}^{\times}_{\mathbb{G}^+}}(\pi, C) = \max_{g \in \mathbb{G}^+} \frac{\overrightarrow{V_g}(\pi, C)}{V_g(\rho^{\pi,C})}.$$

The additive capacity is

$$\overrightarrow{\mathcal{ML}^{+}_{\mathbb{G}\updownarrow}}(\pi, C) = \max_{g \in \mathbb{G}\updownarrow} \overrightarrow{V_g}(\pi, C) - V_g(\rho^{\pi,C}).$$

4.3 The differences between reverse and direct flow

Now that we have the tools to measure direct and reverse flows, we need to explain when they are useful. The short answer is that they are used when analyzing a situation where we care about what each variable reveals about the other.

Let us consider two examples. In the first one, we have a traditional QIF scenario that concerns the privacy of a security system.

Example 1 (Password checker)**.** *A login website has a prompt that receives a password and a username. If the inserted password corresponds to the actual password of the username provided, the user is authenticated. If the password is wrong, a message shows that the password or username is incorrect.*

In this example, the primary concern of a privacy expert is that the displayed message may give information about the secret password. If many queries are made, a malicious agent can discard a user's possible passwords and then guess a password correctly. In this case, the only concern is the information we gain about the password when the message is displayed — the reverse flow of information.

Now, let us consider an example where we care about more than one flow of information.

Example 2 (Census). *A country performs a census on the population and publishes data regarding education and income. The annual income of several people is coupled with their level of education. It is natural that there is a correlation between the two variables, that is, the higher the education, the higher the income.*

The two variables, income and education, influence each other. People with higher incomes have children who can study for more time instead of directly joining the workforce, and people with advanced degrees can get better jobs with better salaries. In this case, there are two distinct (but related) flows of information: from the income to the education level and from the education level to the income.

Our extended QIF framework is made to deal with the second scenario. We want to consider an interaction between variables when both directions of information matter.

We must observe that it makes little sense to define a scenario where the only flow we want to measure is the direct flow. If such a case exists, we can build the joint, marginalize it to build the direct prior and channels, swap the inputs with the outputs, and use traditional QIF. After doing this, direct flow becomes reverse flow, and then we can work with QIF as usual.

4.4 Privacy and fairness in machine learning

Now, let us consider the main focus of the thesis.

Definition 28 (Privacy and fairness as reverse and direct flows). *Consider a machine learning classifier that takes several features of an individual and produces a binary classification $\hat{Y}$ that can have two possible values: $+$ for a positive classification and $-$ for a negative classification. The positive classification is always the preferred one. Let one of the features be a sensitive binary feature with two possible values: s_0 for the unprotected group and s_1 for the protected group. This feature is named S.*

Let C be a QIF channel where the input is the sensitive feature S and the output is the classification $\hat{Y}$. The prior distribution π is the distribution of the groups s_0, s_1.

Pushing π through C is the equivalent of running the classifier.

We will study two flows of information that happen when this channel is executed. The first is the knowledge gained about the input (the sensitive feature) when the output (the classification) is observed. We will call this the privacy flow because we get knowledge about potentially private information.

The other flow is the knowledge gained about the classification when the group is observed. If there is no flow of information, being in a different group does not change the probability of getting a positive classification. However, if this probability is not the same for the protected and unprotected groups, then the system is unfair. Therefore, this is a fairness flow.

Because both flows matter in a sensitive scenario, we cannot disregard one of the flows and focus only on one direction. We must consider both simultaneously. That is why we need the expansion of the QIF framework to deal with the reverse flow instead of just switching the inputs for outputs.

Besides being able to model privacy and fairness within the same representation, the QIF framework has other benefits that show that it is a good model for both quantities. We will now show two advantages: The ability to model different scenarios using gain functions and the power to incorporate prior distributions into the model.

4.4.1 Using gain functions

Most fairness measures, like statistical parity, shown in Definition 16, and equal opportunity, shown in Definition 17, use the difference between probabilities to define unfairness. In the QIF framework, average Bayes flow works similarly. However, there are situations where using only the probabilities does not capture the scenario completely. To show this, consider Example 3.

Example 3 (The bigoted employer [1]). *Let the secrets of a channel represent whether an individual is HIV-positive. Suppose that there is a bigoted employer who wants to hire a worker, but they want the worker to be HIV-negative. A possible gain function that represents this employer is shown in Table 4.2.*

Table 4.2: Possible gain function modeling a bigoted employer

g_{HIV}	$y =$ HIV+	$y =$ HIV−
$w =$ hire	-10	10
$w =$ dismiss	5	-1

This gain function shows that the maximum possible gain for the employer is to hire an HIV-negative employee, and the maximum loss is to hire an HIV-positive individual. The other two situations, dismissing employees, are not as bad or as good for the employer.

[1] This example is a slight modification of an example in [60]. In turn, it is inspired by [22].

Using only probabilities, such as $P(\text{hire}|\text{HIV+})$ or $P(\text{dismiss}|\text{HIV-})$ would not capture the fact that hiring an HIV-positive worker is ten times worse than dismissing an HIV-negative person from the perspective of the bigoted employer, for example.

Other fairness scenarios may have similar characteristics. For example, an ML system that decides whether to hire an individual will probably have a high loss on misclassifying someone from the protected group, compared to misclassifying someone from the unprotected group. Such error may lead to a lawsuit or other legal consequences, for example.

4.4.2 Considering the prior distribution

Another advantage of using QIF to model privacy and fairness is the capacity to take the prior distribution into account. This is illustrated in Example 4.

Example 4 (Brazilian Embassy in the United States)**.** *Suppose that the Brazilian Embassy in the United States is looking for employees. It is going to create an admission process that will happen both in Brazil and in the US. For transparency reasons, Brazilian law mandates that the result of the admission process must be publicly available.*

Suppose an ML classifier is used to pre-select part of the candidates. For this example, suppose that this classifier is skewed regarding race. So, the unprotected group has a higher probability of being accepted. Table 4.3 shows a possible channel.

Table 4.3: Channel describing a skewed classifier.

C	$y = \text{hire}$	$y = \text{dismiss}$
$s_0 = \text{non-black}$	0.5	0.5
$s_1 = \text{black}$	0.4	0.6

Now, consider this system being run in two scenarios. In the first one, the Embassy is considering applications from Brazil, so the prior distribution in terms of race is $\pi_{Brazil} = (0.46, 0.54)$ *[53]. In this case, the reverse average multiplicative Bayes flow is* 1.03*. In the second scenario, applications from the US are being processed. The population distribution is* $\pi_{US} = (0.86, 0.14)$ *[13]. In this case, the reverse average multiplicative Bayes flow is* 1.0*.*

Suppose we consider this system being run in terms of statistical parity. In that case, nothing changes from one situation to the other because it only considers the channel, not the population. Nevertheless, this does not happen with Bayes flow. Because it is a measure of both the prior and the channel, the flow of information is different in both scenarios. In the Brazilian case, an observer can change his guess of what group an individual belongs based on the result of the admissions process. However, this does not happen in the American case because the distribution is skewed, and the observer does not gain knowledge when observing the classification.

Although this example concerns privacy, we can create a similar example where the preoccupation is fairness. In this case, we would need to change the distribution of the classifications, not the groups.

In the next chapter, we will show bounds that govern what can happen when the two flows are observed.

Chapter 5

Theoretical bounds

Now that we have defined a new type of metric, direct flow, a new class of problem was created. When the reverse flow is equal to a constant α, what possible direct flow values can be achieved?

In this chapter, we will answer this question for six different cases. For the metrics of average Bayes flow, maximal Bayes flow and capacity, we will show the feasibility region for direct and reverse flow for the multiplicative and additive cases. The results are summarized on table 5.1.

Table 5.1: Feasibility region if α is the reverse flow and β the direct one.

	Multiplicative case	Additive case
Avg. Bayes	$\frac{\beta}{3-\beta} \leq \alpha \leq \frac{3\beta}{\beta+1}, 1 \leq \alpha, \beta \leq 2$	$\frac{-1}{2} + 2\beta \leq \alpha \leq \frac{\frac{1}{2}+\beta}{2}, 0 \leq \alpha, \beta \leq \frac{1}{2}$
Max. Bayes	$1 \leq \alpha, \beta \leq 2$	$0 \leq \alpha, \beta \leq \frac{1}{2}$
Capacity	$1 \leq \alpha, \beta \leq 2$	$0 \leq \alpha, \beta \leq \frac{1}{2}$

All proofs are for binary channels, where $|\mathcal{X}| = |\mathcal{Y}| = 2$. It is easier to write the proofs in terms of the joints, so every combination of prior π and channel C will result in a joint J that will be written as

$$[\pi \triangleright C] = J = \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} \pi_{s_0} C_{s_0,+} & \pi_{s_0} C_{s_0,-} \\ \pi_{s_0} C_{s_1,+} & \pi_{s_0} C_{s_1,-} \end{pmatrix}. \tag{5.1}$$

This joint's entry i, j represents the probability of input i and output j happening together, that is $J_{ij} = [\pi \triangleright C]_{ij}$.

Because a joint uniquely defines a prior and a channel and vice versa, every proof for the flow of a joint is equivalent to a proof for a combination of prior and channel.

We generated 3 million joints for every cell in table 5.1 and computed the direct and reverse flows. These joints account for all possible joints that have all probabilities as multiples of $\frac{1}{2^8}$. Then, we plotted them in a scatter plot to visualize the feasibility region. These plots are going to be shown after each theorem defining the feasibility bounds.

5.1 Average Bayes flow

The Bayes vulnerability of a prior distribution, shown in Definition 4, is the expected probability over all possible observations that an optimal adversary will guess correctly a value drawn from this distribution. The Bayes vulnerability of a prior and a channel is the probability of an optimal adversary guessing the secret correctly after the secret was fed to the channel and they saw the output. Finally, the multiplicative Bayes flow is the fraction between these two quantities and, the additive is the subtraction and the additive Bayes flow is the difference between the two.

We will now deal with the case where both the reverse and direct gain functions are Bayes vulnerability. In this section, we will only talk about average Bayes flow, so we will refrain from saying it is the average and not the maximal case.

5.1.1 Multiplicative Bayes flow

We begin by defining Bayes flow with regard to the joint. We will call the reverse flow α and the direct flow β. This will make the notation more clear for the proofs. Thus, we have

$$\alpha = \mathcal{L}^{\times}_{id}(J) = \frac{\max(a,c)+\max(b,d)}{\max(a+b,c+d)}$$

and

$$\beta = \overrightarrow{\mathcal{L}^{\times}_{id}}(J) = \frac{\max(a,b)+\max(c,d)}{\max(a+c,b+d)},$$

Where a, b, c, d are the elements of the joint, as in equation 5.1.

We want to describe all possible situations, that is, all combinations of direct and reverse flow values simultaneously. To do this, we present the following lemma that was proved in the Appendix A.1.1.

Lemma 1. *For every point in the set*

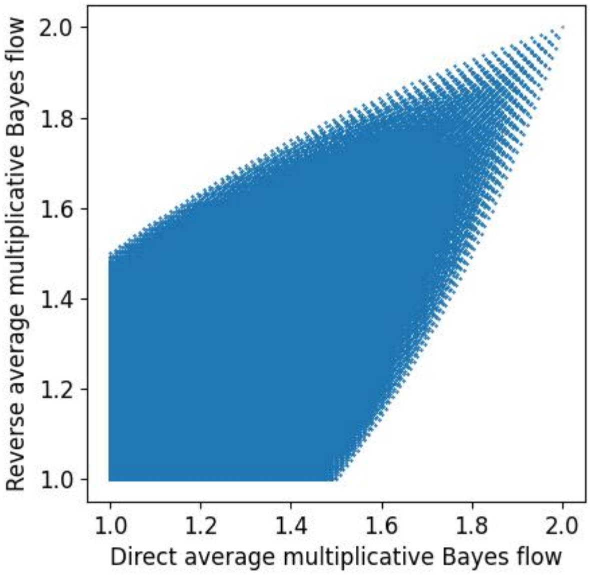
$$\left\{(\alpha,\beta)|\frac{\beta}{3-\beta} \leq \alpha \leq \frac{3\beta}{\beta+1}, 1 \leq \alpha,\beta \leq 2\right\}$$

there is a prior and a channel with α as the reverse average multiplicative Bayes flow and β as the direct average multiplicative Bayes flow. There are no joints with values of direct and reverse flows that are not in this set.

Figure 5.1 shows the feasibility region described in Lemma 1 generated as we described in 5. It is possible to see that only part of the square defined by the points

$(1,1)$ and $(2,2)$ is feasible. The value of flow has to be in the interval $(1,2)$ [3]. Another interesting property we can see from the figure is that in general the greater one of the flows, the greater the other. And this makes sense. If one of the variables defines the value of the other, it is natural that the other will define the value of the one.

Figure 5.1: The feasibility region for direct and reverse average multiplicative Bayes flow.



Source: created by the author.

This region guarantees that in a situation where the multiplicative Bayes flow is the measure of interest, guaranteeing a high enough direct flow guarantees a high reverse flow. The same can be said of low-flow scenarios.

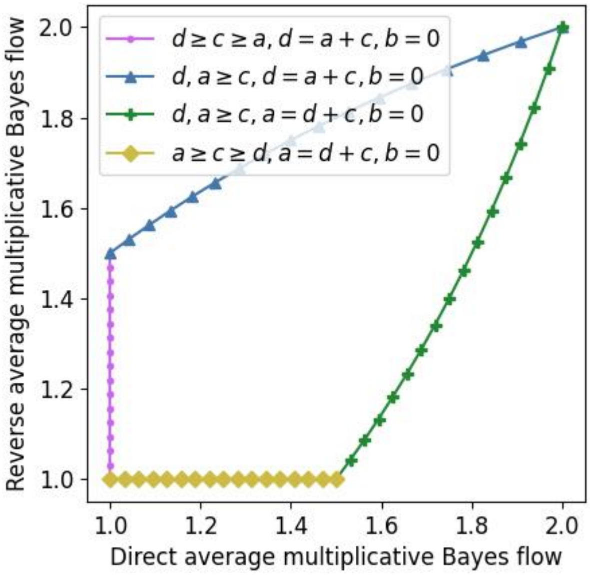
Joints in the Pareto curve.

Another question we may ask is, what are the joints (or priors and channels) in the Pareto curves of direct and reverse flows?

There are four Pareto curves that define this region. The first one is highlighted in blue in Figure 5.2. If there is a joint in this region, there is no other joint with a higher reverse flow with the same amount of direct flow. The second one is the pink one. If there is a joint in this region, there is no other joint with less direct flow and the same amount of reverse flow. The other two curves, in yellow and in green, are analogous.

Figure 5.2 shows this division and a characterization of each component. This is not the only possible characterization, and others can be made. In Appendix A.1.1.1, there is another proof showing that these joints are, in fact, in the Pareto curve.

Figure 5.2: Pareto curves in the multiplicative case. The legend shows the characteristics of each part of the curve.



Source: created by the author.

5.1.1.1 An observation about differential privacy

To prove the feasibility region shown on Figure 5.1, we use the following lemma.

Lemma 2. *The joint*

$$\pi \triangleright C = J = \begin{pmatrix} \frac{\alpha\beta+\alpha-\beta}{\alpha\beta+\alpha+\beta} & 0 \\ \frac{-\alpha\beta+\alpha+\beta}{\alpha\beta+\alpha+\beta} & \frac{\alpha\beta-\alpha+\beta}{\alpha\beta+\alpha+\beta} \end{pmatrix}$$

has reverse average multiplicative Bayes flow equal to α and direct average multiplicative Bayes flow equal to β for every α, β in the set

$$\{(\alpha, \beta) \mid \frac{\beta}{3-\beta} \leq \alpha \leq \frac{3\beta}{\beta+1}, 1 \leq \alpha \leq 2, 1 \leq \beta \leq 2\}$$

This joint has an interesting property: one of the entries is a zero. This may seem unimportant, but it matters.

One of the many notions of privacy is differential privacy. As we said in Chapter 3, a mechanism $\mathcal{M}$ is ϵ-differentially private if no input makes an output ϵ times more likely than another input (if they differ slightly).

Suppose an entry of the joint is zero, for example, $b = 0$. In that case, when the output corresponding to the second column is shown, the input corresponding to the second row is infinitely more likely than the output corresponding to the first row. So, the mechanism represented by the joint cannot be ϵ-differentially private for any value of ϵ, which may not be acceptable in scenarios where differential privacy is required.

Considering this scenario, we propose a conjecture.

Conjecture 1. *For any values of* α *and* β *in* $[1, 2]$, *the joint*

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

where

$$b \leq \begin{cases} \frac{3\alpha-\beta-\alpha\beta}{4\alpha} & \text{if } \alpha < \beta, \\ \frac{\alpha+\beta-\alpha\beta}{4\alpha} & \text{otherwise,} \end{cases}$$

$$a = \frac{-\alpha + b\alpha + \alpha\beta + b\beta - b\alpha\beta}{-\alpha + \beta + \alpha\beta},$$

$$c = 1 - \alpha + a\alpha - b + b\alpha$$

$$d = 1 - a - b - c$$

has reverse Bayes flow equal to α *and direct Bayes flow equal to* β*.*

Our belief that the conjecture holds comes from the fact that we have tested it for 56 million different values of α, β and b, and they were all correct. That is, the reverse flow was α, the direct flow was β, and the values of a, b, c, d formed a probability distribution.

One interesting characteristic about this joint is that the value b is not exact. It is limited from below by zero (it is part of a joint probability distribution), and our conjecture provides a bound from above. Thus, it is possible to choose a value for it, and the other values will change accordingly.

Having all entries of the joint being non-zero is an interesting property because of differential privacy. Having a zero entry makes the mechanism ∞-differentially private. If all elements are non-zero, than the mechanism is ϵ-differentially private for a constant ϵ.

5.1.2 Additive Bayes flow

Now, let us redefine α, β to use it to discuss the additive case. Let

$$\alpha = \mathcal{L}^{+}_{id}(J) = \max(a, c) + \max(b, d) - \max(a + b, c + d)$$

and

$$\beta = \overrightarrow{\mathcal{L}^{+}_{id}}(J) = \max(a, b) + \max(c, d) - \max(a + c, b + d).$$

With these definitions, we can present another lemma. It was also proved in Appendix A.1.2.

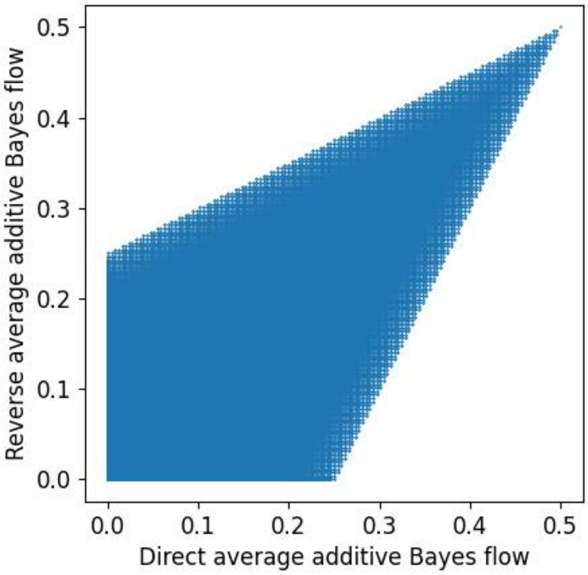
Lemma 3. *For every point in the set*

$$\left\{(\alpha,\beta)| -\frac{1}{2}+2\beta \leq \alpha \leq \frac{1}{4}+\frac{\beta}{2}, 0 \leq \alpha \leq \frac{1}{2}, 0 \leq \beta \leq \frac{1}{2}\right\}$$

there is a prior and channel with α as the reverse average additive Bayes flow and β as the direct average additive Bayes flow. There are no joints with values of direct and reverse flows that are not in this set.

Figure 5.3 shows the plot of the feasibility region, it was generated as described in 5. Like in the multiplicative case, only part of the square defined by the maximal and minimum flows is feasible. But now, the edges of the region are straight lines.

Figure 5.3: The feasibility region for direct and reverse average additive Bayes flow.

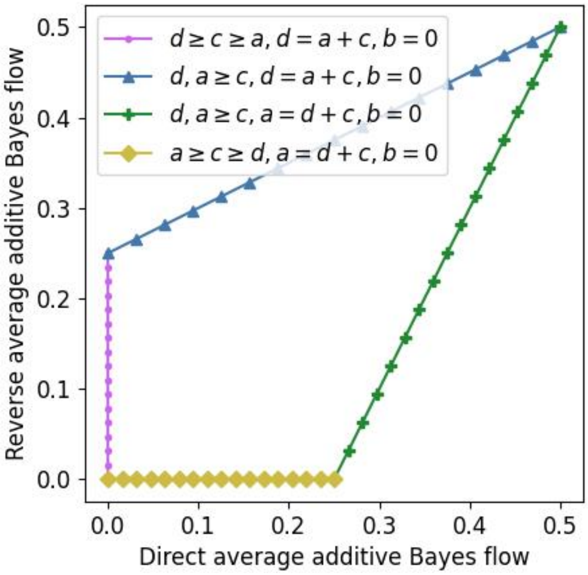


Source: created by the author.

This gives similar guarantees as the multiplicative case. The reverse flow will also be high if the direct flow is high enough. And this makes sense, both the multiplicative and additive flows are measuring the same quantities, prior and posterior Bayes vulnerabilities, the only thing that changes is how they are combined.

Joints in the Pareto curve Again, we show joints in the Pareto curve in Figure 5.4. These joints are not unique; other joints can achieve these values of direct and reverse flow. In Appendix A.1.2.1, we prove that these joints are, in fact, in the Pareto curves.

Figure 5.4: Pareto curves in the additive case. The legend shows the characteristics of each part of the curve.



Source: created by the author.

5.2 Maximum Bayes flow

Again, because we will only talk about maximum flow in this section, we sometimes omit the word maximum.

5.2.1 Multiplicative Bayes flow

We begin by defining Bayes flow with regard to the joint. We will call the reverse flow α and the direct flow β. This will make the notation more clear. Thus, we have

$$\alpha = \mathcal{L}_{id}^{\times,\max}(J) = \frac{\max\left(\frac{\max(a,c)}{a+c}, \frac{\max(b,d)}{b+d}\right)}{\max(a+b, c+d)}$$

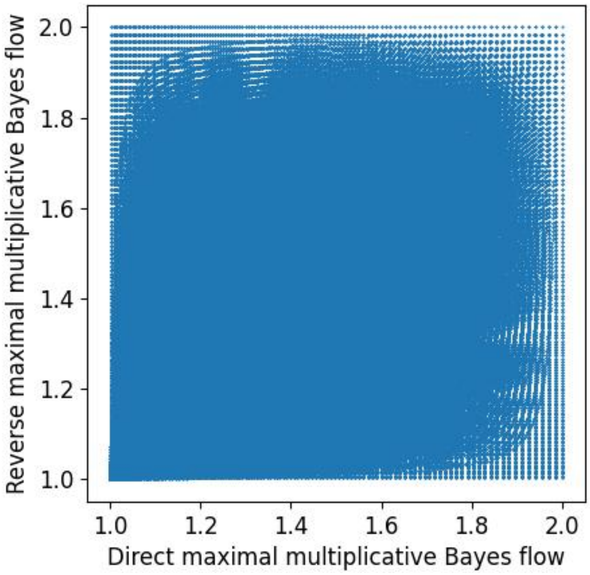
and

$$\beta = \overrightarrow{\mathcal{L}}_{id}^{\times,\max}(J) = \frac{\max\left(\frac{\max(a,b)}{a+b}, \frac{\max(c,d)}{c+d}\right)}{\max(a+c, b+d)}.$$

Now, let us analyze the feasibility region of this metric. Figure 5.5 shows the direct and reverse flow of various joints generated according to the specification given in Chapter

5. Unlike the average case, it is possible to get almost every combination of α and β. We show this in lemma 4.

Figure 5.5: The feasibility region for direct and reverse maximum multiplicative Bayes flow.



Source: created by the author.

Lemma 4. *There is at least one joint that has reverse maximum multiplicative Bayes flow α and direct maximum multiplicative Bayes flow β for every pair in the set:*

$$\{(\alpha,\beta)|1<\alpha<2,1<\beta<2\}.$$

We now give a simple proof of this lemma.

Proof. Consider the joint

$$J=\begin{pmatrix} 0 & \frac{\alpha-1}{\alpha} \\ \frac{\beta-1}{\beta} & \frac{\alpha+\beta-\alpha\beta}{\alpha\beta} \end{pmatrix}.$$

We will show that this is, in fact, a joint probability distribution and then that it has direct flow equal to α and reverse flow equal to β.

To prove it is a joint, we must show that all values sum to one. So, let us sum them

$$a+b+c+d=0+\frac{\alpha-1}{\alpha}+\frac{\beta-1}{\beta}+\frac{\alpha+\beta-\alpha\beta}{\alpha\beta}=\frac{\alpha\beta}{\alpha\beta}=1.$$

The second part is to show that all the values are greater or equal to zero. Both b and c are greater than zero because $\alpha,\beta>1$. d is also greater to zero because $\alpha\beta<\alpha+\beta$ when $1<\alpha,\beta<2$.

Now, let us compute the reverse flow

$$\frac{\max\left(\frac{\max(a,c)}{a+c}, \frac{\max(b,d)}{b+d}\right)}{\max(a+b, c+d)}$$

Because $a = 0$, we have that $\frac{max(a,c)}{a+c} = \frac{c}{c} = 1$. And this is always going to be at least as big as $\frac{\max(b,d)}{b+d}$. So, the reverse flow is

$$\frac{1}{\max(a+b, c+d)}.$$

The value of $a + b$ is equal to b, and the greatest value it can take is when $\alpha = 2$. Then $b = 0.5$. And, because a, b, c, d form a joint, they sum to one. Thus, $c + d$ must be at least 0.5 and $c + d \geq a + b$. Computing this sum, we get

$$c + d = \frac{\beta - 1}{\beta} + \frac{\alpha + \beta - \alpha\beta}{\alpha\beta} = \frac{\alpha\beta - \alpha + \alpha + \beta - \alpha\beta}{\alpha\beta} = \frac{\beta}{\alpha\beta} = \frac{1}{\alpha}.$$

Finally, the reverse flow is

$$\frac{1}{\max(a+b, c+d)} = \frac{1}{\frac{1}{\alpha}} = \alpha.$$

Now, we need to compute the direct flow. The reasoning is the same, so we will be more direct now. The direct flow is

$$\frac{\max\left(\frac{\max(a,b)}{a+b}, \frac{\max(c,d)}{c+d}\right)}{\max(a+c, b+d)} = \frac{1}{b+d} = \frac{1}{\frac{1}{\beta}} = \beta.$$

□

An interesting consequence is that there is no way to be safe if the modelled scenario is captured under the maximum multiplicative Bayes flow. There can always be a joint probability distribution that can lead the channel to leak the most information possible. The direct flow does not bound the reverse flow at all. No matter the value of one of the flows, the other can attain any value.

5.2.2 Additive Bayes flow

We will redefine α, β so they refer to additive Bayes flow. Now, we have

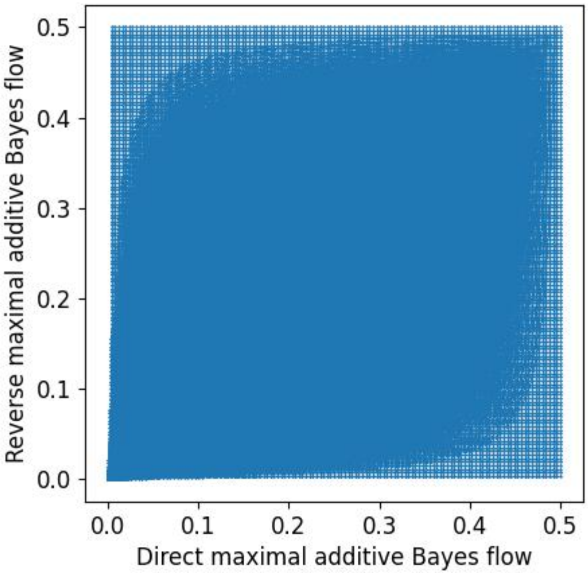
$$\alpha = \mathcal{L}_{id}^{\times,\max}(J) = \max\left(\frac{\max(a,c)}{a+c}, \frac{\max(b,d)}{b+d}\right) - \max(a+b, c+d)$$

and

$$\beta = \overrightarrow{\mathcal{L}}_{id}^{\times,\max}(J) = \max\left(\frac{\max(a,b)}{a+b}, \frac{\max(c,d)}{c+d}\right) - \max(a+c, b+d).$$

Figure 5.6 shows the direct and reverse flow for several joints, according to the specification given on 5. We can see from the image that all the points in the square defined by the points $(0,0)$ and $(0.5,0.5)$ should be reached if we choose the correct joint. This is shown in lemma 5.

Figure 5.6: The feasibility region for direct and reverse maximum additive Bayes flow.



Source: created by the author.

Lemma 5. *There is at least one channel that has reverse maximum additive Bayes flow α and direct maximum additive Bayes flow β for every pair in the set:*

$$\{(\alpha,\beta)|0<\alpha<\frac{1}{2}, 0<\beta<\frac{1}{2}\}.$$

Now, let us prove this statement.

Proof. Consider the joint

$$\pi \triangleright C = J = \begin{pmatrix} 0 & \alpha \\ \beta & 1-\alpha-\beta \end{pmatrix}.$$

We will prove it is a joint with reverse flow α and direct flow β.

We begin by showing that it is a joint. The sum of all values is 1, and because $0 < \alpha, \beta < 0.5$, all values are greater or equal to zero.

Now, computing the reverse flow, we have

$$\max\left(\frac{\max(a,c)}{a+c}, \frac{\max(b,d)}{b+d}\right) - \max(a+b, c+d).$$

Because $a = 0$, we have that $\frac{\max(a,c)}{a+c} = 1$, and this will always be chosen as the max, so the reverse flow is

$$1 - \max(a+b, c+d).$$

We are only interested in the region where $\alpha < 0.5$, so $a + b = b = \alpha < 0.5$. Thus, we have that $c + d > a + b$ and the reverse flow is

$$1 - (c + d) = 1 - (\beta + 1 - \alpha - \beta) = \alpha,$$

as we wanted.

Now, we only need to show that the direct flow is β. Because the arguments are similar to the ones in the direct flow part, we will make it shorter. The direct flow is

$$1 - (b + d) = 1 - (\alpha + 1 - \alpha - \beta) = \beta,$$

and this concludes the proof. □

Just like in the multiplicative case, we have that all of the possible combinations of direct and reverse flow are possible. The main conclusion is that the direct and reverse maximum flows do not bound each other.

5.3 Capacity

The third group of metrics we will analyze is different. Suppose that a prior and a channel that model a system will be deployed, but the exact situation is not exactly clear in the sense that the gain function that models the adversary is not known. In this case, we cannot analyze the Bayes flow or any other gain function and be satisfied. We have to analyze the capacity.

In this thesis, the capacity (shown in Definitions 12 and 13) will be applied to a prior and a channel and is defined by the flow regarding the gain function that maximizes it. So, it represents a scenario where the joint is fixed, but the gain function can change. (In other circumstances, capacity may refer to a fixed gain function and a variable channel or prior (or both), but only the gain function can change here.)

The capacity is the result of maximizing the flow. However, we can maximize the average flow or the maximal flow. Here, we will focus only on the average flow because it is the most common metric.

5.3.1 Multiplicative capacity

As always, we begin by defining capacity with regard to the joint. The reverse capacity is α, and the direct is β. So, we have

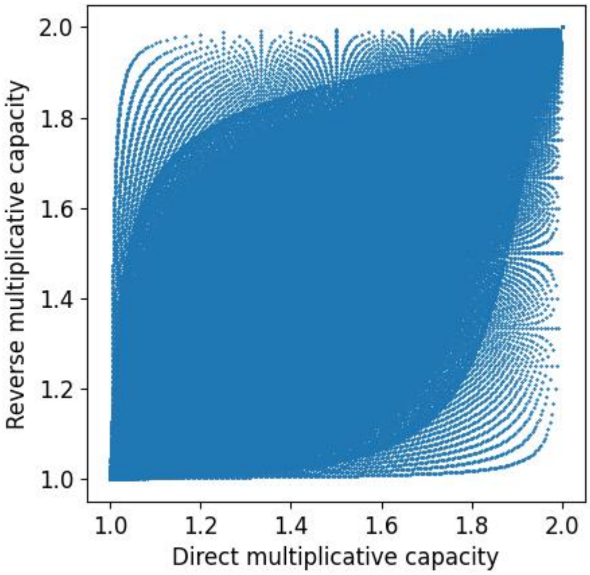
$$\alpha = \mathcal{ML}^{\times}_{\mathbb{G}^+}(\pi, C) = \max\left(\frac{a}{a+b}, \frac{c}{c+d}\right) + \max\left(\frac{b}{a+b}, \frac{d}{c+d}\right)$$

and

$$\beta = \overrightarrow{\mathcal{ML}^{\times}_{\mathbb{G}^+}}(\pi, C) = \max\left(\frac{a}{a+c}, \frac{b}{b+d}\right) + \max\left(\frac{c}{a+c}, \frac{d}{b+d}\right).$$

Figure 5.7, generated according to the specification given on 5, shows that the feasibility region for the multiplicative capacity encompasses all possible values of direct and reverse capacity. This is shown formally in lemma 6.

Figure 5.7: The feasibility region for direct and reverse multiplicative capacity.



Source: created by the author.

Lemma 6. *There is at least one channel that has reverse multiplicative capacity α and direct multiplicative capacity β for every pair in the set:*

$$\{(\alpha, \beta) | 1 < \alpha < 2, 1 < \beta < 2\}.$$

Now, let us prove this lemma.

Proof. Consider the joint

$$\begin{pmatrix} \frac{(\alpha-2)(\beta-1)}{\alpha\beta-\alpha-\beta} & 0 \\ \frac{-(\alpha-2)(\beta-2)}{\alpha\beta-\alpha-\beta} & \frac{(\alpha-1)(\beta-2)}{\alpha\beta-\alpha-\beta} \end{pmatrix}$$

We begin by proving that it is, in fact, a joint probability distribution. The sum must be one:

$$a+b+c+d = \frac{(\alpha\beta-\alpha-2\beta+2)+0+(-\alpha\beta+2\alpha+2\beta-4)+(\alpha\beta-2\alpha-\beta+2)}{\alpha\beta-\alpha-\beta},$$

$$a+b+c+d = \frac{\alpha\beta-\alpha-\beta}{\alpha\beta-\alpha-\beta} = 1.$$

And all elements must be greater than zero. First, note that because $1 < \alpha, \beta < 2$ we have that $\alpha\beta - \alpha - \beta$, the denominator, is always smaller than zero. $(\alpha-2)(\beta-1)$ is the product of a positive number and a negative one, so it is negative, and the value of a will be positive. The same can be said for d. c will be positive because it is a product of two negative terms divided by something negative and multiplied by -1, so it is positive in the end.

Now, we will check if the capacities are α and β. The reverse capacity is:

$$\mathcal{ML}^{\times}_{\mathbb{G}^+}(J) = \max\left(\frac{a}{a+b}, \frac{c}{c+d}\right) + \max\left(\frac{b}{a+b}, \frac{d}{c+d}\right).$$

Because $b = 0$, then $\max\left(\frac{a}{a+b}, \frac{c}{c+d}\right) = \frac{a}{a+b} = 1$ and $\max\left(\frac{b}{a+b}, \frac{d}{c+d}\right) = \frac{d}{c+d}$. Thus,

$$\mathcal{ML}^{\times}_{\mathbb{G}^+}(J) = 1 + \frac{d}{c+d} = 1 + \frac{(\alpha-1)(\beta-2)}{(\alpha-1)(\beta-2)-(\alpha-2)(\beta-2)},$$

$$\mathcal{ML}^{\times}_{\mathbb{G}^+}(J) = 1 + \frac{\alpha\beta-2\alpha-\beta+2}{\beta-2} = \frac{\alpha\beta-2\alpha}{\beta-2} = \frac{\alpha(\beta-2)}{\beta-2} = \alpha,$$

The reverse capacity is

$$\overrightarrow{\mathcal{ML}^{\times}_{\mathbb{G}^+}}(J) = \max\left(\frac{a}{a+c}, \frac{b}{b+d}\right) + \max\left(\frac{c}{a+c}, \frac{d}{b+d}\right).$$

Because $b = 0$, then $\max\left(\frac{a}{a+c}, \frac{b}{b+d}\right) = \frac{a}{a+c}$ and $\max\left(\frac{c}{a+c}, \frac{d}{b+d}\right) = \frac{d}{b+d} = 1$. Thus,

$$\overrightarrow{\mathcal{ML}^{\times}_{\mathbb{G}^+}}(J) = 1 + \frac{\alpha\beta-\alpha-2\beta+2}{\alpha-2} = \frac{\alpha\beta-2\beta}{\alpha-2} = \frac{\beta(\alpha-2)}{\alpha-2} = \beta.$$

This completes the proof. □

When we were looking at the Bayes flow, the question we were trying to answer is: in a system where the Bayes vulnerability gain function models the adversary, what are all possible values of direct and reverse flow?

Now, we are looking at a scenario we do not know which gain function will model the situation, so we use the capacity of the joint as a conservative estimate of risk. And the question we ask now is: In a system where the measure of risk is the capacity, what are all possible values of direct and reverse flow?

We conclude that we cannot guarantee any value of direct and reverse flow. All possible combinations are valid scenarios, and one value does not bound the other, unlike the average Bayes flow case.

5.3.2 Additive capacity

Now, the final bound is the additive capacity. Again, we redefine α and β as reverse and direct additive capacities.

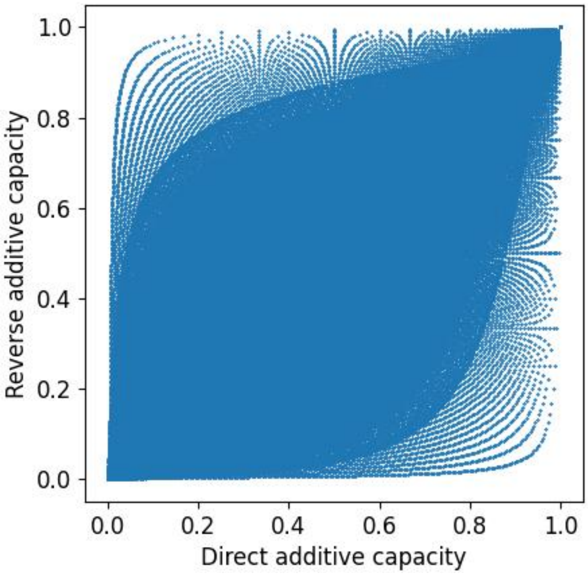
$$\alpha = \mathcal{ML}^{+}_{\mathbb{G}\updownarrow}(J) = 1 - \min\left(\frac{a}{a+b}, \frac{c}{c+d}\right) - \min\left(\frac{b}{a+b}, \frac{d}{c+d}\right)$$

and

$$\beta = \overrightarrow{\mathcal{ML}^{+}_{\mathbb{G}\updownarrow}}(J) = 1 - \min\left(\frac{a}{a+c}, \frac{b}{b+d}\right) - \min\left(\frac{c}{a+c}, \frac{d}{b+d}\right).$$

Figure 5.8, again generated using the specification explained on 5, shows possible direct and reverse additive capacity values. They occupy the whole possible region one more time, and one does not bound the other. We formalize this on lemma 7.

Figure 5.8: The feasibility region for direct and reverse additive capacity.



Source: created by the author.

Lemma 7. *There is at least one channel that has reverse additive capacity α and direct additive capacity β for every pair in the set:*

$$\{(\alpha, \beta)|0 < \alpha < \frac{1}{2}, 0 < \beta < \frac{1}{2}\}.$$

Now, we are going to prove lemma 7.

Proof. Consider the joint

$$\begin{pmatrix} \frac{(\alpha-1)\beta}{\alpha\beta-1} & 0 \\ \frac{-(\alpha-1)(\beta-1)}{\alpha\beta-1} & \frac{(\beta-1)\alpha}{\alpha\beta-1} \end{pmatrix}$$

We will show that it is a joint distribution with reverse capacity α and direct capacity β.

First, the elements sum to 1:

$$a+b+c+d = \frac{(\alpha-1)\beta}{\alpha\beta-1} + \frac{-(\alpha-1)(\beta-1)}{\alpha\beta-1} + \frac{(\beta-1)\alpha}{\alpha\beta-1} = a+b+c+d = \frac{\alpha\beta-1}{\alpha\beta-1} = 1$$

To check that they are all positive, first note that the denominator is negative. Because $\alpha, \beta < 1$, all numerators are also negative, so the values are positive.

Now, let us compute the reverse capacity. It is

$$1 - \min\left(\frac{a}{a+b}, \frac{c}{c+d}\right) - \min\left(\frac{b}{a+b}, \frac{d}{c+d}\right).$$

The minimum between $\frac{b}{a+b}$ and $\frac{d}{c+d}$ is the first term because $b = 0$. This also implies that $\frac{a}{a+b} = \frac{a}{a} = 1$ and we have

$$1 - \frac{c}{c+d} = 1 - \frac{\alpha+\beta-\alpha\beta-1}{\beta-1} = \frac{\alpha(\beta-1)}{\beta-1} = \alpha.$$

We can use the same arguments to compute the direct capacity, and we get

$$1 - \frac{c}{a+c} = 1 - \frac{\alpha+\beta-\alpha\beta-1}{\alpha-1} = \frac{\alpha\beta-\beta}{\alpha-1} = \beta.$$

As we wanted.

□

Just like in the multiplicative capacity, we conclude that if we use additive capacity as a measure of risk, there are no guarantees we can give with respect to the values of direct and reverse flow. All the values can be achieved, and direct capacity does not bound reverse capacity.

Chapter 6

On QIF and fairness

We will now show the relationship between fairness and QIF in two ways. We begin by revisiting Definition 28, where we modeled fairness and privacy as direct and reverse flows, and interpret this using our results from the previous chapters. After that, we show that QIF can capture existing notions of fairness from the literature.

6.1 A QIF model to privacy and fairness

This section will analyze the model described in Definition 28.

As a brief reminder, this model consists of a channel in which the binary input indicates if an individual is part of a sensitive group, and the output is the binary classification given by a machine learning classifier. There is always a protected group and a preferred classification class. The direct flow of information measures the system's fairness, while the reverse flow is a privacy metric.

We can further analyze several aspects of this model using what we described in the previous chapter. We begin with the duality aspect.

6.1.1 The duality aspect

Note that, using this definition, fairness and privacy become interlinked. If we replace the input set with the output set and vice versa, the flow we measured to assess fairness now assesses privacy. And the same happens with the measure that was analyzing privacy. It now measures fairness. In mathematical terms, these two aspects are duals of one another. If we switch the inputs for the outputs, reverse and directed flow are also switched.

This has several benefits. First, it aids with interpretation. Both privacy and fairness can be interpreted similarly. An interpretation used in only one context can be expanded to explain the other problem. Besides that, the tools we have developed to study and improve each of these concepts can also be modified to deal with the other. All of the mathematical properties that have been proved about fair datasets, models and systems can be adapted to guarantee that they are also private.

6.1.2 Interpretation of the theoretical bounds

We can now look at the theoretical bounds we studied in the previous chapter in a new light. The reverse flow can be seen as "unprivacy", and the direct flow is unfairness.

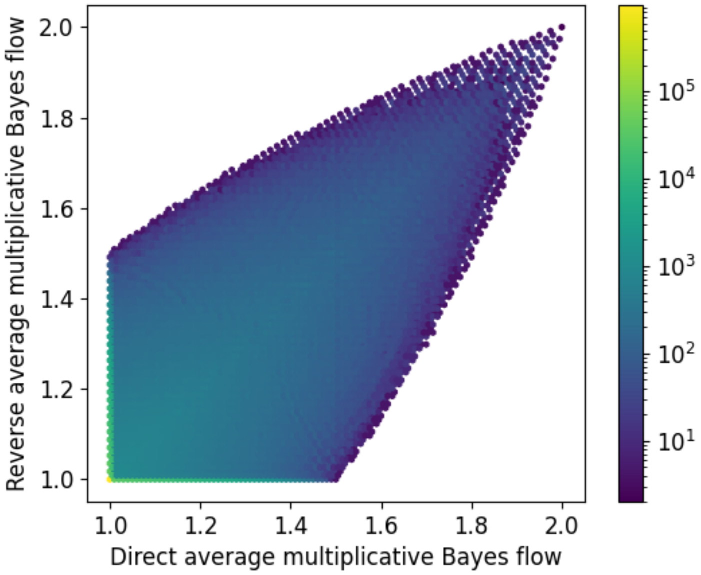
One of the main characteristics of the feasibility regions we describe is that all of them can be arbitrarily close to the point of minimal direct and reverse flows. This means that, under these metrics, there is no trade-off between fairness and privacy. It is always possible to create a classifier that has minimum unfairness and "unprivacy" (not considering utility metrics, such as accuracy).

To show two more characteristics, we need another plot. Figure 6.1 is a heatmap of all possible joints up to 2^{-8} precision and has direct and reverse average multiplicative Bayes flow on both axes. It shows that many points are near the point with minimal flow of information. In fact, around a third of points are in this region. We can also see that the plot is symmetrical in relation to the identity line. This makes sense because both axes are duals that measure the same thing. Both distributions of points are equal and symmetric.

Pareto curves One interesting aspect is what the Pareto bounds mean regarding fairness and privacy. Let us consider the upper part of the curve in the plot of Figures 5.2 and 5.4, the blue part. For every level of direct flow, meaning unfairness, we have the greatest amount of reverse flow possible, that is, "unprivacy". So, the Pareto bound guarantees how bad one aspect can be given the value of the other. Now consider the lower part of the curve, the yellow part. We have the best-case scenario regarding privacy for every level of direct flow. Considering the four curves, we have the complete characterization of the bounds.

Zeros in the joints In the proofs of the feasibility regions, we frequently have used joint distributions without full support. In the context of ML systems, this implies that one of the groups has one classification that never happens. This is extremely unlikely in this context and does not model the scenario well. However, it is still helpful as a tool to show what can happen in other scenarios.

Figure 6.1: Heatmap showing the density of direct and reverse average multiplicative Bayes flow for all joints up to a precision of 2^{-8}.



Source: created by the author.

6.1.3 Interpretation of the metrics

We have discussed the bounds with respect to our new measure of fairness and privacy just using “information flow” as our core concept. However, in the previous chapter, we describe six different metrics. So let us go over them quickly to analyze them in fairness and privacy terms.

Average Bayes flow The average Bayes flow is the most straightforward measure. It computes how much the probability of guessing the group or classification increases when the other value is known. From the feasibility region, we can see that, in general, when one of the measures increases, the other increases, so the more private a system is, the more fair it is and vice versa. Moreover, when fairness or privacy is at its best level, the minimum flow, the other measure is bounded from above to at most half of the range of values, so there is a strong guarantee.

Maximal Bayes flow and capacity Maximal Bayes flow and capacity are slightly more complicated. The maximal Bayes flow measures how much the probability of guessing an individual’s group increases in the worst-case scenario when the other value is known. The capacity measures the greatest gain of a possible adversary when one of the variables is observed.

They measure different things, but the guarantees we have for them are the same: there are no guarantees. All possible combinations of direct and reverse flow are possible in these cases, and, as a consequence, fairness does not bound privacy and vice versa. So, knowing the value of one of them does not give information about the other.

6.2 Modeling existing notions of fairness with QIF

Besides the new modeling of fairness we have just shown, several existing notions of fairness exist in the literature. This section will model some of them using quantitative information flow.

6.2.1 Statistical parity

As we have introduced in Chapter 2, statistical parity is defined as

$$\gamma = |P(+|s_0) - P(+|s_1)|.$$

It is one of the simplest fairness measures and falls under the WAE categorization, defined in subsection 2.3.3.1.

Now, we will show two ways that QIF can model it. First, we will show how the reverse Bayes flow can capture statistical parity exactly. After that, we prove that direct flow can also be equal to statistical parity under the right circumstances.

6.2.1.1 Direct flow and statistical parity

We begin by showing the relation between direct flow and statistical parity.

Theorem 3. *Let $C : \mathcal{X} \rightarrow \mathbb{D}\mathcal{Y}$ be a channel and $\pi : \mathbb{D}\mathcal{X} = (p, 1-p)$ a prior. If the statistical parity of this channel is equal to γ, then the maximal value that the direct average additive Bayes flow of channel C and prior π can attain is*

$$2\gamma p(1-p), \forall p \in [0,1].$$

The maximal value that the direct average multiplicative Bayes flow can assume is

$$4p\gamma - 4p^2\gamma + 1, \forall p \in [0,1].$$

Proof. Let us define the prior distribution of the unprotected and protected groups to be

$$\pi = (p, 1-p).$$

The channel C that receives the group as input and produces a classification is defined as

Table 6.1: Channel C

C	$+$	$-$
s_0	$a+\gamma$	$1-a-\gamma$
s_1	a	$1-a$

With this, we can compute the joint probability J as

Table 6.2: Joint J

J	$+$	$-$
s_0	$p(a+\gamma)$	$p(1-a-\gamma)$
s_1	$(1-p)a$	$(1-p)(1-a)$

Taking the marginal on $\mathcal{Y}$, we have the prior distribution of the classes. We are going to write this as

$$\overrightarrow{\pi} = (a+p\gamma, 1-a-p\gamma).$$

This enables us to compute the direct channel:

Table 6.3: Direct channel $\overrightarrow{C}$

$\overrightarrow{C}$	s_0	s_1
$+$	$\frac{pa+p\gamma}{a+p\gamma}$	$\frac{a-pa}{a+p\gamma}$
$-$	$\frac{p-pa-p\gamma}{1-a-p\gamma}$	$\frac{1-a-p+pa}{1-a-p\gamma}$

The Bayes vulnerability of the direct prior is

$$V(\overrightarrow{\pi}) = \begin{cases} a+p\gamma, & \text{if } a \geq \frac{1-2p\gamma}{2}, \\ 1-a-p\gamma, & \text{if } a < \frac{1-2p\gamma}{2} \end{cases}$$

The posterior Bayes vulnerability of passing the direct prior through the direct channel is

$$V[\overrightarrow{\pi} \triangleright \overrightarrow{C}] = \begin{cases} a+p\gamma, & \text{if } a \geq \frac{1-2\gamma}{2} \text{ and } a \geq \frac{1}{2}, \\ 2pa+p\gamma-p-a+1, & \text{if } a \geq \frac{1-2\gamma}{2} \text{ and } a < \frac{1}{2}, \\ p-p\gamma-2pa+a, & \text{if } a < \frac{1-2p\gamma}{2} \text{ and } a \geq \frac{1}{2}, \\ 1-a-p\gamma, & \text{if } a < \frac{1-2p\gamma}{2} \text{ and } a < \frac{1-2\gamma}{2}, \end{cases}$$

Table 6.4: Values of posterior vulnerability and flow in function of a.

	$a < \frac{1-2\gamma}{2}$	$\frac{1-2\gamma}{2} \leq a < \frac{1-2p\gamma}{2}$	$\frac{1-2p\gamma}{2} \leq a < \frac{1}{2}$	$a \geq \frac{1}{2}$
$V(\overrightarrow{\pi})$	$1-a-p\gamma$	$1-a-p\gamma$	$a+p\gamma$	$a+p\gamma$
$V[\overrightarrow{\pi} \triangleright \overrightarrow{C}]$	$1-a-p\gamma$	$2pa+p\gamma-p-a+1$	$2pa+p\gamma-p-a+1$	$a+p\gamma$
$\mathcal{L}^{+}$	0	$p(2a+2\gamma-1)$	$(1-p)(1-2\gamma)$	0
$\mathcal{L}^{\times}$	1	$\frac{2pa+p\gamma-p-a+1}{1-a-p\gamma}$	$\frac{2pa+p\gamma-p-a+1}{a+p\gamma}$	1

We can split a into intervals and compute the direct Bayes flows. This is shown in table 6.4.

When $\frac{1-2\gamma}{2} \leq a < \frac{1-2p\gamma}{2}$, the derivative of the additive flow with respect to a is $2p$, and this is positive. When $\frac{1-2p\gamma}{2} \leq a < \frac{1}{2}$, the derivative with respect to a is $-2*(1-p)$, and this is negative. Thus, the largest value of flow is achieved when $a = \frac{1-2p\gamma}{2}$, and that is $2\gamma p(1-p)$.

When $\frac{1-2\gamma}{2} \leq a < \frac{1-2p\gamma}{2}$, the derivative of the multiplicative flow with respect to a is $\frac{p(1+2e-2pe)}{(a+pe-1)^2}$. The numerator is greater or equal to zero, and $(a+p\gamma-1)$ is smaller or equal to zero. Thus, the multiplicative flow grows when we increase a in this interval.

When $\frac{1-2p\gamma}{2} \leq a < \frac{1}{2}$, the derivative with respect to a is $\frac{(p-1)(2px+1)}{(a+px)^2}$, and this is negative. Thus, the largest value of flow is achieved when $a = \frac{1-2p\gamma}{2}$, and that is $4p\gamma - 4p^2\gamma + 1$. As we wanted.

□

This shows a clear relationship between our new notion of fairness and one of the existing notions, indicating that our measure is an adequate metric.

These bounds are a function of the statistical parity and the prior distribution. We can change the prior to maximize the flow and get a bound that only depends on the statistical parity of the channel.

Corollary 1. *Let $C : \mathcal{X} \rightarrow \mathbb{D}\mathcal{Y}$ be a channel. If the statistical parity of this channel is equal to γ, the prior is $\pi = (p, 1-p)$ and $a = \frac{1-2p\gamma}{2}$ then the maximal direct average additive Bayes flow is*

$$\frac{\gamma}{2}.$$

And the maximal direct average multiplicative Bayes flow is

$$\gamma + 1.$$

Proof. For the additive case, the derivative of $2\gamma p(1-p)$ with respect to p is $2\gamma - 4\gamma p$. Pluggint the value of a and making it equal to 0, we get $p = \frac{1}{2}$. The value of flow when $p = \frac{1}{2}$ is $\frac{\gamma}{2}$, as we wanted.

In the multiplicative case, the derivative of $4p\gamma - 4p^2\gamma + 1$ with respect to p is $4\gamma - 8p\gamma$. Making it equal to 0, we get $p = \frac{1}{2}$. The value of flow, in this case, is $\gamma + 1$. □

6.2.1.2 Reverse flow and statistical parity

Similarly to what we did concerning direct flow, we can relate reverse flow and statistical parity.

Theorem 4. *Let C be a binary channel and π a prior with full support. Let the statistical parity of the channel be γ. The reverse additive capacity over all gain functions is equal to γ, and the reverse multiplicative capacity is equal to $\gamma + 1$.*

Proof. Let $\pi = (\pi_{s_0}, \pi_{s_1})$. The reverse additive capacity is

$$\begin{aligned}
\mathcal{ML}^{+}(C,\pi) &= 1 - \sum_{y\in\mathcal{Y}} \min_{x\in\mathcal{X}} C_{x,y} \\
&= 1 - \min(P(+|s_0), P(+|s_1)) - \min(P(-|s_0), (-|s_1)) \\
&= 1 - \min(P(+|s_0), P(+|s_1)) - \min(1 - P(+|s_0), 1 - P(+|s_1)) \\
&= 1 - \min(P(+|s_0), P(+|s_1)) - 1 - \min(-P(+|s_0), -P(+|s_1)) \\
&= -\min(P(+|s_0), P(+|s_1)) + \max(P(+|s_0), P(+|s_1)) \\
&= |P(+|s_0) - P(+|s_1)| = \gamma
\end{aligned}$$

The reverse multiplicative capacity is

$$\begin{aligned}
\mathcal{ML}^{\times}(C,\pi) &= \sum_{y\in\mathcal{Y}} \max_{x\in\mathcal{X}} C_{x,y} \\
&= \max(P(+|s_0), P(+|s_1)) + \max(P(-|s_0)), P(-|s_1)) \\
&= \max(P(+|s_0), P(+|s_1)) + \max(1 - P(+|s_0)), 1 - P(+|s_1)) \\
&= \max(P(+|s_0), P(+|s_1)) + 1 + \max(-P(+|s_0)), -P(+|s_1)) \\
&= \max(P(+|s_0), P(+|s_1)) + 1 - \min(P(+|s_0)), P(+|s_1)) \\
&= 1 + \max(P(+|s_0), P(+|s_1)) - \min(P(+|s_0)), P(+|s_1)) \\
&= 1 + |P(+|s_0) - P(+|s_1)| = 1 + \gamma
\end{aligned}$$

As we wanted. □

The capacity is a function of both the channel and the prior, given that we need to know the support of the prior to compute the capacity. But statistical parity is a function of only the channel. This indicates that we may be able to use a function of only the channel to get statistical parity. We do this on the following lemma.

Lemma 8. *Let C be a channel with statistical parity γ. If we push a uniform prior through this channel, the reverse average additive Bayes flow is $\frac{\gamma}{2}$, and the multiplicative one is $\gamma + 1$.*

Proof. The Bayes vulnerability of a binary uniform prior is $\frac{1}{2}$.

Now, let us compute the posterior Bayes vulnerability of this function.

$$\begin{aligned}
V_1[\pi \triangleright C] &= \sum_{y \in \mathcal{Y}} \max_{x \in \mathcal{X}} \pi_x C_{x,y} \\
&= \frac{1}{2} \sum_{y \in \mathcal{Y}} \max_{x \in \mathcal{X}} C_{x,y} \\
&= \frac{1}{2}(\max(P(+|s_0), P(+|s_1)) + \max(P(-|s_0)), P(-|s_1))) \\
&= \frac{1}{2}(1+\gamma)
\end{aligned}$$

The additive flow is

$$\frac{1}{2}(1+\gamma) - \frac{1}{2} = \frac{\gamma}{2}$$

and the multiplicative flow is

$$\frac{\frac{1}{2}(1+\gamma)}{\frac{1}{2}} = 1+\gamma,$$

as we wanted. □

These theorems let us conclude that both flows are closely related and can be used to measure fairness.

6.2.1.3 Binary statistical parity

So far, our measure of statistical parity has been the absolute value of a difference of probabilities. We can also interpret statistical parity as a binary value, that is, whether or not the probabilities are the same. In this way, a channel that satisfies statistical parity is of the form shown in 6.5. It is called the null channel.

Definition 29. *The* null channel *is a channel that has all lines equal. It is of the form represented in channel 6.5.*

Table 6.5: Channel that satisfies statistical parity. a can be any value in $[0, 1]$.

	$+$	$-$
s_0	a	$1-a$
s_1	a	$1-a$

We can then ask if, for every null channel and every prior, the reverse channel also satisfies statistical parity. And the answer is yes. This result is on lemma 9.

Lemma 9. *If a reverse channel C satisfies statistical parity, the direct channel obtained using any prior π also satisfies statistical parity.*

Proof. C is of the form shown in table 6.5. Let $\pi = (p, 1-p)$. Then, the joint is

$$\begin{pmatrix} ap & (1-a)p \\ a(1-p) & (1-a)(1-p) \end{pmatrix}$$

The direct prior is then $\overrightarrow{\pi} = (a, 1-a)$, and the direct channel is in table 6.6.

Table 6.6: Channel that satisfies statistical parity. a can be any value in $[0, 1]$.

	s_0	s_1
$+$	p	$1-p$
$-$	p	$1-p$

This direct channel also satisfies statistical parity, as we wanted. □

So the reverse channel of a null channel is also going to be a null channel. What changes is that the probability distribution for the outputs given an input is the prior π, instead of the prior $(a, 1-a)$.

6.2.2 Other measures

Analogous results to the ones we presented can be shown to work for other fairness metrics. We begin with equal opportunity.

6.2.2.1 Equal opportunity

As we have introduced in the Chapter 2, the definition of equal opportunity is

$$\gamma = |P(\hat{Y} = +|s_0, Y = +) - P(\hat{Y} = +|s_1, Y = +)|.$$

We can build the channel in table 6.7.

Note that this channel and the definition of equal opportunity are equal to their counterparts from the statistical parity section. Thus, if we analyze the dataset with only positive classifications as the ground truth, we conclude that all results valid for statistical parity are also valid for equal opportunity. They are Theorem 3, Corollary 1, Theorem 4, Lemma 8 and Lemma 9.

Table 6.7: channel to analyze equal opportunity.

	$+$	$-$
s_0	$P(\hat{Y}=+\|s_0,Y=+)$	$P(\hat{Y}=-\|s_0,Y=+)$
s_1	$P(\hat{Y}=+\|s_1,Y=+)$	$P(\hat{Y}=-\|s_1,Y=+)$

6.2.2.2 Equalized odds

As mentioned before, equalized odds is usually described as a binary measure where

$$P(\hat{Y}=y|s_0,Y=y)=P(\hat{Y}=y|s_0,Y=y).$$

There are different ways we can enforce this using QIF. We begin by building two channels of the form shown in Figure 6.8.

Table 6.8: channel to analyze equalized odds. y is in the set $\{+,-\}$.

	$+$	$-$
s_0	$P(\hat{Y}=+\|s_0,Y=y)$	$P(\hat{Y}=-\|s_0,Y=y)$
s_1	$P(\hat{Y}=+\|s_1,Y=y)$	$P(\hat{Y}=-\|s_1,Y=y)$

If these two channels are null channels, then equalized odds are satisfied. Otherwise, we must create a measure of how much it is deviating from the fair scenario. The simplest way is to use a similar technique to the one used in statistical parity and sum both of them for the two values of y, that is:

$$\gamma=|P(\hat{Y}=+|s_0,Y=+)-P(\hat{Y}=+|s_1,Y=+)|+|P(\hat{Y}=+|s_0,Y=-)-P(\hat{Y}=+|s_1,Y=-)|.$$

After this, we can also use the results from the previous subsection. Again, they are Theorem 3, Corollary 1, Theorem 4, Lemma 8 and Lemma 9.

6.2.2.3 Conditional statistical parity

The last metric we are going to study is conditional statistical parity. As it was mentioned in the background, for a set of valid features L, conditional statistical parity is satisfied when

$$P(+|s_0,L=\ell)=P(+|s_1,L=\ell),\forall\ell.$$

This is similar to the scenario of equalized odds, where we have multiple conditions to satisfy. The solution is the same: we build one channel for every ℓ, as shown in 6.9.

Table 6.9: channel to analyze conditional statistical parity. ℓ can assume any value that is in the domain of L.

	$+$	$-$
s_0	$P(\hat{Y} = +\|s_0, L = \ell)$	$P(\hat{Y} = -\|s_0, L = \ell)$
s_1	$P(\hat{Y} = +\|s_1, L = \ell)$	$P(\hat{Y} = -\|s_1, L = \ell)$

The problem is that we do not have a fixed number of channels. So, in the case where we do not want conditional statistical parity to be a binary characteristic, we want it to describe how fair a system is from being fair, we must be more careful. We must choose an aggregation function that can deal with multiple existing channels, such as max or average. If we sum all of the deviations

$$|P(+|s_0, L = \ell) - P(+|s_1, L = \ell)|,$$

a scenario with more possible values of ℓ can be described as more unfair, but only because it has more terms. This does not happen with sum or average.

Chapter 7

Experiments

In this chapter, we will measure several metrics presented in this work. The goal is to check how they perform on real-world data.

7.1 The setup

To test our metrics, we assembled four different datasets with different sensitive features and tested them with four different algorithms. In this section, we will briefly describe them.

During the preparation of the dataset, several numerical features had to be transformed into binary features so that we could apply the techniques described in the thesis. In general, we chose to transform all the values smaller than the median of the feature into false and the other into positive. We chose the median so that the proportion of positive and negative classes was close to half, and we would not need to worry about unbalanced classes.

We used a computer system running Ubuntu 20.04.6 LTS. All the intervals shown in the images have a 95% confidence using a two-sided normal distribution. We repeated the experiment 30 times with different random seeds for every combination of parameters. The dataset was divided into 50% for the training data and 50% for the test data randomly for every run. In all executions, the test data was used to estimate the accuracy and the direct and reverse flows of information.

To compute the joint distribution, we used the frequentist approach. That is, the probability of classification i and group j is equal to the fraction of times in the test data that the classification was i and the group was j. This idea was taken from [60].

7.1.1 Datasets

Now, let us briefly present the datasets.

Adult dataset: This dataset contains data from the 1994 American Census. Each row corresponds to a group of the population. The objective is to predict if the average salary of this group is higher than fifty thousand dollars per year. There are 45 thousand rows after removing the ones that have missing entries. Categorical features have been transformed using one-hot encoding [9]. The sensitive features we use are whether the age is higher than the median, whether the person is white, and the sex. Table 7.1 gives some more information about the dataset.

Table 7.1: Information on the adult dataset

Number of rows	45222
Number of columns	96
Positive classifications	22654
Size of first protected group	23027 (age)
Size of second protected group	30527 (sex)
Size of third protected group	38903 (race)

German dataset: This dataset contains data for 1000 German individuals with employment, credit history, housing, and education information. The objective is to predict if each person has a good or bad credit score. The sensitive features we chose are whether the age is above or below the median age and the sex of the individual. It is the only dataset that does not contain information about race, so this will not be used. Table 7.2 describes the main characteristics of the dataset.

Table 7.2: Information on the German credit dataset

Number of rows	1000
Number of columns	48
Positive classifications	700
Size of first protected group	516 (age)
Size of second protected group	452 (sex)

Compas dataset: COMPAS (an acronym for Correctional Offender Management Profiling for Alternative Sanctions) is a tool created by a company called Northpointe that is used in some parts of the United States to predict if a defendant is going to commit crimes after being released from prison. A report made by ProPublica showed that the tool had higher chances of giving a high-risk score to black defendants than white ones [42]. We used the data from the report as a third dataset. It contains information about 3000 defendants after removing missing data. The sensitive features we use are whether the age is higher than the median, whether the person is white, and the sex. Table 7.3 gives some more information about the dataset.

Table 7.3: Information on the COMPAS dataset

Number of rows	3172
Number of columns	12
Positive classifications	2809
Size of first protected group	3164 (age)
Size of second protected group	4997 (sex)
Size of third protected group	2103 (race)

Communities dataset: The communities and crimes dataset contains information about several communities in the United States. There are 123 relevant features, and the objective is to predict whether the crime rate in the community is higher or lower than the median. After removing columns and rows with blank data, 1993 rows, and 100 columns are left. The sensitive features used are whether the percentage of people older than 65 is higher than the median, whether the percentage of white people is higher than the median, and whether the percentage of divorced women is higher than the median. Table 7.4 gives some more information about the dataset.

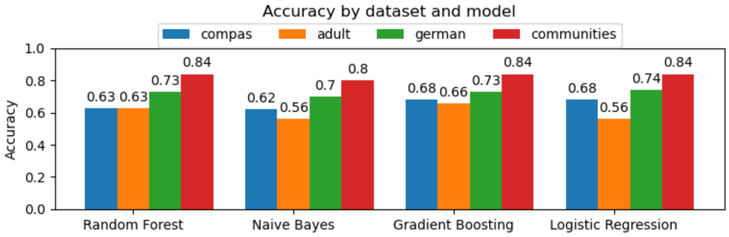
Table 7.4: Information on the communities and crimes dataset

Number of rows	1993
Number of columns	100
Positive classifications	992
Size of first protected group	966 (age)
Size of second protected group	969 (sex)
Size of third protected group	992 (race)

7.1.2 Algorithms

The algorithms we used were the ones described in the Chapter 2. The naive Bayes and logistic regression algorithms represent a relatively simple class of algorithms with few hyperparameters. The random forest and gradient boosting algorithms are more complicated. There are many hyperparameters to be set, and we decided to use the default values implemented in the Scikit-learn library [52]. This library was used to run all experiments.

Figure 7.1: Accuracy for every dataset and every algorithm.



Source: created by the author.

7.2 Results

In this section, we will describe several experiments performed to measure the flow of information under different circumstances. The goal is to measure the effect of the datasets, the algorithms, and the sensitive features in the flow.

There are several possible measures of flow to use. We mainly used average multiplicative Bayes flow because it has a good interpretation, and the feasibility region is more restricted. At the end of the chapter and in Appendix B, we present some plots using different measures.

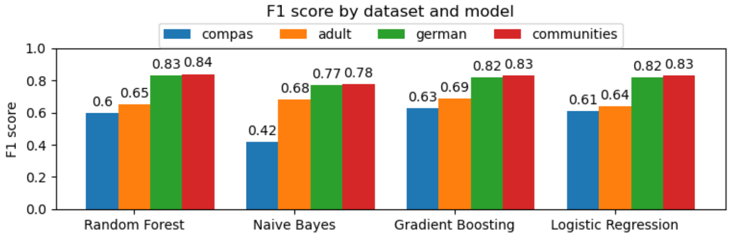
7.2.1 Performance of the algorithms

We begin by testing the accuracy of the different algorithms on different datasets. For every combination of algorithm and dataset, we performed 30 executions and measured the accuracy. The confidence intervals were extremely small, so they did not appear on the images. Figure 7.1 shows the results.

Figure 7.1 shows that the most relevant factor on the accuracy is the dataset, not the algorithm being used. However, some algorithms perform slightly better than the others. Random forest, for example, has accuracy values strictly greater than the accuracy of naive Bayes.

The problem is that accuracy can be misleading. An algorithm that always predicts true in a problem with more true examples has high accuracy, although it is trivial. To avoid this, we will use F1-score, a metric that considers the precision and the recall.

Figure 7.2: F1-score for every dataset and every algorithm



Source: created by the author.

Figure 7.2 shows the results.

Again, the most relevant factor on the performance is the dataset, not the algorithm. However, there are some differences here. The algorithm influences a lot more. The F1-score of naive Bayes on the COMPAS dataset is extremely bad, for example, but not so bad in the rest.

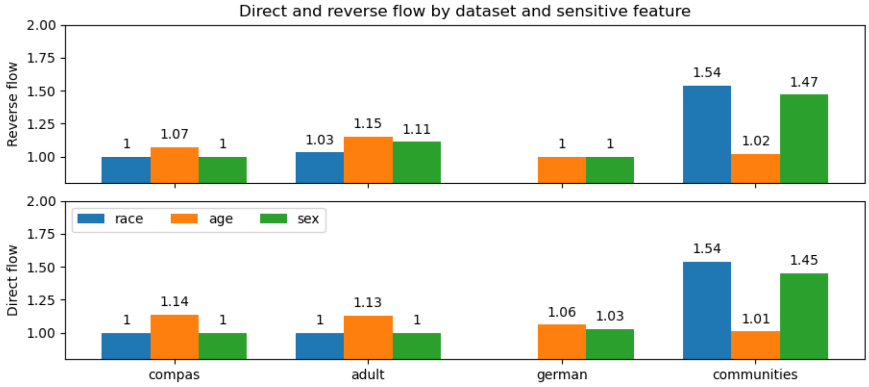
The general result that both plots show is that the algorithms perform well. Random Forest and Gradient Boosting are slightly better than the rest, while naive Bayes performs poorly. This may be because these algorithms have a greater capacity [9] and can be more precise in complicated prediction tasks. The conclusion is that these algorithms may be used in these datasets in real-life scenarios, so we can get valid conclusions from these experiments. If the accuracy or F1-scores were extremely low, these algorithms would not be used, and we would not be able to get any meaningful conclusions from them.

7.2.2 Baseline flow

Before we start measuring the flow of every experiment, there is one baseline we should assess: the flow of information between the sensitive variable and the target classification when no classifier is involved, that is, the flow of the original dataset. Average multiplicative Bayes flow can be measured between any two variables, one of them does not need to be a prediction from an algorithm. So, we can use the correct classification and the sensitive feature to get a baseline for what we should expect from an algorithm trying to make a good prediction. Figure 7.3 shows the results.

We can see that the dataset that flows the most is the communities one, although the flow concerning the age is relatively small. Besides that, the German and COMPAS

Figure 7.3: Direct and reverse average multiplicative Bayes flow in the channel where the input is the sensitive variable and the output is the classification in the original datasets. The dataset is the main factor behind the amount of flow, and the communities dataset is the one with the highest flow. The direct and reverse flows seem to be always close to one another.



Source: created by the author.

datasets flow is also small. The adult dataset is somewhere in the middle. Overall, the direct and reverse flows are quite similar. Indeed, the Pearson correlation between direct and reverse flow is 0.9395.

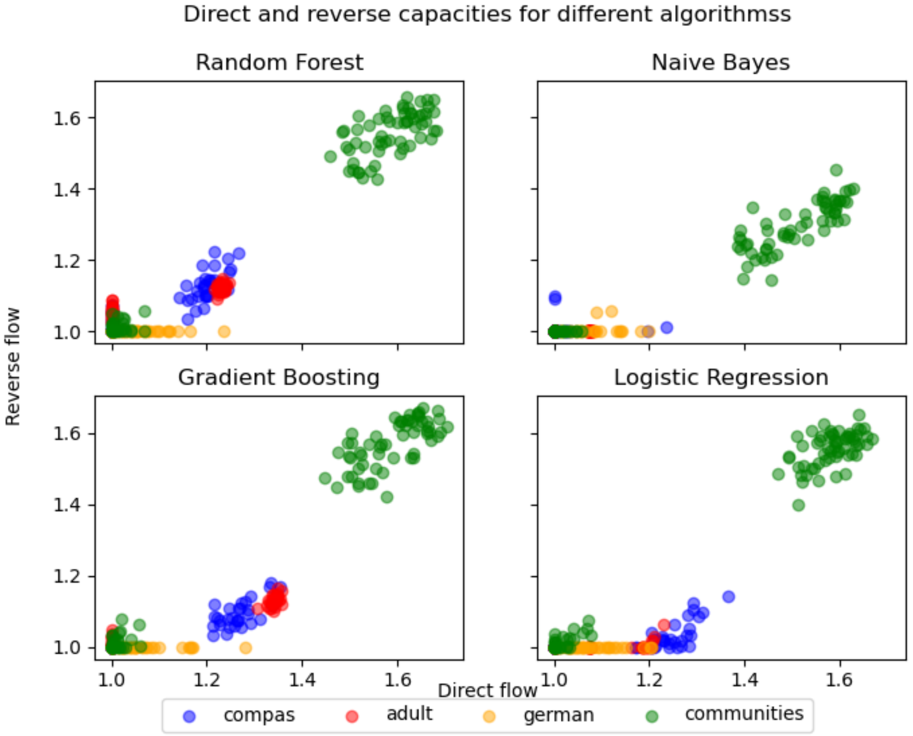
7.2.3 How does the dataset influence the results

Now, we will analyze the effect of different factors on the flow. We are going to begin by studying the datasets. For every dataset and algorithm, we ran 30 executions with different random seeds, such that the training and test datasets were always different. Then, for every sensitive feature, we measure the direct and reverse average multiplicative Bayes flow. Figure 7.4 shows the result in a scatter plot.

We can see that the direct and reverse flows are correlated because most points are near the identity line. There are three main clusters in the image: The first is the green cluster with high flow, which represents the communities dataset. The second cluster is formed by the blue and red points, with low flow but greater than one. The third cluster, near the origin, where several points, mainly from the German dataset, have very low flow. All the points are inside the region delimited by the black line, representing the feasibility region proved in Chapter 5.

Figure 7.5 shows another way to compare the different datasets. For every dataset, we plot the boxplot showing the flow difference in relation to the mean of all measures from all combinations of algorithms and datasets. We see clearly how the communities dataset has large flow and variance, while the German dataset has a pretty consistent

Figure 7.4: Scatter plot showing the different direct and reverse average multiplicative Bayes flow for different datasets. Every point is a different partition of training and test set. The communities dataset has the highest flow, while COMPAS and adult are in the middle, and German has the lowest flow of information. The points are all near the identity line, showing that one flow is highly correlated with the other.



Source: created by the author.

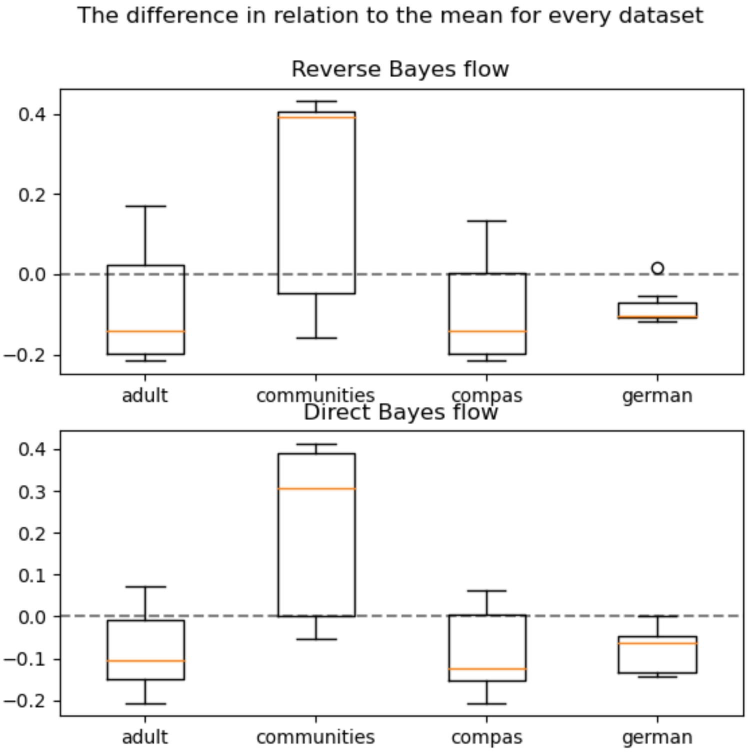
value of flow, mainly in the reverse flow. The COMPAS and adult datasets have lower means but higher variance.

These aggregations of results let us do some interesting analyses, but there is a limitation. We cannot compare how different datasets behave when executed in the same setup, that is, the same algorithm and sensitive feature. We are running all experiments, aggregating them, and then comparing the aggregated results. There may be a situation where the behavior of the datasets is very different, but the aggregate is similar.

So, we perform another experiment. For every pair of datasets, we run experiments with the same algorithm and sensitive feature. Then, we compute the difference between the two datasets' direct and reverse flow values. In the end, we plot every point in a scatter plot.

Figure 7.7 is a scatter plot comparing the difference between the communities and COMPAS datasets. In most cases, the flow on the communities dataset is much higher, but sometimes COMPAS has a higher flow. We could not see these examples in the previous plots. In all cases, the dataset with higher direct flow also has higher reverse flow, one more evidence that these variables are highly correlated. To explain that COMPAS has a lower mean, even though the flow is higher in about one third of the

Figure 7.5: Both boxplots show how the flow of each dataset compares to the mean of all datasets. They are all normalized by the mean of all executions. The communities dataset has the highest variance and mean in both cases. The German dataset has the least variance, but the mean is slightly higher than that of adult and COMPAS. The communities dataset has such a high flow that all others are below the mean.



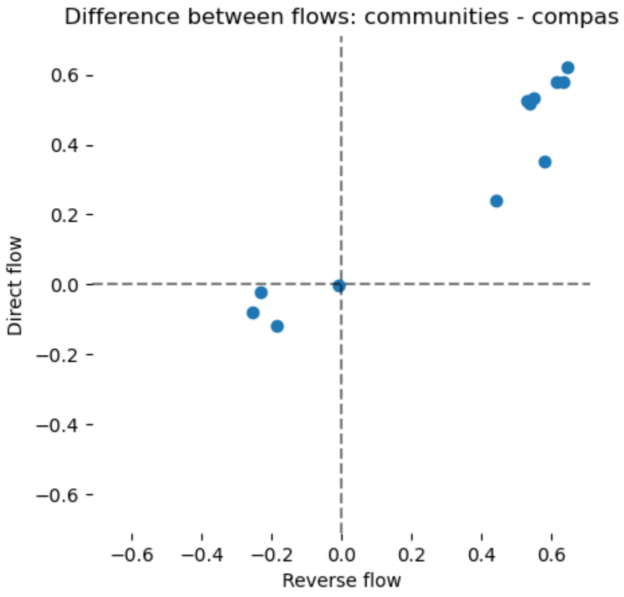
Source: created by the author.

cases, note that when communities has higher flow, it is much higher.

Figure 7.7 is different. The adult and COMPAS datasets are similar, so the difference in flow is much smaller. We can see this by noting how different the values on the axes are on Figures 7.7 and 7.7. While the greatest difference between communities and COMPAS was close to 0.6, the maximal difference between COMPAS and adult was only 0.1, so they are similar. Some examples do not lie where both flows are negative, or both are positive. At first, this may be evidence that the flows are not correlated, but considering that the differences were small, it is not relevant.

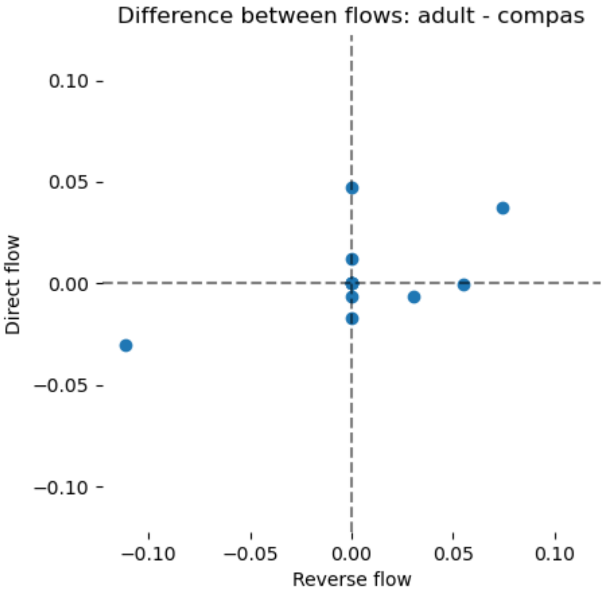
We conclude that some datasets can have more information flow than others. Moreover, when there is a lot of flow in one direction (direct or reverse), the other direction will probably also have high flow.

Figure 7.6: Scatter plot showing the difference in flow between the communities and COMPAS dataset case by case. The input of the channel is the sensitive feature and the output is the prediction of the classifier. In most cases, the communities dataset has a higher flow, but in some cases, the opposite happens. In all cases, the dataset with higher direct flow also has higher reverse flow. When the communities dataset has a higher flow, it is much higher.



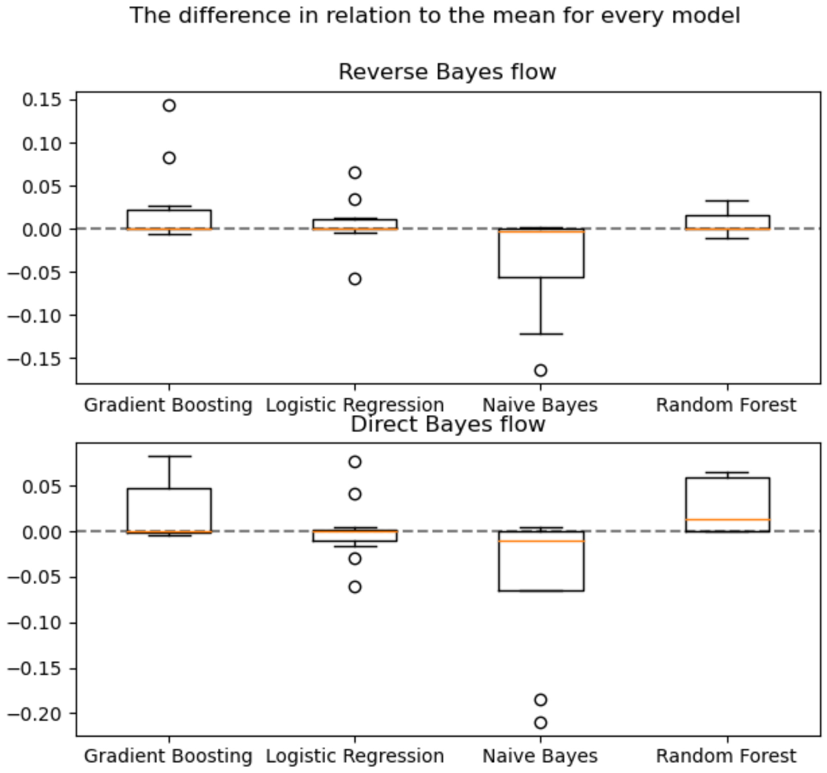
Source: created by the author.

Figure 7.7: Scatter plot showing the difference in flow between the adult and COMPAS dataset case by case. Both datasets' behavior is similar, so most points are near the origin. The points distant from the origin have a high direct and reverse flow difference. There are a few points with one dimension close to zero while the other is larger.



Source: created by the author.

Figure 7.8: Boxplot showing the difference in relation to the mean of different algorithms. The median of all of them is close to zero, meaning that they are all very similar. However, naive Bayes is almost completely below the mean, while Random Forest and Gradient Boosting are above the line. This is true for both direct and reverse flow. Logistic Regression is almost evenly distributed above and below the mean.



Source: created by the author.

7.2.4 How does the algorithm influence the results

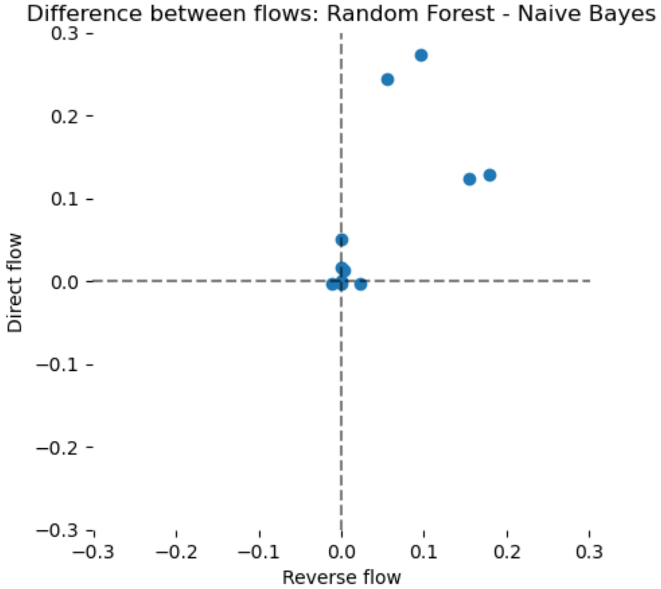
Now, we will do a similar analysis for the different algorithms we tested.

Figure 7.8 is a boxplot showing how the different algorithms deviate from the mean. They all have the median close to zero, meaning the overall mean. This can be attributed to the fact that the dataset, not the algorithm, mainly defines the flow. So, they will have similar values for all datasets and similar values overall. However, the naive Bayes algorithm is below the mean for the rest of the cases, while random forest and gradient boosting are almost all above it. This may be due to their accuracy or higher capacity than the other algorithms.

Now, we will analyze some algorithms pair by pair like we did in the previous section. Again, we will use scatter plots that show how two different algorithms behave when executed in the same scenario.

Figure 7.9 is a scatter plot where each point represents the difference between an

Figure 7.9: Scatter plot showing the difference in flow between the random forest and naive Bayes algorithms case by case. They have similar flows in about half the cases, but random forest has much more flow in the other half. There is no case where naive Bayes has more flow of information.



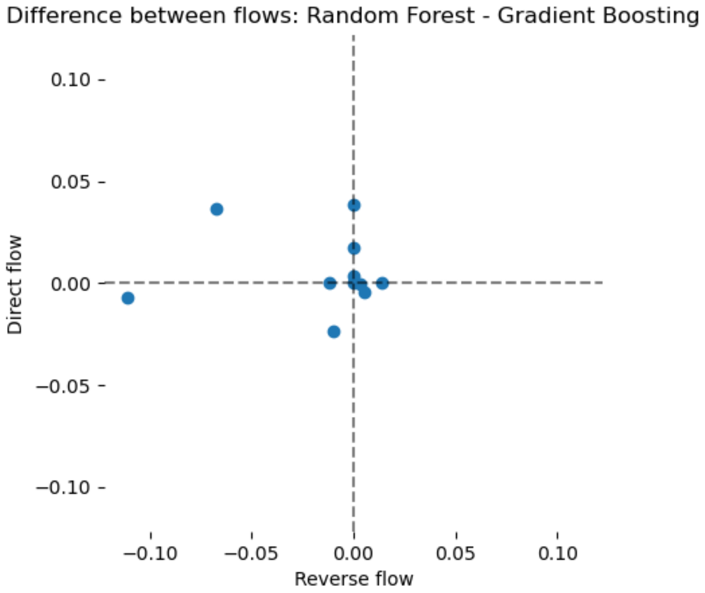
Source: created by the author.

execution with random forest and one with naive Bayes and all else equal. About half of the points are near the origin, showing a similar response. Nevertheless, the other half has a large direct and reverse flow. This shows that the random forest algorithm flows more in some cases. Again, both flows being larger support the idea that they are related. The reason that makes some flows the same and others really different, is because most of the flow is defined by the dataset, not the algorithm. So, when both algorithms deal with a dataset with almost no flow, they will have information flow close to zero. When dealing with a dataset with a large flow of information, random forest can have a much larger information flow than naive Bayes.

Figure 7.10 shows a similar plot but for the case of random forest and gradient boosting. These are similar methods, both relying on ensemble techniques, so their performance is also similar. Both algorithms have similar plots in Figure 7.8, and the scatter plot in Figure 7.10 shows that they have similar behavior in almost all cases, not only in the average.

In conclusion, the algorithms can make the system have larger information flow, thus higher privacy and fairness risks. Algorithms with high capacity, such as gradient boosting and random forest, have more information flow than low-capacity algorithms, such as naive Bayes. However, overall, the effect of the algorithm is smaller than the effect of the dataset.

Figure 7.10: Scatter plot showing the difference in flow between the random forest and gradient boosting algorithms case by case. Most points are close to the origin, showing that the algorithms are similar. There is a point where the reverse flow is significantly larger than the direct one. In this point, gradient boosting is the algorithm with higher flow.



Source: created by the author.

7.2.5 How does the sensitive feature influence the results

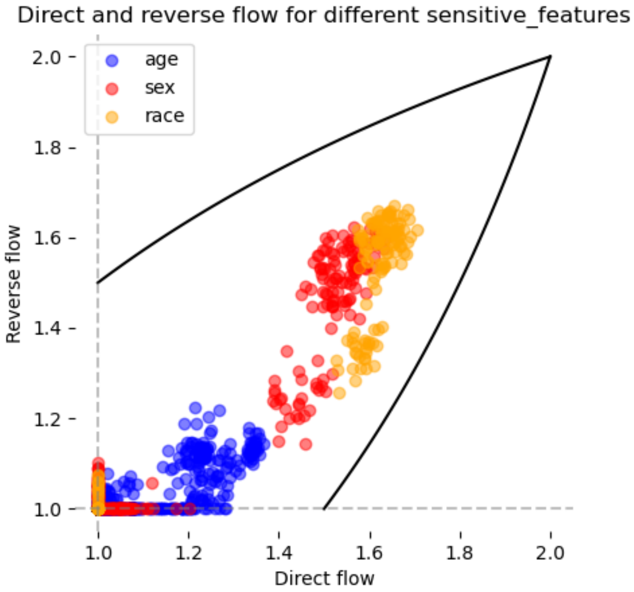
Now, the last analysis of what affects the flow value we will make is regarding the sensitive feature.

Figure 7.11 shows a scatter plot discerning different experiments by the sensitive feature. A large cluster near the origin represents that experiments with all features have had a small flow in some settings. There are three other clusters: one for age, one for sex, and one for race. This may show that the sensitive feature is extremely relevant to the flow, but if we compare this plot with Figure 7.4, we see that the clusters of sex and race, the ones with higher flow, correspond to the cluster of the communities dataset.

From this, the effect of the sensitive feature depends on the dataset, so there is little we can explore because every dataset can have a feature that will flow more information than the other.

In short, comparing different sensitive features is similar to comparing different datasets.

Figure 7.11: Scatter plot showing different flow values for different sensitive features. Several examples are near the origin of all three sensitive features, but there is a cluster of points with slightly higher flow with age as the sensitive feature. Higher on the plot, two other clusters represent sex and race.



Source: created by the author.

7.2.6 Comparison with statistical parity

Another aspect of our metrics we have to check is how they compare to existing metrics.

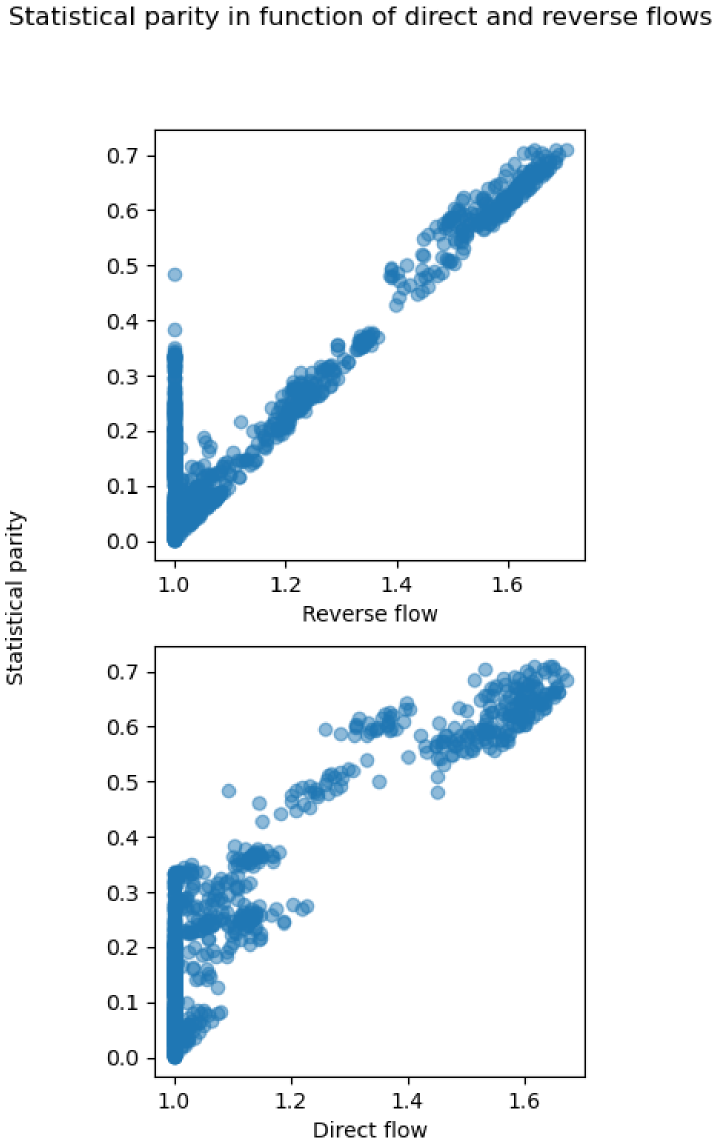
Some of the existing metrics are qualitative, that is, they can only be satisfied or not satisfied, they do not indicate how far they are from being satisfied. One measure that is different from this is statistical parity. It can give a value on how fair the algorithm is. The lower, the better. So, we have decided to compare statistical parity with direct and reverse flow.

Figure 7.12 shows two scatter plots comparing the value of statistical parity and direct and reverse flow. In both cases, the flow and statistical parity are correlated. The correlations between the variables are high. They are shown in table 7.2.6. This shows that the average multiplicative Bayes flow is a reasonable fairness metric.

Variables	**Correlation**
Direct and reverse flow	0.9395
Statistical parity and reverse flow	0.9257
Statistical parity and direct flow	0.9110

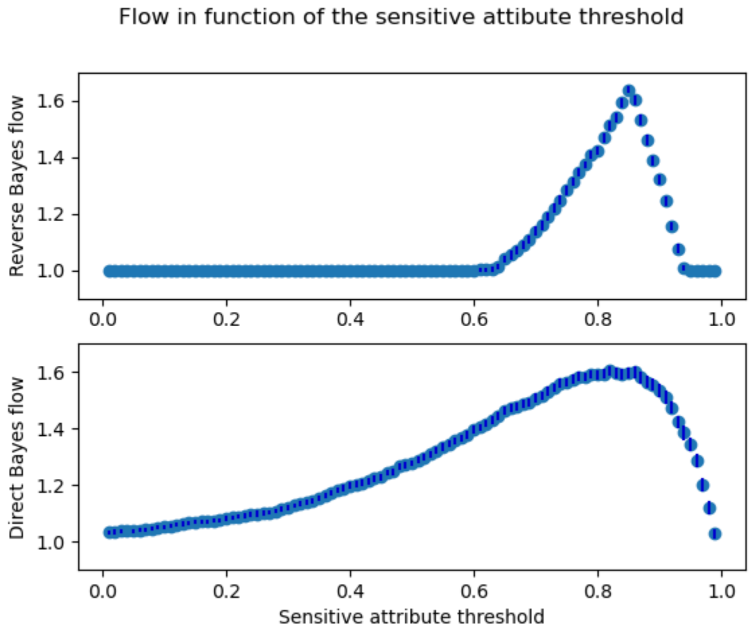
The fact that the correlation is higher between statistical parity and reverse flow,

Figure 7.12: Two scatter plots show the relation between direct and reverse flows with statistical parity. On both of them, the greater the flow, the greater the statistical parity.



Source: created by the author.

Figure 7.13: Two scatter plots showing how direct and reverse flow change when the threshold for the sensitive attribute changes. When it approaches the median, 0.85, in this case, both flows increase significantly. The reverse flow of points with the threshold far from the median goes to the minimum possible value, 1. The direct flow is above zero for most points but peaks near the median. The lines on top of the points are confidence intervals. A point with no dot represents an experiment where all the results were equal.



Source: created by the author.

instead of direct flow, is another argument to say that statistical parity is, in fact, a privacy metric, not a fairness one. However, the correlation between reverse flow and statistical parity above 0.9 shows that it can also be used as a fairness measure.

Nevertheless, there are some problems. On the left of both plots of Figure 7.12, there are points with high statistical parity but with no flow. In the following subsection, we will investigate why this can happen.

7.2.7 Limitations of average Bayes flow

Consider the communities dataset. To check if a community was in the protected group, we checked if the percentage of white people was below 85%, the feature's median. However, we do not need to split the groups in the median. We can define the protected group as communities with less than 70% of white people, for example. Nevertheless, this threshold affects the amount of information that flows through the system. To study this, we measured it for 100 different thresholds. The result is in Figure 7.13.

Note that when the threshold is near the median, both flows increase until they reach their maximum value. However, when the threshold is far from the median, the reverse flow reaches the minimum possible value, and the direct flow decreases significantly.

To explain why this happens, consider the 2×2 joint depicted below:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}.$$

Suppose, without loss of generality, that $c+d \geq a+b$. This makes the prior Bayes vulnerability equal to $c+d$. Now, consider the posterior Bayes vulnerability. It is equal to $\max(a,c)+\max(b,d)$. If both $c \geq a$ and $d \geq b$, then $\max(a,c)+\max(b,d) = c+d$ and the posterior is equal to the prior vulnerability. To make the posterior bigger than the prior, we need to have either $a > c$ or $b > d$. However, if the classes are not well distributed, for example, if the threshold is far from the median, this will not happen because most of the probability mass will be on c and d. So, if the protected and unprotected groups are not roughly the same size, the prior and posterior vulnerabilities will be the same, and the reverse flow will be minimal.

But this only affects the reverse flow a little. The prior reverse Bayes vulnerability is $\max(a+c, b+d)$ and the posterior is $\max(a,b)+\max(c,d)$. In the scenario of most of the probability mass being on c and d, this will not affect any of these terms a lot. The only difference is that $\max(c,d)$ will probably be bigger than $\max(a,b)$, but this does not change the flow a lot.

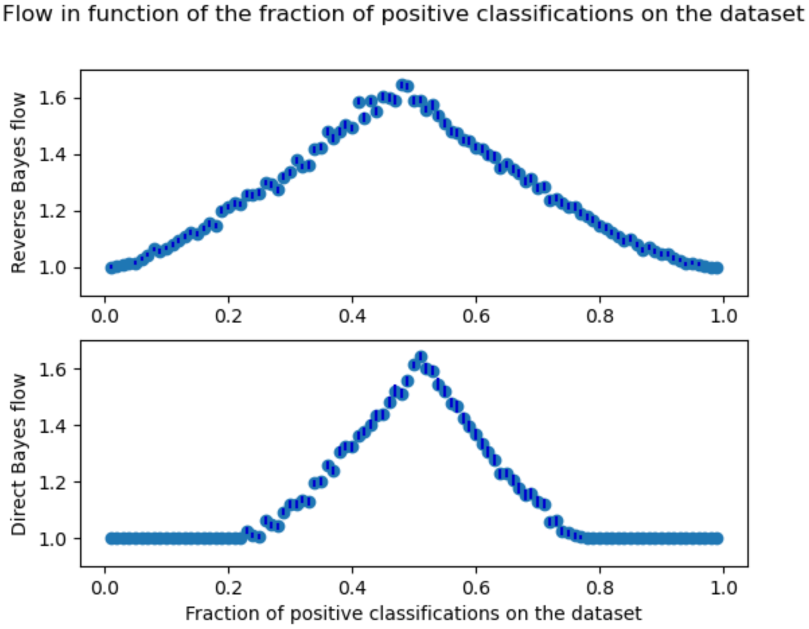
In this explanation, we have supposed that the rows of this joint distribution correspond to the protected or unprotected groups, and the columns are the classification. If we flip this, we will have a similar effect. Suppose one of the classes is much larger than the other. Then, the prior Bayes vulnerability will be close to the posterior, and the flow will be small.

To test this, we performed 100 experiments. On the i-th experiment, some rows of the dataset were removed such that the probability of someone being on the positive class is equal to $i\%$. The results are in Figure 7.14.

The result is exactly as we predicted. When the fraction of positive examples is close to 0.5, the flow is maximal in both directions. When it gets close to 0 or 1, the direct flow goes to zero, while the reverse flow decreases significantly.

The conclusion is that the average Bayes flow can be used as a fairness and privacy metric. However, it works best when the protected and unprotected groups are roughly the same size, as are the positive and negative classes.

Figure 7.14: Two scatter plots showing how the direct and reverse flow change when the fraction of positive classifications in the dataset changes. In both cases, the flow increases when the fraction is close to zero. The direct case is zero on multiple scenarios where the fraction is close to 0 or 1. The reverse flow is almost always greater than 1 but approaches 1 when the classes are unbalanced. The lines on top of the points are confidence intervals. A point with no dot represents an experiment where all the results were equal.



Source: created by the author.

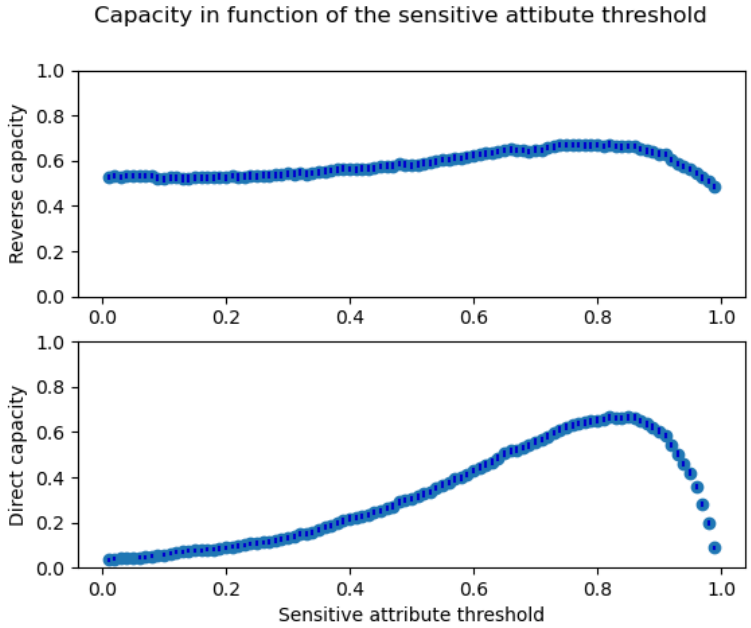
7.2.8 Using capacity

The problem with using Bayes flow is that it uses the joint instead of the channel. When the joint is unbalanced, the Bayes flow approaches 1 in the multiplicative case and 0 in the additive one, the smallest values possible. So, one way to avoid this is to use the capacity. The capacity of a channel uses only the channel instead of the joint (prior and channel). So, there will not be a problem with the classes being unbalanced. To show this, Figure 7.15 is analogous to Figure 7.13, but it uses capacity instead of Bayes flow.

Figure 7.15 shows the capacity when the sensitive attribute threshold changes. The reverse capacity stays approximately the same because what changes is the fraction of examples that are in each group, protected or unprotected. The fraction of individuals getting positive or negative classifications inside these groups is the same. The capacity takes into account only this fraction, not the absolute number. Because this fraction is near constant, the capacity is near constant. This does not happen in the reverse capacity, so the behavior is similar to the Bayes flow scenario.

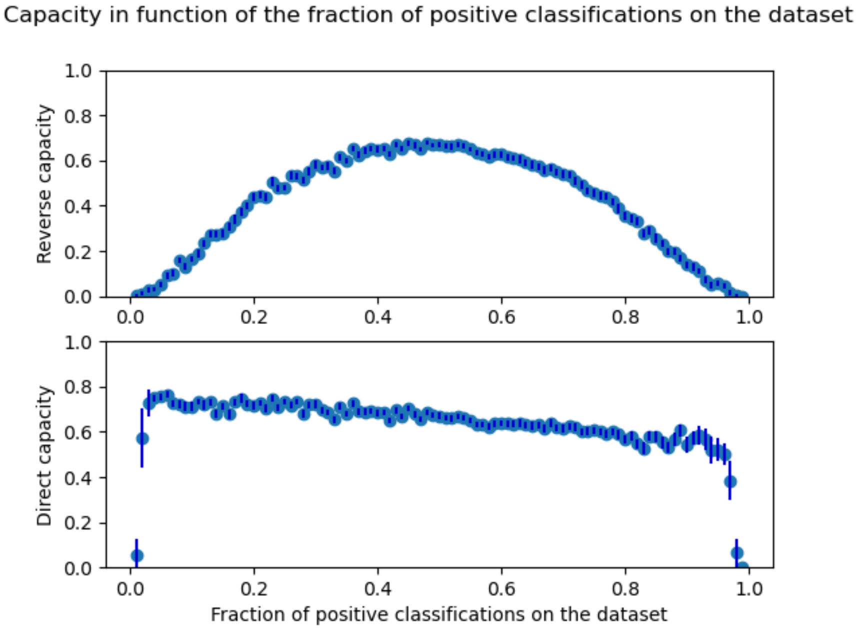
Figure 7.16 shows an analogous plot to Figure 7.14. Because we are changing the fraction of positive classifications in the dataset, the reverse capacity is close to constant.

Figure 7.15: Two scatter plots showing how direct and reverse capacity change when the threshold for the sensitive attribute changes. The reverse capacity stays roughly the same through all thresholds. The direct capacity increases when it is close to the median of the sensitive attribute. The lines on top of the points are confidence intervals.



Source: created by the author.

Figure 7.16: Two scatter plots showing how direct and reverse capacity change when the fraction of positive classifications change. The direct capacity stays approximately the same through all fractions. The reverse capacity increases when the fraction is close to 0.5 and goes to zero when it approaches 0 or 1. The lines on top of the points are confidence intervals. Some examples near the border have very large errors because the sample size is not big enough, making the results unstable.



Source: created by the author.

The reverse capacity behaves similarly to the reverse flow, although it is smoother.

This shows that using the capacity in cases where the classes or groups are not well-balanced works well. Appendix B has a few plots showing the capacity instead of Bayes flow in the plots shown in this chapter.

In conclusion, capacity can capture the same thing as Bayes flow, but its usage is not constrained to a situation where classes and groups are balanced. But Bayes flow can be a good measure in a scenario where we want to know how much information is gained about the input when the output is observed in a specific population.

Chapter 8

Conclusion

As was said in previous chapters, the widespread use of machine learning algorithms in sensitive contexts requires that these algorithms are safe from two perspectives. They have to be fair and adequately treat different groups. Furthermore, they must be private and protect the information about individuals fed to the system. This work has studied these two problems simultaneously.

We begin by expanding the quantitative information flow framework. Although this has been done in previous works, we formalized it and derived theoretical bounds in this framework. We showed that not every combination of direct and reverse average Bayes flow can happen. However, maximal Bayes flow and capacity with respect to gain functions allow for every value of direct and reverse flows.

Then, we used this framework and these results to model fairness and privacy in machine learning. We first interpreted the theoretical bounds from the perspective of fairness and privacy. Then, we showed that both flows can capture existing notions of fairness, such as statistical parity, equalized odds, equal opportunity, and conditional statistical parity.

Finally, we created experiments that indicate that our new metrics make sense when we run them in real-world datasets and standard machine learning algorithms.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[2] Sushant Agarwal. Trade-offs between fairness, interpretability, and privacy in machine learning. Master's thesis, University of Waterloo, 2020.

[3] Mário S Alvim, Konstantinos Chatzikokolakis, Annabelle McIver, Carroll Morgan, Catuscia Palamidessi, and Geoffrey Smith. *The Science of Quantitative Information Flow.* Springer, 2020.

[4] Mário S Alvim, Andre Scedrov, and Fred B Schneider. When not all bits are equal: Worth-based information flow. In *POST*, pages 120–139, 2014.

[5] UN General Assembly et al. Universal declaration of human rights. *UN General Assembly*, 302(2):14–25, 1948.

[6] Haniel Barbosa, Clark Barrett, Martin Brain, Gereon Kremer, Hanna Lachnitt, Makai Mann, Abdalrhman Mohamed, Mudathir Mohamed, Aina Niemetz, Andres Nötzli, et al. cvc5: A versatile and industrial-strength smt solver. In *Tools and Algorithms for the Construction and Analysis of Systems: 28th International Conference, TACAS 2022, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2022, Munich, Germany, April 2–7, 2022, Proceedings, Part I*, pages 415–442. Springer, 2022.

[7] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research*, 50(1):3–44, 2021.

[8] DR Bhandari. Plato's concept of justice: An analysis. In *The Paideia Archive: Twentieth World Congress of Philosophy*, volume 3, pages 44–47, 1998.

[9] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[10] Dan Bogdanov, Liina Kamm, Sven Laur, and Ville Sokk. Implementation and evaluation of an algorithm for cryptographically private principal component analysis on

genomic data. *IEEE/ACM transactions on computational biology and bioinformatics*, 15(5):1427–1432, 2018.

[11] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29, 2016.

[12] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR, 2018.

[13] United States Census Bureaus. Race and ethnicity in the united states: 2010 census and 2020 census. *United States Census Bureaus Library*, 2021.

[14] Toon Calders and Sicco Verwer. Three naive bayes approaches for discrimination-free classification. *Data mining and knowledge discovery*, 21(2):277–292, 2010.

[15] L Elisa Celis, Damian Straszak, and Nisheeth K Vishnoi. Ranking with fairness constraints. *arXiv preprint arXiv:1704.06840*, 2017.

[16] Irene Chen, Fredrik D Johansson, and David Sontag. Why is my classifier discriminatory? *Advances in neural information processing systems*, 31, 2018.

[17] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. Algorithmic decision making and the cost of fairness. In *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining*, pages 797–806, 2017.

[18] Rachel Cummings, Varun Gupta, Dhamma Kimpara, and Jamie Morgenstern. On the compatibility of privacy and fairness. In *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*, pages 309–315, 2019.

[19] Fida Kamal Dankar and Khaled El Emam. Practicing differential privacy in health care: A review. *Trans. Data Priv.*, 6(1):35–67, 2013.

[20] Akash Dhasade, Nevena Dresevic, Anne-Marie Kermarrec, and Rafael Pires. Tee-based decentralized recommender systems: The raw data sharing redemption. *arXiv preprint arXiv:2202.11655*, 2022.

[21] Shayan Doroudi, Philip S Thomas, and Emma Brunskill. Importance sampling for fair policy selection. *Grantee Submission*, 2017.

[22] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012.

[23] Cynthia Dwork and Christina Ilvento. Fairness under composition. *arXiv preprint arXiv:1806.06122*, 2018.

[24] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[25] Danielle Ensign, Sorelle A Friedler, Scott Neville, Carlos Scheidegger, and Suresh Venkatasubramanian. Runaway feedback loops in predictive policing. In *Conference on fairness, accountability and transparency*, pages 160–171. PMLR, 2018.

[26] Zekeriya Erkin, Thijs Veugen, Tomas Toft, and Reginald L Lagendijk. Generating private recommendations efficiently using homomorphic encryption and data packing. *IEEE transactions on information forensics and security*, 7(3):1053–1066, 2012.

[27] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067, 2014.

[28] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015.

[29] Sorelle A Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. On the (im) possibility of fairness. *arXiv preprint arXiv:1609.07236*, 2016.

[30] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems*, 33:16937–16947, 2020.

[31] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[32] Mehmet Emre Gursoy, Ali Inan, Mehmet Ercan Nergiz, and Yucel Saygin. Differentially private nearest neighbor classification. *Data Mining and Knowledge Discovery*, 31(5):1544–1575, 2017.

[33] David J Hand and Keming Yu. Idiot's bayes—not so stupid after all? *International statistical review*, 69(3):385–398, 2001.

[34] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

[35] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

[36] Muneeb Ul Hassan, Mubashir Husain Rehmani, and Jinjun Chen. Differential privacy techniques for cyber physical systems: a survey. *IEEE Communications Surveys & Tutorials*, 22(1):746–789, 2019.

[37] István Hegedűs, Gábor Danner, and Márk Jelasity. Decentralized learning works: An empirical comparison of gossip learning and federated learning. *Journal of Parallel and Distributed Computing*, 148:109–124, 2021.

[38] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[39] Hailong Hu and Jun Pang. Membership inference attacks against gans by leveraging over-representation regions. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 2387–2389, 2021.

[40] Dzmitry Huba, John Nguyen, Kshitiz Malik, Ruiyu Zhu, Mike Rabbat, Ashkan Yousefpour, Carole-Jean Wu, Hongyuan Zhan, Pavel Ustinov, Harish Srinivas, Kaikai Wang, Anthony Shoumikhin, Jesik Min, and Mani Malek. Papaya: Practical, private, and scalable federated learning. In D. Marculescu, Y. Chi, and C. Wu, editors, *Proceedings of Machine Learning and Systems*, volume 4, pages 814–832, 2022.

[41] Shahin Jabbari, Matthew Joseph, Michael Kearns, Jamie Morgenstern, and Aaron Roth. Fairness in reinforcement learning. In *International conference on machine learning*, pages 1617–1626. PMLR, 2017.

[42] Surya Mattu Julia Angwin, Jeff Larson and Lauren Kirchner. Machine bias. *ProPublica*, 2016.

[43] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International Conference on Machine Learning*, pages 2564–2572. PMLR, 2018.

[44] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[45] Lydia T Liu, Sarah Dean, Esther Rolf, Max Simchowitz, and Moritz Hardt. Delayed impact of fair machine learning. In *International Conference on Machine Learning*, pages 3150–3158. PMLR, 2018.

[46] Google LLC. Tensorflow privacy. https://github.com/tensorflow/privacy, 2020.

[47] Kristian Lum and William Isaac. To predict and serve? *Significance*, 13(5):14–19, 2016.

[48] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[49] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.

[50] Tom Michael Mitchell et al. *Machine learning*, volume 1. McGraw-hill New York, 2007.

[51] Olga Ohrimenko, Felix Schuster, Cédric Fournet, Aastha Mehta, Sebastian Nowozin, Kapil Vaswani, and Manuel Costa. Oblivious {Multi-Party} machine learning on trusted processors. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 619–636, 2016.

[52] 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.

[53] Eunício Prudente. Dados do ibge mostram que 54% da população brasileira é negra. *Jornal da USP*, 2020.

[54] John Rawls. Justice as fairness: Political not metaphysical. In *Equality and liberty*, pages 145–173. Springer, 1991.

[55] Sheldon M Ross. *A first course in probability.* Pearson London, 2014.

[56] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

[57] Camelia Simoiu, Sam Corbett-Davies, and Sharad Goel. The problem of inframarginality in outcome tests for discrimination. *The Annals of Applied Statistics*, 11(3):1193–1216, 2017.

[58] Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. *arXiv preprint arXiv:2003.10595*, 2020.

[59] Abdel Aziz Taha and Allan Hanbury. Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool. *BMC medical imaging*, 15(1):1–28, 2015.

[60] Thiago Vieira. *On the relationship of privacy and fairness in Machine Learning*. PhD thesis, Universidade Federal de Minas Gerais, 2019. unpublished thesis.

[61] Teng Wang, Xuefeng Zhang, Jingyu Feng, and Xinyu Yang. A comprehensive survey on local differential privacy toward data statistics and analysis. *Sensors*, 20(24):7030, 2020.

[62] Depeng Xu, Shuhan Yuan, and Xintao Wu. Achieving differential privacy and fairness in logistic regression. In *Companion Proceedings of The 2019 World Wide Web Conference*, WWW '19, page 594–599, New York, NY, USA, 2019. Association for Computing Machinery.

[63] Jian Xu and Shao-Lun Huang. Byzantine-resilient decentralized collaborative learning. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5253–5257. IEEE, 2022.

[64] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th international conference on world wide web*, pages 1171–1180, 2017.

[65] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International conference on machine learning*, pages 325–333. PMLR, 2013.

[66] Ying Zhao and Jinjun Chen. A survey on differential privacy for unstructured data content. *ACM Computing Surveys (CSUR)*, 54(10s):1–28, 2022.

[67] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[68] Indrė Žliobaitė. On the relation between accuracy and fairness in binary classification. In *The 2nd Workshop on Fairness, Accountability, and Transparency in Machine Learning (FAT/ML 2015) workshop at ICML*, volume 15, 2015.

Appendix A

Proofs of the theoretical bounds

A.1 Appendix

In this appendix, we are going to prove some of the results shown in the main text.

Multiple times, we are going to use SMT solvers to prove several small statements at once. The SMT solver we used is cvc5 [6] and we are going to show the snippets of code used to prove the results shown.

A.1.1 Proof for the average multiplicative Bayes flow

We begin the proof of the feasibility region of the average multiplicative Bayes region by showing a joint distribution that can be modified to have every value of direct and reverse flow. We will use the fact that α is the reverse flow and β is the direct flow, in the same way as section 5.1.1. Because we are trying to prove only that values in the feasible region can be achieved, we will only consider these values.

Lemma 10. *The joint*

$$\pi \triangleright C = J = \begin{pmatrix} \frac{\alpha\beta+\alpha-\beta}{\alpha\beta+\alpha+\beta} & 0 \\ \frac{-\alpha\beta+\alpha+\beta}{\alpha\beta+\alpha+\beta} & \frac{\alpha\beta-\alpha+\beta}{\alpha\beta+\alpha+\beta} \end{pmatrix}$$

has reverse average multiplicative Bayes flow equal to α and direct average multiplicative Bayes flow equal to β for every α, β in the set

$$\{(\alpha,\beta)|\frac{\beta}{3-\beta} \leq \alpha \leq \frac{3\beta}{\beta+1}, 1 \leq \alpha, \beta \leq 2\}$$

Proof. The reverse flow is

$$\mathcal{L}^{\times}_{id}(\pi, C) = \frac{\max(a,c)+\max(b,d)}{\max(a+b,c+d)}.$$

The maximum between a and c is always a, because $\alpha\beta \geq \beta$, given that $1 \leq \alpha, \beta \leq 2$.The maximum between b and d is always d, because $b = 0$. To prove that $\max(a+b, c+d) = c+d$, consider

$$(c+d) - (a+b) = \frac{3\beta - \alpha\beta - \alpha}{\alpha\beta + \alpha + \beta}.$$

Now, using an SMT solver, we can check if this can be smaller than zero in the viable region.

```
alpha, beta = Reals('alpha beta')
s = Solver()
#givens
s.add(alpha >= 1, alpha <= 2, beta >= 1, beta <= 2)
s.add(beta/(3-beta) <= alpha, alpha <= 3*beta/(beta+1))
s.add(alpha/(3-alpha) <= beta, beta <= 3*alpha/(alpha+1))
#by contradiction suppose that
s.add(3*beta - alpha - alpha*beta < 0)
print(s.check())
```

The output of the code is `unsat`, so this is always larger than zero. So $c+d >= a+b$ and

$$\mathcal{L}^{\times}_{id}(\pi, C) = \frac{\max(a,c) + \max(b,d)}{\max(a+b, c+d)} = \frac{a+d}{c+d} = \frac{2\alpha\beta}{2\beta} = \alpha.$$

So the reverse flow is equal to α, as wanted.

The direct flow is

$$\mathcal{L}^{\times}(J^T) = \frac{\max(a,b) + \max(c,d)}{\max(a+c, b+d)}.$$

The maximum between a and b is a, because $b = 0$. The maximum between c and d is always d because $\alpha\beta > \beta$. To prove that $\max(a+c, b+d) = a+c$, consider that

$$(a+c) - (b+d) = \frac{3\alpha - \alpha\beta - \beta}{\alpha\beta + \alpha + \beta}.$$

The proof is exactly the same as in the previous case. So,

$$\mathcal{L}^{\times}_{id}(\pi, C) = \frac{\max(a,b) + \max(c,d)}{\max(a+c, b+d)} = \frac{a+d}{a+c} = \frac{2\alpha\beta}{2\alpha} = \beta.$$

The only thing left is to prove that J is in fact a joint probability matrix. The first part is straightforward:

$$a+b+c+d = \frac{\alpha\beta + \alpha - \beta}{\alpha\beta + \alpha + \beta} + 0 + \frac{-\alpha\beta + \alpha + \beta}{\alpha\beta + \alpha + \beta} + \frac{\alpha\beta - \alpha + \beta}{\alpha\beta + \alpha + \beta} = \frac{\alpha\beta + \alpha + \beta}{\alpha\beta + \alpha + \beta} = 1.$$

To prove that $a \geq 0$, consider only the numerator

$$\alpha\beta + \alpha - \beta = \alpha + \beta(\alpha - 1) \geq \beta(\alpha - 1) \geq \beta \geq 0.$$

The same argument shows that $d \geq 0$. Now, consider the denominator of c:

```
alpha, beta = Reals('alpha beta')
s = Solver()
#givens
s.add(alpha >= 1, alpha <= 2, beta >= 1, beta <= 2)
s.add(beta/(3-beta) <= alpha, alpha <= 3*beta/(beta+1))
s.add(alpha/(3-alpha) <= beta, beta <= 3*alpha/(alpha+1))

#by contradiction, suppose that
s.add(alpha + beta - alpha*beta < 0)

print(s.check())
```

The output is also `unsat`, so the element c is also non-negative.

The result is a joint with reverse flow α and direct flow β. □

Now, let us prove another lemma.

Lemma 11. *Any point that is not in the set*

$$\{(\alpha,\beta)|\frac{\beta}{3-\beta}\leq\alpha\leq\frac{3\beta}{\beta+1},1\leq\alpha,\beta\leq 2\}$$

can not be achieved as a pair of direct and reverse flow for average multiplicative Bayes flow.

Proof. We begin by noting that, according to [3], the posterior vulnerability is never smaller than the prior vulnerability, so the flow is at least one. By the definition of Bayes vulnerability, the smallest value possible for the prior is $\frac{1}{2}$ for a binary channel, in the case of a uniform prior, and the greatest value for the posterior is 1, when it is guaranteed that the observer can guess the secret correctly. So the maximum value for the flow is 2. This proves that $1\leq\alpha,\beta\leq 2$.

Using the fact that $2\max(a+b) = a+b+|a-b|$, it is possible to rewrite the definition of reverse flow as

$$\alpha=\frac{a+c+|a-c|+b+d+|b-d|}{a+b+c+d+|(a+b)-(c+d)|}$$

a,b,c,d form a distribution, so their sum is 1, and the previous equation can be simplified to

$$\alpha=\frac{1+|a-c|+|b-d|}{1+|(a+b)-(c+d)|}$$

Because flow is invariant to transpositions of rows and columns, we can write, without loss of generality, that $a+b\geq c+d$. Thus,

$$\alpha=\frac{1+|a-c|+|b-d|}{1+(a+b)-(c+d)},$$

$$\alpha = \frac{1 + |a - c| + |b - d|}{2(a + b)}.$$

Now we can break these absolute values into cases.

$$1 + |a - c| + |b - d| = \begin{cases} 1 + a - c + b - d & \text{if } a \geq c \text{ and } b \geq d \\ 1 + a - c - b + d & \text{if } a \geq c \text{ and } b < d \\ 1 - a + c + b - d & \text{if } a < c \text{ and } b \geq d \\ 1 - a + c - b + d & \text{if } a < c \text{ and } b < d \end{cases}$$

The last of these cases can never happen because $a + b \geq c + d$. The remaining cases can be written as a max:

$$1 + |a - c| + |b - d| = \max(1 + a - c + b - d, 1 + a - c - b + d, 1 - a + c + b - d).$$

Using the fact that $a + b + c + d = 1$, we can write

$$1 + |a - c| + |b - d| = 2\max(a + b, a + d, c + b).$$

$$\alpha = \frac{\max(a + b, a + d, c + b)}{a + b}$$

Analogously, we can say without loss of generality that $a + c \geq b + d$ and conclude

$$\beta = \frac{\max(a + c, a + d, c + b)}{a + c}.$$

To prove that only points in the set S are feasible, we can assume that $\frac{\beta}{3-\beta} > \alpha$ and derive a contradiction. (There is no need to do this in the other equation because they are symmetric). But this is tedious work because there are nine cases, 3 ways to choose the first max and 3 ways to choose the latter. So, we are going to use a computer assisted proof.

```
a, b, c, d = Reals('a b c d')
alpha, beta = Reals('alpha beta')


def new_solver():
    s = Solver()
    #givens
    s.add(a >= 0, b >= 0, c >= 0, d >= 0, a + b + c + d == 1)
    s.add(alpha >= 1, alpha <= 2, beta >= 1, beta <= 2)
    #wlog
    s.add(a+b >= c+d, a+c >= b+d)
    return s


l1 = [a+d, b+c, a+b]
```

```
l2 = [a+d, b+c, a+c]

i = 0
for t1 in l1:
    for t2 in l2:
            i+=1
            print(t1, '/', t2)
            if i in [1, 5]:
                    print('undetermined')
                    continue

            s = new_solver()
            for t in l1:
                    s.add(t1 >= t)
            for t in l2:
                    s.add(t2 >= t)
            s.add(alpha == t1 / (a+b))
            s.add(beta  == t2 / (a+c))
            s.add(3*beta / (beta+1) < alpha)
            print(s.check())
```

This code runs through the Cartesian product of the sets $\{a+b, a+d, c+d\}$ and $\{a+c, a+d, c+b\}$ and for each pair assumes that they are the largest of the set and checks if there is a way for this conditions to be met and the pair of direct and reverse flow be outside the bound. It is impossible for all cases and, because these are all of the cases, then it is impossible.

Observation: There is an `if` statement in the code to avoid two cases because in some computers, if there is not enough memory, the program may crash. We have the following proofs for these two cases.

1. $a+d \geq (b+c), (a+b)$ and $a+d \geq (b+c), (a+c)$
2. $b+c \geq (a+d), (a+b)$ and $b+c \geq (a+d), (a+c)$

Let us begin by proving 1. We have that

$$\frac{\beta}{3-\beta} = \frac{a+d}{3(a+c)-(a+d)}.$$

Suppose $\alpha < \frac{\beta}{3-\beta}$, then

$$\frac{a+d}{a+b} < \frac{a+d}{2a+3c-d}$$
$$2a+3c-d < a+b$$

$$[(a+c)-(b+d)]+2c<0$$

This is a contradiction because $a+c \geq b+d$ and $c \geq 0$.

Now, let us prove 2. We have

$$\frac{\beta}{3-\beta} = \frac{b+c}{3(a+c)-(b+c)}.$$

Suppose $\alpha < \frac{\beta}{3-\beta}$, then

$$\frac{b+c}{a+b} < \frac{b+c}{3a-b+2c}$$
$$3a-b+2c < a+b$$
$$2[(a+c)-b] < 0$$

This is a contradiction because $a+c \geq b+d$. □

Lemma 1 follows directly from lemma 10 and 11.

A.1.1.1 Pareto bounds

Now, we have to prove that the joints we showed in Figure 5.2 are in fact in the Pareto curves.

We begin with the left curve in the plot.

Lemma 12. *The joint where $d \geq c \geq a, d = a+c, b = 0$ has reverse average multiplicative flow equal to* 1.

Proof. The reverse flow is

$$\mathcal{L}^{\times}_{id}(\pi, C) = \frac{\max(a,c)+\max(b,d)}{\max(a+b,c+d)}.$$

Because $c \geq a$, $d \geq b = 0$ and $c+d \geq a+b = a$, we have

$$\mathcal{L}^{\times}_{id}(\pi, C) = \frac{c+d}{c+d} = 1.$$

□

Now, the lower part of the plot.

Lemma 13. *The joint where $a \geq c \geq d, a = d+c, b = 0$ has direct average multiplicative flow equal to* 1.

Proof. The direct flow is

$$\overrightarrow{\mathcal{L}_{id}^{\times}}(\pi, C) = \frac{\max(a,b) + \max(c,d)}{\max(a+c, b+d)}.$$

Because $c \geq d$, $a \geq b = 0$ and $a + c \geq b + d = d$, we have

$$\overrightarrow{\mathcal{L}_{id}^{\times}}(\pi, C) = \frac{a+c}{a+c} = 1.$$

□

Now, for the upper part of the curve.

Lemma 14. *Consider a joint such that $a, d \geq c, d = a + c, b = 0$. If the reverse flow is α, then the direct flow is $\frac{3\alpha}{\alpha+1}$.*

Proof. Because $a \geq c$, $d \geq b = 0$ and $c + d \geq a + b$, we have

$$\mathcal{L}_{id}^{\times}(\pi, C) = \alpha = \frac{a+d}{c+d}$$

for the reverse flow.

Because $a \geq b = 0$, $d \geq c$ and $a + c = b + d = d$, we have

$$\overrightarrow{\mathcal{L}_{id}^{\times}}(\pi, C) = \frac{a+d}{d}$$

as the direct flow.

Now, let us compute $\frac{3\alpha}{\alpha+1}$,

$$\frac{3\alpha}{\alpha+1} = \frac{3\frac{a+d}{c+d}}{\frac{a+d}{c+d}+1} = \frac{3\frac{a+d}{c+d}}{\frac{a+c+2d}{c+d}} = \frac{3(a+d)}{a+c+2d} = \frac{3(a+d)}{3d} = \frac{a+d}{d},$$

and this is the direct flow, as we wanted. □

The only part left is the right part of the plot.

Lemma 15. *Consider a joint such that $a, d \geq c, a = d + c, b = 0$. If the reverse flow is α, then the direct flow is $\frac{\alpha}{3-\alpha}$.*

Proof. Because $a \geq c$, $d \geq b = 0$ and $c + d = a + b = a$, we have

$$\mathcal{L}_{g}^{\times}(\pi, C) = \alpha = \frac{a+d}{c+d}$$

for the reverse flow.

Because $a \geq b = 0$, $d \geq c$ and $a + c \geq b + d = d$, we have

$$\overrightarrow{\mathcal{L}_{g}^{\times}}(\pi, C) = \frac{a+d}{a+c}$$

as the direct flow.

Now, let us compute $\frac{\alpha}{3-\alpha}$,

$$\frac{\frac{a+d}{c+d}}{3 - \frac{a+d}{c+d}} = \frac{a+d}{3c+2d-a} = \frac{a+d}{2c+d} = \frac{a+d}{a+c},$$

and this is the direct flow, as we wanted. □

A.1.2 Proof for the average additive Bayes flow

We will now prove lemma 3. Let α be the reverse average additive Bayes flow and β the direct one. We begin by showing a joint that has reverse flow α and direct flow β.

Lemma 16. *The joint*

$$\pi \triangleright C = J = \begin{pmatrix} \frac{1-\alpha-\beta}{3} & \frac{1+2\beta-\alpha}{3} \\ \frac{1+2\alpha-\beta}{3} & 0 \end{pmatrix}$$

has reverse average additive Bayes flow equal to α and direct average additive Bayes flow equal to β for every α, β in the set

$$\{(\alpha, \beta) | -\frac{1}{2} + 2\beta \leq \alpha \leq \frac{1}{4} + \frac{\beta}{2}, 0 \leq \alpha, \beta \leq 0.5\}$$

Proof. First, we have to prove that this is in fact a joint. The sum of all values is

$$\frac{1-\alpha-\beta}{3} + \frac{1+2\beta-\alpha}{3} + \frac{1+2\alpha-\beta}{3} + 0 = \frac{3}{3} = 1,$$

which is necessary. Now, we have to prove that every value is larger or equal to zero. Because the maximum value of α, β is 0.5, $\frac{1-\alpha-\beta}{3}$ is always at least zero. To prove, that the value of $\frac{1+2\alpha-\beta}{3}$ is also valid, we run the following program

```
alpha, beta = Reals('alpha beta')
s.add(alpha >= 0, alpha <= 1/2)
s.add(beta >= 0, beta <= 1/2)
s.add(beta <= 1/4 + alpha/2)
s.add(beta >= -1/2 + 2*alpha)
s.add(1 + 2*alpha - beta < 0)
s.check()
```

which give us the output `unsat`, as we wanted.

Now, let us compute the reverse flow

$$\mathcal{L}^{+}_{id}(\pi, C) = \max(a, c) + \max(b, d) - \max(a + b, c + d)$$

$$\mathcal{L}^{+}_{id}(\pi, C) = c + b - (a + b) = c - a = \frac{3\alpha}{3} = \alpha,$$

as we wanted.

The direct flow is

$$\overrightarrow{\mathcal{L}^{+}_{id}}(\pi, C) = \max(a, b) + \max(c, d) - \max(a + c, b + d)$$

$$\overrightarrow{\mathcal{L}^{+}_{id}}(\pi, C) = b + c - (a + c) = b - a = \frac{3\beta}{3} = \beta.$$

□

Now, we just need to prove that every α and β outside of the region that was describe can not be reached.

Lemma 17. *Any point that is not in the set*

$$\{(\alpha,\beta)| -\frac{1}{2}+2\beta \leq \alpha \leq \frac{1}{4}+\frac{\beta}{2}, 0 \leq \alpha, \beta \leq 0.5\}$$

can not be achieved as a pair of direct and reverse flow for average additive Bayes flow.

Proof. We are also going to use an SMT for this proof. The program that checks this is

```
a, b, c, d = Reals('a b c d')
alpha, beta = Reals('alpha beta')

def new_solver():
    s = Solver()
    #givens
    s.add(a >= 0, b >= 0, c >= 0, d >= 0, a + b + c + d == 1)
    s.add(alpha >= 0, alpha <= 1/2, beta >= 0, beta <= 1/2)
    #wlog
    s.add(a+b >= c+d, a+c >= b+d)
    return s

def maxx(x, y, l):
    return x if l.index(x) < l.index(y) else y

for perm in permutations([a, b, c, d]):
    i = list(perm)
    s = new_solver()
    s.add(alpha == maxx(a, c, i) + maxx(b, d, i) - (a + b))
    s.add(beta == maxx(a, b, i) + maxx(c, d, i) - (a + c))

    for t1, t2 in zip(i[:-1], i[1:]):
        s.add(t1 >= t2)

    s.add(beta > 1/4 + alpha/2)
    print(s.check())
```

What this proof does is test all possible permutations in the sense of which value of the probability distribution is the largest, the second largest, and so on. Then, it checks if it is possible for the direct and reverse flow to be outside the stated region. The output is `unsat` for every permutation. Because this is an exhaustive list of all cases, it is not possible. □

A.1.2.1 Pareto bounds

Now, we need to prove the Pareto bound for additive case. Lemma A.1.1.1 shows that the additive reverse flow is 0, and lemma A.1.1.1 shows that the additive direct flow is zero, as we want. Now we need to prove the upper part and the right part. We begin with the upper one.

Lemma 18. *Consider a joint such that $a, d \geq c, d = a + c, b = 0$. If the reverse flow is α, then the direct flow is $\frac{1}{4} + \frac{\alpha}{2}$.*

Proof. Because $a \geq c$, $d \geq b = 0$ and $c + d \geq a + b$, we have

$$\mathcal{L}^{+}_{id}(\pi, C) = \alpha = a + d - (c + d) = a - c$$

for the reverse flow.

Because $a \geq b = 0$, $d \geq c$ and $a + c = b + d = d$, we have

$$\overrightarrow{\mathcal{L}^{\times}_{id}}(\pi, C) = a + d - d = a$$

as the direct flow.

Because $b = 0$, we have that $a + c + d = 1$. But if $a + c = d$, then $a + c = \frac{1}{2}$ and $c = \frac{1}{2} - a$.

Now, let us compute $\frac{1}{4} + \frac{\alpha}{2}$,

$$\frac{1}{4} + \frac{\alpha}{2} = \frac{1}{4} + \frac{a - c}{2} = \frac{1}{4} + \frac{2a - \frac{1}{2}}{2} = a,$$

and this is the direct flow, as we wanted. □

The only part left is the right part of the plot.

Lemma 19. *Consider a joint such that $a, d \geq c, a = d + c, b = 0$. If the reverse flow is α, then the direct flow is $-\frac{1}{2} + 2\alpha$.*

Proof. Because $a \geq c$, $d \geq b = 0$ and $c + d = a + b = a$, we have

$$\mathcal{L}^{\times}_{g}(\pi, C) = \alpha = a + d - (c + d) = a - c$$

for the reverse flow.

Because $a \geq b = 0$, $d \geq c$ and $a + c \geq b + d = d$, we have

$$\overrightarrow{\mathcal{L}^{\times}_{g}}(\pi, C) = a + d - (a + c) = d - c$$

as the direct flow.

Because $b = 0$, we have that $a + c + d = 1$. But if $a = d + c$, then $d + c = \frac{1}{2}$ and $c = \frac{1}{2} - d$.

Now, let us compute $2\alpha - \frac{1}{2}$,

$$2\alpha - \frac{1}{2} = 2(a - c) - \frac{1}{2},$$

because $a = d + c$, we get:

$$2\alpha - \frac{1}{2} = 2(d + c - c) - \frac{1}{2} = d + d - \frac{1}{2}.$$

Now, using the fact that $c = \frac{1}{2} - d$, we have

$$2\alpha - \frac{1}{2} = d - c,$$

and this is the direct flow, as we wanted. □

Appendix B

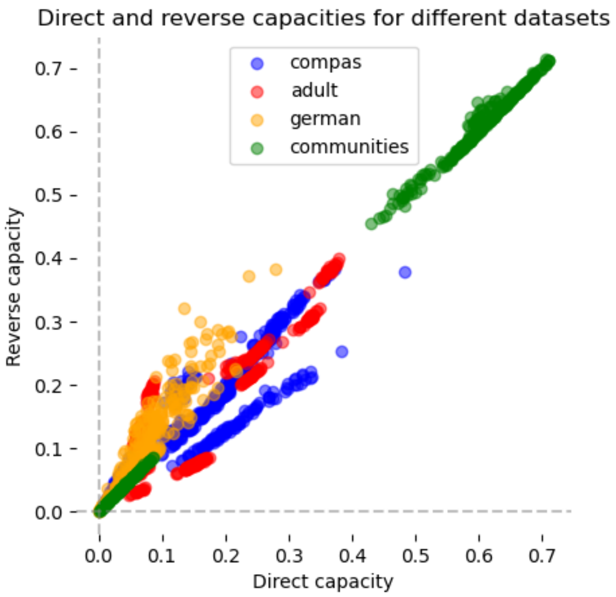
Complementing experiments

B.1 Appendix

This appendix presents some of the plots from Chapter 7 replacing average multiplicative Bayes flow with multiplicative capacity. The conclusion from most plot is very similar, so we will discuss them all briefly, because the discussion from the chapter is valid here as well.

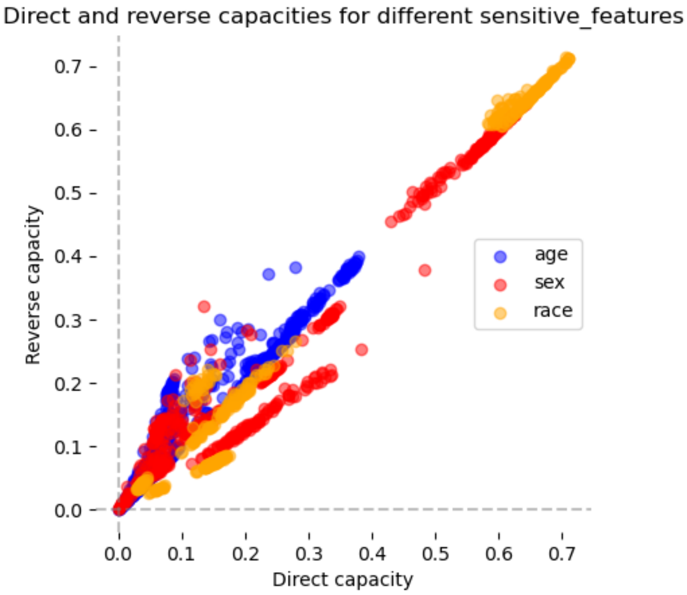
The first two plots are scatter plots showing the values for direct and reverse capacity by dataset and sensitive features. They are in Figures B.1 and B.2. Both plots are similar to the ones in Chapter 7, we can see that the same groups that have higher flow are the ones who have higher capacity. One thing that changes is that direct and

Figure B.1: Scatter plot showing the value of direct and reverse capacity by dataset. The values for the communities dataset are the larger ones, while the German dataset has the smallest ones. Most points are distributed near the identity line.



Source: created by the author.

Figure B.2: Scatter plot showing the value of direct and reverse capacity by sensitive attribute. Race is the sensitive feature with the higher values of capacity. Most points are distributed near the identity line.



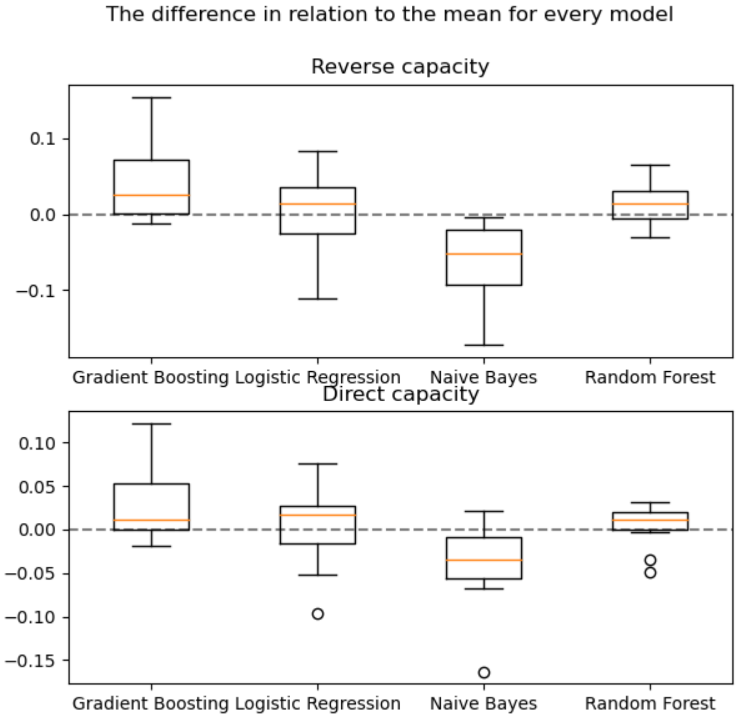
Source: created by the author.

reverse capacity are more correlated than direct and reverse flow, so he points are closer to the identity line. Another difference is that there are not so many points with minimal capacity as there are with minimal flow.

Figure B.3 shows the direct and reverse capacities for every model in relation to the mean. Just like in the Bayes flow case, Naive Bayes has the smallest flows of information, while gradient boosting and random forest have the largest values.

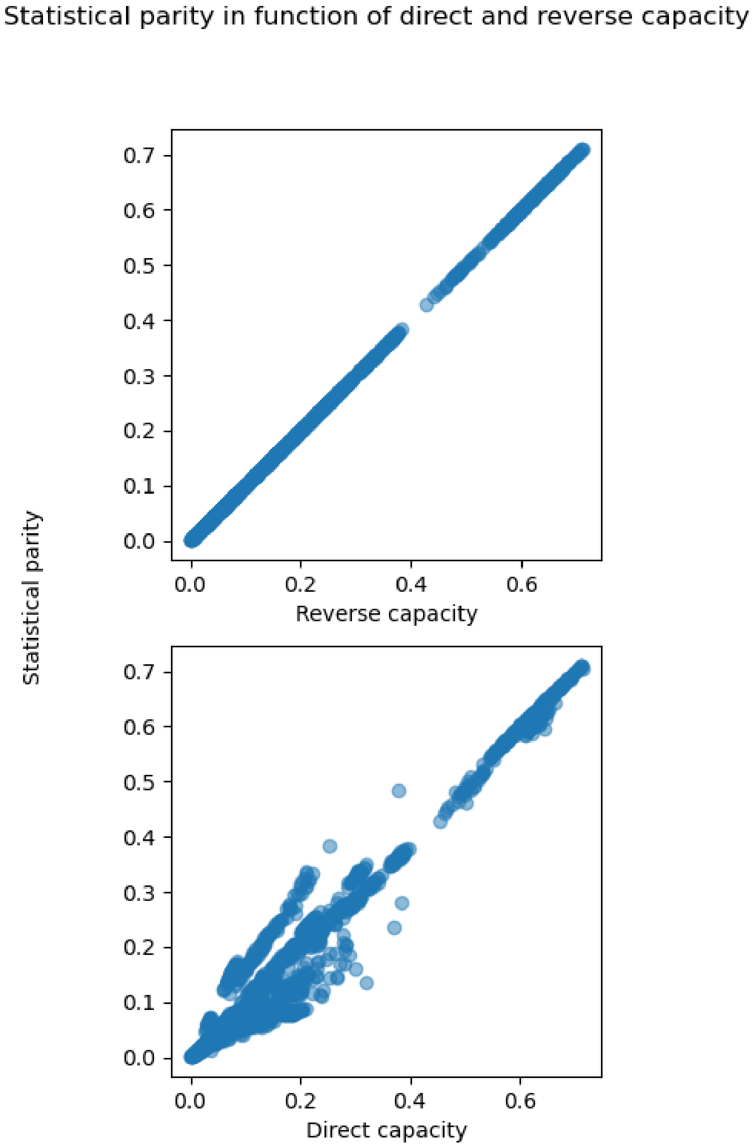
Figure B.4 has two scatter plots. One comparing mean difference and direct capacity, and the other showing mean difference and reverse capacity. The second plot is very close to the previous plot of Chapter 7, that shows that mean difference and reverse flow of information are highly correlated. But there is a small difference in the first one. As was proved in Chapter 6, mean difference is equal to multiplicative capacity, so all points are on the identity line.

Figure B.3: Boxplot showing the difference of multiplicative capacity in relation to the mean of different algorithms. The median of all of them is close to zero, meaning that they are all very similar. But naive Bayes is almost completely below the mean, while Random Forest and Gradient Boosting are above the line. This is true for both direct and reverse flow. Logistic Regression is almost evenly distributed above and below the mean.



Source: created by the author.

Figure B.4: Two scatter plots show the relation of capacity with mean difference. In the first one, we see that direct capacity is equal to mean difference, so all points are in the identity line. In the second one, we see that they are correlated, so, the greater one of them is, the greater the second one is as well.



Source: created by the author.