A bayesian approach for Uplift modeling : application on biased data

Mina Rafla

Normandie Université

UNIVERSITÉ CAEN NORMANDIE

THÈSE

Pour obtenir le diplôme de doctorat

Spécialité **INFORMATIQUE**

Préparée au sein de l'**Université de Caen Normandie**

A Bayesian Approach for Uplift Modeling: Application on Biased Data

Présentée et soutenue par

MINA RAFLA

Thèse soutenue le 06/11/2023

devant le jury composé de :

MME SIHEM AMER-YAHIA	Directeur de recherche - UNIVERSITE GRENOBLE ALPES	Rapporteur du jury
M. SZYMON JAROSZEWICZ	Professeur - Institute of Computer Science	Rapporteur du jury
M. TIAS GUNS	Professeur - LEUVEN - KATHOLIEKE UNIVERSITEIT	Membre du jury
MME MARIE-JEANNE LESOT	Professeur des universités - UNIVERSITE PARIS 4 PARIS-SORBONNE	Membre du jury
M. NICOLAS VOISINE	Ingénieur de recherche - ORANGE INNOVATION	Membre du jury
M. PHILIPPE LERAY	Professeur des universités - UNIVERSITE NANTES	Président du jury
M. BRUNO CREMILLEUX	Professeur des universités - Université de Caen Normandie	Directeur de thèse

Thèse dirigée par **BRUNO CREMILLEUX** (GREYC ALGORITHMIQUE)

MIIS

GREYC
Laboratoire de recherche en sciences du numérique

Normandie Université

THÈSE

Pour obtenir le diplôme de doctorat

Spécialité INFORMATIQUE

Préparée au sein de l'Université de Caen Normandie

A Bayesian Approach for Uplift Modeling: Application on Biased Data

Présentée et soutenue par

Mina RAFLA

Thèse soutenue le 06/11/2023
devant le jury composé de

Mme SIHEM AMER-YAHIA	Directrice de recherche, Université de Grenoble Alpes	Rapportrice du jury
M. SZYMON JAROSZEWICZ	Professeur, Institue of Computer Science, Polish Academy of Sciences	Rapporteur du jury
M. TIAS GUNS	Professeur, KU LEUVEN	Membre du jury
M. PHILIPPE LERAY	Professeur des universités, Université de Nantes	Membre du jury
Mme MARIE-JEANNE LESOT	Professeure des universités, Sorbonne Université	Membre du jury
M. BRUNO CREMILLEUX	Professeur des universités, Université de Caen Normandie	Directeur de thèse
M. NICOLAS VOISINE	Ingénieur de recherche, Orange Innovation	Encadrant de thèse

Thèse dirigée par BRUNO CREMILLEUX, Groupe de recherche en informatique, image, automatique et instrumentation

UNIVERSITÉ CAEN NORMANDIE

ED 590 MIIS
Mathématiques, Information, Ingénierie des Systèmes

GREYC

orange™

Remerciements

Tout d'abord, je tiens à exprimer ma profonde gratitude à Sihem AMER-YAHIA et Szymon JAROSZEWICZ pour avoir accepté de rapporter cette thèse. Je tiens également à remercier Marie-Jeanne LESOT, Tias GUNS et Philippe LERAY pour leur participation au sein du jury de thèse.

Cette thèse n'aurait pas atteint ce niveau sans l'accompagnement constant de mes encadrants, Bruno CREMILLEUX et Nicolas VOISINE. Avoir une telle équipe d'encadrement a été une véritable chance.

Bruno : Ton accueil chaleureux à Caen, nos nombreux dîners et discussions, ton soutien pendant la période difficile du covid, et ton engagement pour que je vive pleinement mon doctorat ont été essentiels. Je te suis très reconnaissant pour tout cela et pour ton accompagnement dans la planification de ma carrière post-thèse.

Nicolas : Ta motivation pour mon travail, ta disponibilité constante, nos longues réunions et même nos échanges du quotidien ont été essentiels. Nos moments partagés lors des différentes conférences, enrichis de discussions et de nombreuses photos, resteront toujours de très beaux souvenirs.

Je tiens également à remercier Marc BOULLÉ. Ta collaboration a été très précieuse pour plusieurs parties de cette thèse. Merci pour nos discussions sur les thématiques Bayésiennes et pour ton accueil dans ton bureau à chaque fois.

Un grand merci également à Fabrice Clerot pour nos échanges stimulants, notamment autour de la causalité.

Je tiens à remercier l'équipe PROF : Fabrice, Elias, Tanguy, Marc, Bruno, Carine, Hugo, Anaïs, Raphael, Frank, Laurent, Romain, Xihui, Charbel, Betty et Valentina pour la merveilleuse ambiance de travail, nos pauses café, nos discussions et tous les moments de rire partagés. Une mention spéciale à la Dream Team Orange: Vincent, Nicolas, Françoise, Victor, Colin et Pierre, avec lesquels j'ai eu le plaisir d'assister à plusieurs conférences scientifiques.

Je tiens à adresser mes remerciements à toute l'équipe CODAG. Un merci particulier au responsable de l'équipe, Bertrand CUISSART. De plus, je souhaite remercier les gestionnaires du laboratoire GREYC

pour leur accueil et leurs efforts dans l'organisation des différentes missions.

Je tiens à exprimer ma profonde gratitude à mon père, ma mère et ma sœur Gina. Votre présence constante a été précieuse. Gina, merci d'avoir fait le nécessaire pour assister spécialement à ma soutenance.

Enfin, je souhaite remercier du fond du cœur Aline, ma fiancée. Tu as été ma source de joie, de réconfort et d'encouragement tout au long de ces trois années. Merci pour ta motivation constante, ta patience et ta compréhension, surtout durant les moments les plus éprouvants. Je ne te promets pas que j'arrêterai de me plaindre lorsque mon code ne marche pas.

Résumé

La modélisation de l'uplift vise à estimer l'impact d'un traitement, comme une campagne marketing ou un médicament, sur le comportement d'un individu. Cette approche est très utile dans de nombreuses applications, comme la médecine personnalisée et la publicité, car elle permet de cibler la sous-population sur laquelle le traitement aura le plus grand impact. La modélisation de l'uplift est une tâche ardue car les données disponibles ne sont que partiellement connues (pour un individu, les réponses aux traitements alternatifs ne peuvent pas être observées).

Cette thèse, réalisée en collaboration avec la société française de télécommunications *Orange*, est une contribution au domaine de la modélisation de l'uplift. Plus précisément, elle traite de trois défis majeurs rencontrés dans toute approche de modélisation d'uplift: 1. La paramétrisation des algorithmes existants. 2. Le biais des données. 3. La haute dimensionalité des données.

Cette thèse répond à ces défis en définissant une approche bayésienne sans paramètre utilisateur pouvant être appliquée à une variété d'algorithmes d'uplift. Nous introduisons d'abord une approche de discrétisation bayésienne de l'uplift pour le prétraitement des données. Nous l'étendons ensuite à la sélection des variables. Nous montrons que les méthodes que nous proposons pour la transformation et la sélection de variables sont efficaces pour la modélisation de l'uplift.

Puis, nous présentons une méthode sans paramètre utilisateur de construction d'un nouvel arbre de décision. Cette méthode, appelée UB-DT, transforme le problème d'apprentissage de l'arbre de décision en un problème d'optimisation, avec pour objectif de trouver l'arbre de décision le plus probable sachant les données. De plus, nous étendons UB-DT aux forêts aléatoires et démontrons sa performance par des évaluations expérimentales.

Nous répondons au défi du biais de sélection en concevant un protocole expérimental de simulation sous contrôle de jeux de données biaisés selon le biais de non affectation aléatoire. Cette démarche nous a permis de tester rigoureusement nos méthodes ainsi que les solutions existantes de l'état de l'art face à ce type de biais et de mieux déterminer lesquelles employer face à ce type de biais.

Enfin, nous avons évalué nos méthodes en les confrontant à des jeux de données télécom réels. Chaque méthode a été évaluée de façon individuelle et dans le cas d'une chaîne de traitement d'un problème d'uplift. Nous avons implémenté toutes nos approches proposées dans une nouvelle bibliothèque Python nommée 'Kuplift' que nous présentons.

Abstract

Uplift modeling aims to estimate the incremental impact of a treatment, such as a marketing campaign or a drug, on an individual's behavior. This approach is very useful in a lot of applications such as personalized medicine and advertising, as it allows targeting the specific population on which the treatment will have the greatest impact. Uplift modeling is a challenging task because data are only partially known (for an individual, responses to alternative treatments cannot be observed).

This thesis, carried out in collaboration with the French telecommunications company *Orange*, is a contribution to the field of uplift modeling. More specifically, it addresses three major challenges encountered in any uplift modeling approach.

1. The parameterization of existing algorithms.
2. Data bias.
3. The high dimensionality of the data.

This thesis achieves this by defining a user parameter-free Bayesian approach that can be applied to a variety of uplift algorithms. We first propose a Bayesian uplift discretization method that can be used as a data preprocessing approach. We then extend it to the case of feature selection. We show that both the variable transformation and feature selection approaches are powerful and important for the case of uplift modeling.

We then design a new user-parameter-free Bayesian decision tree method. This approach, named UB-DT, transforms the decision tree learning problem into an optimization problem, where the goal is to find the decision tree that is most likely given the data. In addition, we extend UB-DT to the case of random forests and demonstrate its performance through experimental evaluations.

We then tackle the challenge of selection bias by developing an experimental protocol specifically designed to simulate non-random assignment bias in uplift datasets. This allowed us to rigorously test both our methods and existing state-of-the-art solutions against this type of bias.

Finally, we conducted comprehensive evaluations of our proposed techniques using real-world telecom datasets. Each method was evaluated both in isolation and in combination. We implemented all of our proposed approaches. We introduce them in a new Python package called 'Kuplift'.

To Aline, Imane, Wagdi, and Gina, for all the love they give me.

Contents

CHAPTER 1

Introduction

Contents

1.1 Context and motivation

'Would a teenage boy buy the same clothes as his grandmother? Probably not. But when they get sick, they're likely to receive the same medical treatment, despite their many differences. And so will everyone else' [43]. This highlights the importance of personalised medicine, which tailors medical treatment to a patient's unique characteristics and genetic information. However, estimating the effect of medical treatments at the individual level is a complex task because it requires observing an individual's behaviour with and without treatment, which is impossible to do simultaneously. This is because it is impossible to treat and not treat the same person at the same time and to observe the difference between the two alternative behaviours.

This problem exists in many fields, including marketing, medicine and the social sciences. We often face the challenge of identifying the individuals who are most likely to benefit from a particular intervention (treatment), i.e. the individuals on whom the treatment will have the most positive effect. In marketing, for example, the goal is to design a campaign that effectively motivates customers to buy a particular product. This is one of the current challenges faced by *Orange*, a French telecommunications company where this thesis was carried out. Finding the most effective marketing campaign, i.e. the optimal treatment, for each customer that will result in the maximum number of purchases would be extremely

valuable. Here, an optimal treatment is defined as the treatment among several options that maximises the probability of a desired outcome. In economics, another example is the analysis of the impact of funded training programmes on the earnings of trainees and non-trainees in the labour market, as described in [2].

To identify the individuals who should receive a particular intervention (also called a treatment or action), a first simple strategy is to compare the average outcomes of the treatment group (those who receive the intervention) and the control group (those who do not). This comparison helps determine whether or not the intervention was beneficial on average, however such a comparison is not sufficient in a lot of situations. To illustrate, let's take a study of 200,000 individuals, half of them, i.e. 100,000 individuals, were contacted about an internet offer while the other half did not. (cf. Fig. 1.1). 70% of the contacted group purchased the offer, while only 50% of the non-contacted group did. This suggests a positive impact of the marketing campaign, with a 20% increase in purchases. However, if we examine the data more closely, we will see that the treatment effect varies across different subgroups. In our example, as shown in Fig. 1.1, the intervention had a negative impact on younger people: only 30% of young customers made a purchase when contacted, compared to an 80% purchase rate when not contacted. This suggests that the intervention may have discouraged them from purchasing the product. For seniors, the opposite effect was observed. All of the contacted seniors took advantage of the internet offer, but only 20% of them took advantage of the internet offer when no one contacted them. With this information, a marketing team would not have targeted everyone, but only a subset of customers, each with their optimal treatment.

1.2 From conventional methods to uplift modeling

1.2.1 Response Modeling and A/B Testing

Traditional methods, particularly in marketing, have been used to either identify potential targets for marketing campaigns or to determine the optimal treatment to assign to all customers. This has been done using response modeling and A/B testing, respectively. In this section, we provide a brief discussion of each of these techniques and highlight their limitations.

Response modeling [46] has long been used to predict the outcome of individuals after treatment. It has been used to predict whether or not a person will buy a product after receiving a marketing campaign. However, the disadvantage

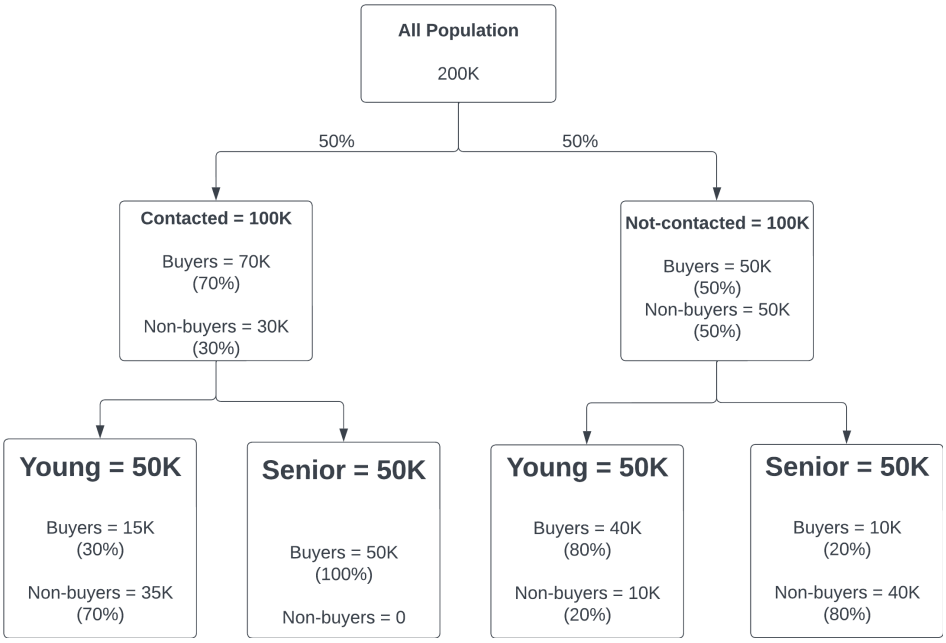


Figure 1.1: Example of treatment effect estimation

of this method is that it tries to predict the probability of a particular outcome without considering the effect of the treatment on that outcome. For example, a person may decide to buy the product regardless of whether they receive the marketing campaign. In this case, contacting them is an unnecessary cost. They also might decide not to buy the product if they receive the campaign, whereas their decision would be different if they do not receive it (like the young clients in our previous example).

A/B testing [69] is also often used in these situations. It is most commonly used to compare two versions of a web page of a software system on different users. It involves randomly dividing a sample of users into two groups, a treatment group and a control group, and then measuring a metric of interest, such as the number of clicks or the conversion rate. By comparing the results of the two groups with statistical tests, researchers can determine which version of the two treatments (the variations of the web page) is more effective. The disadvantage of A/B testing is that it does not search for individuals to target with a marketing campaign, but it searches for the treatment that will be assigned to everyone and will yield the greatest profit in average. So, it cannot discover groups that should be avoided like the senior group in our previous example (cf. Fig. 1.1).

1.2.2 The need for uplift modeling

To address the limitations of traditional approaches, it is necessary to take into account different categories of customers:

1. The *persuadables*: customers who respond positively to a marketing campaign only because they have received the treatment ⇒*Treatment has a positive effect.*

2. The *sure-things*: customers who will always respond positively to a marketing campaign ⇒*Treatment has no effect in this case because they would buy anyway.*

3. The *lost causes*: customers who would not respond positively anyway ⇒*Treatment has no effect in this case either.*

4. The *do-not-disturb*: customers who would respond positively but did not because of the marketing campaign ⇒*The treatment has a negative effect in this case.*

Estimating the treatment effect per subgroup as shown in Fig. 1.1 may be complex in real scenarios, particularly in high-dimensional data where it may not be possible to determine the relevant subgroups. This is where ***uplift modeling*** can be useful. In marketing, "uplift" refers to the treatment effect, and uplift modeling seeks to estimate the effect of a treatment on an outcome variable at the individual level. It comes to overcome the drawbacks of response modeling and A/B testing.

A formal difference between supervised learning and uplift modeling

Uplift modeling should not be confounded with supervised learning. Supervised learning algorithms, such as those used in classification and response modeling, aim to estimate a *single* probability distribution for a target variable and can help avoiding *lost causes* (as presented above). This can be interpreted as training a classifier to predict the individuals that are most likely to have a positive response.

On the other side, the goal of the treatment effect estimation, for example in marketing applications, is not to predict likely buyers but to predict the people who will buy *only because* they received a treatment. In uplift modeling, we distinguish two different groups, the treatment group and the control group. The treatment group consists of people who received a treatment, and the control group consists of people who did not receive a treatment. The goal is then to create a model that

learns not the probability of a particular response, but the difference between two outcome probabilities: the outcome probability in the treatment group and the outcome probability in the control group.

1.3 Background on Uplift Modeling

In this section, we will introduce the basic concepts and formal definitions related to uplift modeling.

We begin with an example that we will refer to throughout this section to illustrate the formal definitions. Consider the case of a telecom company with customers subscribing to a basic internet package. The company is evaluating a new promotional campaign offering selected customers a free one-month upgrade to a premium internet package. It wants to predict the probability that customers will retain the premium package (i.e. become paying customers) after the promotional period based on this campaign. In this example, the treatment, denoted T, is whether the customer receives the promotional offer or not. This is coded as '1' if the customer receives the offer and '0' if she does not. The outcome, denoted Y, is also binary, whether the customer upgrades to the premium package and pays for it after the promotion period ('1' if yes, '0' if no). Each customer can have two potential outcomes: $Y(T = 0)$, the potential outcome if she had received no treatment; and $Y(T = 1)$, the potential outcome if she had received the treatment.

The ITE, which corresponds to the *Individual Treatment Effect*, can then be defined as:

$$ITE = Y(T = 1) - Y(T = 0)$$

The goal of treatment effect estimation is to calculate the ITE. However, the ITE can never be calculated because only one of $Y(T = 1)$ and $Y(T = 0)$ can be observed, i.e. a client cannot be treated and not treated at the same time. This issue is known as the ***Fundamental Problem of Causal Inference***. The unobserved potential outcome can also be called the counterfactual of the observed outcome.

Since we cannot calculate the ITE, two communities simultaneously tackled this challenge: (a) The Heterogeneous Treatment Effect community that focuses on calculating the Conditional Average Treatment Effect (CATE) and (b) the uplift modeling community. The objectives of each of these two communities are discussed below.

Conditional Average Treatment Effect. Since the ITE cannot be calculated, we can instead, under some assumptions that we state below, estimate the conditional average treatment effect (CATE). A client being described by a vector x, the CATE can then be defined as:

$$CATE : \hat{\tau}(\boldsymbol{x}) := \mathbb{E}[Y(T=1) - Y(T=0) \mid X = \boldsymbol{x}] \tag{1.1}$$

where X is a random variable describing a set of features.

It has been shown that the CATE is the best estimator for the ITE in terms of the mean squared error [42].

Observational data. The CATE estimation approaches are often developed for observational data. Observational data refers to data collected without the use of a controlled experiment, where treatments are assigned to the individuals or clients without randomization. Observational data are collected without the company assigning the promotional offer to specific customers. Instead, customers may themselves decide whether or not to take up the offer, or the promotional offer may be available only to a subset of customers who meet certain criteria (such as being on a particular current plan or having a particular usage behaviour). The company then collects data on outcomes (i.e. whether customers retain the premium package after the promotional period) and any relevant characteristics of the individuals. As a consequence, CATE estimation meets the challenge of non-random assignment of the treatment leading to selection bias. Uplift data bias will be described in Section 2.5 and non-random assignment will be particularly discussed in Chapter 5.

Uplift Modeling. Uplift modeling is a practical branch of the CATE estimation basically developed for the applications of the marketing field. Uplift modeling assumes a randomized control trial, where data is collected in a controlled experiment and customers are randomly assigned to treatment and control groups. In other words, there is no dependence between the characteristics of the instances (customers) and the treatment assignment. In this case the uplift of an individual described by a vector x, denoted Uplift(x), is defined by:

$$\text{Uplift}(\boldsymbol{x}) = \mathbb{E}[Y \mid T = 1, X = \boldsymbol{x}] - \mathbb{E}[Y \mid T = 0, X = \boldsymbol{x}]. \tag{1.2}$$

The link between the CATE and Uplift estimation tasks. Both of the CATE estimation task and the uplift estimation task are equivalent under a set of assumptions:

- **Conditional independence assumption (CIA)** This assumption is also referred to as the *unconfoundedness* assumption or the *strong ignorability* assumption). It implies that the treatment assignment is independent of the two potential outcomes:

$$(Y(T=1), Y(T=0)) \perp T \mid X$$

  In the context of the earlier example, the CIA would assume that there is no unobserved variable, such as customer satisfaction, that affects both the customer's decision to accept the offer and their decision to remain on the free plan. If such a variable exists and is not accounted for, it may confound the relationship between treatment and outcome and thus bias the estimate of the treatment effect. This assumption is untestable and its validity is based on expert knowledge of the data.

- **Stable Unit Treatment Value (SUTVA) assumption** The treatment given to one subject has no effect on other subjects, i.e. subjects do not interfere with each other. Again, in the context of our earlier example on telecom data, this assumes that whether a client takes the offer does not affect the decision of another customer(s), for example his neighbours or friends, to buy the premium package. This is an important assumption, because otherwise the treatment effect estimation will not be correct.

- **Overlap assumption** Each subject has a non-zero probability of being in the treatment or control group. In other words, no sub-population is entirely in the treatment or control group:

$$0 < P(T=1 \mid X=x) < 1$$

  The overlap assumption in our previous scenario means that for any given customer, regardless of their characteristics (such as age, loyalty to the company, current plan, past usage data, etc.), there should be a non-zero probability that they will take up the promotional offer (treatment) and a non-zero probability that they will not.

Given these assumptions, we can consider the tasks of CATE (Conditional Average Treatment Effect) estimation (see Equation 1.1) and uplift estimation (see Equation 1.2) to be equivalent (for a mathetmatical proof please refer to [77, 36]). There is significant overlap in the methodologies used for uplift modeling and CATE estimation, to the extent that certain approaches have been independently reinvented within each community. *Throughout this thesis, the terms 'Uplift' and 'CATE' will be used interchangeably to refer to the treatment effect.*

1.4 Challenges we tackle in this thesis

This thesis basically tackles 3 problems that already exist in the uplift modeling literature: (1) Automating Uplift Models (2) Data bias (3) High dimensionality. We discuss below each of them.

1.4.1 Automating Uplift Models

According to [82], Automated machine learning (AutoML) aims to reduce the demand for data scientists by enabling domain experts to build machine learning applications automatically without extensive knowledge of statistics and machine learning. To the best of our knowledge, the challenge of automating uplift approaches has not been tackled in the uplift modeling research area.

As we will see in Section 2.2, there is a wide range of uplift methods [77] such as meta-learners and direct approaches. A meta-learner is an algorithm that combines traditional supervised learning algorithms for uplift estimation while direct approaches are a set of algorithms specifically designed for uplift modeling. The main drawback of all these approaches is that they require parameters to be set. Meta learners also present an additional requirement, which is the choice of the machine learning algorithm to be used. All of these are clear limitations for non-machine learning experts to use these tools. Even for machine learning experts, they need to test different parameter values and different learning algorithms with meta learners to find the optimal combination that fits the data at hand. That is why automatic parameter-free uplift modeling algorithms are needed.

1.4.2 Data Bias

Uplift modeling assumes that treatment and control groups are drawn from the same distribution. While this strong assumption is potentially valid in experimental data and controlled trials, it often does not hold in real-world scenarios. The CATE estimation example given above in Fig. 1.1 was relatively straightforward because the treatment and control groups were of equal size. In addition, treatment assignment did not depend on the features of the instances. This was reflected in the equal representation of young and old people in each of the treatment groups. In real-world scenarios, it is easier to collect control data than to collect treatment data. That is why the treatment group tends to be more biased: it is difficult to apply treatments to individuals and collect the corresponding data, often due to ethical, political or economic constraints. This often leads to unequal sizes of treatment and control groups, which has been referred to in recent work as *Imbalanced Treatment Conditions (ITC)* [7], and can complicate the estimation

of CATE. Also, even when the two treatment groups are of equal size, there are often dependencies between the treatment assignment and the characteristics of the instances. For example, the treatment group may consist mostly of young individuals, while the control group may consist mostly of older individuals.

1.4.3 High Dimensionality

Identifying subgroups with different treatment effects when there are hundreds to thousands of features is a more complicated task. Telecom companies like *Orange* often has this type of problem with marketing and telecom data. This data has information generated and collected through their network infrastructure, customer interactions, and billing systems. It includes various dimensions, customer demographics, usage patterns, call records, service subscriptions, customer interactions, and billing information. Uplift modeling algorithms can suffer when there are large numbers of features, leading to overfitting and computational and interpretability problems. Since uplift modeling is a different problem than supervised learning, traditional feature selection approaches (which have been extensively studied in the literature) are not applicable.

1.5 Contributions

The contributions of this thesis are as follows:

1. We propose a user parameter free Bayesian approach for uplift discretisation that we called UMODL. It uses a density estimation approach based on the MDL principle. UMODL defines a space of discretization models and a prior distribution. From this model space, a Bayesian optimal evaluation criterion is defined to evaluate a discretization model. A search algorithm is then used to find the model with the optimal criterion. An experimental protocol evaluates the discretisation approach as a univariate uplift estimator. We show that UMODL is a good uplift estimator resistant to overfitting.

2. While a discretization approach is basically designed to handle continuous data, we show how to take advantage of UMODL to handle categorical data to do *value grouping* in order to be able to separate the different values of the categorical variable with distinct uplift and group the ones with similar behavior (similar treatment effect).

3. While the feature selection approaches for uplift modeling are very limited, we introduce a new feature selection approach for uplift modeling called UMODL-FS based on UMODL. Once the intervals dividing a variable X

are found using the UMODL discretization approach, UMODL-FS calculates an importance score on the found intervals. An experimental protocol shows that UMODL-FS is very resistant to noise and is able to find the set of variables that lead to the best uplift model.

4. We propose a Bayesian decision tree algorithm for uplift modeling, UB-DT. We transform the uplift tree learning problem into an optimisation problem. The goal is to find the uplift tree model that is most probable given the data according to Bayes law. So a global evaluation criterion for an uplift tree model is described, and a search algorithm is presented to search for the optimal uplift decision tree according to the global evaluation criterion. A random forest extension, that we called UB-RF, is also presented. A benchmark study shows the efficiency of our method against state-of-the-art modeling algorithms.

5. We study a type of bias in the uplift modeling process called the *Non-Random Assignment (NRA)* bias. We carry out an experimental study of the effect of the NRA bias on the different uplift modeling approaches and UB-DT. We propose a reweighting method to improve an uplift modeling method called the *class transformation* approach, which our study found to be the most sensitive to the NRA bias.

6. Finally, we provide an introduction to telecom data and show how uplift modeling can be performed to deal with it. We apply our feature selection, discretization, decision trees and random forests on real world telecom data. We introduce Kuplift, a new Python package that we have developed to implement our Bayesian algorithms.

1.6 Thesis outline

This thesis is structured as follows. **Chapter 2** presents an overview of the literature made in both the uplift modeling and CATE estimation communities, including a description of the modeling approaches, evaluation metrics and feature selection techniques. An overview of a density estimation approach called *MODL* is then presented. Finally, an overview of possible biases in the uplift modeling process is given. **Chapter 3** presents UMODL, our Bayesian approach to uplift discretisation and density estimation. We also show how UMODL can be used for feature selection for uplift. In **Chapter 4** we present UB-DT, a new Bayesian decision tree for uplift modeling. We present our global evaluation criterion for an uplift tree and a detailed proof. We discuss the algorithm used to find the uplift

tree with the best criterion. We then extend it to a random forest algorithm. In **Chapter 5** we present the experimental study we conducted to evaluate the uplift modeling approaches and UB-DT against the *Non-Random Assignment* bias. In **Chapter 6** we perform additional evaluation experiments for the UMODL discretization approach, UMODL-FS, UB-DT, and UB-RF on real telecom data. Finally, in **Chapter 7** we summarise the contributions of the thesis and discuss research perspectives.

1.7 Corresponding Articles

International Conferences

- Rafla, M., Voisine, N., & Crémilleux, B. (2023, May). Parameter-free Bayesian Decision Trees for Uplift Modeling : 27th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD).
- Rafla, M., Voisine, N., Crémilleux, B., & Boullé, M. (2022, September). A Non-Parametric Bayesian Approach for Uplift Discretization and Feature Selection. In European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD).
- Rafla, M., Voisine, N., & Crémilleux, B. (2022, April). Evaluation of Uplift Models with Non-Random Assignment Bias. In Advances in Intelligent Data Analysis XX: 20th International Symposium on Intelligent Data Analysis, (IDA)

International Workshop

- Rafla, M., Voisine, N., & Crémilleux, B. (2023, September). A Parameter-Free Bayesian Framework for Uplift Modeling - Application on Telecom Data. In *Uplift Modeling and Causal Machine Learning for Operational Decision Making* workshop, co-located with European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD).

National Conferences

- Rafla, M., Voisine, N., Cremilleux, B., & Boullé, M. (2023, January). Une approche bayésienne non paramétrique de sélection de variables pour la modélisation de l'uplift. In 23ème Journées Francophones Extraction et Gestion de Connaissances (EGC)

- Rafla, M., Voisine, N., & Cremilleux, B. (2022, January). Evaluation de l'uplift sur des données biaisées dans le cas du Non-Random Assignment. In 22ème Journées Francophones Extraction et Gestion de Connaissances (EGC)

CHAPTER 2

State-of-the-art

Contents

2.1 Introduction

The work presented in this thesis lies at the intersection of several topics (cf. Fig. 2.1): *uplift modeling*, *Bayesian approaches* and *data bias*. In this chapter we provide background information on each of these topics.

We start by addressing several points in uplift modeling. First, we review state-of-the-art uplift modeling approaches in Section 2.2 and evaluation metrics for uplift models in Section 2.3. In addition, we discuss in Section 2.4 feature selection techniques developed in the literature specifically for uplift modeling problems.

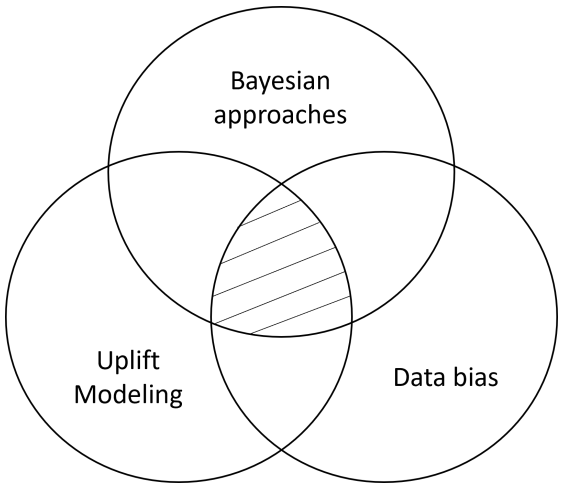


Figure 2.1: The structure of this chapter is around three topics: Uplift modeling, Data bias and a Bayesian approach called *MODL*

Next, in Section 2.5 we discuss potential biases that may exist in the uplift modeling process. We will divide them into two main categories: 1. modeling bias, which occurs mainly during the training phase of an uplift model. 2. deployment bias, which occurs when we apply a learned uplift model to a real-world scenario during the deployment phase.

A large part of the contribution of this thesis is to propose Bayesian approaches for uplift discretisation, uplift feature selection and uplift decision trees based on a density estimation technique known as *Minimum Optimised Description Length (MODL)* [8]. In the last section of this chapter, we will present the MODL density estimation and the MODL decision tree approaches. These will serve as preliminary knowledge for our contributions presented in Chapter 3 and Chapter 4.

2.2 Review of existing Uplift modeling approaches

The uplift modeling literature and a branch of the causal inference literature have recently approached each other [29]. In this section, we review uplift approaches developed in both literatures. We divide them into two categories [77]: *Metalearners*, whose building blocks are traditional supervised ML algorithms, and *Direct approaches*, which are algorithms tailored specifically for uplift modeling.

2.2.1 Metalearners

Meta-learning, or learning to learn, is the science of systematically observing how different machine learning approaches perform on a wide range of learning tasks, and then learning from this experience [70]. Following the same idea, in uplift modeling meta-learners are a set of algorithms that exploit traditional supervised learning algorithms to estimate the CATE. One of the main advantages of these algorithms is that they are constructed by merging off-the-shelf algorithms in a specific way. They include simple and intuitive techniques such as the single-model and two-model approaches, as well as more sophisticated methods such as the X-Learner, R-Learner and DR-Learner. In this section, we will take a look at each of these approaches.

The Single-model Approach [42, 5]

As its name implies, the Single-model approach (also called the S-learner) consists of learning a single model using the treatment variable as an additional feature, without giving it a special role. In other words, the Single-model approach considers the concatenation of the treatment and the covariates (T, X) as the features. Thus a response function $\hat{\mu}(x,t)$ is defined as: $\hat{\mu}(x,t) := \mathbb{E}\left[Y \mid (X = x, T = t)\right]$. This function can be learnt using any ML algorithm. CATE estimation $\hat{\tau}(x)$ is then calculated as:

$$\hat{\tau}(x) = \hat{\mu}(x,1) - \hat{\mu}(x,0)$$

Although the *Single-model* approach is simple, it may not be able to predict the difference between the two potential outcomes (i.e. uplift). It can have very poor performance [42] and is not often used in practical problems. This is because the approach relies on a single algorithm that is trained solely to learn the estimation of the output variable. Also, in high dimensional data, the treatment variable can have a less importance for the learnt model. In addition, algorithms such as LASSO or decision trees that perform variable selection internally may not select the treatment variable during the training phase.

The Two-model Approach [31]

The Two-model approach, also known as the T-learner [42], is a simple and intuitive approach for estimating the conditional average treatment effect (CATE). The idea is to create two predictive models for the treatment and control groups to estimate $\hat{\mu}_1(x) = \mathbb{E}[Y|X, T = 1]$ and $\hat{\mu}_0(x) = \mathbb{E}[Y|X, T = 0]$. The CATE is then estimated as the difference between the predictions of these two models:

$$\hat{\tau}(x) = \hat{\mu}_1(x) - \hat{\mu}_0(x)$$

One advantage of the Two-model approach is that it can use any supervised learning algorithm to construct the predictive models. However, a problem with this approach is that it tries to predict the outcomes separately in each of the treatment and control groups, rather than the uplift itself. This can be problematic if the average response is weak or noisy. Additionally, if the data used to estimate the treatment effect is biased, the global estimator will be biased. A study on the limitations of the *Two-model approach* can be found in [57].

Inverse Propensity Weighting (IPW) [59]

The idea behind inverse propensity weighting (IPW) is to estimate the treatment effect while accounting for dependencies between the treatment and the instances' features. Using the sampling probability in the treatment and control groups, subjects are weighted by the inverse probability. That will give higher weights to instances that are under-represented in the treatment group, and lower weights for the ones who are over-represented. It's like creating a "pseudo population" in which the treatment is independent of the variables. The sampling probability in the treatment group is called the propensity score [60] denoted $e(x) = P(T = 1|X = x)$. Economists used the inverse propensity weighting for estimating the Average Treatment Effect [1](ATE). Assuming that the real propensity scores are known, the ATE for a population of size N indexed by i is defined as:

$$ATE = \frac{1}{N}\sum_i \frac{T_i Y_i}{e(X_i)} - \frac{1}{N}\sum_i \frac{(1-T_i)Y_i}{(1-e(X_i))}$$

Also, the same principle can be used for $CATE$ estimation. Then the *inverse-probability-weighted outcome* for an individual i denoted Y_i^{IPW} is:

$$Y_i^{IPW} = \frac{(T_i - e(X_i))Y_i}{e(X_i)(1-e(X_i))}$$

A regression of Y_i^{IPW} would behave as an oracle estimator of the treatment effect where counterfactuals are present in the data. However, the problem is that the propensity score is usually unknown and must be estimated from the data. As a result, Y_i^{IPW} estimate may be incorrect if the propensity score function is not correctly specified. Additionally, it can suffer from high variance, especially when the estimated propensity score is small.

[1]The Average Treatment Effect (ATE) measures the difference in the average outcome between the treated group and the control group.

Doubly-robust learner [39]

The Doubly-robust learner (also known as the DR-learner) combines the Two-model approach and the inverse propensity weighting. Data is divided into three parts of equal size. Conditional mean outcomes $\hat{\mu}_1(x) = \mathbb{E}[Y|X, T = 1]$ and $\hat{\mu}_0(x) = \mathbb{E}[Y|X, T = 0]$ are learnt on the first part of the data. Propensity scores $e(X)$ are learnt on the second part of the data. The outcome is then transformed to be :

$$Y_i^{DR-L} = \frac{T_i - e(X_i)}{e(X_i)\left(1 - e(X_i)\right)} \left(Y_i - \hat{\mu}_{T_i}\left(X_i\right)\right) + \hat{\mu}_1\left(X_i\right) - \hat{\mu}_0\left(X_i\right)$$

and regressed on the third part of the data.

This approach is called "doubly robust" because it is unbiased if either the propensity score or the conditional mean outcomes are correctly specified. However, it can be more computationally intensive to implement than other methods.

X-learner [42]

The X-Learner is a meta-learner that estimates the treatment effect separately for each treatment group. This may be helpful in case of imbalanced treatment and control conditions. The X-Learner is composed of several stages. The first one (identical to the Two-model approach) estimates the response functions $\hat{\mu}_0(x)$ and $\hat{\mu}_1(x)$ using any supervised learning algorithm. The second step consists of estimating the imputed treatment effect for each individual in the control and treatment groups, denoted by $\tilde{D}_i^0$, $\tilde{D}_i^1$ respectively. This is done by assigning treatment effects to individuals in one group based on the outcome estimator of the other group, that is:

$$\tilde{D}_i^1 := Y_i - \hat{\mu}_0\left(X_i\right), \text{where subject } i \text{ belongs to the treatment group}$$
$$\tilde{D}_i^0 := \hat{\mu}_1\left(X_i\right) - Y_i, \text{where subject } i \text{ belongs to the control group}$$

The third step is to estimate uplift in two ways: by modeling the imputed treatment effects in the treatment group and those of the control group. Thus we obtain $\hat{\tau}_1(x) = \mathbb{E}[\tilde{D}^1 \mid X = x]$ and $\hat{\tau}_0(x) = \mathbb{E}[\tilde{D}^0 \mid X = x]$. Finally, the CATE estimation $\hat{\tau}(x)$ is the weighted average of these two estimates:

$$\hat{\tau}(x) = \hat{e}(x)\hat{\tau}_0(x) + (1 - \hat{e}(x))\hat{\tau}_1(x)$$

where $\hat{e}(x) = P(T = 1 \mid X = x)$ is the propensity score estimation [60]. The advantage of the X-learner is that it combines information from the control group to estimate the treatment effect in the treatment group and vice versa. So it may perform better than the *Two-model* approach. However, it requires learning four models, increasing complexity and parameters tuning.

R-learner [52]

The R-Learner is a two step algorithm that estimates the treatment and control outcomes $\hat{\mu}(x)$ and the propensity score $e(x)$. The CATE is estimated by minimizing the following loss function:

$$L[\hat{\tau}(\boldsymbol{x})] = \frac{1}{n}\sum_{i=1}^{N}\left\{\left[\left(Y_i - \hat{\mu}^{(-i)}\left(\boldsymbol{X}_i\right)\right] - \left[T_i - \hat{e}^{(-i)}\left(\boldsymbol{X}_i\right)\right]\hat{\tau}\left(\boldsymbol{X}_i\right)\right\}^2$$

where $\hat{e}^{(-i)}$ and $\hat{\mu}^{(-i)}$ denote the out-of-fold held-out predictions made without using the ith training sample.

Class-Transformation approach [38]

The principle of this approach is to map the uplift modeling problem to a usual supervised learning problem. The outcome variable Y is transformed into a variable Z as illustrated in Eq. 2.2.1. Then a machine learning algorithm is used to learn a model and to predict $P(Z|X)$. The estimated uplift of an individual i is $\hat{\tau}_i = 2 \times P(Z = 1|X_i) - 1$

$$Z = \begin{cases} 1, & \text{if } T = 1 \text{ and } Y = 1 \\ 1, & \text{if } T = 0 \text{ and } Y = 0 \\ 0, & \text{otherwise.} \end{cases}$$

Several studies [19, 38] show that this approach has a better performance than the two-model approach. However, as shown in the experiments we conduct in Chapter 5, the Class-Transformation approach may be very sensitive to the *Non-random assignment bias* (see Section 2.5.1).

2.2.2 Direct Approaches

Unlike metalearners, direct approaches are specifically designed for treatment effect estimation and uplift modeling. Various algorithms have been proposed in the literature, such as tree-based methods, SVM-based methods, and deep learning methods. Random forest-based methods were also proposed as a natural extension of tree-based methods by combining several uplift tree models into a single uplift model. In this section, we will examine these methods, focusing on tree-based methods because they are one of the best learning approaches on tabular data, while being interpretable, which is crucial in many fields such as marketing, especially when dealing with customers, as is the case with the *Orange* Group.

Tree-based and Random forest methods

Tree-based methods for uplift modeling build decision trees for estimating the CATE or uplift. Unlike traditional decision trees, the goal is not to find leaves with pure class distributions, but to find leaves that estimate the treatment effect. The main advantage of tree-based methods lies in their interpretability, which is very important for many applications. However, a major drawback is that the induction of an optimal uplift decision tree from a data set is NP-hard [51], and the tree-learning process is usually greedy.

To overcome the issue of high variance in decision trees, random forests have been proposed for uplift modeling. They typically perform better when a large number of trees are included in the forest. However, unlike decision trees, random forests lack interpretability.

Uplift Decision Tree algorithm [63] is similar to traditional decision tree algorithms in machine learning, but it introduces a new splitting criterion based on information theory. It can handle multiple treatments and an arbitrary number of classes. The proposed splitting criterion is based on distribution divergences, with the goal of maximizing the differences between the class distributions in the treatment and control sets. For each non-leaf node, the criterion for a split test is calculated as follows:

$$D_{\text{gain}}(A) = D\left(P(Y|T=1) : P(Y|T=0) \mid A\right) - D\left(P(Y|T=1) : P(Y|T=0)\right)$$

where A represents a split test performed on a non-leaf node (for example $x < v$, where v is a real number, presents a test A) and where $D(p : q)$ is a divergence measure between two probabilities p and q such as Kullback-Leibler divergence (KL), squared Euclidean Distance (ED), Chi-squared divergence (Chi). The criterion selects a test that leads to the most divergent class distributions in each branch. The gain $D_{\text{gain}}(A)$ obtained from a test A is calculated by subtracting the divergence between class distributions on the entire dataset from the divergence between class distributions due to test A. The class divergence between treatment and control groups of a test A is simply the sum of the class divergence between treatment groups for each value of the test A (each child node, denoted by a). In other words:

$$D\left(P(Y|T=1) : P(Y|T=0) \mid A\right) = \sum_a \frac{N(a)}{N} D\left(P(Y|T=1,a) : P(Y|T=0,a)\right)$$

where $N(a)$ denotes the number of instances for the outcome of the test A is a. The authors argued that the ED divergence measure may perform better since it's symmetric and more stable.

A normalization step is performed in order to prevent bias toward tests with a large number of outcomes and tests that tend to separate treatment and control groups. The algorithm is followed by a pruning step to avoid overfitting. Additionally, a number of parameters must be set, such as the maximum depth and the minimum number of samples required to perform a split.

Uplift Incremental Value modeling (UpliftIVM) [31] is one of the earliest tree-based uplift approaches proposed in the uplift modeling community. Unlike the Uplift decision tree approach that tries to maximize the estimated treatment effect in each child node, the UpliftIVM tries to maximize the difference between the treatment effect of the left and right child nodes. UpliftIVM searches for the split s that maximizes the following splitting criterion:

$$\Delta\mu(s) := |\hat{\tau}_L - \hat{\tau}_R|$$

Each of $\hat{\tau}_L$ and $\hat{\tau}_R$ are estimated as the treatment effect in the left and right nodes. More precisely:

$$\hat{\tau}_L = \frac{\sum_{i=1}^{n_L} T_i Y_i}{\sum_{i=1}^{n_L} T_i} - \frac{\sum_{i=1}^{n_L} (1 - T_i) Y_i}{\sum_{i=1}^{n_L} (1 - T_i)},$$

and

$$\hat{\tau}_R = \frac{\sum_{i=1}^{n_R} T_i Y_i}{\sum_{i=1}^{n_R} T_i} - \frac{\sum_{i=1}^{n_R} (1 - T_i) Y_i}{\sum_{i=1}^{n_R} (1 - T_i)}$$

where n_L and n_R denote the number of instances in the left and right child nodes.

Causal trees [4] are different from traditional decision trees in that they are designed to be an *honest* approach. According to [72], an *honest* approach requires that we do not use the same set of data to both learn and to conduct inference. In a causal tree, the training data (of size n_s) is used to create tree splits (to build the entire tree) and the estimation data is used to estimate the uplift values in the leaves. When learning a causal tree, the goal is to find the split s that maximizes the following splitting criterion:

$$\Delta\mu(s) := \underbrace{\left(\frac{n_L}{n}\hat{\tau}_L^2 + \frac{n_R}{n}\hat{\tau}_R^2\right)}_{\text{Rewards treatment effect heterogeneity}} - \underbrace{\left(\frac{1}{n} + \frac{1}{n_s}\right)\left(\frac{S_{1L}^2}{p} + \frac{S_{0L}^2}{1-p} + \frac{S_{1R}^2}{p} + \frac{S_{0R}^2}{1-p}\right)}_{\text{Penalizes splits leading to small leaf nodes}}$$

where S_{0L}, S_{1L}, S_{0R} and S_{0R} denote sample variances in treatment and control groups in each of the left and right leaves. p denotes treatment probability in the data. The CATE estimation in a Causal tree is done using inverse propensity score in each leaf node.

The approaches mentioned previously belong to the category of tree-based methods. In the literature, forest-based methods have also been developed as a natural extension of these approaches [67]. Typically, a forest is constructed by combining multiple trees and then computing their average predictions.

Causal forests [72] Causal forests is a random forest algorithm that uses Causal trees as its base learner. Similar to random forest-like algorithms, k causal trees are trained and then used to provide a treatment effect estimation $\hat{\tau}_t(x)$ for each example x in a test set. The prediction of the Causal forest is then the average of the predictions provided by the Causal trees, i.e., $\hat{\tau}(x) = \frac{1}{k}\sum_t \hat{\tau}_t(x)$. The authors showed the estimations of the causal forests are asymptotically Gaussian and unbiased.

The Contextual Treatment Selection (CTS) [79] algorithm is a random forest algorithm designed to directly maximize a new performance measure called *the expected performance* through its splitting criterion.

- Expected response: The expected response is an evaluation measure where multiple treatments and/or outcomes can be considered. It defines a new random variable z_i, such that:

$$z_i = \sum_{k=1}^{N} \frac{Y_i}{P_{T=k}} \mathbb{I}\left\{h\left(x_i\right) = k\right\} \mathbb{I}\{T = k\}$$

  where $P_{T=k}$ is the prior probabilities of the treatment and $h\left(x_i\right)$ is an uplift model resulting in the optimal treatment and $\mathbb{I}\left(.\right)$ is the Iverson bracket (the 0/1 indicator function), equal to one if the predicted optimal treatment is equal to the assigned treatment, and zero otherwise.

  When the predicted optimal treatment is equal to the observed treatment,z_i becomes equal to the outcome scaled by the probability of the treatment. Thus the expected response of an uplift model is then the expectation of z_i, $\mathbb{E}[z_i]$, calculated as follows:

$$\mathbb{E}[z_i] = \mathbb{E}[Y \mid T = h(x_i)] = \frac{1}{N}\sum_{i=1}^{N} z_i$$

- Splitting criterion: Suppose s is a candidate split that divides a space ϕ into left and right subspaces, resp. ϕ_l and ϕ_r. The goal is to perform the split s that leads to the greatest increase in the expected response. The increase in expected response is calculated as:

$$\begin{aligned}\Delta\mu(s) =& P\left\{\mathbf{X} \in \phi_l \mid \mathbf{X} \in \phi\right\} \max_{t_l=0,\ldots,K} \mathrm{E}\left[Y \mid \mathbf{X} \in \phi_l, T = t_l\right] \\ &+ P\left\{\mathbf{X} \in \phi_r \mid \mathbf{X} \in \phi\right\} \max_{t_r=0,\ldots,K} \mathrm{E}\left[Y \mid \mathbf{X} \in \phi_r, T = t_r\right] \\ &- \max_{t=0,\ldots,K} \mathrm{E}[Y \mid \mathbf{X} \in \phi, T = t]\end{aligned}$$

  Note that we subtract the maximum response of the parent node, to be able to calculate the gain achieved by s.

Note that all splits in the CTS approach are binary splits. Similarly to the Uplift decision tree approach, a number of parameters should be set by the user, such as the minimum number of samples required to split a node and a regularization term.

Other forest-based methods were also proposed. [27] studied decision trees for uplift modeling and pointed to their high variance problem. So, they presented the *Uplift Random Forest* algorithm. Later, [28] presented the *Causal Conditional Inference Forest* to solve both the variable selection bias in the splitting criterion and the overfitting problem of the *Uplift Random Forest* algorithm. [67] performed an extensive study on ensemble methods and proposed a bagging algorithm for uplift modeling.

Support vector machine-based methods

The support vector machine algorithm has been adapted for uplift modeling problems [76, 75]. [76] proposed an SVM-based approach called L_1-*USVM*, where the main idea is to use two parallel hyperplanes that divide the sample space into three regions of different treatment effects: positive, neutral and negative treatment effects. In this way, the uplift modeling problem becomes a three-class classification problem. The two hyperplanes are:

$$H_1 : \langle \mathbf{w}, \mathbf{x} \rangle - b_1 = 0, \quad H_2 : \langle \mathbf{w}, \mathbf{x} \rangle - b_2 = 0$$

where b_1, b_2 are the intercepts and w is the normal vector to the hyperplanes. The CATE predictions are then obtained according to the following equations:

$$\hat{\tau}(x) = \begin{cases} +1 & \text{if } \langle \mathbf{w}, \mathbf{x} \rangle > b_1 \text{ and } \langle \mathbf{w}, \mathbf{x} \rangle > b_2, \\ 0 & \text{if } \langle \mathbf{w}, \mathbf{x} \rangle \leq b_1 \text{ and } \langle \mathbf{w}, \mathbf{x} \rangle > b_2, \\ -1 & \text{if } \langle \mathbf{w}, \mathbf{x} \rangle \leq b_1 \text{ and } \langle \mathbf{w}, \mathbf{x} \rangle \leq b_2 \end{cases}$$

The same authors then proposed L_p-*USVM* [75], which extends L_1-*USVM* by using the L_p norm for the regularisation for w instead of the L_1 norm. Improved optimisation algorithms were also proposed in the same paper. According to [75], L_p-*USVM* does not suffer from discontinuity problems unlike L_1-*USVM* and improves convergence and efficiency.

The main drawback of this approach is its complexity due to the additional hyperplane and its variables.

Deep Learning-based methods

Deep learning based methods have also been proposed in the literature. We briefly review the different contributions. *The Causal Effect Variational Autoencoder (CEVAE)* proposed by [47], is a neural network latent variable model for causal effect estimation. It learns a latent set of confounders from the observed covariates. *The Treatment Effect with Disentangled Autoencoder (TEDVAE)* [78] improves CEVAE by taking into account not only the confounding variables that are correlated with both the treatment and outcome variables (see Section 2.5.1), but also the instrumental factors that affect only the treatment and the risk factors that affect only the outcome. *Generative Adversarial Network for Individualised Treatment Effects (GANITE)* [74] attempts to learn the counterfactual distributions using a Generative Adversarial Network (GAN) while generating CATE estimates for the instances. *Counterfactual Regression (CFR)* [64] extends the two-model approach. First, a representation learning is performed to minimise the discrepancy between the two distributions $P(X|T = 1)$ and $P(X|T = 0)$, then two neural networks are trained on each of the treatment and control groups to estimate the CATE. *DeepTreat* [3] is a single-model based approach. First, it consists of a bias-removing auto-encoder to control the trade-off between bias reduction and information loss. It learns a new representation for the covariates X where the treatment groups are balanced. It then trains a single neural network to predict the outcome Y using the concatenated features and the treatment variables.

Deep learning-based methods excel at learning large datasets, but are less effective for small datasets (as in medical applications). In addition, they lack interpretability, which is sometimes crucial in some applications. Their parameters are also difficult to tune.

2.3 Evaluation metrics

An important part of building a machine learning model is being able to evaluate it. An evaluation metric is used to evaluate the predictive performance of a

learning algorithm. They help to evaluate model results in order to select the best model among several. For example, in supervised learning, such as classification or regression, metrics like the F1 score, accuracy, area under the curve (AUC) can be used. They all depend on two values for each instance in the data set, the predicted value and the actual value.

However, as mentioned earlier, one of the main problems with uplift modeling is that the actual uplift values cannot be observed. We cannot simultaneously observe both outcomes for a given individual with and without treatment. That's why performance measures of the supervised setting are inoperative.

In this section we present the performance measures used in the case of uplift modeling to assess the quality of the predicted treatment effects.

2.3.1 Group-level uplift based metrics

Uplift per decile [46, 68] Evaluating the quality of the estimated uplift values is a challenge, as it is not possible to do it directly or define a loss function. One solution is to rank the instances based on their estimated uplift values and evaluate the resulting ranking. Since comparing true uplift values is infeasible, researchers proposed comparing the estimated uplift within bins or groups. Specifically this is done by sorting the instances in descending order by their predicted uplift values (separately for each of the treatment and control groups), then dividing them into deciles, and calculating uplift per decile d,denoted $\hat{\mu}(X_d)$, such that:

$$\hat{u}\left(X_d\right) = \mathbb{E}\left[Y \mid X_d, T = 1\right] - \mathbb{E}\left[Y \mid X_d, T = 0\right]$$

where X_d denotes the individuals of a particular decile d.

The predictions of a good uplift model will yield a decreasing uplift-per-decile bins as presented on the left figure of Fig. 2.2. On the contrary, a bad uplift model will yield increasing uplift-per-decile bins or random bins as shown in the right figure of Fig. 2.2.

The uplift-per-decile chart can help practitioners (for example in the marketing field) to target the subjects in the first deciles (with higher predicted uplift values) since they are persuadables and to avoid subjects in other bins. However, the uplift-per-decile chart does not allow to compute the *Gain* of uplift targeting a particular ratio of subjects and does not give us a numerical evaluation metric for uplift models.

Qini Curve The Qini curve was first proposed in [56] to plot the absolute incremental responses of the treated group compared to the control group. Let D_T, D_C be respectively the treatment and control groups, ordered by the predictions of an uplift model; N_T, N_C be the total number of instances in D_T and D_C

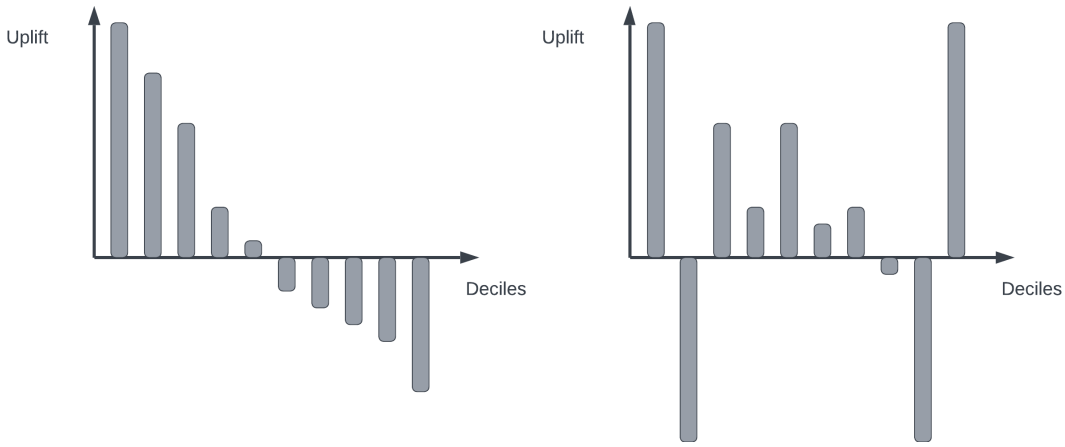


Figure 2.2: Results of a good uplift model (left) and a bad uplift model (right) represented by an uplift-per-decile chart.

respectively; $R^T(k)$ and $R^C(k)$ be the number of treated and control responders, respectively, among the top k percent of instances in each of D_T and D_C. The values of the Qini curve $V(k)$ are then obtained by varying the value of k between 0 and 100, such that:

$$V(k) = R^T(k) - R^C(k)\frac{N_T}{N_C}$$

Let's take an example of qini curves similar to the one presented by [56]. Assume a dataset containing 100K treated individuals and 100K non-treated individuals. The number of positive responses in the treatment group is 30K against 10K in the control group. Fig. 2.3 shows two Qini curves presenting the performance of two different uplift models, a Qini curve V_{random} showing the performance of the random model [56] and a Qini curve V_* showing the performance of an optimal uplift model. A random model is the model that assigns treatment randomly to subjects. An optimal model (in yellow) assigns higher scores to all treated responders than all non-responders. It is a theoretical curve that assumes that all treated responders have positive outcomes because of the treatment. Thus it climbs at 45°, assuming that positive outcomes are due to the treatment, then proceeds horizontally and finally goes down due to the negative effects of the treatment.

By targeting the top 50% of instances according to model A, the gain will be 20K, while targeting the top 50% of instances according to model B the uplift gain will be 25K It's clear that the uplift model B outperforms model A. Note that targeting the whole population will yield a gain equivalent to the average treatment effect (ATE), which is 20K in our case.

Several variants of the qini curve were used in the literature. Also the uplift curve [62], which is quite similar to the qini curve, was introduced.

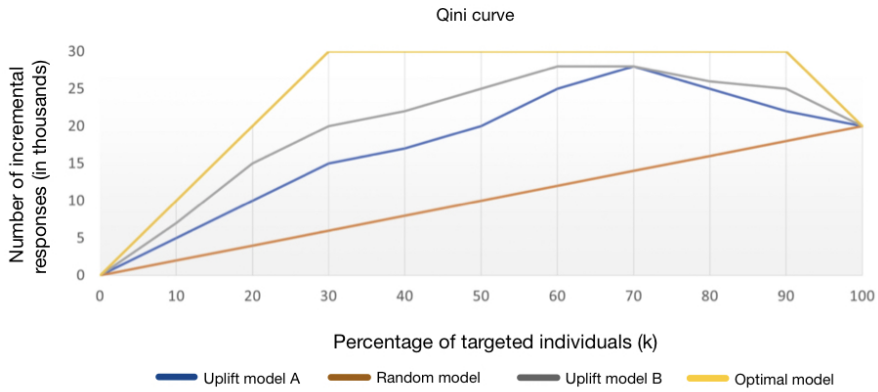


Figure 2.3: Example of qini curves. X-axis shows the percentage of targeted individuals k, sorted by their predicted uplift values. Y-axis present the percentage of the cumulative uplift.

Uplift curve As just mentioned, the uplift curve is an another variant of the qini curve that was also widely used in the literature [62, 38, 50, 67]. To our knowledge, [62] were the first to introduce the uplift curve. It is obtained by subtracting the lift curve obtained on the control set from the lift curve obtained on the treatment set. The values of the uplift curve can then be calculated by:

$$U(k) = R^T(k) - R^C(k)$$

In [63], the authors renormalized the curves by the number of examples of their respective treatment groups. So $U(k)$ can then be calculated by:

$$U(k) = \frac{R^T(k)}{N_T} - \frac{R^C(k)}{N_C}$$

To compare several uplift models, we need a numerical value describing the performance of each model. Mainly, two measures were proposed in the literature, named the *Qini coefficient* and the *AUUC*:

- **Qini Coefficient** Q [56] The qini coefficient is a generalization of the Gini coefficient. It provides an evaluation of how far the qini curve $V(k)$ is from the random curve V_{random} and how close it is from the optimal curve V_*. The qini curve is generated by varying k from 0 to 100. Subsequently, the

area under the qini curve, denoted as $AUC(V)$, can be computed. The qini coefficient Q of an uplift model can then be calculated as follows:

$$Q = \frac{AUC(V) - AUC\left(V_{random}\right)}{AUC\left(V_{*}\right) - AUC\left(V_{random}\right)}$$

The Qini coefficient will be referred to as 'qini' or 'qini value' throughout the remainder of the thesis.

- **Area Under the Uplift Curve (AUUC)** [62] As the name implies, it is area under the uplift curve.

$$AUUC = \int_{0}^{100} U(k)dk$$

Sometimes the area under the diagonal line is subtracted from this quantity [19, 38]. Several variants of the metrics described above were proposed in the literature. For the interested reader, please refer to [18].

2.3.2 Precision in the Estimation of Heterogeneous Effects (PEHE)

When the ground truth uplift values are observed, which happens in synthetic datasets, where data are simulated, the precision in the estimation of heterogeneous effect (PEHE) [32] can be used. It acts as a mean-squared error (RMSE) for uplift problems.

$$\text{PEHE} = \frac{1}{n}\sum_{i}^{n}\left(\hat{\tau}\left(\boldsymbol{x}_i\right) - \tau\left(\boldsymbol{x}_i\right)\right)^2$$

2.4 Feature selection for uplift models

The telecommunications industry collects huge amounts of data [14]. Data that can be collected includes the location of phones, call records, timestamps and call duration. In addition, SMS records such as SMS length, frequency and timestamps can be collected. The industry also collects information about internet usage, such as user IDs, the types of websites visited and the number of bytes transferred. It often has hundreds to thousands of features. This may cause serious challenges to machine learning models such as the curse of dimensionality [6]. Large number (and often noisy) features may lead machine learning algorithms to overfitting and decreased performance. In addition, interpretability of machine learning decisions in the presence of a large number of features is not practical.

That's why dimensionality reduction techniques, including feature selection, is a crucial step in machine learning for improving the efficiency of models. Feature selection help to simplify models, making them easier to interpret, reducing computation power, cost and training time, and facilitating data visualization.

The literature of feature selection is extensive [30]. In this section, we will first provide a high-level overview of the main feature selection families of methods, explain why they are not well-suited for uplift modeling problems, and then take a look at the contributions of feature selection methods specifically for uplift modeling.

2.4.1 Feature selection in classical machine learning

Feature selection is a very large research domain that was Different feature selection techniques were designed for supervised, unsupervised and semi supervised learning. Supervised feature selection can be broadly categorized as filter, wrapper, and embedded methods. Filter methods select subsets of variables as a preprocessing step, independently of the chosen machine learning algorithm. Wrapper methods evaluate subsets of features, allowing for the evaluation of interactions between different variables. Embedded methods perform feature selection as part of the learning process; for example, decision trees such as CART [13] perform internal feature selection, and LASSO method constructs linear models while penalizing regression coefficients. Other feature selection techniques were also developed for unsupervised learning. A review of unsupervised feature selection can be found in [1]. Finally, when high-dimensional data is available with only a subset of labeled samples, semi-supervised feature selection techniques [80] were developed to face a new challenge.

A new problem, a new challenge Conventional feature selection techniques try to find the most relevant features for outcome prediction. However, as previously presented, in uplift modeling we try to estimate the difference between two outcome distributions, which make traditional feature selection techniques innoperative.

2.4.2 Feature selection for uplift modeling

To the best of our knowledge, only two articles in the uplift modeling literature that discuss feature selection. Zhao et al. [81] propose filter and embedded feature selection methods for uplift. They compare them with conventional feature selection methods. Their results show that traditional feature selection approaches are

not effective in the uplift modeling context. In [34], the author suggests performing a second step in the feature selection process to remove redundant features. The article calculates a correlation coefficient between each variable and all other variables. If two features have a correlation coefficient greater than 0.8, one of them is removed. While removing redundant features is an important step, this approach may not be feasible on real data due to computational constraints.

Following, we briefly present the filter and embedded methods that were proposed by [81].

Filter methods

Filter methods are used in a pre-processing step independently of an uplift model.

F-Filter The F filter method uses the *F statistic* to test the significance of the interaction between the treatment variable and a feature in a linear regression. An interaction in linear regression occurs when the effect of an independent variable (for instance a feature X) on the outcome variable Y depends on another independent variable (in our case the treatment variable T). In order to capture non-linear interactions, the authors extended this approach by adding higher-order terms of the feature in the regression. More specifically, a linear regression model with interaction is presented as:

$$Y = \alpha + \delta T + \sum_{r=1}^{R} \beta_r X^r + \underbrace{\sum_{r=1}^{R} \theta_r T X^r}_{\text{The interaction term}} + \epsilon$$

where X is the feature for which a score is to be calculated. α, δ, β and θ are the coefficients and ϵ represents the error term. R is the higher-order term (a hyperparameter set by the user). Since θ is the coefficient of the interaction term, its significance indicates the strength of the treatment effect for the feature X. To examine the significance of θ, we should contrast the model with interaction above with the linear regression model without an interaction term. A linear regression model without interaction can be presented as:

$$Y = \alpha' + \delta' T + \sum_{r=1}^{R} \beta'_r X^r + \epsilon'$$

The F-filter feature selection calculates the importance score of a feature X as the F-statistic for the coefficient of the interaction term θ:

$$F = \frac{(RSS - RSS'/R)}{RSS'/(N - R - 2)}$$

where RSS and RSS' are the Residual Sum of Squares for the fitted model with and without interaction respectively. The value of F can be used with an F-distribution calculator with degrees of freedom $(R, N - R - 2)$ to calculate a p-value. The null hypothesis states that there is no statistical difference between the linear regression model with and without interaction.

Likelihood ratio (LR) filter Similar to the F-filter, the LR filter uses the *likelihood ratio test statistic* for the interaction coefficient in a logistic regression. The likelihood test static measures whether adding a parameter to our model , e.g. the interaction term, will make our model fit the data significantly better. Again, this is achieved by comparing the models that include the interaction term with those that do not.

Bin-based divergence Filter Bin based Divergence filter approach comes from the split criteria of the uplift trees. The bin-based method first divides a feature into equally sized bins then estimates the divergence between the outcome distributions in each of the treatment groups. Let p_i and q_i denote the outcome distribution in each of the treatment and control groups respectively for the bin i:

$$\sum_{i=1}^{I} \frac{N_i}{N} D(p_i : q_i)$$

where N_i is the number of instances in the bin i and D corresponds to a distribution divergence measure. Three measures were used: the Kullback-Leibler divergence (KL), the squared Euclidean Distance (ED) and the chi-squared divergence (Chi).

Again, the number of bins is a hyperparameter to be set by the user.

Embedded methods

Embedded methods for feature selection generate importance scores by training an uplift model. The authors proposed to generate importance scores from an uplift decision tree model. At each split, the gain in the distribution divergence is calculated:

$$\Delta = \sum_{k \in \{\text{ left, right }\}} \frac{n_k}{n} D\left(p_k, q_k\right) - D(p, q)$$

where n is the number of instances in the parent node.

The importance score of a feature X can be calculated by summing all the gains of all the splits where X was used.

The filter feature selection approaches described above are all parametric, while embedded methods can be time consuming as they rely on the performance of an uplift model. The authors showed that bin-based approaches outperformed other methods. The F and LR filters showed improved performance when the hyperparameter R was set to a value greater than 1, which allows non-linear patterns in the data to be detected.

2.5 Biases in uplift modeling

Most machine learning algorithms work well when the training and test data come from the same distribution. They guarantee their performance if the deployment data has the same distribution as the data on which the learning algorithm was trained. However, when the distributions are different, the performance of these models deteriorates [40]. In other words, a machine learning model will not "generalise" well if the training data does not reflect the population on which the model is tested. This phenomenon is also called *data bias*.

To overcome this, it's important to ensure that the training data matches the distribution of the test data by collecting new data. However, in many applications it is impossible or expensive to collect new data to rebalance the training and test sets [54]. That's why research areas such as *domain adaptation* have grown to develop techniques to bring the training and test distributions closer together [40].

Like any other machine learning algorithm, uplift modeling can also be prone to data bias. Additionally, as it is a distinct problem from conventional supervised learning, there are specific types of bias that are unique to the uplift modeling problem.

In this section we discuss different sources and types of bias that can be observed in an uplift problem. We divide them into two categories: *modeling bias* and *deployment bias*. As their names imply, a *modeling bias* occurs during the training phase of an uplift model while the *deployment bias* occurs during the deployment of the uplift model in a real world application.

2.5.1 Modeling bias

Modeling bias typically occurs during the training phase of an uplift model. In principle, the ideal scenario for performing uplift modeling is within a randomised control trial setting, where the data is generated under controlled conditions and biases are minimised between the data from different treatments. However, in practical applications, data is collected from observational studies, i.e. gathered

without the subject of a research experiment, where the data generation process is not controlled, and biases are often present.

In this section, we present three types of modeling bias: non-random assignment bias (NRA), non-response bias and confounding variables.

Non-random assignment bias

Non-random assignment (also known as selection bias) [77] happens when there are differences between treatment and control groups. Formally, this bias occurs when $P(T = 1|X) \neq P(T = 0|X)$ (which also means $P(X|T = 1) \neq P(X|T = 0)$). This can be seen as a *covariate shift* (see Section 2.5.2) between treatment and control groups. Usually it is easier to collect control data and the treatment group is the most biased because it is more challenging to apply a treatment to individuals and collect the corresponding data due to ethical, political or economic constraints.

This bias problem has been studied in the literature on clinical studies where the goal is to estimate the "Average Treatment Effect" (ATE) defined as $\mathbb{E}[Y_i(T = 1) - Y_i(T = 0)]$. In order to estimate it, the "Propensity Score Matching" (PSM) [61] is used to extract balanced treatment groups on which ATE is estimated. Similarly, in the uplift literature, since uplift methods assume the homogeneity between treatment groups, PSM is used to extract an unbiased sample from a biased dataset. Uplift modeling is applied subsequently as carried in [53]. However, this procedure clearly suffers from a loss of data.

This type of bias is discussed and an experimental evaluation of its effect is carried out in Chapter 5.

Non-response bias

Let's consider a marketing campaign for an internet offer. The treatment group receives the campaign by e-mail, while the control group is not contacted. Non-response bias occurs when some individuals in the treatment group are considered to have received the treatment, however, they did not really receive it. For instance, some individuals do not check their emails regularly, and hence they did not read the received email. If there is a common pattern among the people who do not read the email, this leads to non-response bias. For instance, if all older persons in the treatment group do not check regularly their emails, we cannot attribute their buying behaviour to the assigned treatment.

This type of bias is called non-response bias and refers to the case where individuals do not respond to the treatment (e.g. do not answer the phone call, do

not read the received email or sms). These individuals are part of the treatment group even though they did not actually receive the treatment.

Non-response bias is a phenomenon that also occurs in medical contexts, particularly in clinical trials. For example, consider a clinical trial in which researchers give a drug to one group of people (the treatment group) and a placebo to another group of people (the control group). Some people may not consume the drug, perhaps because they experience more unwanted side effects. This can lead to non-response bias.

The pattern in which non-response occurs can have a significant impact on the data, introducing noise. Rubin's noise taxonomy [44] provides a systematic approach to classifying different types of noise and missing data. Through this lens, non-response bias is identified as a form of noise that can be categorized:

- *Missing not at random (MNAR)*: happens when the non-response depends on unobserved causes (not included in the variables).
- *Missing at random (MAR)*: occurs when non-response is conditional on covariates X. In other words, when the probability of being a non-response depends only on the observed attributes.
- *Missing completely at random (MCAR)*: occurs when the non-response is completely independent of the initial variable and the covariates. Thus, subjects with MCAR non-response are a random sample.

The specific patterns in which non-response bias occurs can affect the data in distinct ways, presenting unique challenges in each case.

Confounding variables

Confounding variables are not exactly a type of bias, but can be considered a problem in treatment effect estimation. They violate the conditional independence assumption (or unconfoundedness assumption) described in Section 1.3. Confounding variables are variables that are correlated with both the outcome and the treatment variables. In other words, when individuals who are more likely to have a particular outcome are more likely to receive a treatment. As an example [55], let's take a medical study investigating the relationship between coffee drinking and lung cancer. If the people (coffee drinkers) in the treatment group of the study were also smokers (without taking the effect of smoking into account), the study might conclude that coffee drinking increases the risk of lung cancer.

According to [55, 37], there are 3 methods to reduce the effect of confounding variables and distribute them evenly between the treatment groups:

- **Randomization** helps prevent selection bias. It consists of randomly assigning subjects to treatment and control groups.
- **Matching** can be used to match individuals from the treatment group with another from the control group based on one or more selection criteria. For example if age, sex and eating habits are the matching variables then a vegeterian male of 25 years in the treatment group is matched with another from the control group. Also, techniques like *Propensity score matching* can be useful. This was also illustrated in an uplift modeling benchmark study by [53]
- **Restriction** is used especially in clinical trials. If the confounder variable is known, the idea is to eliminate variations in the study by restricting the study for example on the same age or sex for example.

In summary, modeling bias primarily arises from the data used for training an uplift model. They can be caused by differences in the distribution of the treatment and control groups, as is the case with non-random assignment, by the presence of confounding variables or non-response bias. Such biases have the potential to yield poor learning models and erroneous interpretation of the impact of a treatment on the behavior of subjects.

2.5.2 Deployment bias

As with any machine learning problem, uplift modeling can face the problem of deployment bias. Deployment bias occurs when the available data is not representative of the general population, also known as "data shift" [40, 48]. A data shift occurs between a source domain (where we have access to training data) and a target domain (where we apply our machine learning model and where labels are typically not available). Reasons for data shifts can include an outdated training set, different or limited data sources, sample selection bias, or changes in the behaviour of individuals.

Before proceeding, note that domain-specific functions given below are denoted by the subscripts S and Ta for the "source" and "target" domains, respectively. For example, $P_S(X|Y)$ and $P_{Ta}(X|Y)$ denote the source and target class conditional distributions, respectively. The most common data shifts are:

- **Prior shift**: refers to changes in the distribution of the output variable. A prior shift occurs when the prior probabilities of the classes are different, i.e. $P_S(Y) \neq P_{Ta}(Y)$, while the posterior distributions are equivalent, $P_S(X|Y) = P_{Ta}(X|Y)$.

- **Covariate shift**: refers to the case where $P_S(X) \neq P_{Ta}(X)$ while $P_S(Y|X) \neq P_{Ta}(Y|X)$. It most commonly occurs when there is a sample selection bias. For example, a face recognition algorithm that has been trained on young faces, but is used on a dataset of older faces, will suffer from the covariate shift problem. The relationship between input and output is the same, but the training data is not representative of the population of interest.

- **Concept shift** (also referred to as "concept drift" [24]) refers to the change in relationships between input and output variables. For example, on a property platform where users mark each listing as "interesting" or "not interesting", changes in the economic situation may change consumers' buying habits over time. In this case, it is not the data distribution or the class distribution that has changed, but the relationship between the data and class variables. Formally, a concept drift occurs if one of the following situations occurs:

  - $P_S(Y|X) \neq P_{Ta}(Y|X)$ and $P_S(X) = P_{Ta}(X)$
  - $P_S(X|Y) \neq P_{Ta}(X|Y)$ and $P_S(Y) = P_{Ta}(Y)$

It is also related to *data drift*, where models are trained and deployed *online* in non-stationary environments.

To summarize, deployment bias can arise when the distribution of labels, covariates, or their relationship differs between the training and deployment data. It is also possible for multiple types of shifts to occur simultaneously. Deployment bias is not specific to uplift modeling problems, it is a commonly recognised issue in various domains of machine learning.

2.6 MODL: Minimum Optimized Description Length

This thesis proposes an uplift discretization approach and a new Bayesian decision tree algorithm for uplift modeling based on the Minimum Optimized Description Length (MODL). MODL is a Bayesian approach for density estimation through discretization for supervised learning. It is founded on the Minimum description length (MDL) principle.

In this section, we introduce the MDL principle and present the MODL discretization approach [8] and the MODL decision trees [71] as preliminaries for the rest of the thesis.

2.6.1 MDL: Minimum Description Length principle

The Minimum Description Length (MDL) principle [58] is derived from Shannon's information theory [65] and allows finding the simplest model that best describes the data. The MDL principle is used to select the best model, among a family of models, by taking into account the complexity of the models and the complexity of the data according to the model. According to the MDL principle, the best model $\mathbb{M}$ that describes the data D is the model with the minimum description length $L(M, D)$, s.t. $L(M, D) = L(M) + L(D|M)$, where $L(M)$ is the model's description length and $L(D|M)$ is the description length of the data encoded by the model.

2.6.2 The MODL approach for discretization

The MODL (Minimum Optimized Description Length) [8] approach is a non-parametric Bayesian approach for discretization and conditional probability estimation, based on the Minimum Description Length (MDL) principle. Let us first introduce the link between a Bayesian and a MDL model selection problem.

A Bayesian approach for model selection From a Bayesian perspective, the best model M, among a family of models, is found by maximizing the posterior probability $P(M|D)$, i.e., to find the one that is most likely given the data. Using Bayes rule, maximizing $P(M|D)$, while taking into account that $P(D)$ is constant for all the candidate models, is equivalent to maximizing the product of the prior and the posterior probabilities: $P(M)P(D|M)$.

A MDL approach for model selection The previous approach can also be seen from an information theory perspective by using the MDL approach [58]. As previously mentioned, the goal of the MDL approach is to select the model with the minimal description length $L(M|D)$. Replacing the previously introduced probabilities by their negative log, allows interpreting them as Shannon's code length [65], s.t.: $-\log P(M|D)$ corresponds to $L(M|D)$.

Parameters of a discretization model The MODL approach comes to apply the MDL approach to the discretization problem to help find the best discretiza-

tion model M that maximizes the posterior probability $P(M|D)$. First the data values of a variable are sorted. Then a space of discretization models is defined. A discretization model is described by a set of parameters:

- the number of intervals I
- the boundaries of the intervals, i.e. the number of instances in each interval i denoted N_i
- the frequencies of the classes in each interval, i.e., the number of instances in each interval i with class j denoted N_{ij}

An Evaluation criterion for a discretization model Using these parameters, the MODL approach defines a prior distribution on a discretization model M. It exploits the hierarchy of the parameters and assumes a uniform distribution at each stage of the hierarchy with independence across intervals. The MODL approach defines then the cost of a model $C(M)$, i.e., which is the negative log of the posterior probability by:

$$C(M) = \log N + \log \binom{N+I-1}{I-1} + \sum_{i=1}^{I} \log \binom{N_i+J-1}{J-1} + \underbrace{\sum_{i=1}^{I} (1-W_i) \log \frac{N_i!}{N_{i.1}!..N_{i.J}!}}_{Likelihood}$$

Using a search algorithm the MODL approach can score all possible discretization models and selects the one with the minimal criterion.

2.6.3 The MODL classification trees

The MODL approach can also be applied to classification trees [71]. It allows for the definition of a global criterion for a decision tree model, enabling the selection of the tree model with the minimal tree criterion among a family of tree models. It is distinguished from other tree approaches by being a user parameter-free approach and by defining a global criterion to evaluate a tree model.

As a Bayesian approach, it tries to select the most probable tree model given the data by maximizing the posterior probability $P(Tree|Data)$ of a tree model $Tree$. This is achieved by maximizing the product of the prior probability of the tree and the likelihood of the data given the model, i.e., $P(Tree)P(Data|Tree)$.

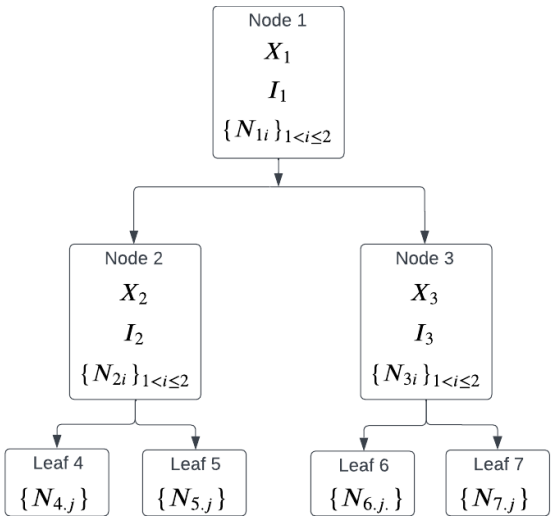


Figure 2.4: Parameters of a MODL decision tree

A decision tree model is defined by its structure, the distribution of instances in this structure and the distribution of the class values:

- The structure consists of:
  - the set of internal nodes $\mathbb{S}_{Tree}$ (an internal node is a node with at least two children).
  - the set of of leaf nodes $\mathbb{L}_{Tree}$
  - the subset of variables $\mathbb{K}_{Tree}$ used by $Tree$ chosen among K variables in the dataset, where K_{Tree} is the number of variables in $\mathbb{K}_{Tree}$.
- The distribution of instances in the structure is described by:
  - the choice of the test variable X_s (also called segmentation variable) for each internal node s
  - its number of partitions I_s, where $I_s > 2$ for all internal nodes.
  - the distribution of instances in each partition i for each internal node s, denoted by: $\{N_{si.}\}_{1\leq i\leq I_s}$
- The distribution of the classes in the structure is defined by the class frequency in each leaf node l denoted by: $\{N_{l.j}\}_{1\leq j\leq J}$

An Evaluation criterion for a tree model Using these parameters, the prior distribution and the likelihood of $Tree$ are defined. An evaluation criterion $C(Tree)$ is then presented as the negative logarithm of the posterior probability, s.t.:

$$\begin{aligned}
C(Tree) &= -\log P(Tree)P(Data|Tree) \\
&= \log(K+1) + \log \begin{pmatrix} K + K_{Tree} - 1 \\ K_{Tree} \end{pmatrix} + \\
&+ \sum_{s \in \mathbb{S}_{T_n}} \log K_{Tree} + C_{Ris}\left(I_s\right) \log 2 + \log \begin{pmatrix} N_{s.} + I_s - 1 \\ I_s - 1 \end{pmatrix} + \\
&+ \sum_{s \in \mathbb{S}_{T_c}} \log K_{Tree} + C_{Ris}\left(I_s\right) \log 2 + \log B\left(V_{X_s}, I_s\right) + \\
&+ \sum_{l \in \mathbb{L}_T} C_{Ris}(1) \log 2 + \log \begin{pmatrix} N_{l.} + J - 1 \\ J - 1 \end{pmatrix} + \\
&+ \sum_{l \in \mathbb{L}_T} \log \frac{N_{l.}!}{N_{l.1}! N_{l.2}! \ldots N_{l.J}!}
\end{aligned}$$

where V_{X_s} is the number of values of a categorical variable X_s, $B\left(V_{X_s}, I_s\right)$ is the number of possible divisions of V_{X_s} into I_s groups and $C_{Ris}(I_s)$[2] is Rissanen optimal encoding of an integer I_s [73]. A detailed proof can be found in [71].

Once the tree criterion $C(Tree)$ is defined, a greedy search algorithm is then used to find the tree model that minimizes $C(Tree)$. The algorithm starts from the root node and looks for the best partition according to the tree criterion presented above. The leaves are partitioned as long as the tree criterion is improved. Each leaf is partitioned with MODL discretization presented earlier in Section 2.6.2.

The authors of [71] claimed that this algorithm may create under-fitted trees and proposed a post-pruning algorithm to find trees that improve $C(Tree)$.

2.7 Conclusion

The work of this thesis is at the intersection of *Bayesian approaches*, *Uplift modeling* and *Data bias*.

In this chapter, we have first presented in Section 2.2 the state of the art in uplift modeling approaches. We presented two types of approaches: the metalearners and the direct approaches. The metalearners divide the uplift estimation problem into several steps, each of which can be performed using any supervised

[2] $C_{Ris}\left(I_s\right) = \log_2(2.865) + \log_2\left(I_s\right) + \log_2\left(\log_2\left(I_s\right)\right) + \ldots$

machine learning algorithm. The direct approaches, on the other hand, are a set of algorithms specifically designed for uplift modeling. Several algorithms have been proposed in the literature such as decision trees, random forests, SVM and deep learning based methods.

Since one of the main problems with uplift modeling is that the actual uplift values cannot be observed, special metrics have been developed to evaluate its learning algorithms. We presented in Section 2.3 the evaluation metrics used to assess the performance of uplift modeling algorithms. Among the most well-known metrics are the AUUC and the Qini coefficient.

We thereafter introduced the feature selection approaches proposed in the uplift modeling literature. As uplift modeling is a different problem from supervised learning, new feature selection algorithms are required to correctly find the features that contain the treatment effect information. This is accentuated by the fact that the applications of uplift estimation, such as telecommunications data, contain a large number of collected features.

We went on to present the problem of data bias in the uplift problem. Again, as this is a different problem to supervised learning, different types of bias occur in each of the modeling and deployment phases. The modeling bias includes the NRA bias, the non-response bias and the confounding bias.

Finally, the MODL approach is presented. The MODL (Minimum Optimized Description Length) approach is a Bayesian density estimation approach developed in Orange. The MODL approach was designed for the supervised setting and was later extended to the unsupervised setting [22], sequence mining [21], and applied to the design of several learning algorithms such as Naive Bayes [10] and Decision Trees [71]. This thesis proposes an uplift Bayesian approach based on MODL.

CHAPTER 3

A Parameter-free Approach for Uplift Discretization and Feature Selection

Contents

3.1 Introduction

In this chapter, we present a parameter-free feature selection method for uplift modeling founded on a Bayesian approach. Following a part of literature on feature selection that performs a discretization of numerical features [45, 66] as a basis for feature selection, we first describe an automatic feature discretization method for uplift modeling that we call UMODL - for Uplift MODL (c.f. Section 2.6).

As a popular data preprocessing technique in data mining, data discretization converts continuous data into a set of categories that appropriately retains as much information as possible from the original continuous attribute. While data

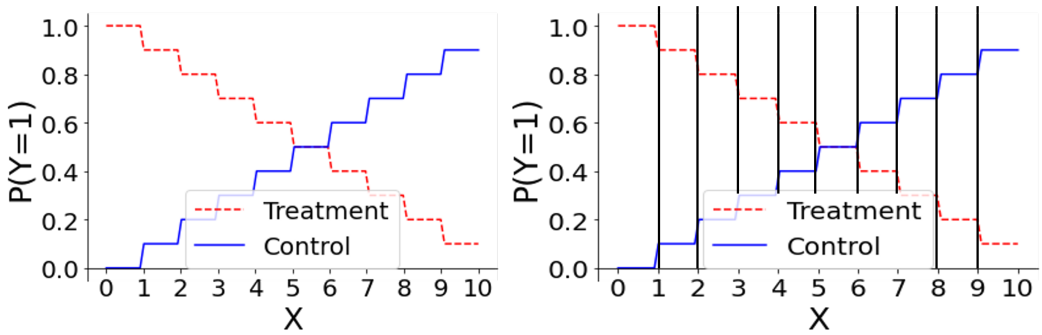


Figure 3.1: On the left figure, a variable X along with the output distributions. On the right figure, the optimal discretization

discretization tasks have been extensively studied in the supervised learning literature, they have not yet been addressed in uplift modeling. Therefore, in this chapter, we present a novel discretization technique specifically tailored to the problem of uplift.

The newly proposed discretization technique can also be used as a univariate uplift estimator. The uplift modeling problem can be viewed as a density estimation challenge, where the goal is to estimate the regions in a variable space where the density of the target variable Y significantly differs between the treatment and control groups. A discretization method can then be useful. Our newly proposed method, UMODL (Uplift Minimum Optimized Description Length), is based on the MODL approach [8], a discretization approach that aims to split a continuous feature into a list of intervals. UMODL discretizes a variable by taking into account the presence of treatment and control groups and facilitates the estimation of the density of the outcome variable for each treatment group within each interval. This is achieved by simply counting the number of instances in each interval and the number of positive outcomes for each treatment group.

For an intuitive understanding, consider the left part of Fig. 3.1, which shows the variable X next to the output distribution in the treatment and control groups. Our goal is to determine the optimal discretization shown in the right of Fig. 3.1. The discretization process can be seen as an estimate of the treatment effect, as it isolates regions with different treatment effect values. Consequently, the uplift value can be calculated individually for each interval by calculating $CATE_i = P_i(Y = 1 \mid T = 1) - P_i(Y = 1 \mid T = 0)$ for each interval i.

UMODL is based on a space of discretization models and a prior distribution. From this model space, we define a Bayesian optimal evaluation criterion of a discretization model for uplift. We then propose an optimization algorithm that finds a near-optimal discretization for uplift estimation in $O(n \log n)$ time. Experiments demonstrate the high performance of this new discretization method.

We then describe UMODL-FS a parameter-free feature selection method for uplift built upon UMODL. Once UMODL identifies the Bayesian optimal discretization for a feature, UMODL-FS is employed to assess the difference in outcome distributions between the treatment and control groups. It is a filter-based method (c.f Section 3.5) that can be used as a pre-processing step before training an uplift model to eliminate features that are not relevant to the uplift estimation. Lastly, we conduct an experimental protocol that validates the effectiveness of UMODL-FS in eliminating irrelevant features and helping the uplift model in achieving superior performance compared to state-of-the-art techniques.

The chapter is structured as follows: Section 3.2 first presents the evaluation criterion of a Bayesian optimal discretization model for uplift and its proof. It then presents the search algorithm and the post-optimisation steps to find the parameters that lead to the best evaluation criterion. Section 3.3 presents the quality evaluation experiments of the discretization approach on a set of synthetic uplift data samples and concludes with a discussion of the results. Finally, Section 3.5 presents the UMODL feature selection approach, the experimental protocol and the results.

This work is the object of the following publication:
Rafla, M., Voisine, N., Crémilleux, B., & Boullé, M. (2023, March). A non-parametric bayesian approach for uplift discretization and feature selection. In Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2022, Grenoble, France, September 19–23, 2022, Proceedings, Part V (pp. 239-254). Cham: Springer Nature Switzerland.

3.2 UMODL

This section introduces UMODL, a novel approach for uplift discretization. We first describe the parameters of an uplift discretization model, which will be used to define the prior and the likelihood of an uplift discretization. The defined prior and likelihood are then used to compute an evaluation criterion for an uplift

discretization model.

After establishing the evaluation criterion, we present the search algorithm that identifies the Bayesian optimal uplift discretization for a given variable. In other words, the search algorithm will be used to automatically find the parameters that lead to the best criterion.

3.2.1 UMODL Criterion

In Section 2.6, we presented the MODL (Minimum Optimized Description Length) approach. MODL is a Bayesian approach for density estimation through discretization for supervised learning. It is founded on the Minimum descriptiom length (MDL) principle.

While MODL properly exploits discretization for density estimation, it is not suitable for uplift modeling since uplift deals with two treatment groups and the estimation of the conditional probabilities of the outcome variable Y given an attribute X also depends on the treatment variable T.

We now introduce the new criterion that we propose to define the best discretization model for uplift. Let M be an uplift discretization model and D denotes data. From a Bayesian point of view, the best uplift discretization model is found by maximizing the posterior probability of the model given the data $P(M|D)$. Let us consider the Bayes rule:

$$P(M \mid D) = \frac{P(M)P(D \mid M)}{P(D)} \tag{3.1}$$

Given that $P(D)$ is constant, maximizing $P(M|D)$ is equivalent to maximizing $P(M)P(D|M)$, i.e the prior probability and the likelihood of the data given the chosen model.

Remark: This optimisation problem represents a trade-off between the prior probability and the likelihood. A simple discretization model with an extremely high prior probability can be represented by a single interval model. However, the likelihood associated with such a model is significantly low. On the other hand, a discretization model characterised by a very high likelihood would be an elementary interval model, where each value of the variable has its own distinct interval. However, this type of model is associated with a significantly low prior probability.

Before determining the posterior probability for an uplift discretization model and presenting the UMODL criterion, let us first introduce some notations:

- X : explanatory variable to discretize
- Y : binary outcome variable
- N : number of instances in the dataset
- J : number of classes of Y
- I : number of intervals
- N_i : number of instances in the interval i
- $N_{it.}$: number of instances in the interval i of treatment t
- $N_{i.j}$: number of instances in the interval i of class j
- N_{itj} : number of instances in the interval i of class j and the treatment t
- W_i : boolean term indicating if the treatment has an effect in interval i (W_i=1) or not (W_i=0)

We define an uplift discretization model M by the number of intervals, the bounds of the intervals, the presence or absence of a treatment effect, class frequencies per interval or for each treatment per interval. In other words, a model M is defined by the following parameters (cf. Fig. 3.2):

$$\{I, \{N_i\}, \{W_i\}, \{N_{i.j}\}_{W_i=0}, \{N_{itj}\}_{W_i=1}\}$$

These parameters are exploited according to a particular hierarchy when defining the prior distribution of M denoted $P(M)$. This hierarchy requires the parameters to be selected in a particular order. We will discuss the hierarchy of the parameters in the next section.

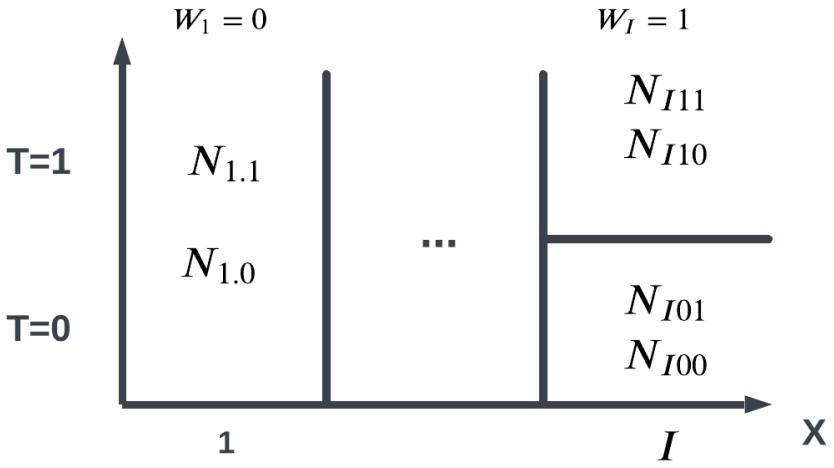


Figure 3.2: Parameters of an uplift discretization model. The presence of a treatment effect ($W_i = 1$) in interval i requires describing the distribution of the outcome variable Y separately for each treatment (part right). In contrast, the absence of a treatment effect ($W_i = 0$) indicates to consider the distribution of the outcome variable Y for the interval i independently of the treatment variable (part left).

The evaluation criterion $C(M)$ which is the cost of an uplift discretization model M is defined then by:

$$C(M) = -\log\Big(P(M) \times P(D|M)\Big)$$

Taking the negative log turns the maximization problem to a minimization one. M is optimal if $C(M)$ is minimal.

How to define the prior distribution ?

To define the prior distribution of the model parameters $P(M)$:

1. We first exploit the ***hierarchy of the parameters*** of a discretization model. This hierarchy requires the parameters to be selected in a particular order. First, we determine the number of intervals I, followed by the location of these I intervals or boundaries. Next, we determine whether each interval contains a treatment effect or not. Finally, we decide the distribution of the outcome variable within each interval or the distribution of the outcome variable for each treatment.

2. Next, we assume a ***uniform distribution at each stage of this hierarchy***. In other words, we assume that

   (a) The number of intervals I is equally likely to be any value between 1 and N.

   (b) Given the number of intervals I, each possible way of dividing the data into I intervals has an equal probability.

   (c) There is an equal probability that an interval i contains a treatment effect or not. Therefore, the value of the term W_i has an equal chance of being either 1 or 0.

   (d) Given an interval i and the value of W_i, every distribution of the class values in the interval is equiprobable, or alternatively, every distribution of the class values for each treatment in the interval is equiprobable.

3. Finally, we assume the ***independence of the distributions across intervals***. This assumption is based on the *IID hypothesis* [11]. This assumption enables the evaluation of the model's prior as a product of multiple terms, which will be demonstrated next concerning the prior definition (Eq. 3.3) and its proof, as well as the likelihood definition in Eq. 3.10. By taking the negative logarithm, the prior can be assessed as the sum of these terms, as illustrated in the UMODL criterion in Eq. 3.2.

The UMODL criterion

Using the components described above (the parameter hierarchy, the uniform distribution assumption and the independence assumption), we express $C(M)$ in terms of the parameters of an uplift discretization model and obtain Eq. 3.2, which we demonstrate below.

$$\begin{aligned} C(M) = \log N + \log \binom{N+I-1}{I-1} + I \times \log 2 \\ + \sum_{i=1}^{I} (1-W_i) \log \binom{N_i+J-1}{J-1} + \underbrace{\sum_{i=1}^{I} (1-W_i) \log \frac{N_i!}{N_{i.1}!..N_{i.J}!}}_{Likelihood} \\ + \sum_{i=1}^{I} W_i \sum_t \log \binom{N_{it.}+J-1}{J-1} + \underbrace{\sum_{i=1}^{I} W_i \sum_t \log \frac{N_{it.}!}{N_{it1}!..N_{itJ}!}}_{Likelihood} \end{aligned} \tag{3.2}$$

Proof of Eq 3.2. We express $P(M)$ and $P(D|M)$ according to the parameters of an uplift discretization model. We introduce a prior distribution by exploiting the hierarchy of the models' parameters. Assuming the independence of the local distributions across the intervals, we obtain:

$$P(M) = P(I) \times P(\{N_i\}|I) \times \\ \prod_i P(W_i|I) \left[(1 - W_i) \times P(\{N_{i.j}\}|I, \{N_i\}) + W_i \times \prod_t P(\{N_{itj}\}|I, \{N_{it.}\}) \right] \tag{3.3}$$

We express each of the terms of Eq. 3.3 according to the parameters of M assuming a uniform distribution for each parameter. Assuming that the number of intervals I is uniformly distributed between 1 and N, the first term in Eq. 3.3 becomes:

$$P(I) = \frac{1}{N} \tag{3.4}$$

Given a number of intervals I, all the discretizations into I intervals (i.e. the choices of the bounds) are equiprobable. Computing the probability of an interval set leads to a combinatorial calculation of the number of all possible interval sets or equivalently the number of ways of distributing the N instances in the I intervals, with counts N_i per interval. The second term of Eq. 3.3 is then:

$$P(\{N_i\}|I) = \frac{1}{\binom{N+I-1}{I-1}} \tag{3.5}$$

For a given interval i, we assume that a treatment can have an effect or not, with equal probability, i.e. $P(W_i|I) = \frac{1}{2}$. We obtain:

$$\prod_i P(W_i|I) = \left(\frac{1}{2}\right)^I \tag{3.6}$$

In the case of an interval i where there is not effect of the treatment ($W_i = 0$), UMODL describes one unique distribution of the outcome variable. Given an interval i, its number of examples N_i is known. Assuming that each of the class distributions is equiprobable, we end up also with a combinatorial problem:

$$P(\{N_{i.j}\}|I, N_i) = \frac{1}{\binom{N_i+J-1}{J-1}} \tag{3.7}$$

In the case of an interval i with an effect of the treatment ($W_i = 1$), UMODL describes two distributions of the outcome variable, with and without the treatment. Given an interval i and a treatment t, we know the number of examples

$N_{it.}$. Assuming that each of the distributions of class values is equiprobable, we get:

$$P(\{N_{itj}\}|I, N_{it.}) = \frac{1}{\binom{N_{it.}+J-1}{J-1}} \tag{3.8}$$

After defining the models' prior, we define the likelihood $P(D|M)$ of the data given the uplift discretization model. For each multinomial distribution of the outcome variable (a single or two distinct distributions per interval depending on whether the treatment has an effect or not), we assume that all possible observed data D_i consistent with the multinomial model are equiprobable. Using multinomial terms, we obtain the following likelihood term:

$$P(D|M) = \prod_i P(D_i|M) \tag{3.9}$$

$$= \prod_i \left[(1 - W_i) \times \frac{1}{(N_i!/N_{i.1}!..N_{i.J}!)} + W_i \times \prod_t \frac{1}{(N_{it.}!/N_{it1}!..N_{itJ}!)} \right] \tag{3.10}$$

Combining the prior $P(M)$ (Eq 3.4 to 3.8) with the likelihood $P(D|M)$ (Eq. 3.10), we obtain $P(M)P(D|M)$. Taking the negative log yields to the UMODL criterion presented in Eq. 3.2. Coming back to Eq. 3.2, the prior terms of the first line come from Eq. 3.4 to 3.6. In the second line of Eq. 3.2 (modeling a situation w/o a treatment effect) and the third line (situation with a treatment effect), the first terms are prior terms (Eqs 3.7- 3.8) and the second terms are likelihood terms (Eq. 3.10).

3.2.2 Search algorithm and post-optimization

We sketch below our search algorithm to find the best model w.r.t. the UMODL criterion. This algorithm finds the optimal values of the parameters that minimize $C(M)$. The principle of this algorithm is inspired by the search algorithm [8] which we adapt to our criterion. As an optimal search algorithm is not practical due to the complexity of the problem, we build a greedy algorithm[1].

Greedy Search algorithm

The search algorithm is a greedy bottom-up algorithm with the following steps:

- The algorithm starts by making an elementary discretization such that all examples with the same value have their own interval,

[1]Our implementation is provided at https://github.com/MinaWagdi/UMODL

- Compute the costs of all possible merges i.e. try to merge adjacent intervals,
- Merge the two adjacent intervals that decrease $C(M)$ the most,
- Recalculate the cost of all possible adjacent merges and select the merge that reduces $C(M)$ the most,
- Repeat until no merge decreases $C(M)$.

While this algorithm is complex, it can be implemented in $O(n \log n)$ time [8].

Post-optimization

This greedy search algorithm can fall into a local minimum, so post-optimization steps are needed to perturb the interval bounds. We used post-optimization steps that consist of recurrent splits, merges, merge splits, and merge merge splits of adjacent intervals, as described in [8] but designed in this work for uplift.

3.2.3 Conclusion

The presented discretization approach is a density estimation approach for uplift modeling. We model the probability of Y conditionally on the explanatory variable X and a binary treatment variable T. The search algorithm we present is looking for the parameters I, $\{W_i\}$, $\{N_i\}$, $\{N_{i.j}\}$, $\{N_{itj}\}$, and $\{W_i\}$ that minimize the cost of the model. In other words, the search algorithm tries to find the optimal discretization in the Bayes sense that best estimates the real densities of the outcome variable Y conditionally on X and T. Once a discretization and its parameters are defined, the estimation of the CATE for each interval is simple. As shown in Fig. 3.2, assuming a binary outcome variable Y and given $W_i = 1$, we have $P_i(Y = 1|T = 1) = N_{i11}/(N_{i11} + N_{i10})$ and $P_i(Y = 1|T = 0) = N_{i10}/(N_{i01} + N_{i00})$, therefore $CATE_i = P_i(Y = 1|T = 1) - P_i(Y = 1|T = 0)$. For intervals with $W_i = 0$, $CATE_i$ is considered insignificant.

The UMODL discretization method has the advantage of not relying on user-specified parameters. All parameters are automatically determined by the search algorithm (Section 3.2.2). In addition, UMODL does not require any prior assumptions about the data distribution. It also facilitates interpretation, as each identified interval represents the distribution of a set of examples.

3.3 UMODL quality evaluation experiments

This section experimentally evaluates whether UMODL is a good estimator of uplift. The principle of the experiments is to generate data with different synthetic uplift patterns in order that results of UMODL can be compared to true uplift. A *synthetic uplift pattern* is a data pattern where $P(Y = 1|X, T = 1)$ and $P(Y = 1|X, T = 0)$ are identified for each example. Therefore several indicators can be observed: (1) the number of intervals founded by UMODL w.r.t. the characteristics of the uplift pattern, (2) the RMSE (root mean squared error) between the real uplift and the estimated uplift by UMODL computed for each instance and (3) the number of instances needed by UMODL to find the uplift pattern. We generate synthetic uplift patterns of different characteristics for simulating various situations.

3.3.1 Description

The experimental protocol is made of the following steps:

1. Define a particular synthetic uplift pattern of one dimension.

2. Generate several train samples according to the defined pattern with 40 different number of instances (also called *data size*) ranging from 10 to 100,000 instances. For each data size, generate ten datasets. All generated data are uniformly distributed on the $[0, 10]$ numerical domain for each of the treatment ($T = 1$) and control groups ($T = 0$).

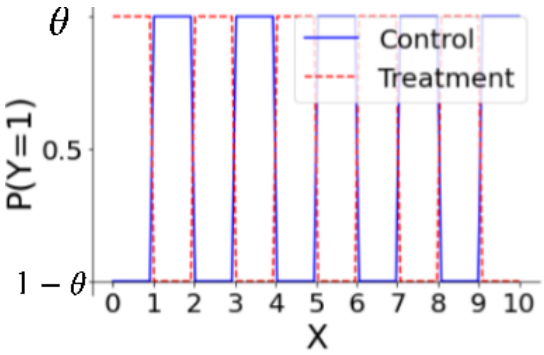
3. Generate a test set of 10,000 instances based on the defined uplift pattern.

4. For each training sample, apply the UMODL approach to search for the best discretization model.

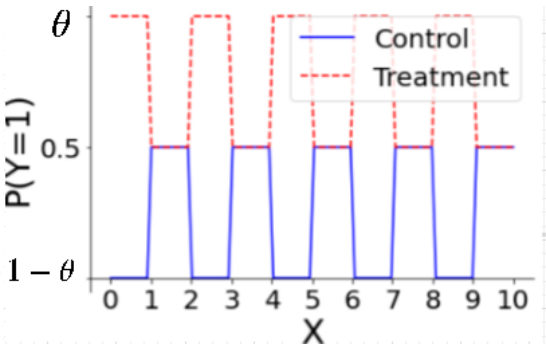
5. For each experiment, the obtained discretization model is then applied to the test set, and RMSE is computed by comparing for each data point: (a) the CATE estimation in the found interval and (b) the real CATE value.

6. By observing both the number of found intervals for each dataset and the RMSE values, we can determine whether the UMODL approach manages to find the synthetic pattern or not.

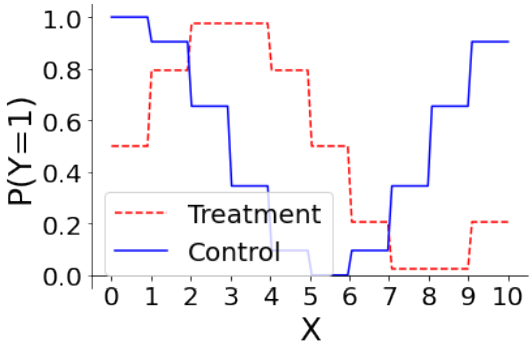
7. Repeat these steps with different synthetic uplift patterns.



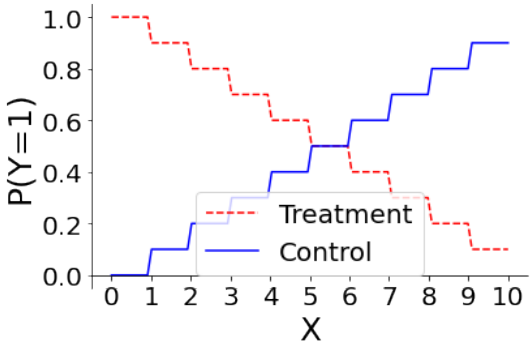
(a) Crenel pattern 1



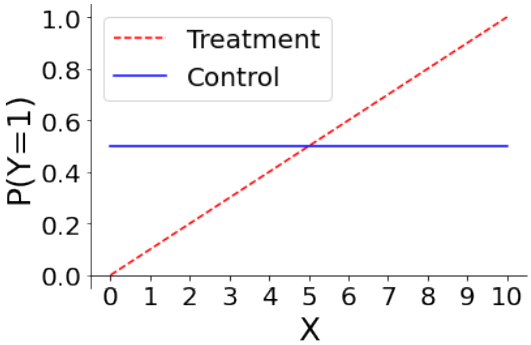
(b) Crenel pattern 2



(c) Trigonometric pattern



(d) Scissors pattern



(e) Continuous pattern

Figure 3.3: Synthetic uplift patterns. The X-axis represents variable X and the Y-axis represents $P(Y = 1)$. For *Crenel Pattern 1* and *Crenel Pattern 2*, five versions are generated with different values of $\theta \in \{0.6, 0.7, 0.8, 0.9, 1\}$. The difference between $P(Y = 1)$ in the treatment and control groups represents the uplift.

3.3.2 Synthetic uplift patterns

We generate four bin-based patterns and one continuous pattern. We use patterns of different characteristics[2] to evaluate how UMODL performs both in various situations and different rates of uplift. The patterns are illustrated in Fig. 3.3 and depicted below.

- *Crenel pattern 1* (cf. Fig. 3.3a): this crenel pattern is made of 10 intervals containing a repeated sequence of a positive treatment effect followed by a negative one. We generated five versions of this pattern with different uplift values. In other words, this pattern was generated with different θ values, where a positive treatment effect is equal to $\theta - (1 - \theta)$, resulting in $2\theta - 1$, and a negative treatment effect is equal to $(1 - \theta) - \theta$, which simplifies to $1 - 2\theta$.

- *Crenel pattern 2* (cf. Fig. 3.3b): is a slightly different crenel pattern similarly made of 10 intervals containing a repeated sequence of a positive treatment effect followed by no treatment effect. We generated five versions of this pattern with different treatment effects (uplift). In other words, this pattern was generated with different θ values, where a positive treatment effect is equal to $\theta - (1 - \theta)$, resulting in $2\theta - 1$.

- *Trigonometric pattern* (cf. Fig. 3.3c) is a particular bin-based pattern with trigonometric shape where: $P(Y = 1|T = 1) = 0.5 + (0.5 \times sin(i \times \frac{2\pi}{10}))$ and $P(Y = 1|T = 0) = 0.5 + (0.5 \times cos(i \times \frac{2\pi}{10}))$

- *Scissors pattern* (cf. Fig. 3.3d) is a bin-based pattern where $P(Y = 1|T = 1) = \frac{i}{10}$ and $P(Y = 1|T = 0) = 1 - \frac{i}{10}$, where i is the interval number.

- *Continuous pattern* (cf. Fig. 3.3e) differs from bin-based patterns. Here $P(Y = 1|T = 1) = X/10$ $P(Y = 1|T = 0) = 0.5$.

3.3.3 Results

Results are given in Figures 3.4, 3.5 and 3.6. We start by the central question "Is UMODL a good estimator of uplift?" and provide complementary observations.

[2]Other patterns can be found using the github link provided previously.

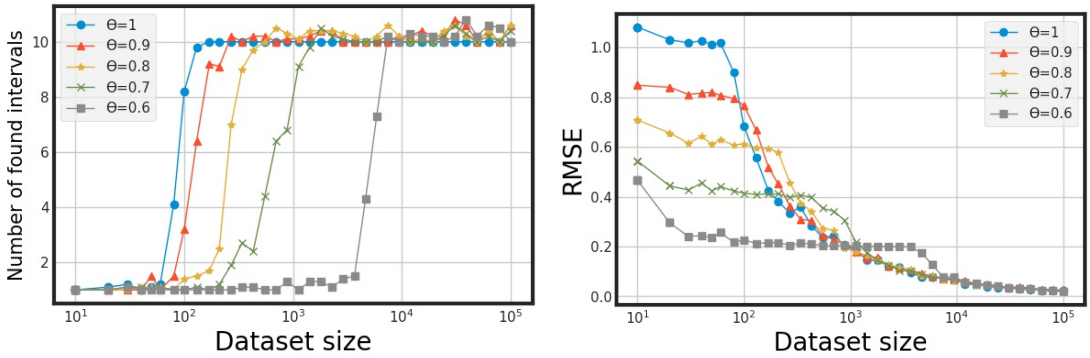


Figure 3.4: Results obtained for *Crenel pattern 1* (cf. Fig. 3.3a). The left (resp. right) figure shows the mean number of found intervals (resp. the mean value of RMSE) on the test set by UMODL according to the dataset size. Different curve colors correspond to different treatment effects. For example, the blue curve corresponds to the crenel pattern of repeated positive uplift ($= 1$) followed by negative uplift ($= -1$).

Is UMODL a good estimator of uplift? From Figures 3.4 (left) and 3.5 (left), we clearly see that even when the treatment effect is very small per interval (grey curves), UMODL is able to find the proper number of intervals of the uplift patterns. This is also illustrated by the RMSE curves (Figures 3.4 (right) and 3.5 (right)) showing that RMSE always converges towards 0 for sufficiently large datasets. Similar performances are reported with the *trigonometric pattern* (cf. Fig. 3.6a), the *scissors pattern* (cf. Fig. 3.6b) and the *continuous pattern* (cf. Fig. 3.6c) except that the number of estimated intervals is not a relevant indicator for the *continuous pattern* because this pattern is continuous.

How many instances are needed to find the uplift pattern according to its characteristics? When the differences of densities between adjacent intervals get smaller, UMODL needs more instances to give prominence to a model with more intervals. This is typically the case with the *scissors pattern* (cf. Fig. 3.6b). Analogous behaviors are observed in Figures 3.4 and 3.5. For example, in Fig. 3.4, the blue curve finds the uplift pattern with less instances than the red curve. Interestingly, UMODL succeeds in finding the appropriate intervals even when there is no treatment effect (for example, identifying the intervals $[1, 2], [3, 4], [5, 6], \ldots$ in the results of the *crenel pattern 2*, as depicted in Fig. 3.5).

Does UMODL overfit? Another important aspect of the UMODL discretization is that the UMODL method does not overfit, i.e. UMODL always finds the

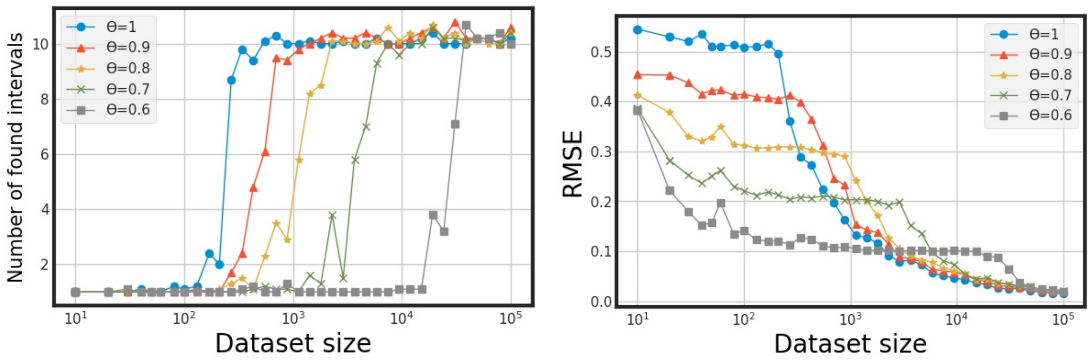


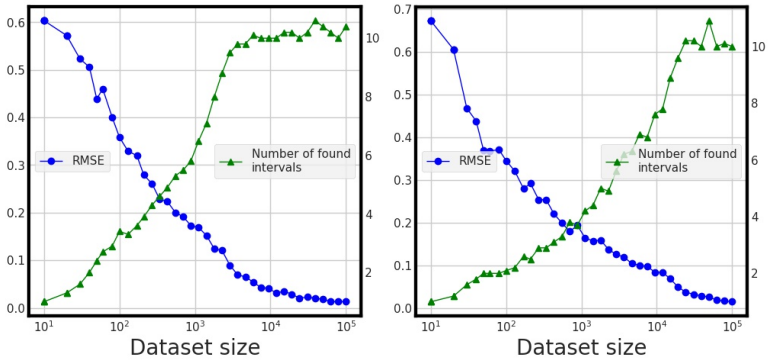
Figure 3.5: Results obtained for *Crenel pattern 2* (cf. Fig. 3.3b). The left (resp. right) figure shows the mean number of found intervals (resp. the mean value of RMSE) on the test set by UMODL according to the dataset size. Different curve colors correspond to separate treatment effects. For example, the blue curve corresponds to the crenel pattern of repeated positive uplift (=1) followed by zero uplift.

ten intervals of the underlying patterns and does not consider extra intervals even when the data size increases significantly (cf. Fig. 3.4 and Fig. 3.5). With the *continuous pattern*, UMODL goes on to consider more intervals as long as the size of the data increases (cf. Fig. 3.6c) which is appropriate since the pattern is continuous and there is no defined intervals.

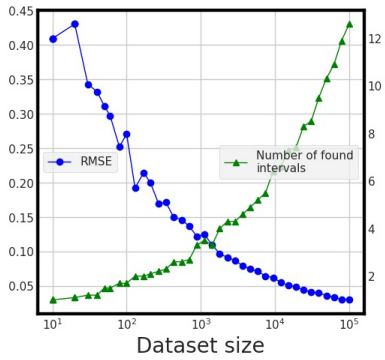
3.4 How to deal with categorical variables?

So far, our experiments have only been carried out on synthetic data with continuous attributes, as this type of data requires discretization when performing density estimation. However, it is crucial to acknowledge the existence of another type of data: categorical data, which can be considered as grouped information covering categories such as job type, phone type, and subscription type, among others.

When dealing with categorical data, the objective shifts from discretizing the variable to performing *value grouping*. In other words, given a categorical variable, the goal is to group the values of the variable that have **similar behaviour**: either the same outcome distribution or the same treatment effect. Assuming a variable '*Subscription*' that has 3 possible values '*Prepaid*', '*Postpaid*', '*Family Plan*', the number of possible groupings of these values is equal to the number of Bell, which counts the number of possible partitions of a set. These sets are:



(a) Trigonometric pattern (b) Scissors pattern



(c) continuous pattern

Figure 3.6: Fig 3.6a, 3.6b, 3.6c present the performances obtained with the *trigonometric pattern (cf. Fig. 3.3c), scissors pattern (cf. Fig. 3.3d) and continuous pattern (cf. Fig. 3.3e)*. Blue curves depict the mean value of the RMSE per dataset size while the green curves indicate the number of found intervals.

- $\{(Prepaid),(Postpaid),(Family\ Plan)\}$
- $\{(Prepaid,Postpaid),(Family\ Plan)\}$
- $\{(Prepaid,Family\ Plan),(Postpaid)\}$
- $\{(Postpaid,Family\ Plan),(Prepaid)\}$
- $\{(Postpaid,Family\ Plan,Prepaid)\}$

When the number of variable values increases, the corresponding *Bell number*, and therefore, the number of potential partitions, grows significantly. The Bell numbers denoted by B_n count the number of different possible ways to partition

a set of n elements. For example, the sequence of Bell numbers displays a rapid increase: $B_0 = 1$, $B_1 = 1$, $B_2 = 2$, $B_3 = 5$, $B_4 = 15$, $B_5 = 52$, $B_6 = 203$, $B_7 = 877$, $B_8 = 4140$ and so on. This represents the complexity of finding the optimal value grouping with slightly higher number of values. A UMODL criterion specifically designed for categorical variables can be designed, we propose this as a topic for future research.

Yet a basic technique for dealing with categorical variables in data mining tasks is *unsupervised label encoding*. Unsupervised label encoding involves assigning an integer value to each instance of a categorical variable. This assignment can be done either randomly or according to the lexicographic order of the categorical values. The UMODL approach can then be used to discretize the variable as if it were a continuous numerical attribute. However, *unsupervised label encoding* alone may not be sufficient for a discretization task because UMODL considers an order for the values, and this order may not be efficient. This inefficiency occurs because the numerical values assigned to each category are random and unrelated to the outcome distribution or the uplift.

In this section, I will show why an *unsupervised label encoding* is not efficient and propose an adequate uplift-based label encoding that allows better *value grouping* and improves the UMODL discretization approach.

3.4.1 Why unsupervised label encoding is not efficient ?

An unsupervised label encoding may be particularly inefficient in cases where there is an imbalance in the size of different values. For instance, consider a scenario with a 'subscription' variable having three distinct values: 'Prepaid', 'Postpaid', and 'Family Plan', each having a different level of uplift (see Table 3.1).

Value	T0Y0	T0Y1	T1Y0	T1Y1	P(Y=1|T=1)	P(Y=1|T=0)	Uplift
Prepaid	2102	883	1214	1771	0.6	0.3	0.3
Postpaid	2098	887	316	2669	0.9	0.3	0.6
Family Plan	21	9	12	18	0.6	0.3	0.3

Table 3.1: Description of the 'subscript' variable. T0Y0 denotes the number of examples with $T = 0$ and $Y = 0$. Similarly, T0Y1, T1Y0, and T1Y1 represent examples with respective T and Y values.

By assigning the integer values 1, 2 and 3 to 'Prepaid', 'Postpaid' and 'Family plan' respectively, in the given order, we can encode these values. In this situation, the resulting density plot is shown in Fig. 3.7.

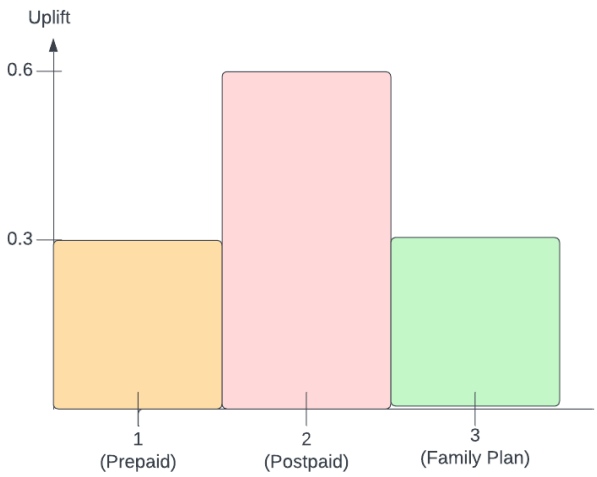


Figure 3.7: An unsupervised label encoding of the categorical variable 'subscription'

In Fig. 3.7, an ideal discretization according to this encoding requires each categorical variable value to be separated into a separate interval, with the parameter $W = 1$ in each interval. However, if there are **very few examples** for a 'Family Plan' value (as shown in Table 3.1), the robust UMODL method is less likely to create an interval where there are very few examples. Although this would improve the likelihood component (see Eq. 3.2), the three-interval model significantly increases the cost of the prior component. Instead, UMODL would choose a two-interval model, with the first interval consisting of 'Prepaid' and the second interval including both 'Postpaid' and 'Family Plan'. This approach achieves a trade-off that balances the costs of the likelihood and the prior components.

However, by reordering the encoding of the values, UMODL can achieve a better trade-off, which will be discussed in the next section.

A numerical illustration For instance, let's consider the the data in Table 3.1 comprising 12,000 examples, which includes 5,970 'prepaid', 5,970 'postpaid', and only 60 'Family Plan'. Each value is equally distributed between the treatment and control groups. Assume we have randomly encoded the values of the categorical variable as depicted in Fig. 3.7. In that case, as discussed before, UMODL will opt the the two interval model named M_1, as shown in Fig. 3.8. Based on the information provided in Table 3.1, we can determine that $C(M_1) = 6737.02$. The prior cost of M_1 is $-\log P(M_1) = 52.1976$, while the likelihood term cost amounts to $-\log P(D|M_1) = 6684.82$. UMODL being a robust method would choose the tradeoff between the prior and the likelihood of M_1 rather than a

three-interval model. A three-interval model, named M_2 in Fig. 3.8 will have a cost of $C(M_2) = 6740.88$, consisting of a prior $-\log P(M_2) = 68.439$ and a likelihood $-\log P(D|M_2) = 6672.4$.

As demonstrated, the cost of M_1 is lower, despite having a worse likelihood. This is because introducing an additional interval with a limited number of examples proved to be expensive.

M_1 also proved to be much better than the *null model* $M_\varnothing$, i.e. the model with only one interval. This model assumes that there is only one probability distribution and treats the variable as a random variable. The null model is shown in 3.9. Calculating the cost of this model yields $C(M_\varnothing) = 8322.96$. The cost of the prior of this model is certainly less than the other models presented $-\log P(M_\varnothing) = 19.47$. However, the cost of the likelihood of our data given this model is very large; $-\log P(D|M_\varnothing) = 8303.4$,

Note: These calculations can simply be done using the criterion equation in Eq. 3.2 and the information of data sizes in Table 3.1.

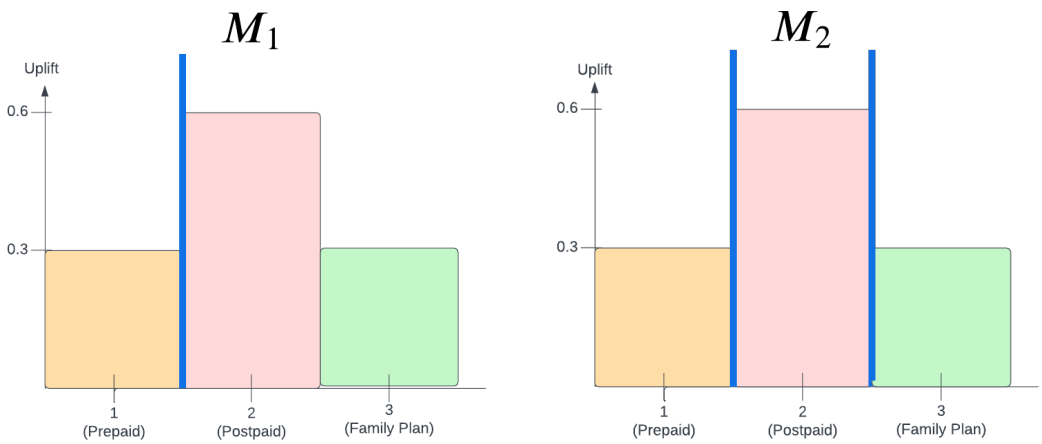


Figure 3.8: Two different discretization models for a categorical variable

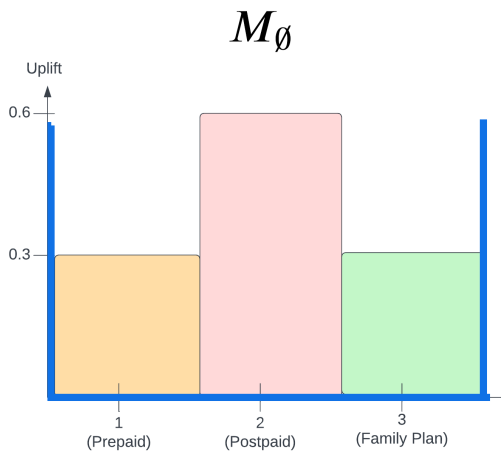


Figure 3.9: The null model: the model with one interval

In summary, an unsupervised approach to label encoding assigns a random ranking to categorical values, which is often not optimal. In the following section, I present an uplift-based label encoding that allows UMODL to find a better trade-off for value grouping for categorical variables.

3.4.2 An uplift-based label encoding for UMODL

In the earlier example, 'Family Plan' and 'Postpaid' values were grouped by UMODL despite having different uplift values. A solution is to intelligently encode categorical values by the order of the uplift values: 'Prepaid' as 1, 'Family Plan' as 2, and 'Postpaid' as 3. This groups similar uplift values, as seen in Fig. 3.10. Based on this encoding, the UMODL discretization method selects the two-interval model, M_3, as optimal. This model accurately groups $\{Prepaid, FamilyPlan\}$ and $\{Postpaid\}$ together, effectively grouping values with similar uplifts.

The cost of M_3 is given by $C(M_3) = 6728.4$, with a prior's cost of $-\log P(M_3) = 52.19$ and a likelihood of $-\log P(D|M_3) = 6676.2$.

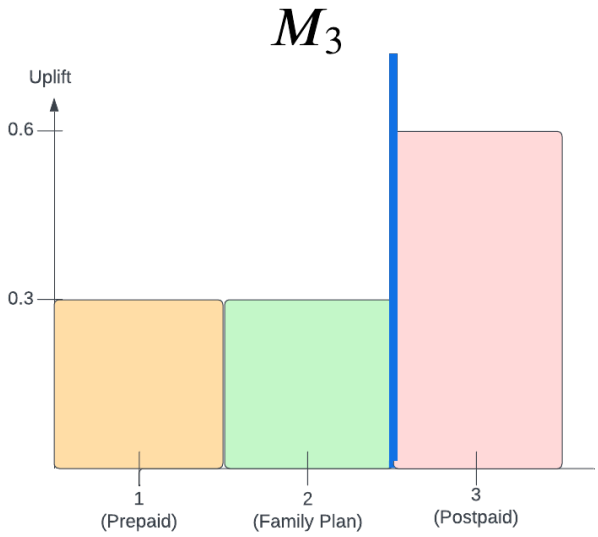


Figure 3.10: The resulting discretization model by encoding the values of the 'subscription' variable by ascending uplift values

The proposed encoding does not always yield improvements, though it doesn't negatively affect discretization. If 'Family Plan' had an uplift value of 0.9, re-encoding and applying UMODL discretization would result in a two-interval model grouping $Postpaid, FamilyPlan$. While not highly accurate, it is the best achievable trade-off given the data.

Note: The examples above involve a situation with parameter W has the value 1 for each interval, indicating a treatment effect. The UMODL criterion (Eq. 3.2) identifies intervals with significant uplift and unique outcome variable distributions, with the search algorithm determining parameter W for each interval. In cases where the uplift for each categorical variable value is minimal ($W = 0$), encoding values based on uplift is ineffective. Instead, we focus on encoding values according to the outcome distribution within each interval. I propose that the ideal solution would involve developing an algorithm that combines both encoding approaches (ranking by uplift values and ranking by outcome distribution). Designing such an algorithm presents a promising perspective for this thesis.

3.5 UMODL Feature Selection

As previously discussed in Section 2.4, the telecommunications sector collects significant amounts of data (containing hundreds to thousands of features) derived from services such as mobile internet, home internet, SMS and phone calls. This data contains a significant amount of noise and irrelevant features, which can cause significant challenges for supervised machine learning models, such as the curse of dimensionality. Feature selection serves as an essential step to increase model efficiency and improve interpretability.

In this section I describe how the UMODL discretization method can be used to develop a novel feature selection technique, which I have named 'UMODL-FS', tailored to uplift modeling. I then present the experimental protocol we used to evaluate whether UMODL-FS efficiently improves the uplift models.

3.5.1 Description of UMODL feature selection

In this section I will explain the UMODL feature selection technique. The UMODL Feature Selection (UMODL-FS) calculates the importance of a feature X by first discretizing it using the UMODL discretization approach. The method then computes the summed Euclidean distances between the outcome distributions within the treatment and control groups over the intervals found. To elaborate further:

1. Given a feature X, we apply the UMODL discretization method to find the optimal uplift discretization model as presented in Section 3.2.1.

2. Compute for X an importance score (described below), denoted by $imp.s(X)$, which is the divergence measure of the treatment effect over the found intervals.

3. We repeat these steps for each feature of the dataset.

4. All features with $imp.s(X) > 0$ are considered relevant for the uplift estimation, while any feature with $imp.s(X) = 0$ is eliminated.

We define $imp.s(X)$ as follows. Assuming $p_i = P_i(Y = 1|T = 1)$ and $q_i = P_i(Y = 1|T = 0)$. We define:

$$imp.s(X) = \begin{cases} \sum_{i=1}^{I} \frac{N_i}{N} D(p_i : q_i), & \text{if } I > 1 \\ 0, & \text{otherwise .} \end{cases} \tag{3.11}$$

where the distribution divergence measure D is the squared euclidean distance. We choose the squared euclidean distance for the divergence since it is symmetric and stable [63]. UMODL-FS considers irrelevant for the uplift estimation any feature with $imp.s(X) = 0$ and keeps for the uplift modeling any feature with $imp.s(X) > 0$. When UMODL finds a single interval for a feature, it means there is only one distribution for all instances and thus a non-informative feature (i.e . $imp.s(X) = 0$). Unlike feature selection methods of the literature [81], our approach does not require parameters to set, and there is no need to give the number of features to keep or delete.

3.5.2 Experimental Protocol

To compare UMODL-FS to the state-of-art uplift feature selection methods (cf. Section 3.5), we design the following experimental protocol:

1. For each dataset, we generate eleven variants of the dataset, each with an incremental total number (from 0 to 100) of noise features. Noise features are sampled from $\mathcal{N}(0, 1)$ for each of the treatment and control groups.

2. For each variant, we apply the following feature selection methods (previously described in Section 2.4): (a) KL-filter (b) Chi-filter (c) ED-filter (d) LR-filter (e) F-filter (f) UMODL-FS.

   For KL-filter, Chi-filter and ED-filter, we set the number of bins to 10.

3. To have the same number of features for each feature selection method and perform a fair comparison, we pick the M most important features, where M is the number of all features deemed informative by UMODL-FS.

4. With these sets of features, we build uplift models: a two-model approach with logistic regression [33] and X-Learner with linear regression [36].

5. The learning process is done with stratified ten-fold cross-validation. Test samples are used to evaluate the performance of uplift models based on the selected features.

6. The qini coefficient metric [18] is used to evaluate the performance of the uplift model.

3.5.3 Datasets

Experiments are conducted on two publicly available datasets which are usual on the uplift community:

1. Criteo dataset [19]: a real large scale dataset constructed by assembling data resulting from several incrementality tests in advertising. In the experiments, we use a random sample of 10,000 instances with the 'visit' variable as outcome variable.

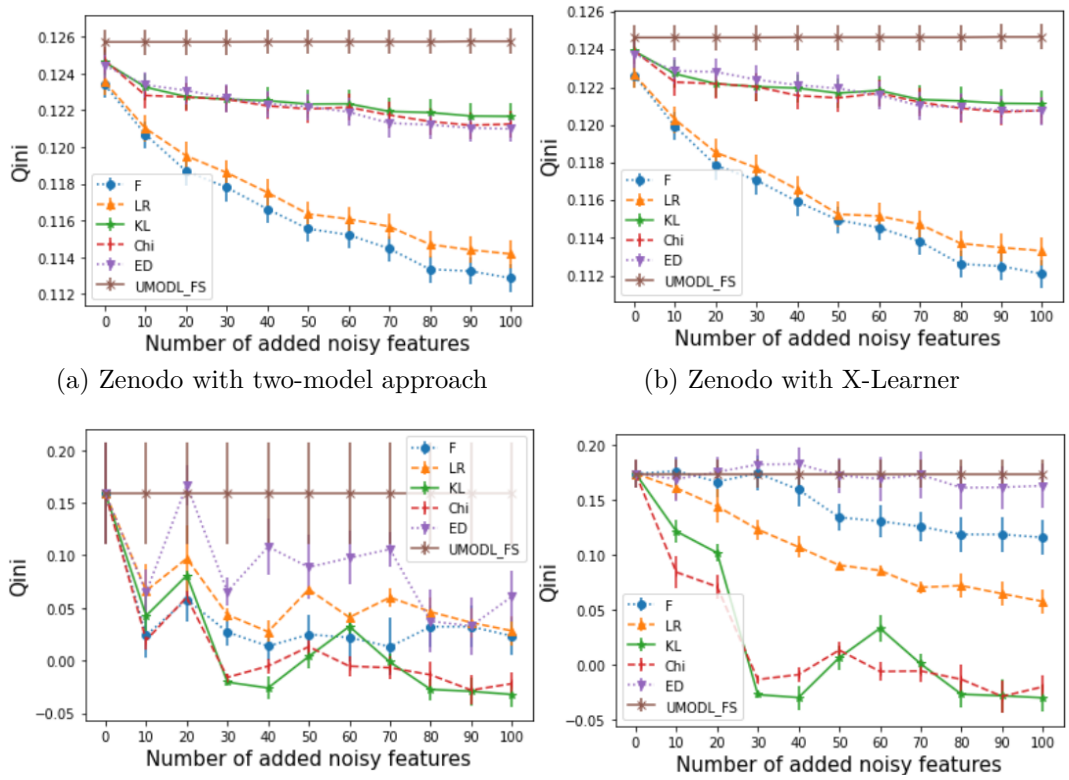
2. Zenodo synthetic dataset [3]: this dataset was created for evaluating feature selection methods for uplift modeling. It has three types of features: (a) uplift features influencing the treatment effect on the conversion probability (outcome variable is 'conversion'); (b) classification features influencing the conversion probability independent of the treatment effect; (c) irrelevant features. This dataset consists of 100 trials of different patterns. Each trial has 10,000 instances and 36 features.

3.5.4 Results

Fig. 3.11 presents the results on the use of UMODL-FS for uplift modeling. In all experiments, UMODL-FS selects the set of features leading to the uplift model with the best qini coefficient (therefore the best uplift model) whatever the used uplift approach. Remarkably, the more noisy features are added, the more the qini difference between UMODL-FS and other feature selection methods increases.

Fig. 3.12 indicates the percentage of added noisy features which are selected by the different feature selection methods according to the number of added noisy features. UMDOL-FS never selects a noisy feature. It illustrates the clear ability of UMODL-FS to remove noisy features. On the contrary, all other methods select noisy features and the percentage of the selected noisy ones increases as the number of added noisy features increases. To sum up, the more the number of added noisy features, the more the feature selection methods of the literature select irrelevant features as informative. In contrast, UMODL-FS always neglects irrelevant features and has the most stable qini coefficients. Moreover, UMODL-FS does not require to set a parameter giving the number of features to keep.

[3] https://doi.org/10.5281/zenodo.3653141



(a) Zenodo with two-model approach

(b) Zenodo with X-Learner

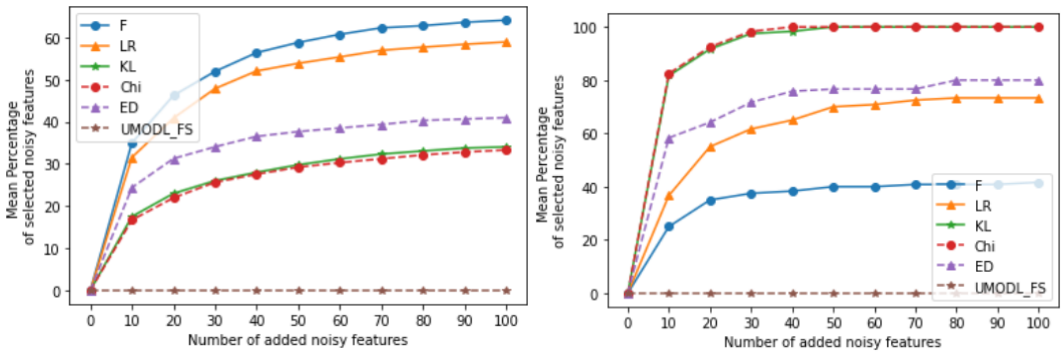
(c) Criteo with two-model approach

(d) Criteo with X-Learner

Figure 3.11: Average qini coefficients and their variances according to the number of added noisy features. The X-axis indicates the total number of added noisy features. Y-axis represents the qini coefficients achieved by uplift models.

3.6 Conclusion

In this chapter, we have proposed a new non-parametric Bayesian approach for uplift discretization and feature selection. We have defined UMODL, a Bayes optimal evaluation criterion of a discretization model for uplift modeling and a search algorithm to find the best model. We have conducted an experimental protocol to assess UMODL as an uplift estimator through discretization. We defined different synthetic uplift patterns and generated accordingly several datasets with several data sizes. The use of synthetic data gave us the advantage to know the true uplift value and thus be able to compare the estimated uplift value by our approach and the true one. By observing the RMSE of the predicted uplift values and the number of found intervals by data size, we were able to infer the following chacarteristics:



(a) Zenodo (b) Criteo

Figure 3.12: Percentage of selected noisy features according to the number of added noisy features.

1. UMODL is a good uplift estimator through discretization.
2. UMODL does not overfit
3. It needs sufficient number of instances to give prominence to a model with more intervals

We have also shown that UMODL can effectively handle categorical variables, and we have introduced an adequate label encoding technique that helps UMODL to identify more appropriate intervals, especially when there is an imbalance in the values of a variable.

Finally, we have presented UMODL-FS, a feature selection method for uplift. We conducted an experimental protocol on real and synthetic datasets, where the idea was to gradually add noisy features and build several uplift models, each with a different feature selection method as a preprocessing step. Experiments show that UMODL-FS removes irrelevant features and clearly outperforms state of the art methods by providing uplift models with the highest and most stable qini coefficients. The method is parameter free, making it easy to use.

CHAPTER 4

Parameter-free Bayesian Decision Trees for Uplift modeling

Contents

4.1 Introduction

Decision tree algorithms have been developed in state-of-the-art uplift modeling approaches (see Section 2.2.2). These algorithms aim to partition the feature space into distinct branches with the aim of identifying leaf nodes that have the most important difference in the outcome distribution between the treatment and control groups. The main advantage of these algorithms is their interpretability, which is very important for telecom companies when dealing with clients.

However, a significant drawback of state-of-the-art uplift decision tree algorithms is the need for user-defined parameters to train a decision tree model [63]. Examples of these parameters include the *maximum depth* of the tree, the *maximum number of features* to be used, and the *minimum number of instances* required in a leaf node. These decision tree algorithms depend only on local independent splits at each node. Once a tree model has been trained, a pruning step is performed to avoid overfitting and improve the predictive performance of the model.

This chapter introduces a novel user parameter-free decision tree algorithm called UB-DT that is specifically designed for uplift modeling. UB-DT is characterized by a Bayesian global criterion for an uplift decision tree that evaluates the quality of an induced uplift decision tree. The Bayesian evaluation global criterion for uplift decision trees $\mathbb{T}$ is defined by the posterior probability of $\mathbb{T}$ given uplift data.

In this chapter, we first define the parameters of an uplift decision tree model and demonstrate how to derive an uplift tree global criterion using these parameters. Our aim is to convert the uplift tree learning problem into an optimization problem, in which the goal is to search for the parameters that yield the best possible uplift tree model (i.e., with the highest global criterion score). Then, we present a search algorithm and introduce an extension for random forests, namely UB-RF.

Finally, we evaluate our proposed approaches using large scale experiments on real and synthetic datasets. These experiments show the efficiency of our methods over other state-of-art uplift modeling approaches.

The chapter is structured as follows: Section 4.2 presents the evaluation criterion for an uplift decision tree $\mathbb{T}$. It starts by presenting the parameters of $\mathbb{T}$ and the notations associated. Then it presents the evaluation criterion with a detailed proof in Section 4.2.3. We then present the decision tree algorithm that searches for the Bayesian optimal uplift tree model with the best criterion and its extension to a random forests algorithm. Finally in Section 4.3, we compare UB-DT and UB-RF with several uplift approaches on real and synthetic datasets.

This work is the object of the following publication:
Rafla, M., Voisine, N., & Crémilleux, B. (2023, May). Parameter-Free Bayesian Decision Trees for Uplift Modeling. In Advances in Knowledge Discovery and Data Mining: 27th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2023, Osaka, Japan, May 25–28, 2023, Proceedings, Part II (pp. 309-321). Cham: Springer Nature Switzerland.

4.2 Uplift Bayesian Decision Tree approach

UB-DT is made up of two ingredients: a global criterion $C(\mathbb{T})$ for a binary uplift decision tree $\mathbb{T}$ and a tree search algorithm to find the most probable optimal tree. We start by presenting the structure of an uplift tree model. Then we describe the new global criterion for an uplift decision tree and the algorithm to give the best tree. Finally we show how the approach is straightforwardly extended to random forests.

4.2.1 Parameters of an uplift tree model

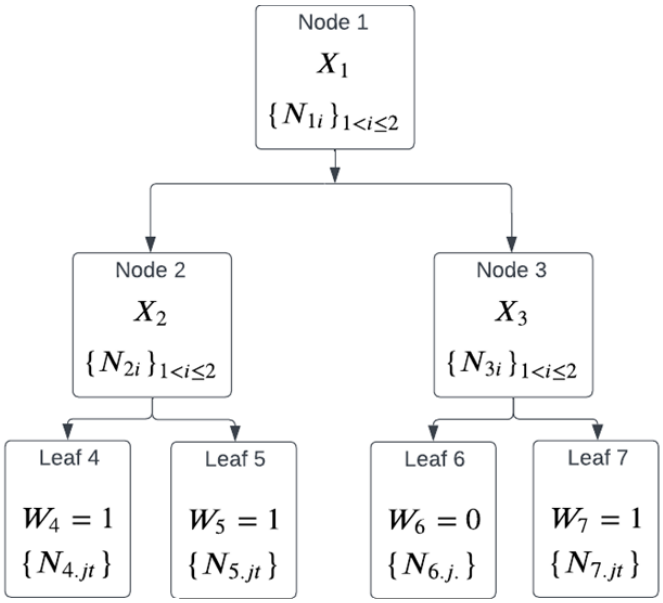


Figure 4.1: Example of an uplift tree model. Internal nodes are described by the segmentation variable X_s and the distribution of instances in each of the two children $\{N_{si}\}$. Leaf nodes containing a treatment effect (i.e $W_l = 1$) are described by the class distribution for each treatment. This applies to leaves 4, 5 and 7. Leaf nodes containing no treatment effect (i.e $W_l = 0$) are only described by the class distribution (this is the case of leaf 6).

We define a binary uplift decision tree model $\mathbb{T}$ by its structure and the distribution of instances and class values in this structure. The structure of $\mathbb{T}$ consists of the set of internal nodes $\mathbb{S}_\mathbb{T}$ and the set of leaf nodes $\mathbb{L}_\mathbb{T}$. The distribution of the instances in this structure is described by the partition of the segmentation variable X_s for each internal node s, the class frequency in each leaf node where there is no treatment effect, and the class frequency on each treatment in the leaf nodes with a treatment effect. More precisely, $\mathbb{T}$ is defined by:

- the subset of variables $\mathbb{K}_\mathbb{T}$ used by model $\mathbb{T}$. This includes the number of the selected variables $K_\mathbb{T}$ and their choice among a set of $\mathbb{K}$ variables provided in a dataset, we note $K = |\mathbb{K}|$.

- a binary variable I_n indicating the choice of whether each node n is an internal node ($I_n = 1$) or a leaf node ($I_n = 0$).

- the distribution of instances in each internal node s, which is described by the segmentation variable X_s of the node s and how the instances of s are distributed on its two child nodes.

- a binary variable W_l indicating for each leaf node l if there is a treatment effect ($W_l = 1$) or not ($W_l = 0$). If $W_l = 0$, l is described by the distribution of the output values $\{N_{l.j.}\}_{1 \leq j \leq J}$, where $N_{l.j.}$ is the number of instances of output value j in leaf l. If $W_l = 1$, l is described by the distribution of the class values per treatment $\{N_{l.jt}\}_{1 \leq j \leq J, 1 \leq t \leq 2}$, where $N_{l.jt}$ is the number of instances of output value j and treatment t in leaf l.

These parameters are automatically optimized by the search algorithm (presented in Section 4.2.4) and not fixed by the user. In the rest of the paper, the following notations $N_{s.}$, $N_{si.}$, $N_{l.}$ and $N_{l..t}$ will additionally be used to respectively designate the number of instances in node s, in the ith child of node s, in the leaf l and treatment t in leaf l.

4.2.2 Uplift tree evaluation criterion

We now present the new global criterion $C(\mathbb{T})$ which is an uplift tree model evaluation criterion. UB-DT applies a Bayesian approach to select the most probable uplift tree model $\mathbb{T}$ that maximizes the posterior probability $P(\mathbb{T}|Data)$. Let us consider the Bayes rule:

$$P(\mathbb{T}|Data) = \frac{P(\mathbb{T})P(Data|\mathbb{T})}{P(Data)}$$

Following the same approach of the UMODL discretization model criterion in Chapter 3, giving that $P(Data)$ is constant, maximizing $P(\mathbb{T}|Data)$ is equivalent to maximizing the product of the prior and the likelihood i.e. $P(\mathbb{T}) \times P(Data \mid \mathbb{T})$. This product represents a trade-off problem. On one hand, maximising the likelihood component requires a tree model with a number of leaf nodes equal to the number of data points. However, this leads to a smaller prior component. On the other hand, optimising the prior probability requires a decision tree model consisting of only a root node, but this results in a significantly reduced likelihood component.

Taking the negative log turns the maximization problem into a minimization one:

$$C(\mathbb{T}) = -\log\left(P(\mathbb{T}) \times P(Data|\mathbb{T})\right)$$

where $C(\mathbb{T})$ is the cost of the uplift tree model $\mathbb{T}$. $\mathbb{T}$ is optimal if $C(\mathbb{T})$ is minimal.

How to define the prior distribution $P(\mathbb{T})$?

To define the prior distribution of a tree model $\mathbb{T}$, we first exploit the hierarchy of the presented uplift tree parameters. This hierarchy describes the dependence relationships between parameters and requires the parameters to be selected in a particular order. The hierarchy of the tree model is described from the root node to its children and recursively to the leaves.

We also assume the independence of the distribution of the outcome values between children nodes. This assumption allows the prior of the model to be evaluated as a product of several terms. Taking the negative log, the prior can be evaluated as the sum of these terms (cf. Eq. 4.1).

Furthermore, we assume a *uniform distribution of parameters at every stage of the hierarchy*, as described in Section 4.2.3.

Using the components of the prior term described above (the parameter hierarchy, the uniform distribution assumption and the independence assumption), we show next the global evaluation criterion for an uplift decision tree and its detailed proof.

The Bayesian decision tree criterion

Combining the prior term $P(\mathbb{T})$ and using the likelihood terms on the tree leaves, we express the negative log of the posterior probability, our criterion $C(\mathbb{T})$, as follows (cf. Eq. 4.1):

$$
\begin{aligned}
C(\mathbb{T}) = & \underbrace{\log(K+1) + \log \begin{pmatrix} K + K_{\mathbb{T}} - 1 \\ K_{\mathbb{T}} \end{pmatrix}}_{\text{Variable selection}} \\
& + \underbrace{\sum_{s \in \mathbb{S}_{T_n}} \log 2 + \log K_{\mathbb{T}} + \log(N_{s.} + 1)}_{\text{Prior of internal nodes}} + \underbrace{\sum_{l \in \mathbb{L}_{\mathbb{T}}} \log 2}_{\text{Treatment effect W}} \\
& + \underbrace{\sum_{l \in \mathbb{L}_{\mathbb{T}}} \log 2 + \sum_{l \in \mathbb{L}_{\mathbb{T}}} (1 - W_l) \log \begin{pmatrix} N_{l.} + J - 1 \\ J - 1 \end{pmatrix} + \sum_{l \in \mathbb{L}_{\mathbb{T}}} W_l \sum_t \log \begin{pmatrix} N_{l..t} + J - 1 \\ J - 1 \end{pmatrix}}_{\text{Prior of leaf nodes}} \\
& + \underbrace{\sum_{l \in \mathbb{L}_{\mathbb{T}}} (1 - W_l) \log \frac{N_{l.}!}{N_{l.1.}! N_{l.2.}! \ldots N_{l.J.}!} + \sum_{l \in \mathbb{L}_{\mathbb{T}}} W_l \sum_t \log \frac{N_{l..t}!}{N_{l.1t}! .. N_{l.Jt}!}}_{\text{Tree Likelihood}}
\end{aligned}
\tag{4.1}
$$

The next section explains the different terms shows the proof of the criterion in.Eq. 4.1.

4.2.3 $C(\mathbb{T})$ proof of Equation 4.1

We express the prior and the likelihood of a tree model, resp. $P(\mathbb{T})$ and $P(Data|\mathbb{T})$ according to the hierarchy of the uplift tree parameters. Assuming the independence between all the nodes, the prior probability of an uplift decision tree is thus defined as:

$$
\begin{aligned}
P(\mathbb{T}) = & \underbrace{P(\mathbb{K}_{\mathbb{T}})}_{\text{Variable selection}} \times \\
& \underbrace{\prod_{s \in \mathbb{S}_{\mathbb{T}}} P(I_s) P(X_s \mid \mathbb{K}_{\mathbb{T}}) P(N_{si.} \mid \mathbb{K}_{\mathbb{T}}, X_s, N_{s.}, I_s)}_{\text{Prior of internal nodes}} \times \underbrace{P(\{W_l\})}_{\text{Treatment effect W}} \times \\
& \underbrace{\prod_{l \in \mathbb{L}_{\mathbb{T}}} P(I_l) \left[(1 - W_l) \times p(\{N_{l.j}\} \mid \mathbb{K}_{\mathbb{T}}, N_{l.}) + W_l \times \prod_t P(\{N_{l.jt}\} \mid \mathbb{K}_{\mathbb{T}}, N_{l..t}) \right]}_{\text{Prior of leaf nodes}}
\end{aligned}
\tag{4.2}
$$

The first line is the prior probability of the variable selection, the second line the prior of internal nodes and the third line the prior of the leaf nodes.

Variable selection probability

A hierarchical prior is chosen: first the choice of the number of selected variables $K_{\mathbb{T}}$, then the choice of the subset $\mathbb{K}_{\mathbb{T}}$ among $\mathbb{K}$ variables. By using a uniform prior the number $K_{\mathbb{T}}$ can have any value between 0 and K in an equiprobable manner. For the choice of the subset $\mathbb{K}_{\mathbb{T}}$, we assume that every subset has the same probability. Then the prior of the variable selection can be defined as:

$$P(\mathbb{K}_{\mathbb{T}}) = \frac{1}{K+1} \frac{1}{\binom{K + K_{\mathbb{T}} - 1}{K_{\mathbb{T}}}}$$

Prior of internal nodes

Each node can either be an internal node or a leaf node with equal probability. This implies that:

$$P(I_s) = \frac{1}{2}$$

The choice of the segmentation variable is equiprobable between 1 and $K_{\mathbb{T}}$. We obtain:

$$P(X_s|\mathbb{K}_{\mathbb{T}}) = \frac{1}{K_{\mathbb{T}}}$$

All splits of an internal node s to two intervals are equiprobable. We then obtain:

$$P\left(N_{si.} \mid \mathbb{K}_{\mathbb{T}}, X_s, N_{s.}, I_s\right) = \frac{1}{N_s + 1}$$

Prior of leaf nodes

Similar to the prior of internal nodes, each node can either be internal or a leaf node with equal probability leading to

$$P(I_l) = \frac{1}{2}$$

For each leaf node, we assume that a treatment can have an effect or not, with equal probability, we get:

$$P(\{W_l\}) = \prod_l \frac{1}{2}$$

In the case of a leaf node l where there is not effect of the treatment ($W_l = 0$), UB-DT describes one unique distribution of the class variable. Assuming that

each of the class distributions is equiprobable, we end up also with a combinatorial problem:

$$P\left(\{N_{l.j}\} \mid \mathbb{K}_{\mathbb{T}}, N_{l.}\right) = \frac{1}{\binom{N_{l.} + J - 1}{J - 1}}$$

In a leaf node with an effect of the treatment ($W_i = 1$), UB-DT describes two distributions of the outcome variable, with and without the treatment. Given a leaf l and a treatment t, we know the number of instances $N_{l..t}$ Assuming that each of the distributions of class values is equiprobable, we get:

$$P\left(\{N_{l.jt}\} \mid \mathbb{K}_{\mathbb{T}}, N_{l..t}\right) = \frac{1}{\binom{N_{l..t} + J - 1}{J - 1}}$$

Tree likelihood

After defining the tree's prior probability, we establish the likelihood probability of the data given the tree model. The class distributions depend only of the leaf nodes. For each multinomial distribution of the outcome variable (a single or two distinct distributions per leaf depending on whether the treatment has an effect or not), we assume that all possible observed data D_l consistent with the multinomial model are equiprobable. Using multinomial terms, we end up with:

$$\begin{aligned} &P(Data \mid \mathbb{T}) = \prod_{l \in L} P(Data_l | \mathbb{T}) = \\ &\prod_{l \in L} \left[(1 - W_l) \times \frac{1}{N_{l.}!/N_{l.1.}!N_{l.2.}! \ldots N_{l.J.}!} + W_l \times \prod_t \frac{1}{(N_{l..t}!/N_{i.1t}!..N_{i.Jt}!)} \right] \end{aligned} \tag{4.3}$$

By combining the prior and the likelihood (resp. Eq. 4.2 and 4.3) and by taking their negative log, we obtain $C(\mathbb{T})$ and thus Eq. 4.1 is proved.

4.2.4 Search algorithm

The induction of an optimal uplift decision tree from a data set is NP-hard [51]. Thus, learning the optimal decision tree requires exhaustive search and is limited to very small data sets. As a result, heuristic methods are required to build uplift decision trees. Algorithm 1 (see below) selects the best tree according to the global criterion. Algorithm 1 chooses a split among all possible splits in all terminal nodes only if it minimizes the global criterion of the tree. The algorithm continues as long as the global criterion is improved. Since a decision tree is a partitioning of the feature space, a prediction for a future instance is then the

average uplift in its corresponding leaf. This algorithm is deterministic and thus it always leads to the same local optimum. In Section 4.3 we perform several experiments that show the quality of the trees that are built. The next section shows how to extend this algorithm to build random forests.

Algorithm 1: UB-DT algorithm

```
input  : 𝕋 the root tree
output: the tree 𝕋* which minimizes the proposed criterion
𝕋* ← 𝕋
while C(𝕋*) decreases:
    𝕋' ← 𝕋*
    for leaf l in 𝕃_𝕋:
        for X in 𝕂:
            Get the best Split S_X(l) according to UMODL
            𝕋_X ← 𝕋* + S_X(l)
            if C(𝕋_X)<C(𝕋'):
                𝕋' ← 𝕋_X
    if C(𝕋')<C(𝕋*):
        𝕋* ← 𝕋'
Prediction: The output of a tree is a partition of the feature space. The
 predicted uplift for each instance is the average uplift of its leaf node.
```

4.2.5 UB-RF

Random forests are an ensemble machine learning algorithm consisting of multiple decision trees. They were first proposed by [12], where the author showed their efficiency against other classifiers such as support vector machines. Random forests were shown to have better performance and lower variance. However, one problem is that they lack the interpretability provided by a single decision tree.

UB-DT can be easily extended to random forests. In this extension, a split is randomly selected from all possible splits that improve the global criterion. The number of individual trees included in the random forest is determined by the analyst, and the overall prediction of the forest is calculated as the mean of all individual tree predictions. Algorithm 2 provides a detailed description of the Random Forest algorithm, hereafter referred to as UB-RF.

Algorithm 2: UB-RF algorithm

```
input  : training data, number of trees ntree
output: A forest of ntree trees
for n ← 1 in ntree:
    T_n ⟵ root tree
    S ⟵emply list
    while True:
        for l in L_{T_n}:
            for X in K:
                Get the best split S^n_X(l) according to UMODL
                /* add S^n_X(l) to S if it improves the global
                   criterion                                    */
                if C(T_n + S^n_X(l)) < C(T_n):
                    Add S^n_X(l) to S
        Split ⟵ rand(S)
        /* Stop when there is no more splits to improve the
           criterion                                            */
        if Split ⟵ ∅:
            break
        /* Perform the split in the tree T_n                    */
        T_n ⟵ T_n + Split
Predictions: The predicted uplift is the average of the predictions of the
 ntree trees
```

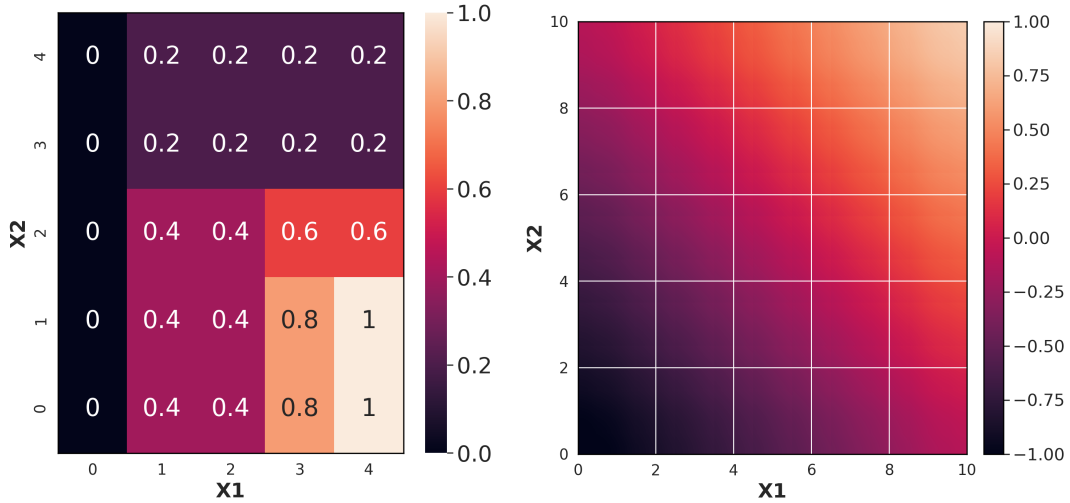
4.3 Experiments

We experimentally evaluate the quality of UB-DT as an uplift estimator in Section 4.3.1 and compare UB-DT and UB-RF versus state-of-art uplift modeling approaches in Section 4.3.2.

We use the following state-of-art methods: (1) metalearners: two-model approach (2M), X-Learner and R-Learner, each with Xgboost; (2) uplift trees: CTS-DT,KL-DT, Chi-DT, ED-DT; (3) uplift random forests: CTS-RF,KL-RF, Chi-RF, ED-RF [63]; (4) and causal forests. All approaches were used with 10 trees (Other experiments with 50 trees are shown in Appendix A.1).

4.3.1 Is UB-DT a good uplift estimator?

As we have done previously in Chapter 3, we start our experiments with synthetic datasets. Synthetic datasets are useful because they allow us to create the real



(a) Grid pattern

(b) Continuous pattern

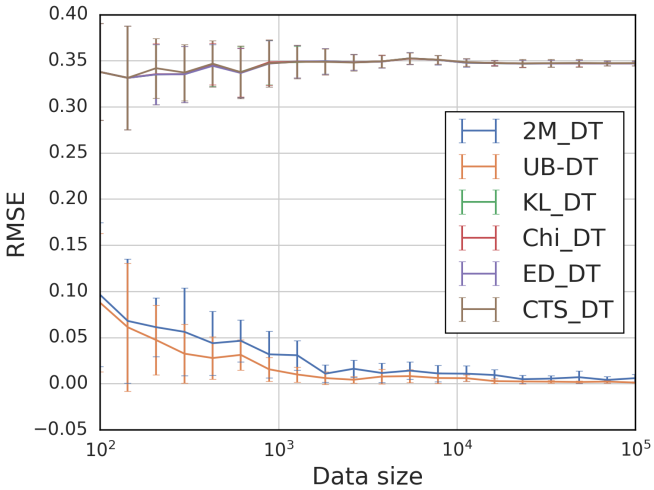
Figure 4.2: Uplift for 2 synthetic patterns. Fig. 4.2a (grid pattern): uplift values for each cell. Fig. 4.2b (continuous pattern): uplift values are $P(Y|T = 0, x1, x2) = 1 - (x1 + x2)/20$ while $P(Y|T = 1, x1, x2) = (x1 + x2)/20$.

uplift for each example and thus assess the quality of the estimated uplift by an uplift modeling algorithm.

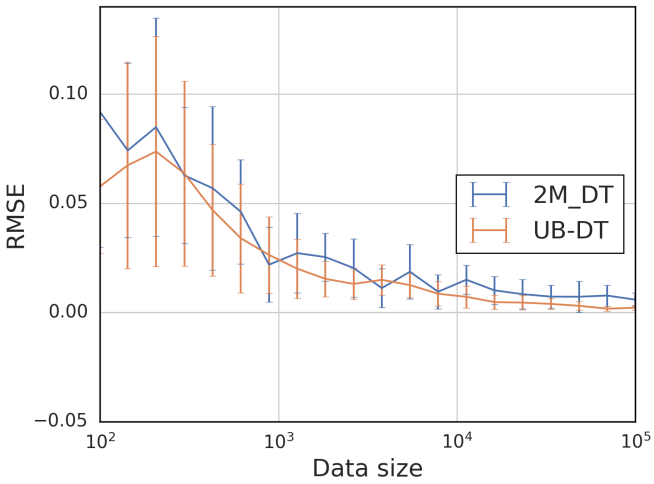
Fig. 4.2 depicts two synthetic uplift patterns where $P(Y = 1|X, T = 1)$ and $P(Y = 1|X, T = 0)$ are identified for each instance. The grid pattern can be considered as a tree-friendly pattern whereas the continuous pattern is much more difficult. We generated several datasets according to these patterns with several different numbers of instances (also called data size) ranging from 100 to 100,000 instances. Uplift models were built using 10-fold stratified cross validation and the RMSE (Root Mean Squared Error) was used to evaluate the performance of the models.

Results: Fig. 4.3 gives the RMSE for the two synthetic patterns according to the data size for different uplift methods. We see that UB-DT is a good estimator for uplift. With UB-DT, RMSE decreases and converges to zero when data sizes increase both for the grid and continuous patterns. This is the expected behavior of a good uplift estimator. This also means that UB-DT, thanks to its global criterion, avoids overfitting of uplift trees. The two-model approach with decision trees also shows competitive performance. UB-DT clearly outperforms the other tree-based methods, these latter having similar performances. With the continuous pattern, KL-DT, Chi-DT, ED-DT and CTS-DT approaches have lower

performances (their RMSE are around 0.5). To avoid a cluttered visualisation, their performances are not included in Fig. 4.3b.



(a) Grid Pattern



(b) Continuous Pattern

Figure 4.3: RMSE obtained by training tree-based methods.

4.3.2 UB-DT and UB-RF versus state of the art methods

Datasets We conducted experiments on 8 real and synthetic datasets widely used in the uplift modeling community:

[1]For efficiency purposes, model learning was conducted on a random sample of 200,000 instances for each fold.

Dataset	No. Rows	No. Columns	Treatment ratio	Outcome Ratio	Average Uplift	Treatment variable	Outcome variable
Hillstrom-m	42,613	10	0.5	0.145	0.076	*'mens'*	*'visit'*
Hillstrom-w	42,693	10	0.5	0.128	0.045	*'womens'*	*'visit'*
Hillstrom-mw	64,000	10	0.67	0.146	0.06	*'mens' & 'womens'*	*'visit'*
Gerber-N	229,444	16	0.166	0.31	0.081	*'neighbour'*	*'voted'*
Geber-S	229,461	16	0.166	0.304	0.04	*'self'*	*'voted'*
Starbucks	84,534	9	0.5	0.012	0.009	*'promotion'*	*'purchase'*
Information	20,000	69	0.5	0.2	0.0018	*'treatment'*	*'purchase'*
Bank-tel	15,926	17	0.18	0.05	0.09	*'telephone'*	*'Y'*
Bank-cell	42,305	17	0.6	0.115	0.11	*'cellular'*	*'Y'*
Bank-tel-cel	45,211	17	0.71	0.116	0.107	*'telephone'&'cellular'*	*'Y'*
Megafon	600,000	52	0.5	0.2	-0.18	*'treatment'*	*'conversion'*
Criteo-v[1]	13,979,592	12	0.85	0.047	0.68	*'treatment'*	*'visit'*
Criteo-c[1]	13,979,592	12	0.85	0.0029	0.37	*'treatment'*	*'conversion'*
RHC	5735	62	0.38	0.35	-0.05	*'RHC'*	*'swang1'*

Table 4.1: Summary of datasets specifications.

1. *Hillstrom*[2]: a classical dataset for uplift modeling with data of customers who either received emails featuring men's or women's products, or received no emails,

2. *Criteo* [19] (previously introduced in Section 3.5.3): a usual marketing dataset for uplift modeling,

3. *Bank* [49]: a marketing campaign conducted by a bank,

4. *Information*[3]: a marketing dataset in the insurance domain, a part of the Information R package,

5. *Megafon*[4]: a synthetic dataset created for uplift modeling. It is generated by telecom companies in order to reproduce the situations encountered by these companies,

6. *Starbucks*[5]: an advertising promotion tested to improve customers purchases,

7. *Gerber* [25]: a policy-relevant dataset used to study the effect of social pressure on voter turnout,

[2] http://blog.minethatdata.com/2008/03/minethatdata-e-mail-analytics-and-data.html

[3] https://cran.r-project.org/web/packages/Information/index.html

[4] https://ods.ai/tracks/df21-megafon/competitions/megafon-df21-comp/data

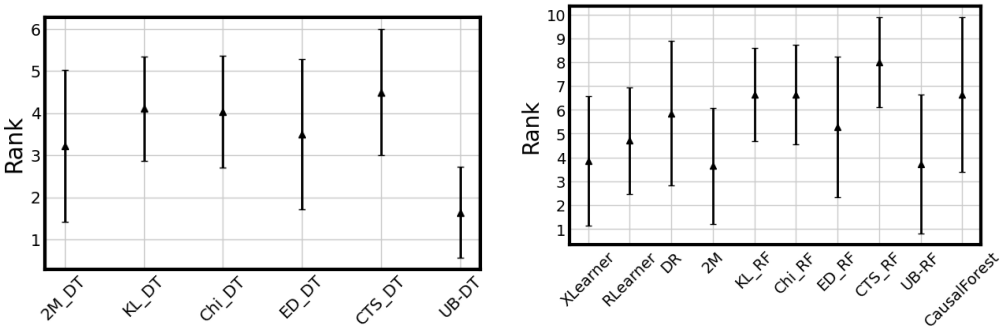
[5] https://github.com/joshxinjie/Data_Scientist_Nanodegree/tree/master/starbucks_portfolio_exercisejoshxinjie

8. *Right Heart Catheterization (RHC)* [16]: a real dataset from the medical domain, the treatment indicates whether a patient received a RHC and the outcome is whether the patient died at any time up to 180 days after admission to the study.

Each dataset was used with different settings of treatment and outcome variables. For all datasets, each treatment and outcome variables are binary. Table 4.1 provides the most relevant specifications about the data sets.

Results [6] We evaluate the uplift models by using the qini coefficient metric [18]. Fig. 4.4a (resp. Fig. 4.4b) shows the overall average ranking of tree based methods (resp. meta-learners and forest-based methods) according to its qini coefficient performance against each dataset. Compared to other tree-based methods and to the two-model approach with decision trees, Fig. 4.4a shows that UB-DT achieves the best performance. Table 4.2 reports the results of the experiment for the qini coefficient. This table shows that UB-DT is also a good estimator of the uplift on real data. Fig. 4.4b shows that both UB-RF, X-Learner and 2M have the best rank. Table 4.3 indicates that the random forest strategy improves the performance of the uplift models (the values of the qini coefficients are higher with UB-RF than UB-DT). UB-RF has the best performance on 4 out of the 14 experiments. With no altering to the main conclusions, a comparative study between the uplift approaches with 10 and 50 trees is shown in Appendix A.1. UB-RF has the best average ranking when the number of trees is increased to 50.

[6] Supplementary results with different uplift approaches can be found in https://github.com/MinaWagdi/UB-DT



(a) tree-based methods (b) meta-learners and forest-based methods

Figure 4.4: Overall average ranking of the uplift approaches

Dataset	2M-DT	KL-DT	Chi-DT	ED-DT	CTS-DT	UB-DT
Hillstrom-m	0.3(1.0)	1.1(1.9)	1.0(1.9)	0.0(1.4)	0.2(1.0)	**1.6(1.6)**
Hillstrom-w	0.8(1.6)	5.2(2.5)	5.2(2.6)	**6.4(1.2)**	-0.4(2.0)	4.8(2.3)
Hillstrom-mw	-0.6(0.8)	-0.1(1.2)	-0.8(1.1)	**4.4(2.7)**	-0.0(1.0)	-0.4(1.4)
Gerber-n	**5.6(0.8)**	1.3(0.8)	1.2(0.8)	1.1(0.6)	1.3(0.8)	1.9(0.6)
Gerber-s	**5.5(1.1)**	0.4(0.5)	0.4(0.6)	0.5(0.3)	0.4(0.4)	0.8(0.6)
Criteo-c	8.0(1.5)	4.1(1.4)	4.8(1.5)	**15.2(0.3)**	1.7(0.3)	13.7(3.2)
Criteo-v	0.4(0.3)	-1.2(0.2)	-1.1(0.3)	-1.3(0.3)	0.4(1.1)	**3.6(1.2)**
Megafon	5.1(0.6)	4.5(0.9)	4.7(0.9)	4.7(0.9)	4.9(0.8)	**7.8(0.8)**
Bank-tel	5.4(7.6)	-12.5(2.8)	-10.8(7.0)	-10.2(7.8)	-12.8(2.9)	**12.8(8.0)**
Bank-cell	11.1(3.0)	-2.0(1.5)	-1.4(2.5)	-2.2(1.5)	-3.7(1.5)	**38.4(3.4)**
Bank-tel-cell	10.3(1.6)	-1.9(1.2)	-1.2(2.1)	-1.8(1.2)	-3.4(1.4)	**37.1(2.6)**
Information	4.6(3.4)	-6.3(2.8)	-6.3(2.8)	-2.8(1.5)	-5.4(1.5)	**11.8(2.4)**
Starbucks	1.4(1.4)	20.1(3.0)	18.3(3.4)	19.9(3.2)	13.9(3.9)	**20.2(3.5)**
RHC	12.8(1.9)	18.4(3.8)	19.9(4.2)	18.4(3.8)	16.7(2.5)	**20.7(5.0)**

Table 4.2: Average qini coefficients and standard deviation (multiplied by 100). The best qini coefficient for each dataset is marked in bold.

Dataset	XLearner	RLearner	DR	2M	KL-RF	Chi-RF	ED-RF	CTS-RF	UB-RF	CausalForest
Hillstrom-m	0.3(2.3)	0.3(1.8)	1.2(1.6)	0.7(2.3)	-0.0(2.1)	-0.9(1.5)	0.7(1.5)	1.1(1.9)	**1.8(1.6)**	-0.2(1.6)
Hillstrom-w	6.2(1.7)	6.2(1.4)	6.0(1.4)	4.9(1.1)	6.2(1.1)	**7.0(1.0)**	6.2(1.1)	5.7(1.3)	6.7(1.1)	2.1(1.9)
Hillstrom-mw	3.7(2.3)	**3.9(2.7)**	3.8(2.8)	3.0(2.0)	3.0(1.3)	2.8(1.5)	3.6(2.5)	2.3(2.4)	3.1(1.7)	0.1(1.7)
Gerber-n	**3.7(0.6)**	1.9(0.7)	0.5(0.9)	3.1(0.6)	1.8(1.0)	2.1(1.1)	1.9(0.5)	1.4(1.0)	2.7(0.7)	2.9(1.0)
Gerber-s	2.4(0.9)	1.7(0.7)	0.6(0.9)	2.2(0.8)	1.3(1.0)	1.4(0.6)	1.6(0.8)	1.4(0.7)	1.8(0.8)	**3.1(0.5)**
Criteo-c	**22.3(1.8)**	19.4(1.0)	20.0(0.6)	19.5(1.6)	14.6(3.5)	12.4(4.3)	21.1(2.3)	7.3(3.9)	18.7(1.5)	10.9(2.4)
Criteo-v	0.3(0.8)	5.3(0.5)	4.8(1.5)	3.9(0.5)	5.4(1.2)	4.8(1.7)	**6.1(1.0)**	2.4(0.8)	5.7(0.7)	0.4(0.4)
Megafon	**18.2(0.6)**	2.6(0.5)	2.2(0.9)	16.6(0.9)	11.2(0.7)	11.0(1.2)	10.8(0.8)	9.2(1.1)	12.8(1.0)	9.7(0.7)
Bank-tel	14.5(7.6)	2.8(8.8)	16.0(9.0)	21.1(11.6)	-15.5(6.3)	-6.1(12.6)	-15.8(5.6)	-18.7(2.9)	**26.7(7.2)**	25.4(5.3)
Bank-cell	18.8(4.7)	23.3(3.6)	17.4(6.5)	31.0(3.9)	0.4(2.3)	1.5(2.5)	-2.5(2.6)	-1.0(1.9)	**45.5(2.7)**	20.8(2.6)
Bank-tel-cell	16.2(5.6)	23.8(2.5)	17.0(3.4)	30.5(2.7)	1.4(3.4)	-0.4(5.7)	-1.7(3.1)	-0.5(2.3)	**46.1(2.1)**	23.5(2.9)
Information	**14.9(3.3)**	10.0(3.1)	4.1(2.3)	13.7(4.1)	9.6(2.0)	9.7(3.1)	11.2(2.9)	10.6(2.9)	12.0(3.1)	10.5(3.2)
Starbucks	22.3(4.5)	22.4(3.9)	22.4(3.7)	22.7(4.1)	22.4(2.1)	21.4(3.4)	**23.4(3.2)**	20.8(3.1)	20.2(3.3)	8.1(3.7)
RHC	32.4(3.5)	31.3(4.3)	30.3(5.0)	**34.6(4.3)**	29.6(4.2)	29.7(5.0)	30.0(4.1)	29.1(3.7)	27.2(5.0)	27.6(4.5)

Table 4.3: Average qini coefficients and standard deviation (multiplied by 100) across datasets and uplift approaches. In bold, the best value for each dataset

Computation time In this part, we compare the computation time of different uplift modeling algorithms. However, comparing the computation time of these uplift methods should be performed carefully, as it is strongly influenced by the quality of the implementation and the programming language used. For meta-learners, it also depends on the complexity of the chosen supervised learning algorithm. We present in Table 4.4 the computation time for each uplift approach with respect to the first fold in each dataset. This allows us to get a general idea of the time consumption of the current implementation of our approach compared to the other state-of-the-art approaches, and to determine whether it is tractable and can still be used by the research community.

Table 4.4 shows that causal forests are the fastest learning approach of all tested uplift modeling approaches. We also note that the tested version is implemented in the C programming language. Their computation times range from 1.4 seconds to 16.9 seconds. X-learner, R-learner, DR-learner and 2M come in second place, their computation times range from 7.6 seconds (for the 2M approach) to 33.2 seconds (for the DR-learner).

The uplift random forests (KL-RF, Chi-RF, ED-RF, CTS-RF, UB-RF) have the longest computation times. Especially UB-RF is computationally expensive. UB-RF is based on the UMODL discretization approach (see Chapter 3), for which the complexity is $N \log N$. The search algorith of the UMODL discretization depends on the size of the data, the type of columns and the number of unique values in the columns. The UB-RF algorithm takes longer to find the optimal split for continuous variables. The more values a variable has, the longer the algorithm takes. For example, UB-RF takes the longest time on the criteo dataset, which contains 14 continuous variables, half of which have more than 1000 values. On the contrary, the gap between the time consumption of UB-RF and other forest-based approaches is the smallest on the Hillstrom and Starbucks datasets, which each contain a small number of continuous variables.

In addition, our implemented version of UB-RF is in Python and its implementation is sub-optimal. This observation leads to future works to optimize the algorithm and its implementation.

Dataset	XLearner	RLearner	DR	2M	KL-RF	Chi-RF	ED-RF	CTS-RF	UB-RF	CausalForests	Cont. cols	Cont. cols > 1000 values
Hillstrom-m	14.769	14.926	15.624	7.966	64.8	66.202	66.728	44.485	60.609	2.023	5	1
Hillstrom-w	14.486	14.614	15.629	7.664	63.912	61.765	63.633	41.815	65.537	2.03	5	1
Hillstrom-mw	15.614	14.857	15.78	7.914	99.732	97.967	102.941	69.53	102.351	2.275	5	1
Gerber-n	19.194	18.79	20.257	9.202	398.158	503.291	350.923	346.62	1139.27	5.37	6	2
Gerber-s	30.319	29.136	34.821	15.233	432.648	381.878	505.065	300.064	1009.788	16.936	6	2
Criteo-c	25.204	20.918	21.782	14.061	382.942	375.397	468.533	132.642	2944.905	9.041	14	7
Criteo-v	24.423	21.084	21.837	13.612	382.189	348.73	337.542	145.127	3013.66	9.218	14	7
megafon	26.844	24.472	26.519	12.519	205.696	195.138	253.749	229.131	2443.51	24.945	50	50
Bank-tel	18.47	17.566	17.596	8.826	26.652	28.367	24.809	20.773	61.075	1.413	7	2
Bank-cell	19.167	19.279	19.328	9.471	91.581	96.933	80.902	53.773	250.195	2.032	7	2
Bank-tel-cell	19.251	18.681	33.251	15.924	57.693	62.99	52.842	37.069	279.54	2.102	7	2
Information	25.163	24.581	40.833	21.038	29.226	30.676	28.871	21.899	601.167	1.561	67	13
Starbucks	22.398	24.225	26.289	7.942	130.392	86.709	69.791	99.973	84.962	5.406	2	1
RHC	23.142	20.442	20.648	9.644	15.301	15.295	15.663	12.427	166.619	2.153	7	7

Table 4.4: Computational time (in seconds) per Uplift approach for the first fold in each dataset. The last two columns represent the number of continuous columns and those with over 1000 values.

4.4 Conclusion

In this chapter, we have presented a new parameter-free method called UB-DT for uplift decision trees. We have designed a Bayesian approach to select the most probable uplift tree model $\mathbb{T}$ that maximizes the posterior probability $P(\mathbb{T}|Data)$. Contrary to state-of-art uplift decision tree approaches, UB-DT is characterized by a global criterion to build a tree, so the splits in one node depend on the splits in the other nodes. This approach avoids overfitting and the need for a pruning step. A search algorithm finds the tree that optimizes this criterion. We have shown that our approach is easily extended to random forests and we have defined UB-RF. Evaluations on real and synthetic data sets show that UB-DT is a good uplift estimator and our tree and forests methods perform competitively with state-of-art uplift modeling approaches including non tree methods.

CHAPTER 5

Evaluation of Uplift Models with Non-Random Assignment Bias

Contents

5.1 Introduction

Uplift modeling algorithms assume that the treatment and control groups are drawn from the same distribution. While this strong assumption is potentially valid in experimental data and controlled trials, it often does not hold in real-world scenarios. The nature of real-world data is mainly observational, which means that it is collected without conducting a controlled experiment. Consequently, we cannot guarantee that the treatment and control groups have the same distribution.

As outlined in Section 1.3, two distinct communities are attempting to address the problem of treatment effect estimation: the heterogeneous treatment effect estimation community, via the CATE estimation task, and the uplift modelling community. In the former, CATE estimation algorithms take into account non-random assignment bias (described in Section 2.5.1) and any difference between the distributions of the treatment and control groups. Examples of these algorithms include DR-Learner and X-Learner, as discussed in Section 2.2.1. Conversely, uplift modeling assumes equivalence of the two distributions. This is a strong assumption that rarely holds in real-world data, such as the telecom data.

In this chapter, we revisit the definition of the non-random assignment bias and we address the following research questions:

1. What is the impact of NRA bias on the main uplift modeling and CATE estimation approaches [1]?

2. Does our newly proposed Uplift Bayesian Decision Tree (UB-DT) algorithm perform well in the presence of NRA bias?

3. How can the bias effect be reduced?

To answer the first and second questions, we design an experimental protocol to evaluate the impact of NRA bias on state-of-the-art uplift methods. Our study allows us to identify several behavioural aspects of uplift methods. To address the third question, we define a reweighting method based on the inverse propensity weighting (IPW) approach to reduce the impact of NRA bias on the performance of the class transformation approach, which is found to be one of the methods most affected by NRA bias. Our experimental results show that this bias reduction method significantly improves the performance of the class transformation approach in the presence of NRA bias.

The remainder of this chapter is organized as follows. Section 5.2 describes the problem setting and our experimental protocol for evaluating the impact of NRA bias. We present our reweighting method in Section 5.3 then conclude in Section 5.4

This work is the object of the following publication:
Rafla, M., Voisine, N., & Crémilleux, B. (2022, April). Evaluation of Uplift Models with Non-Random Assignment Bias. In Advances in Intelligent Data Analysis XX: 20th International Symposium on Intelligent Data Analysis, IDA 2022, Rennes, France, April 20–22, 2022, Proceedings (pp. 251-263). Cham: Springer International Publishing.

5.2 Evaluation of uplift with biased data

This section presents the NRA bias and the experimental protocol that we designed to assess performance of uplift methods under this bias.

[1] For simplicity, both methods will be referred to as 'uplift modeling approaches'.

5.2.1 Problem setting

Some state-of-art uplift methods assume that data are unbiased and that the treatment group comes from the same distribution as the control group, which is not true for real data. In practice, there are often differences between treatment and control groups, also known as Non-Random Assignment bias, a prevalent type of bias in uplift modeling [77]. Formally, this bias occurs when $P(T = 1|X) \neq P(T = 0|X)$ (which also means $P(X|T = 1) \neq P(X|T = 0)$). Usually it is easier to collect control data and the treatment group is the most biased because it is more challenging to apply a treatment to individuals and collect the corresponding data due to ethical, political or economic constraints.

This bias problem has been studied in the literature on clinical studies where the goal is to estimate the "Average Treatment Effect" (ATE) defined as $\mathbb{E}[Y_i(T = 1) - Y_i(T = 0)]$. In order to estimate it, the "Propensity Score Matching" (PSM) [61] is used to extract balanced treatment groups on which ATE is estimated. Similarly, in the uplift literature, since uplift methods assume the homogeneity between treatment groups, PSM is used to extract an unbiased sample from a biased dataset. Uplift modeling is applied subsequently as carried in [53]. However, this procedure clearly suffers from a loss of data.

5.2.2 Designing of the experimental protocol

This section describes the experimental protocol that we designed to evaluate the behavior of uplift methods under the NRA bias. The principle, to create a NRA bias and observe its impact, is to introduce imbalances in the data regarding the initial distribution of the variables. We do this by modifying proportions of individuals in a non-random way (for example, decreasing the proportion of specific socio-professional categories or ages till it disappears in the data). Such a protocol must satisfy several conditions to correctly evaluate the impact of NRA in order to avoid introducing a bias due to the protocol itself.

- The chosen variables to introduce bias have to be correlated with the outcome Y or Y given the treatment T, otherwise the bias will not affect the uplift modeling.

- In contrast, the choice of the values of the variables, according to which the proportions of individuals vary, is random. If not, the construction of the populations $E1$ and $E2$ (which will be explained below) may be biased.

- The bias must be tunable in order to change its rate and quantify its impact on the uplift methods.

- The created bias is only in the treatment group in order to imitate the natural phenomena as previously explained in Section 5.2.1.
- The total size of each of the biased learning samples is always the same in order to avoid any variation in the performance due to different learning data sizes.

More precisely, as shown in Fig. 5.1, two populations $E1$ and $E2$ are created. This is done by choosing a set of variables V and dividing its values into two groups, $C1$ and $C2$, such that the number of individuals defined by the values of $C1$ is equivalent to the number of individuals defined by $C2$. Let $E1$ (resp. $E2$) be the population whose variables correspond to $C1$ (resp. $C2$) and whose sizes are $N1$ and $N2$ respectively. We use a 10-fold cross-validation. In the first training sample, $E1$ and $E2$ have an equal size (i.e. $N1 = N2$), it is considered unbiased and gives a reference value of the qini coefficient. The NRA bias is gradually introduced in the treatment group by increasing the size of $E1$ and decreasing the size of $E2$ while preserving the total size of the treatment group. We identify the bias rate of a sample by the variable b where $b = (N1 - N2) \times 100/N$. b goes from $b = 0$ in the unbiased situation to $b = 100$ the most biased situation according to the NRA bias. An uplift model is then learned on each biased sample defined by b. All models are then tested on the same test sample and evaluated using the qini coefficient. The evolution of the qini coefficient according to b allows studying the behavior of an uplift method towards the NRA bias.

5.2.3 Experiments

In this section, we introduce the datasets, the uplift modelling approaches and the details of the experiments.

Datasets We use several real and synthetic datasets from different fields that are widely used in the literature. All these datasets have been previously described in Section 3.5.3 and in Section 4.3.2. Below is a brief description of the datasets used:

1. Criteo [19]: a usual marketing dataset for uplift modeling.
2. Hillstrom[2]: another classical marketing dataset for uplift modeling.
3. Gerber [25]: a policy-relevant dataset.

[2] http://blog.minethatdata.com/2008/03/minethatdata-e-mail-analytics-and-data.html/

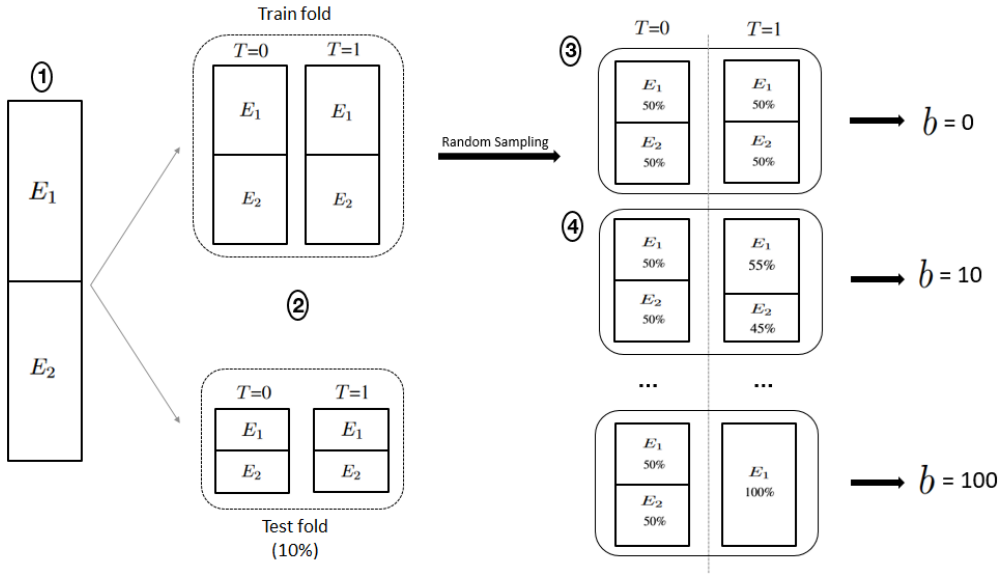


Figure 5.1: Biased samples generation procedure for one fold (a 10-fold cross validation is used): (1) Variable(s) V is chosen to create E1 and E2. (2) Creating training and test sets with 10-fold cross validation. (3) Random sampling of treatment and control groups. (4) The sizes of the treatment and control groups are always the same throughout the biasing process.

4. Retail Hero[3]: a marketing dataset of the X5 sales group.
5. Megafon[4]: a synthetic dataset created by telecom companies for uplift modeling.
6. Zenodo[5]: a synthetic dataset containing trigonometric patterns specifically designed for uplift modeling. This dataset consists of 100 trials of different patterns. We only used the first trial of the dataset.
7. Continuous and Grid patterns: two synthetic datasets, each consisting of two variables.
8. Information: a marketing dataset.
9. Starbucks: an advertising promotion dataset.
10. Bank: a marketing campaign conducted by a bank.

[3] https://ods.ai/competitions/x5-retailhero-uplift-modeling/data
[4] https://ods.ai/tracks/df21-megafon/competitions/megafon-df21-comp/data
[5] https://zenodo.org/record/3653141#.YUCYEufgoW8

Uplift methods We test 18 uplift methods: two-model approach (2M); class-transformation approach (CT), each with Xgboost and logistic regression (LR); DR-learner (DR); X-learner, R-learner and S-learner, each with Xgboost and linear regression (LinR). Direct-approaches based on random forests (RF) and decision trees (DT) are tested as well: KL, ED [63] and CTS [79].

Implementation details For each dataset (except Synth1 and Synth2) and for each uplift method, the experimental protocol is run twice with different compositions of the set of variables V on which the bias is created: once including the most important variable within the dataset and once with the second most important variable (The use of a non-informative or random variable in the bias generation process won't yield a difference between the treatment and control groups). For Synth1 and Synth2, V contains the two variables of these datasets. Moreover, given a set V, the experiment is repeated twice in order to provide different splittings of $C1$ and $C2$.

5.2.4 Results

Qini coefficients variability according to b

Fig. 5.2 illustrates a subset of the results. We observe that the NRA bias strongly affects the performance of uplift models (the higher the bias rate, the more significant the decrease of the qini coefficient).

To provide a global view of the results, we compute for each dataset and each uplift method the *average qini coefficient*, i.e., the average of qini coefficients according to the bias rates going from $b = 0$ to $b = 100$. We show the result of the average qini for all datasets in Table 5.1 and Table 5.2. Each dataset name is followed by the variable V used to introduce the bias. We have tested different splits of V in $C1$ and $C2$. These variations are indicated by an apostrophe after the variable name in both Table 5.1 and Table 5.2.

Overall ranking

To better compare the methods according to their resistance to NRA bias, Fig. 5.3 shows the average rank obtained by each method based on the average qini coefficient (all divisions of V are taken into account).

The results of our experiments show the following conclusions:

- The models with the highest resistance to the NRA bias are mainly the meta-learners: the X-learner, the R-learner, the DR-learner and ***our newly***

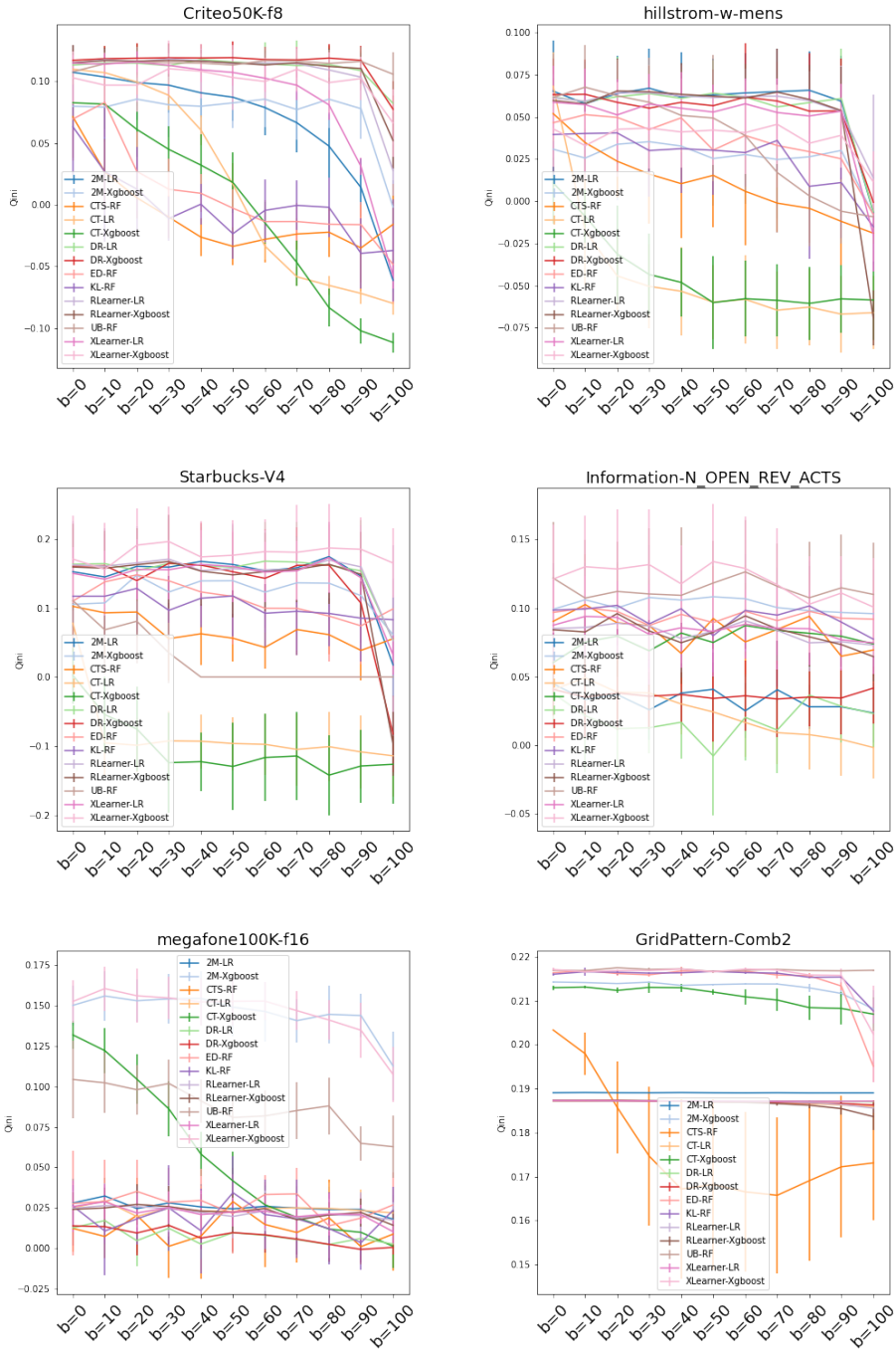


Figure 5.2: Qini coefficients of uplift methods based on NRA bias rates. The figure title consists of the dataset name followed by the variable used to generate the bias. In the legend, a method name is followed by the associated learning algorithm.

Dataset	KL-DT	CTS-DT	ED-DT	UB-DT
zenodo-trial0-x32	6.08(1.45)	1.82(0.36)	**7.02(0.39)**	5.08(0.76)
zenodo-trial0-x32'	**6.18(0.61)**	1.82(0.36)	5.97(0.56)	6.03(1.02)
zenodo-trial0-x34	5.6(0.47)	1.58(0.44)	**5.67(0.49)**	2.88(1.48)
zenodo-trial0-x34'	6.83(0.72)	3.94(1.94)	**7.28(0.76)**	4.65(0.68)
GridPattern-Comb2	16.77(0.77)	12.18(1.3)	17.14(0.2)	**20.7(3.19)**
GridPattern-Comb2'	17.11(0.29)	7.09(0.48)	17.29(0.23)	**21.17(1.6)**
ContPattern-Comb2	24.27(0.12)	23.49(0.16)	24.59(0.34)	**29.0(2.46)**
ContPattern-Comb2'	24.33(0.13)	23.34(0.2)	24.82(0.09)	**29.61(0.65)**
hillstrom-w-mens	2.84(1.77)	0.22(1.29)	**4.2(2.05)**	0.26(2.1)
hillstrom-w-mens'	6.08(0.33)	6.01(1.3)	**6.18(0.17)**	4.79(2.84)
hillstrom-w-womens	-0.7(3.68)	-2.13(2.24)	0.2(4.49)	**2.97(1.57)**
hillstrom-w-womens'	-0.7(3.68)	-2.13(2.24)	0.2(4.49)	**2.97(1.57)**
hillstrom-m-mens	0.66(0.32)	**1.1(0.34)**	0.82(0.48)	0.61(0.41)
hillstrom-m-mens'	1.2(0.18)	1.31(0.23)	1.12(0.3)	**1.66(0.73)**
hillstrom-m-womens	1.0(0.31)	1.14(0.35)	0.96(0.37)	**1.56(0.7)**
hillstrom-m-womens'	1.0(0.31)	1.14(0.35)	0.96(0.37)	**1.56(0.7)**
Criteo50K-f2	-0.64(2.58)	-0.15(1.58)	0.58(3.12)	**9.27(3.29)**
Criteo50K-f2'	7.03(2.89)	8.51(1.97)	8.14(2.39)	**9.43(0.52)**
Criteo50K-f8	-1.39(1.54)	0.52(2.54)	-0.14(1.97)	**9.33(2.17)**
Criteo50K-f8'	-2.31(2.55)	0.15(1.04)	-0.55(3.16)	**9.85(0.99)**
megafone100K-f35	-1.77(0.7)	-2.23(1.44)	-1.81(0.81)	**7.93(0.42)**
megafone100K-f35'	-0.17(0.68)	0.04(0.22)	-0.47(0.63)	**6.28(0.68)**
megafone100K-f16	-0.65(1.22)	-0.26(0.61)	-1.28(1.54)	**5.67(1.62)**
megafone100K-f16'	-0.2(0.29)	-0.18(0.24)	-0.21(0.41)	**7.17(0.4)**
Bank3-duration	-0.91(1.11)	-4.53(0.54)	0.07(0.31)	**14.15(6.78)**
Bank3-duration'	-0.96(1.11)	-2.45(1.81)	-1.59(0.77)	**15.06(2.55)**
Bank3-month	0.17(2.1)	-28.61(27.27)	-1.43(1.47)	**8.79(5.22)**
Bank3-month'	0.53(2.57)	-4.64(1.54)	-5.42(10.13)	**14.98(3.57)**
Information-N-OPEN-REV-ACTS	-2.92(1.9)	0.67(2.82)	-3.32(0.8)	**10.57(1.54)**
Information-N-OPEN-REV-ACTS'	-0.26(3.01)	-0.8(2.03)	-1.27(2.01)	**9.8(0.83)**
Information-PREM-BANKCARD-CRED-LMT	0.15(2.59)	0.46(2.4)	-0.59(1.97)	**10.9(0.67)**
Information-PREM-BANKCARD-CRED-LMT'	-4.86(1.25)	-3.41(1.69)	-3.97(1.27)	**9.81(0.97)**
Starbucks-V4	7.75(2.68)	2.87(1.72)	**12.02(2.57)**	2.69(3.91)
Starbucks-V4'	7.75(2.68)	2.87(1.72)	**12.02(2.57)**	2.69(3.91)
Starbucks-V5	12.08(2.2)	11.77(2.03)	**14.01(2.29)**	1.67(3.24)
Starbucks-V5'	13.38(0.77)	11.03(1.9)	**13.9(0.89)**	11.81(2.19)
Gerber-cluster	**0.28(0.32)**	0.15(0.34)	0.11(0.23)	-1.43(0.21)
Gerber-cluster'	**1.0(0.09)**	0.88(0.12)	0.96(0.13)	-1.65(0.15)
Gerber-yob	**0.85(0.14)**	0.79(0.07)	0.76(0.08)	-2.26(0.3)
Gerber-yob'	0.46(0.22)	**0.51(0.2)**	0.42(0.2)	-1.69(0.34)
Retail-express-spent-mean	-0.09(0.17)	0.0(0.18)	-0.12(0.17)	**0.7(0.25)**
Retail-express-spent-mean'	-0.59(0.1)	-0.55(0.18)	-0.6(0.07)	**0.89(0.22)**
Retail-first-redeem-date	0.11(0.09)	0.29(0.15)	0.05(0.09)	**0.87(0.1)**
Retail-first-redeem-date'	-0.14(0.24)	-0.21(0.37)	-0.15(0.24)	**0.69(0.19)**

Table 5.1: Average qini coefficients with standard deviation for tree. In bold, the best value for each dataset.

	R-Learner		Class Transformation		DR-Learner		X-Learner		2M		RF			
Dataset	Xgboost	LR	Xgboost	LR	Xgboost	LR	Xgboost	LR	Xgboost	LR	KL	ED	CTS	UB
zenodo-trial0-x32	9.78(0.31)	9.32(0.52)	7.06(1.14)	9.65(0.59)	11.14(0.78)	11.27(0.36)	**11.28(0.52)**	9.31(0.28)	10.27(0.3)	9.74(0.28)	8.8(0.95)	9.61(0.7)	5.99(0.92)	10.04(0.84)
zenodo-trial0-x32'	9.38(0.86)	10.32(0.26)	7.51(0.98)	9.7(0.48)	11.01(0.89)	10.96(1.48)	**11.59(0.67)**	9.73(0.61)	9.8(0.54)	10.59(0.57)	8.86(0.85)	9.06(0.98)	4.54(0.62)	9.86(0.62)
zenodo-trial0-x34	9.32(0.32)	8.74(0.87)	7.9(1.22)	8.99(0.75)	10.9(1.09)	10.71(1.73)	**11.59(0.61)**	9.18(0.47)	9.58(1.02)	10.58(0.25)	9.08(0.76)	8.78(0.65)	4.74(1.13)	10.2(0.81)
zenodo-trial0-x34'	9.4(0.41)	10.04(0.47)	8.41(0.85)	10.21(0.21)	10.9(1.09)	**11.64(0.23)**	11.15(1.05)	9.61(0.29)	9.35(0.44)	10.46(0.21)	9.21(0.96)	8.78(0.65)	6.86(1.66)	10.52(1.08)
GridPattern-Comb2	18.65(0.11)	18.69(0.05)	21.1(0.21)	18.69(0.04)	18.7(0.03)	18.72(0.01)	21.54(0.41)	18.72(0.01)	21.31(0.17)	18.91(0.0)	21.54(0.25)	21.41(0.61)	17.67(1.25)	**21.7(0.02)**
GridPattern-Comb2'	18.58(0.24)	18.54(0.21)	20.96(0.38)	18.71(0.02)	18.68(0.07)	18.72(0.01)	21.52(0.41)	18.72(0.01)	21.27(0.37)	18.92(0.0)	**21.64(0.02)**	21.38(0.6)	18.81(0.74)	20.78(2.95)
ContPattern-Comb2	**30.92(0.0)**	30.85(0.11)	30.35(0.02)	30.92(0.0)	30.8(0.37)	30.84(0.13)	30.51(0.38)	30.92(0.0)	30.35(0.09)	30.92(0.0)	30.36(0.3)	30.35(0.29)	29.88(0.26)	30.08(1.18)
ContPattern-Comb2'	**30.92(0.0)**	30.91(0.0)	30.35(0.03)	30.92(0.0)	30.9(0.06)	30.82(0.28)	30.62(0.17)	30.92(0.0)	30.34(0.2)	30.92(0.0)	30.35(0.28)	30.35(0.31)	28.84(0.93)	30.13(0.99)
hillstrom-w-mens	4.95(3.77)	5.68(1.39)	-4.31(2.29)	-4.33(3.78)	5.31(1.73)	5.48(1.95)	3.79(0.91)	4.8(2.17)	2.59(1.13)	**5.7(2.03)**	2.56(1.64)	3.54(1.62)	1.11(1.97)	3.59(2.78)
hillstrom-w-mens'	6.16(0.21)	5.68(1.39)	4.62(1.29)	6.21(0.19)	6.06(0.34)	5.8(1.11)	3.66(0.56)	6.01(0.18)	3.45(0.61)	**6.22(0.28)**	4.77(0.48)	4.49(0.54)	5.72(0.48)	5.54(1.55)
hillstrom-w-womens	4.97(3.75)	**5.95(0.87)**	-4.78(2.54)	-5.43(3.65)	4.81(3.7)	5.54(1.86)	3.38(1.23)	5.47(1.65)	2.71(1.7)	5.27(3.74)	1.32(2.54)	2.03(2.33)	-0.31(2.62)	4.54(2.26)
hillstrom-w-womens'	4.97(3.75)	**5.95(0.87)**	-4.56(2.63)	-5.03(3.72)	4.86(3.76)	5.32(1.94)	4.08(1.29)	5.45(2.4)	3.02(1.65)	4.8(3.65)	1.79(2.16)	2.25(2.2)	0.27(2.68)	5.07(1.92)
hillstrom-m-mens	**0.8(0.67)**	0.66(0.75)	0.2(0.35)	-0.0(0.71)	0.42(0.66)	0.36(0.49)	0.2(0.62)	0.61(0.81)	0.21(0.47)	0.64(1.34)	-0.6(0.56)	-0.43(0.72)	-0.05(0.74)	0.59(1.17)
hillstrom-m-mens'	1.64(0.7)	1.67(0.59)	-0.65(0.28)	-0.05(0.6)	0.97(0.6)	1.03(0.59)	0.63(0.53)	1.82(0.52)	0.36(0.36)	1.75(0.65)	-0.03(0.52)	0.31(0.4)	0.3(0.49)	**1.86(0.4)**
hillstrom-m-womens	0.85(0.67)	1.01(0.44)	-1.48(0.53)	-0.16(0.55)	0.63(0.61)	0.79(0.52)	-0.07(0.38)	1.22(0.53)	-0.66(0.33)	1.51(0.66)	-0.46(0.52)	-0.17(0.55)	0.25(0.41)	**1.77(0.4)**
hillstrom-m-womens'	0.85(0.67)	1.01(0.44)	-1.48(0.53)	-0.16(0.55)	0.63(0.61)	0.79(0.52)	-0.07(0.38)	1.22(0.53)	-0.66(0.33)	1.51(0.66)	-0.46(0.52)	-0.17(0.55)	0.25(0.41)	**1.73(0.42)**
Criteo50K-f2	10.8(1.19)	10.44(2.09)	0.09(6.61)	2.01(7.11)	10.9(2.86)	11.39(0.97)	9.62(1.28)	8.59(4.7)	7.11(2.03)	6.63(4.43)	1.16(3.23)	1.35(3.89)	-0.73(3.23)	**11.7(0.58)**
Criteo50K-f2'	10.64(0.34)	10.71(0.41)	7.27(1.07)	7.07(1.9)	11.16(0.58)	**11.57(0.22)**	7.76(0.58)	10.74(0.36)	7.74(0.41)	10.6(0.28)	8.39(0.92)	9.14(0.44)	8.55(2.36)	11.19(0.22)
Criteo50K-f8	10.91(1.81)	10.62(2.51)	-0.35(6.92)	1.57(7.56)	**11.46(1.18)**	11.1(0.93)	9.96(1.15)	8.44(5.07)	7.38(2.41)	6.64(4.83)	-0.14(2.78)	0.83(3.71)	-0.83(3.06)	11.37(0.34)
Criteo50K-f8'	10.87(0.79)	10.77(0.95)	-2.47(5.1)	-1.9(6.06)	11.05(1.12)	**11.16(0.53)**	9.2(1.65)	9.0(4.47)	6.36(2.3)	6.57(4.14)	1.33(1.21)	2.37(2.84)	0.08(1.53)	10.98(0.93)
megafone100K-f35	2.36(0.51)	2.43(0.23)	11.12(1.65)	1.77(0.44)	1.49(0.32)	1.66(0.34)	**15.44(0.57)**	2.21(0.24)	15.35(0.46)	2.59(0.19)	2.84(0.86)	2.57(0.75)	1.75(1.24)	8.41(0.41)
megafone100K-f35'	2.64(0.26)	2.62(0.19)	7.53(3.44)	1.21(0.67)	1.67(0.26)	1.57(0.27)	**15.6(0.25)**	2.18(0.21)	15.36(0.19)	2.57(0.22)	2.02(0.7)	2.48(1.0)	0.87(0.58)	7.38(0.6)
megafone100K-f16	2.24(0.36)	2.21(0.29)	5.58(4.57)	2.4(0.23)	0.75(0.5)	0.76(0.46)	**14.66(1.41)**	2.18(0.45)	14.6(1.15)	2.54(0.33)	1.85(0.84)	2.7(0.63)	1.18(0.8)	8.74(1.36)
megafone100K-f16'	2.64(0.2)	2.62(0.2)	11.81(0.9)	2.18(0.13)	2.13(0.37)	2.33(0.28)	**15.3(0.53)**	2.61(0.16)	15.04(0.3)	2.86(0.17)	2.34(0.65)	2.2(0.85)	1.5(0.79)	8.68(0.5)
Bank3-duration	-3.45(3.84)	4.86(6.14)	-13.49(5.09)	-8.92(3.61)	5.71(8.14)	-11.63(2.33)	-27.47(4.41)	**14.54(3.08)**	-21.38(4.59)	-6.56(6.02)	-7.87(5.22)	-9.68(14.3)	-13.2(13.59)	6.39(5.59)
Bank3-duration'	-10.74(2.52)	-2.1(1.33)	-22.36(4.2)	-19.22(2.88)	-1.92(1.68)	-6.97(3.46)	-28.34(9.41)	**7.47(0.83)**	-30.6(1.95)	-11.37(1.2)	-12.75(10.78)	-4.64(3.56)	-12.99(10.93)	-5.29(3.65)
Bank3-month	-5.64(13.78)	1.49(2.56)	-25.64(9.16)	-16.51(4.57)	**6.23(1.76)**	-0.81(6.64)	-31.11(8.84)	-0.71(1.99)	-23.21(8.12)	-17.67(9.0)	-10.0(10.66)	-4.59(6.94)	-20.0(18.47)	-2.64(5.73)
Bank3-month'	-1.89(4.27)	-6.98(6.17)	-17.42(3.99)	-14.49(1.38)	6.72(4.73)	-4.01(9.69)	-22.6(6.17)	**11.71(0.78)**	-21.83(4.47)	-11.92(1.74)	-16.26(11.81)	-4.17(9.21)	-12.15(14.25)	1.35(4.26)
Information-N-OPEN-REV-ACTS	8.17(0.85)	8.23(0.58)	7.69(0.72)	2.33(1.64)	3.65(0.25)	1.93(1.22)	**12.03(1.11)**	8.49(0.57)	10.22(0.45)	3.31(0.71)	9.36(0.82)	9.43(0.39)	8.33(1.18)	11.39(0.59)
Information-N-OPEN-REV-ACTS'	7.26(0.87)	7.25(0.54)	6.46(1.0)	2.23(2.27)	3.25(0.37)	0.77(1.39)	**11.65(0.96)**	7.95(0.8)	10.41(0.53)	2.64(1.2)	6.89(1.95)	7.12(1.6)	3.93(3.0)	10.17(0.94)
Information-PREM-BANKCARD-CRED-LMT	7.06(2.82)	7.03(2.39)	6.46(1.07)	-0.05(2.24)	3.29(1.25)	0.79(1.21)	10.39(1.54)	6.83(1.38)	10.67(0.85)	3.23(1.22)	7.64(1.95)	8.0(1.45)	5.69(2.21)	**11.2(1.05)**
Information-PREM-BANKCARD-CRED-LMT'	8.36(0.89)	8.64(0.58)	5.05(1.43)	3.75(0.2)	3.46(0.19)	1.28(0.78)	**11.42(0.76)**	8.7(0.29)	10.26(0.94)	3.86(0.5)	6.72(1.31)	6.97(1.46)	6.37(1.18)	9.84(1.11)
Starbucks-V4	13.4(7.33)	15.14(3.17)	-10.3(4.15)	-8.4(5.17)	13.04(6.93)	15.2(3.15)	**17.87(1.12)**	14.39(3.62)	12.17(2.31)	14.54(4.12)	10.37(1.48)	11.25(2.21)	6.65(2.02)	2.69(3.91)
Starbucks-V4'	13.4(7.33)	15.14(3.17)	-10.3(4.15)	-8.4(5.17)	13.04(6.93)	15.2(3.15)	**17.87(1.12)**	14.39(3.62)	12.17(2.31)	14.54(4.12)	10.37(1.48)	11.25(2.21)	6.65(2.02)	2.69(3.91)
Starbucks-V5	15.46(0.66)	15.01(0.82)	10.96(2.62)	10.43(1.67)	14.21(2.47)	14.5(1.91)	**16.5(2.64)**	15.87(0.85)	15.02(2.22)	16.21(0.75)	13.87(2.22)	13.89(2.5)	14.61(2.65)	1.67(3.24)
Starbucks-V5'	15.67(0.82)	15.65(0.79)	-6.7(3.56)	8.81(2.01)	14.93(0.48)	15.24(0.5)	**17.82(2.28)**	16.18(0.43)	13.86(1.64)	16.09(0.45)	13.29(0.94)	13.22(2.17)	11.32(1.09)	11.81(2.19)
Gerber-cluster	**1.32(0.22)**	1.23(0.24)	-3.71(0.19)	-0.05(0.17)	0.45(0.4)	0.27(0.34)	-2.08(0.39)	1.29(0.18)	-2.43(0.25)	-0.6(0.4)	-2.3(0.54)	-2.1(0.79)	-1.4(0.55)	-1.83(0.26)
Gerber-cluster'	1.49(0.12)	1.52(0.14)	-3.88(0.18)	0.08(0.02)	0.6(0.3)	0.56(0.39)	-2.24(0.44)	**1.61(0.18)**	-1.99(0.23)	-0.27(0.4)	-0.28(0.57)	-0.13(0.63)	0.07(0.55)	-1.77(0.24)
Gerber-yob	1.49(0.13)	**1.5(0.08)**	-3.81(0.25)	0.02(0.01)	0.62(0.34)	0.51(0.29)	-1.88(0.33)	1.45(0.1)	-2.16(0.21)	-0.09(0.47)	-1.26(0.77)	-0.78(0.37)	-0.62(0.35)	-1.64(0.3)
Gerber-yob'	1.23(0.42)	1.38(0.16)	-3.57(0.39)	0.02(0.13)	0.17(0.55)	0.12(0.34)	-2.47(0.33)	**1.52(0.12)**	-2.23(0.21)	-1.31(0.54)	-1.54(0.43)	-1.39(0.35)	-0.79(0.46)	-1.55(0.3)
Retail-express-spent-mean	0.65(0.32)	0.79(0.21)	0.77(0.25)	0.98(0.33)	0.79(0.18)	0.74(0.35)	1.21(0.43)	0.72(0.18)	0.68(0.21)	0.59(0.12)	1.26(0.16)	1.26(0.15)	**1.27(0.18)**	1.05(0.37)
Retail-express-spent-mean'	0.91(0.04)	0.92(0.1)	0.84(0.45)	0.92(0.11)	0.87(0.07)	0.41(0.66)	0.96(0.5)	0.89(0.09)	0.56(0.57)	0.54(0.15)	0.83(0.33)	0.86(0.27)	0.67(0.47)	**1.21(0.17)**
Retail-first-redeem-date	0.59(0.04)	0.89(0.07)	0.44(0.4)	0.82(0.45)	0.57(0.05)	0.58(0.33)	0.41(0.22)	0.84(0.03)	-0.31(0.19)	0.8(0.07)	**1.65(0.12)**	1.64(0.11)	1.63(0.18)	1.07(0.18)
Retail-first-redeem-date'	0.63(0.06)	0.89(0.05)	0.79(0.3)	1.2(0.24)	0.63(0.1)	0.91(0.18)	0.78(0.24)	0.97(0.04)	0.15(0.15)	0.82(0.02)	**1.37(0.28)**	1.34(0.32)	1.28(0.34)	0.91(0.17)

Table 5.2: Average qini coefficients and standard deviation. In bold, the best value for each dataset

Dataset	Ref.qini UB-RF	UB-RF	Ref.qini XLearner-Xgboost	XLearner-Xgboost	Ref.qini XLearner-LR	XLearner-LR	Ref.qini RLearner-LR	RLearner-LR	Ref.qini RLearner-Xgboost	RLearner-Xgboost
zenodo-trial0-x32	10.39(3.44)	10.04(0.84)	11.26(2.74)	**11.28(0.52)**	9.13(3.16)	9.31(0.28)	10.33(3.34)	9.32(0.52)	9.79(2.77)	9.78(0.31)
zenodo-trial0-x32'	10.59(3.04)	9.86(0.62)	12.42(4.1)	**11.59(0.67)**	10.18(3.18)	9.73(0.61)	10.53(2.59)	10.32(0.26)	10.16(3.81)	9.38(0.86)
zenodo-trial0-x34	10.92(3.35)	10.2(0.81)	12.12(3.64)	**11.59(0.61)**	9.64(3.55)	9.18(0.47)	9.21(3.02)	8.74(0.87)	9.91(3.11)	9.32(0.32)
zenodo-trial0-x34'	10.89(2.92)	10.52(1.08)	11.5(3.54)	11.15(1.05)	9.74(3.09)	9.61(0.29)	10.41(3.13)	10.04(0.47)	10.2(4.04)	9.4(0.41)
GridPattern-Comb2	21.69(0.03)	**21.7(0.02)**	21.71(0.04)	21.54(0.41)	18.73(0.01)	18.72(0.01)	18.72(0.0)	18.69(0.05)	18.72(0.0)	18.65(0.11)
GridPattern-Comb2'	21.71(0.01)	20.78(2.95)	21.66(0.01)	**21.52(0.41)**	18.73(0.0)	18.72(0.01)	18.72(0.0)	18.54(0.21)	18.73(0.0)	18.58(0.24)
ContPattern-Comb2	30.47(0.01)	30.08(1.18)	30.68(0.0)	30.51(0.38)	30.92(0.0)	30.92(0.0)	30.92(0.0)	30.85(0.11)	30.92(0.0)	**30.92(0.0)**
ContPattern-Comb2'	30.44(0.01)	30.13(0.99)	30.71(0.0)	30.62(0.17)	30.92(0.0)	30.92(0.0)	30.92(0.0)	30.91(0.0)	30.92(0.0)	**30.92(0.0)**
hillstrom-w-mens	6.15(2.6)	3.59(2.78)	4.3(2.25)	3.79(0.91)	5.87(2.16)	4.8(2.17)	6.15(1.41)	5.68(1.39)	5.98(1.31)	4.95(3.77)
hillstrom-w-mens'	6.71(2.24)	5.54(1.55)	4.21(2.63)	3.66(0.56)	5.95(1.56)	6.01(0.18)	5.73(1.95)	5.68(1.39)	5.86(1.89)	6.16(0.21)
hillstrom-w-womens	6.3(2.37)	4.54(2.26)	3.74(2.59)	3.38(1.23)	5.92(2.19)	5.47(1.65)	6.54(1.23)	**5.95(0.87)**	6.51(1.62)	4.97(3.75)
hillstrom-w-womens'	6.83(2.12)	5.07(1.92)	4.95(2.95)	4.08(1.29)	6.1(1.7)	5.45(2.4)	6.54(1.23)	**5.95(0.87)**	6.51(1.62)	4.97(3.75)
hillstrom-m-mens	1.64(2.25)	0.59(1.17)	-0.23(2.5)	0.2(0.62)	1.18(2.27)	0.61(0.81)	0.94(2.02)	0.66(0.75)	1.3(2.19)	**0.8(0.67)**
hillstrom-m-mens'	1.83(2.21)	**1.86(0.4)**	1.44(2.15)	0.63(0.53)	1.32(3.23)	1.82(0.52)	0.95(2.86)	1.67(0.59)	1.01(2.86)	1.64(0.7)
hillstrom-m-womens	1.43(2.43)	**1.77(0.4)**	0.24(1.23)	-0.07(0.38)	1.65(2.78)	1.22(0.53)	1.06(2.43)	1.01(0.44)	1.02(2.35)	0.85(0.67)
hillstrom-m-womens'	1.52(2.58)	**1.73(0.42)**	0.24(1.23)	-0.07(0.38)	1.65(2.78)	1.22(0.53)	1.06(2.43)	1.01(0.44)	1.02(2.35)	0.85(0.67)
Criteo50K-f2	12.05(1.13)	**11.7(0.58)**	9.89(1.48)	9.62(1.28)	11.23(0.63)	8.59(4.7)	10.95(1.18)	10.44(2.09)	11.0(1.24)	10.8(1.19)
Criteo50K-f2'	11.09(0.74)	11.19(0.22)	7.25(1.59)	7.76(0.58)	11.39(0.75)	10.74(0.36)	11.32(0.94)	10.71(0.41)	10.78(1.13)	10.64(0.34)
Criteo50K-f8	10.78(1.09)	11.37(0.34)	10.29(2.14)	9.96(1.15)	11.48(0.66)	8.44(5.07)	11.49(1.43)	10.62(2.51)	11.52(1.41)	10.91(1.81)
Criteo50K-f8'	11.5(0.98)	10.98(0.93)	10.18(1.96)	9.2(1.65)	10.94(0.87)	9.0(4.47)	11.26(0.5)	10.77(0.95)	11.45(0.49)	10.87(0.79)
megafone100K-f35	7.81(1.1)	8.41(0.41)	15.79(1.19)	**15.44(0.57)**	2.04(1.31)	2.21(0.24)	2.52(0.75)	2.43(0.23)	2.51(0.83)	2.36(0.51)
megafone100K-f35'	8.06(1.07)	7.38(0.6)	16.04(1.63)	**15.6(0.25)**	1.89(1.27)	2.18(0.21)	2.83(1.15)	2.62(0.19)	2.94(1.09)	2.64(0.26)
megafone100K-f16	10.44(2.37)	8.74(1.36)	15.26(1.29)	**14.66(1.41)**	2.58(1.58)	2.18(0.45)	2.47(1.08)	2.21(0.29)	2.41(1.08)	2.24(0.36)
megafone100K-f16'	7.78(1.18)	8.68(0.5)	15.69(1.1)	**15.3(0.53)**	2.37(1.12)	2.61(0.16)	2.76(1.06)	2.62(0.2)	2.75(1.33)	2.64(0.2)
Bank3-duration	9.25(25.69)	6.39(5.59)	-30.93(32.17)	-27.47(4.41)	11.32(14.6)	**14.54(3.08)**	-0.13(41.12)	4.86(6.14)	-6.06(27.94)	-3.45(3.84)
Bank3-duration'	-0.75(33.94)	-5.29(3.65)	-24.36(31.62)	-28.34(9.41)	8.75(16.74)	7.47(0.83)	-2.95(51.76)	-2.1(1.33)	-6.12(33.23)	-10.74(2.52)
Bank3-month	2.94(23.84)	-2.64(5.73)	-27.27(25.41)	-31.11(8.84)	0.15(16.71)	-0.71(1.99)	-1.8(38.59)	1.49(2.56)	-8.53(22.37)	-5.64(13.78)
Bank3-month'	2.54(22.09)	1.35(4.26)	-20.92(30.76)	-22.6(6.17)	12.32(14.5)	11.71(0.78)	2.02(38.51)	-6.98(6.17)	-7.64(29.51)	-1.89(4.27)
Information-N-OPEN-REV-ACTS	12.18(4.08)	11.39(0.59)	12.11(3.94)	**12.03(1.11)**	8.75(3.48)	8.49(0.57)	8.51(2.75)	8.23(0.58)	8.41(2.81)	8.17(0.85)
Information-N-OPEN-REV-ACTS'	11.32(3.09)	10.17(0.94)	11.99(4.1)	**11.65(0.96)**	9.12(2.88)	7.95(0.8)	8.16(2.85)	7.25(0.54)	8.56(2.57)	7.26(0.87)
Information-PREM-BANKCARD-CRED-LMT	10.68(4.19)	**11.2(1.05)**	12.07(3.59)	10.39(1.54)	7.41(2.58)	6.83(1.38)	8.26(2.07)	7.03(2.39)	8.85(2.96)	7.06(2.82)
Information-PREM-BANKCARD-CRED-LMT'	11.0(3.5)	9.84(1.11)	12.31(4.7)	**11.42(0.76)**	8.99(2.62)	8.7(0.29)	9.78(2.72)	8.64(0.58)	8.89(3.09)	8.36(0.89)
Starbucks-V4	11.09(5.37)	2.69(3.91)	17.1(6.31)	**17.87(1.12)**	15.04(3.83)	14.39(3.62)	16.25(6.35)	15.14(3.17)	15.95(6.25)	13.4(7.33)
Starbucks-V4'	11.09(5.37)	2.69(3.91)	17.1(6.31)	**17.87(1.12)**	15.04(3.83)	14.39(3.62)	16.25(6.35)	15.14(3.17)	15.95(6.25)	13.4(7.33)
Starbucks-V5	9.43(5.95)	1.67(3.24)	19.94(7.26)	**16.5(2.64)**	16.52(5.04)	15.87(0.85)	15.63(5.14)	15.01(0.82)	15.48(4.81)	15.46(0.66)
Starbucks-V5'	5.13(5.26)	11.81(2.19)	18.0(6.11)	**17.82(2.28)**	16.01(4.95)	16.18(0.43)	16.12(5.19)	15.65(0.79)	16.14(5.13)	15.67(0.82)
Gerber-cluster	-1.82(2.56)	-1.83(0.26)	-1.32(2.61)	-2.08(0.39)	1.42(1.22)	1.29(0.18)	1.26(1.35)	1.23(0.24)	1.32(1.34)	**1.32(0.22)**
Gerber-cluster'	-2.13(2.6)	-1.77(0.24)	-1.93(2.74)	-2.24(0.44)	1.39(0.63)	**1.61(0.18)**	1.3(1.02)	1.52(0.14)	1.33(0.92)	1.49(0.12)
Gerber-yob	-1.98(2.83)	-1.64(0.3)	-2.21(3.34)	-1.88(0.33)	1.4(1.08)	1.45(0.1)	1.48(1.47)	**1.5(0.08)**	1.45(1.39)	1.49(0.13)
Gerber-yob'	-1.17(2.87)	-1.55(0.3)	-2.16(3.53)	-2.47(0.33)	1.68(0.88)	**1.52(0.12)**	1.49(1.13)	1.38(0.16)	1.47(1.01)	1.23(0.42)
Retail-express-spent-mean	1.12(0.55)	1.05(0.37)	1.67(0.69)	**1.21(0.43)**	0.92(0.46)	0.72(0.18)	0.89(0.43)	0.79(0.21)	0.85(0.37)	0.65(0.32)
Retail-express-spent-mean'	1.41(0.51)	**1.21(0.17)**	1.18(0.9)	0.96(0.5)	0.92(0.45)	0.89(0.09)	0.96(0.54)	0.92(0.1)	0.9(0.44)	0.91(0.04)
Retail-first-redeem-date	1.09(0.39)	**1.07(0.18)**	0.49(1.16)	0.41(0.22)	0.84(0.38)	0.84(0.03)	0.96(0.51)	0.89(0.07)	0.62(0.33)	0.59(0.04)
Retail-first-redeem-date'	1.28(0.43)	0.91(0.17)	0.8(0.97)	0.78(0.24)	0.92(0.54)	**0.97(0.04)**	0.84(0.44)	0.89(0.05)	0.63(0.33)	0.63(0.06)

Table 5.3: Reference qini coefficient and average qini coefficients for the best approaches

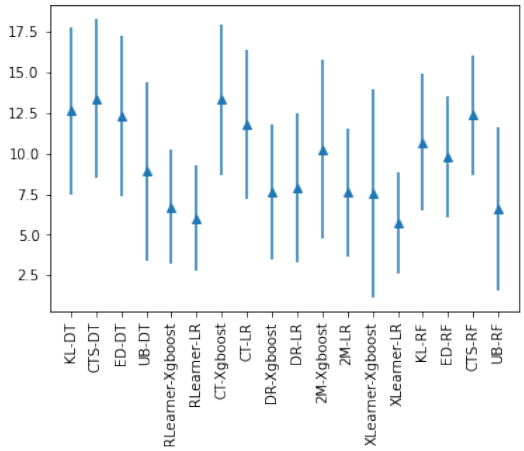


Figure 5.3: Overall ranking for the different uplift approaches.

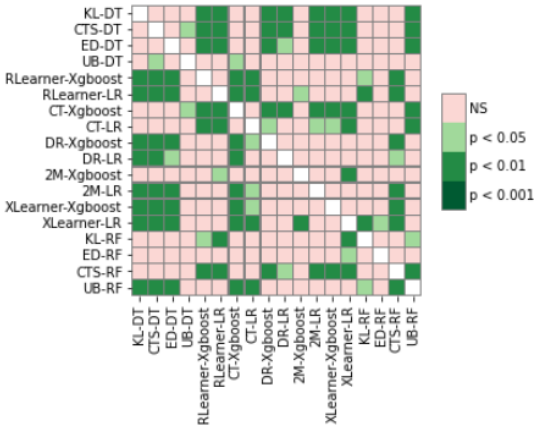


Figure 5.4: Heat map to visualize the comparison between uplift methods. A value of p smaller than 0.05 means that the null hypothesis is rejected.

proposed UB-RF method. The qini coefficient generally decreases only when the bias rate is increased. Table 5.3 shows the reference qini and the average qini coefficients for each of these methods. The reference qini coefficient (denoted Ref.qini) is the qini coefficient when $b = 0$

- While traditional tree-based methods show limited effectiveness, their random forest counterparts show improved performance, especially the ED-RF model.

- In contrast to other tree-based techniques, our UB-DT model shows competitive performance compared to meta-learners and the UB-RF method. Although it does not perform better - which is an expected behaviour for a simple decision tree - the effectiveness of the UB-DT model is remarkable. In fact, it even has a higher average rank than state-of-the-art random forest algorithms.

- The class-transformation approach is particularly sensitive to the bias introduced, as its qini coefficient performance deteriorates significantly even in the presence of minimal bias.

Methods comparison with statistical tests

We study now the significance of the results regarding the comparison of the uplift methods (cf. Table 5.2) by using a statistical test. Following the study [17],

we choose the Friedman test with the post hoc test of Nemenyi to compare the performance (average qini coefficient) of more than two methods across several datasets. Fig. 5.4 depicts the results with a heatmap. The null hypothesis states that there is no significant difference in performance according to the average qini coefficients between two methods across the datasets. With a value of p (p-value) smaller than 0.05, the null hypothesis is rejected (in green in Fig. 5.4). Figures 5.4 and 5.3 confirm our previous conclusions. They show that the class-transformation and tree-based approaches are the least resistant to NRA bias, and their performance differs significantly from the most resistant methods. The methods that show the most resistance to NRA bias include our method UB-RF, the R-learner, X-learner and DR-learner. Notably, there are *no significant* differences between these top performers in terms of their resistance to NRA bias.

5.3 Method to reduce the NRA bias impact

5.3.1 Method Description

In the previous section, our experimental protocol showed that the class-transformation approach is highly sensitive to the NRA bias. In this section, we examine the impact of reweighting individuals using propensity scores on the resilience of the class transformation approach to the NRA bias. Propensity scores represent the probability of an individual receiving a treatment ($T = 1$) based on their vector of observed variables X_i, i.e. $P(T = 1|X_i)$. This strategy is inspired by the literature on domain adaptation, where samples from a source dataset are weighted according to their relevance to a target dataset [40].

The principle of the method is to weight individuals in the treatment group based on their corresponding weights in the control group, thereby making the biased population (treatment group) more similar to the unbiased population (control group). In observational studies, the propensity scores are not directly known, but can be estimated from the data using a regression algorithm. This method weights each individual i of the treatment group by $w(X_i)$ s.t.:

$$w(X_i) = P(T = 0|X_i)/P(T = 1|X_i) \tag{5.1}$$

We estimate the probabilities of Eq. 5.1 by using logistic regression and xgboost. Then the uplift method integrates the weights to amplify the role of the under-represented individuals in the treatment group and estimate $\hat{\tau}_i$. We named wt-1 (resp. wt-2) the use of the logistic regression (resp. xgboost) in the weighting method.

We evaluate our weighting method with the class-transformation approaches since they are the most affected approaches by the NRA bias according to our

experiments and they use traditional machine learning algorithms where weights can be given directly at each line (individual).

5.3.2 Results

Results show a large enhancement in the performance with the class-transformation methods (cf. Fig. 5.5). Table 5.4 details the results with the class-transformation based methods. The same as before, "*Ref.qini*" denotes the *reference qini*, that is the qini value of a method without bias (i.e. $b = 0$) and without weighting. The Mean Absolute Error (MAE), given by the formula $MAE = \frac{1}{n}\sum_{j=1}^{n} |Ref.qini_j - Averageqini_j|$, indicates the difference between the qini obtained by an uplift method and the reference qini. Here n is the total number of experiments performed. The smaller the gap is, the better the weighting.

With the reweighting method, the gap is much smaller especially when the class transformation is combined with the logistic regression (LR). The best average qini values are also obtained with weighting, except for the Zenodo, Bank and Gerber datasets. It is also worth noting that on the Bank and Gerber datasets, the class transformation approach shows poor performance (as indicated by the negative $Ref.qini$ values) even when no bias is introduced ($b = 0$). In these cases, the reweighting method does not improve much or at all the class transformation approach, specially when the dataset is already hard to learn an uplift model.

5.4 Conclusion

In this chapter we define the non-random assignment bias (NRA) in the context of uplift modeling. NRA bias occurs when samples are not equally distributed between the treatment and control groups, i.e. when $P(T = 1) \neq P(T = 0)$. This bias is often observed in real world data. To explore the impact of NRA bias, we have designed an experimental protocol that simulates it in a dataset. The goal of the protocol is to generate NRA bias and study its effects on state-of-the-art uplift methods, as well as our newly proposed decision tree and random forest approaches, UB-DT and UB-RF.

Our experiments revealed different behaviours between the uplift methods. Our novel random forest approach, UB-RF, showed resilience to NRA bias. Several meta-learners, such as the X-learner, R-learner and DR-learner, also maintained strong performance when confronted with NRA bias. Also, our proposed decision tree method UB-DT, showed to have a good performance compared to the state-of-the art decision tree and random forest approaches. In contrast, the class transformation approach proved to be the least resilient to NRA bias.

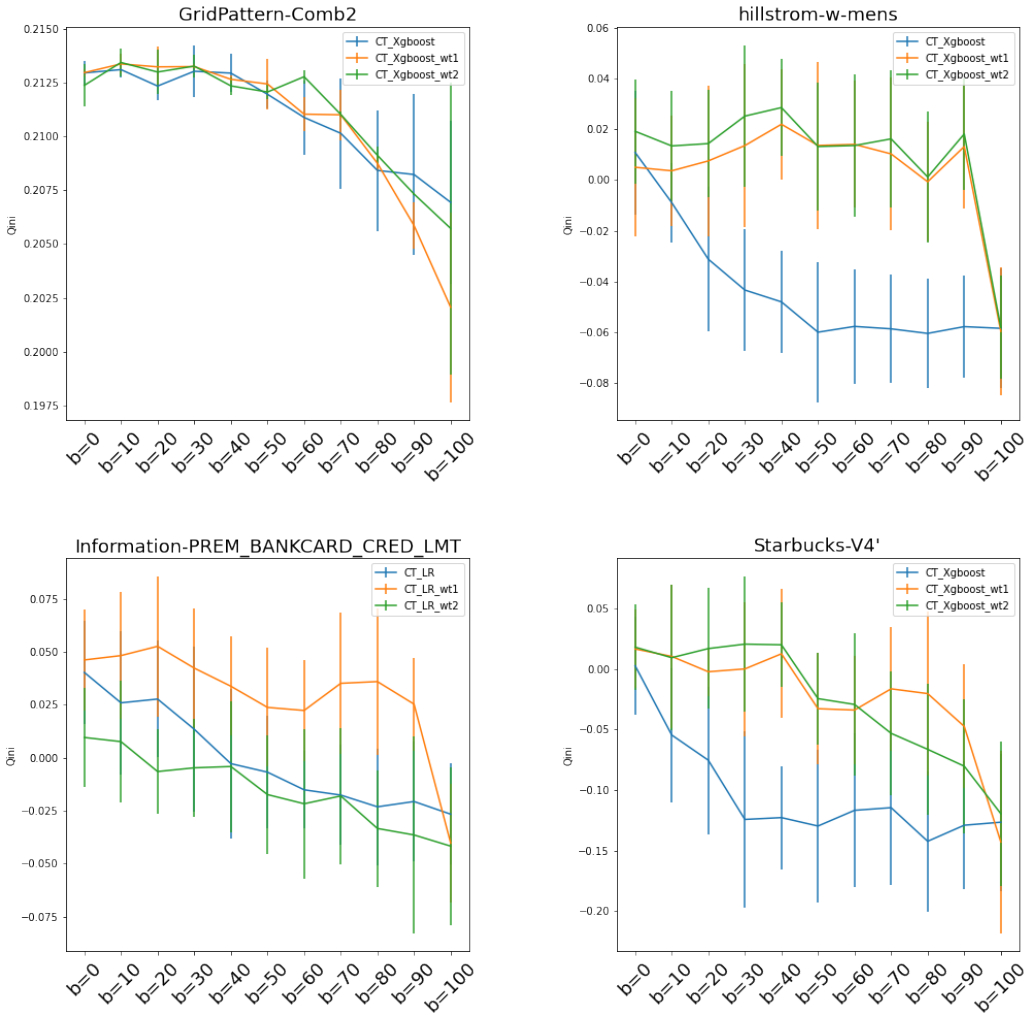


Figure 5.5: The qini values obtained through class-transformation with Xgboost for varying levels of NRA bias rates, both with and without reweighting. Each figure's title is composed of the dataset's name, followed by the specific variable utilized to generate the bias.

Dataset	Class-Transformation with LR				Class-Transformation with Xgboost			
	Ref.qini	w/o weights	wt_1	wt_2	Ref.qini	w/o weights	wt_1	wt_2
zenodo-trial0-x32	8.27(3.23)	**9.65(0.59)**	9.39(0.44)	7.12(0.8)	5.81(2.08)	**7.06(1.14)**	7.03(0.93)	5.84(1.11)
zenodo-trial0-x32'	8.15(2.33)	**9.7(0.48)**	9.15(0.96)	7.2(0.97)	5.81(2.08)	7.51(0.98)	**7.51(0.73)**	5.84(1.11)
zenodo-trial0-x34	7.85(2.31)	8.99(0.75)	**9.66(0.31)**	8.1(0.37)	6.45(3.98)	**7.9(1.22)**	7.57(1.36)	6.97(1.06)
zenodo-trial0-x34'	7.25(1.94)	**10.21(0.21)**	8.79(2.62)	6.29(0.77)	5.12(3.36)	**8.41(0.85)**	7.57(1.36)	4.47(1.36)
GridPattern-Comb2	18.73(0.0)	18.69(0.04)	18.69(0.03)	**18.72(0.03)**	21.24(0.1)	21.1(0.21)	21.06(0.35)	**21.11(0.25)**
GridPattern-Comb2'	18.73(0.0)	18.71(0.02)	**18.71(0.04)**	18.7(0.09)	21.25(0.02)	20.96(0.38)	**21.09(0.26)**	21.08(0.29)
ContPattern-Comb2	30.92(0.0)	30.92(0.0)	**30.92(0.0)**	30.92(0.0)	30.35(0.0)	**30.35(0.02)**	30.32(0.06)	30.23(0.19)
ContPattern-Comb2'	30.92(0.0)	30.92(0.0)	30.91(0.0)	**30.92(0.0)**	30.35(0.0)	**30.35(0.03)**	30.32(0.02)	30.23(0.17)
hillstrom-w-mens	6.07(2.11)	-4.33(3.78)	**4.74(3.55)**	4.36(3.56)	1.93(2.05)	-4.31(2.29)	0.39(2.1)	**0.96(2.25)**
hillstrom-w-mens'	6.29(2.61)	6.21(0.19)	6.15(0.21)	**6.23(0.29)**	1.49(1.64)	**4.62(1.29)**	2.05(1.25)	2.61(1.07)
hillstrom-w-womens	6.02(1.72)	-5.43(3.65)	**4.7(3.38)**	4.1(3.55)	2.89(2.53)	-4.78(2.54)	0.31(2.24)	**1.6(2.56)**
hillstrom-w-womens'	6.02(1.72)	-5.03(3.72)	**4.7(3.38)**	4.1(3.55)	2.89(2.53)	-4.56(2.63)	0.31(2.24)	**1.6(2.56)**
hillstrom-m-mens	1.19(1.64)	-0.0(0.71)	0.89(0.67)	**0.99(0.74)**	-0.45(2.99)	0.2(0.35)	0.07(0.31)	**0.22(0.45)**
hillstrom-m-mens'	1.05(2.71)	-0.05(0.6)	**1.73(0.93)**	1.56(0.85)	0.28(2.42)	-0.65(0.28)	**-0.19(0.67)**	-0.25(0.73)
hillstrom-m-womens	1.51(2.56)	-0.16(0.55)	1.3(0.82)	**1.33(0.91)**	-0.51(2.57)	-1.48(0.53)	**-0.83(0.72)**	-1.03(0.54)
hillstrom-m-womens'	1.51(2.56)	-0.16(0.55)	1.3(0.82)	**1.33(0.91)**	-0.51(2.57)	-1.48(0.53)	**-0.83(0.72)**	-1.03(0.54)
Criteo50K-f2	10.33(1.86)	2.01(7.11)	6.33(7.24)	**8.26(6.97)**	6.48(2.22)	0.09(6.61)	2.75(5.97)	**5.27(5.39)**
Criteo50K-f2'	8.91(2.11)	7.07(1.9)	**7.54(3.53)**	7.31(4.16)	5.97(2.46)	**7.27(1.07)**	5.33(1.11)	4.21(1.38)
Criteo50K-f8	10.35(1.52)	1.57(7.56)	5.58(7.93)	**7.98(7.36)**	6.26(2.32)	-0.35(6.92)	3.05(6.21)	**5.27(5.76)**
Criteo50K-f8'	9.87(1.6)	-1.9(6.06)	6.27(7.31)	**7.44(6.42)**	4.32(2.43)	-2.47(5.1)	0.94(5.53)	**2.69(4.84)**
megafone100K-f35	2.04(1.27)	1.77(0.44)	1.91(0.6)	**2.05(0.25)**	13.35(0.81)	11.12(1.65)	11.07(1.75)	**12.41(1.43)**
megafone100K-f35'	2.34(1.23)	1.21(0.67)	**2.21(0.24)**	2.08(0.56)	13.36(1.62)	7.53(3.44)	7.63(3.59)	**11.6(2.75)**
megafone100K-f16	2.52(1.55)	**2.4(0.23)**	2.23(0.32)	2.33(0.33)	13.1(0.99)	5.58(4.57)	5.53(4.52)	**10.44(3.43)**
megafone100K-f16'	2.35(1.0)	2.18(0.13)	**2.64(0.21)**	2.2(0.25)	12.7(1.24)	11.81(0.9)	11.79(0.9)	**12.43(0.82)**
Bank3-duration	-15.86(15.84)	**-8.92(3.61)**	-40.62(20.65)	-24.42(9.24)	-60.72(61.04)	**-13.49(5.09)**	-22.48(6.23)	-48.71(11.48)
Bank3-duration'	-31.91(58.88)	-19.22(2.88)	-31.59(20.92)	**-17.64(6.13)**	-46.25(36.52)	**-22.36(4.2)**	-30.34(3.35)	-46.31(5.45)
Bank3-month	-19.5(11.11)	-16.51(4.57)	-58.88(15.59)	**-16.23(4.8)**	-48.47(25.77)	**-25.64(9.16)**	-31.7(6.11)	-51.89(10.59)
Bank3-month'	-11.88(17.11)	**-14.49(1.38)**	-40.14(12.24)	-18.8(5.58)	-54.38(43.57)	**-17.42(3.99)**	-26.54(3.27)	-44.99(6.93)
Information-N-OPEN-REV-ACTS	0.31(2.09)	2.33(1.64)	**4.16(0.61)**	-0.2(1.02)	8.26(3.16)	7.69(0.72)	**8.05(0.73)**	7.95(0.86)
Information-N-OPEN-REV-ACTS'	2.04(2.92)	2.23(2.27)	**5.06(0.59)**	0.23(1.69)	7.26(3.34)	6.46(1.0)	6.75(0.63)	**7.18(0.74)**
Information-PREM-BANKCARD-CRED-LMT	0.96(2.33)	-0.05(2.24)	**2.96(2.41)**	-1.51(1.65)	5.71(2.27)	**6.46(1.07)**	5.9(1.0)	6.3(0.9)
Information-PREM-BANKCARD-CRED-LMT'	0.65(2.45)	3.75(0.2)	**4.12(0.37)**	0.71(0.69)	7.77(3.79)	5.05(1.43)	5.63(1.43)	**7.11(1.15)**
Starbucks-V4	10.73(4.72)	-8.4(5.17)	**10.75(7.58)**	-4.14(7.05)	1.79(3.52)	-10.3(4.15)	**-2.34(4.27)**	-2.63(4.6)
Starbucks-V4'	10.73(4.72)	-8.4(5.17)	**10.75(7.58)**	-4.14(7.05)	1.79(3.52)	-10.3(4.15)	**-2.34(4.27)**	-2.63(4.6)
Starbucks-V5	15.22(4.33)	10.43(1.67)	**15.12(1.48)**	11.18(3.51)	4.17(6.69)	**10.96(2.62)**	10.69(2.06)	6.31(1.88)
Starbucks-V5'	14.72(5.44)	8.81(2.01)	**16.63(0.61)**	11.46(3.91)	3.4(5.22)	-6.7(3.56)	-7.12(2.95)	**-0.39(3.98)**
Gerber-cluster	-0.4(1.72)	**-0.05(0.17)**	-0.78(0.3)	-0.64(0.27)	-3.37(3.29)	-3.71(0.19)	**-3.42(0.3)**	-3.72(0.24)
Gerber-cluster'	-0.7(2.22)	**0.08(0.02)**	-0.72(0.39)	-0.29(0.38)	-3.73(3.45)	-3.88(0.18)	**-2.89(0.29)**	-3.81(0.16)
Gerber-yob	-1.18(1.61)	**0.02(0.01)**	-0.29(0.4)	-0.59(0.26)	-4.18(2.8)	-3.81(0.25)	**-2.84(0.47)**	-3.77(0.4)
Gerber-yob'	-0.9(1.98)	**0.02(0.13)**	-1.44(0.32)	-0.64(0.35)	-3.65(3.17)	-3.57(0.39)	**-3.2(0.38)**	-3.6(0.25)
Retail-express-spent-mean	1.32(0.8)	0.98(0.33)	0.61(0.46)	**1.0(0.5)**	0.73(0.88)	0.77(0.25)	**0.92(0.32)**	0.45(0.22)
Retail-express-spent-mean'	1.02(0.72)	0.92(0.11)	0.96(0.12)	**1.06(0.28)**	0.29(0.89)	**0.84(0.45)**	0.73(0.41)	0.37(0.3)
Retail-first-redeem-date	1.22(1.03)	0.82(0.45)	0.68(0.27)	**1.01(0.22)**	0.8(0.98)	0.44(0.4)	**0.51(0.38)**	-0.06(0.42)
Retail-first-redeem-date'	1.14(1.36)	1.2(0.24)	**1.24(0.27)**	1.1(0.29)	0.1(1.06)	0.79(0.3)	**0.79(0.2)**	-0.02(0.26)
MAE	0	0.036	0.03	0.021	0	0.05	0.03	0.014

Table 5.4: Average qini and its variance (shown in brackets) with the class-transformation based methods (in bold, the best value for each dataset). Dataset name is followed by the names of the V variables used to generate the NRA bias.

In the final section of this chapter, we propose to reweight the data using the inverse of the propensity scores when using the class transformation approach. This reweighting technique significantly improves the performance of the class transformation method in the presence of NRA bias.

CHAPTER 6

Application on Telecom Data

Contents

6.1 Introduction

Telecommunications data refers to the vast amount of information generated and collected by telecommunications companies through their network infrastructure, customer interactions, and billing systems. As discussed in Section 2.4, this data includes various dimensions, customer demographics, usage patterns, call records, service subscriptions, customer interactions, and billing information. Leveraging this wealth of telecom data can provide valuable insights into customer behavior and churn prediction, enabling telecom companies to proactively manage customer retention strategies.

That is why data scientists and decision makers are trying to get the most out of telecom data by pre-processing it and learning uplift models to predict uplift scores for each of their future customers. Learning uplift models may seem simple when using state-of-the-art uplift algorithms, but it is often very challenging because of the parameters that should be defined by the user. The main drawback of all the uplift approaches is that they require parameterization. Meta learners also have an additional requirement, which is the choice of the machine learning algorithm to be used. All this is a clear limitation for non-machine learning experts to use these tools. Even for machine learning experts, they need to test different parameter values and different learning algorithms with meta learners to find the optimal combination that fits the data at hand. Therefore, the parameter-free approaches proposed in this thesis are very much needed, especially in industrial contexts.

In this chapter, we first show how uplift modeling can be performed to deal with telecom data. Then we evaluate state-of-the-art uplift approaches for model learning and feature selection and compare them to our proposed approaches.

This work is the object of the following publication:
Rafla, M., Voisine, N., & Crémilleux, B. (2023, September). A Parameter-Free Bayesian Framework for Uplift Modeling - Application on Telecom Data. In Uplift Modeling and Causal Machine Learning for Operational Decision Making workshop, co-located with European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD).

6.2 Uplift methodology

In this section we provide important steps to follow when applying uplift modeling on telecom data:

- Outcome Definition. The outcome variable needs to be defined based on the marketing goal. For example, it could be whether a customer made a purchase or renewed their subscription within a specific time period after receiving the treatment. This outcome will be used to measure the uplift.
- Treatment Assignment. A group of customers needs to be exposed to a marketing treatment, while another group serves as a control group that does not receive the treatment. The treatment can be a specific marketing campaign, promotional offer, or any other intervention. Uplift is then modeled from the data of 2 groups.
- Data Preparation. The telecom data needs to be preprocessed and prepared for uplift modeling. This includes cleaning the data, handling missing values, and transforming variables into a suitable format. Data bias should also be addressed either at this step using propensity score matching or in the next step of model learning.
- Uplift model learning. The model predicts the uplift score, which represents the difference in the probability of a positive outcome between the treatment and control groups.
- Model Evaluation. The uplift model needs to be evaluated using appropriate metrics. These metrics help assess the model's performance in identifying customers who are most likely to respond positively to the marketing treatment.
- Targeting and Decision Making. Once the uplift model is built and evaluated, it can be used to target customers for the marketing intervention. The

model can identify customers who have a high likelihood of being positively influenced by the treatment. These customers can then be prioritized for the marketing campaign, maximizing the impact and return on investment.

- Iterative Refinement. Uplift modeling is an iterative process. The model's performance should be continuously monitored and refined based on the observed outcomes. This helps improve the targeting strategy and optimize the uplift achieved from the marketing interventions.

6.3 An experimental study on telecom data

In this section, we conduct different experiments to evaluate our Bayesian approaches on real data, specifically derived from Orange marketing campaigns conducted in 2013 (data specifications are shown in Table 6.1). The original Orange dataset contains 2700 variables. However, for the purpose of efficient computation, we have selected a subset of this dataset using Khiops software[1], keeping only 101 variables. We also added 50 noise variables to this dataset to better evaluate the feature selection approaches.

First, we compare our uplift Bayesian decision trees and random forests versus the state-of-art uplift modeling algorithms. We then evaluate the impact of UMODL discretization and feature selection as preprocessing steps on these state-of-the-art uplift modeling algorithms.

Size	P(Y=1|T=0)	P(Y=1|T=1)	P(T=1)	No. columns	No. continuous variables
20000	0.13	0.35	0.89	151	44

Table 6.1: Data specifications

6.3.1 UB-DT and UB-RF

In this part, we conduct a study to evaluate the performance of the UB-DT and UB-RF algorithms on our dataset. We also compare their performance with the state-of-art uplift modeling algorithms. More particularly, we evaluate the following algorithms: 1. X-learner 2. R-learner 3. DR-learner 4. Two model approach 5. Random forest algorithm based on the ED criterion 6. UB-RF 7. UB-DT Each of the X-learner, R-learner, DR-learner and the Two model approach was used with a random forest algorithm and a logistic regression as base models. All random forests are learnt using 10 trees. Each model is learnt using a 10-fold cross validation. We use the qini metric [18, 56] to evaluate the uplift models.

[1] https://www.khiops.com/

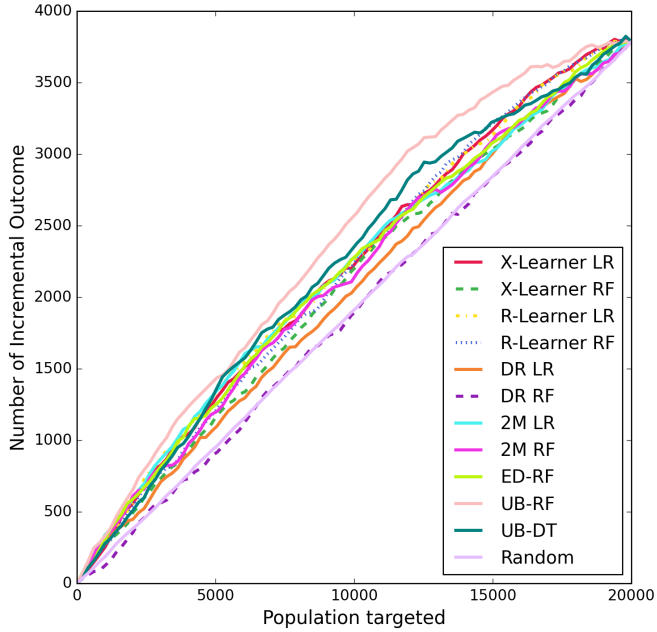


Figure 6.1: Qini curves: The x-axis denotes the number of individuals targeted, while the y-axis shows the number of incremental positive outcomes.

X-learner		R-learner		DR-learner		2M				
LR	RF	LR	RF	LR	RF	LR	RF	ED-RF	UB-RF	UB-DT
8.6(2.6)	5.6(3.8)	8.8(2.5)	8.0(2.5)	4.0(4.6)	-0.5(2.0)	7.9(2.5)	7.5(4.4)	7.8(2.3)	**14.4(5.3)**	10.2(3.6)

Table 6.2: Qini values multiplied by 100 and variance for each uplift model. In bold the biggest qini value.

Results Figure 6.1 shows the qini curves [18] for each model, aggregating data from all test folds. The corresponding qini values are given in Table 6.2. Looking at the results, we see that UB-RF (shown in pale pink color) outperforms all other uplift modeling algorithms. Even a single decision tree with the UB-DT approach shows remarkable efficiency and competes well with other methods.

6.3.2 Variable transformation

This section demonstrates the impact of employing a variable transformation with the UMODL discretization as an initial preprocessing step. When the UMODL discretization is applied to a non-informative variable, the result is a single interval, i.e. the transformed variable follows a uniform distribution for all examples. In such cases, UMODL discretization discards this variable. Consequently, the process of transforming variables can be seen as inherently involving a feature

selection step.

We carry out our experiments in three stages: the first with the original variables, the second with only a feature selection step, and the third with the variable transformation incorporated as a preprocessing step. As just mentioned, it should be noted that this third stage is considered to combine both feature selection and transformation. Again the model is built using a 10-fold cross-validation approach.

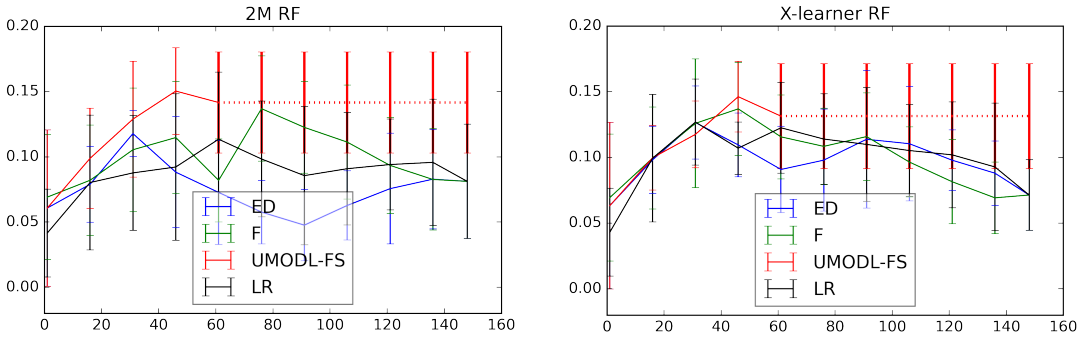
Results Table 6.3 presents the performance of uplift modeling algorithms under the three scenarios given above: 1. without any preprocessing step (i.e. original variables) 2. with UMODL-FS as a preprocessing step 3. with variable transformation as a preprocessing step. The results indicate that feature selection significantly improves the performance of all uplift modeling algorithms compared to the original dataset (without any preprocessing). Applying variable transformation yields similar improvements for all uplift modeling algorithms, with the exception of ED-RF. When comparing the impact of feature selection and variable transformation, the benefits appear to depend on the uplift approach used. For example, when logistic regression is used as the base learner, variable transformation appears to offer better improvements. In the contrary, when random forests are used as the base learner, feature selection shows to perform better. By performing feature selection and variable transformation, the state-of-the-art uplift models achieved the best results, close to those obtained with UB-RF (see Table 6.2).

	X-learner		R-learner		DR-learner		Two Model		
	LR	RF	LR	RF	LR	RF	LR	RF	ED-RF
w/o preprocessing	8.6(2.6)	5.6(3.8)	**8.8(2.5)**	8.0(2.5)	4.0(4.6)	-0.5(2.0)	7.9(2.5)	7.5(4.4)	7.8(2.3)
w/ feature selection	14.2 (3.3)	13.1(4.0)	14.1(3.1)	13.6(3.6)	12.3(5.2)	-0.06(1.0)	12.8(1.7)	14.1(3.8)	**14.3(4.2)**
w/ transformation	14.3(4.6)	11.6(2.7)	14.0(5.1)	13.6(5.0)	14.3(4.8)	1.7(7.5)	**14.8(5.5)**	10.0(3.1)	0.4(3.1)

Table 6.3: Qini values and variance (multiplied by 100) for each uplift model. In bold the biggest qini value among all the values.

6.3.3 Feature Selection

We evaluate the following feature selection methods: KL, LR filter, F filter and UMODL-FS. Each feature selection method gives a score to each variable in the data. The principle of the experiment is to feed the top-k features selected by each of the feature selection method into an uplift modeling algorithm and then observe the performance of the model. We use an incremental approach, first introducing only the top-1 feature. We then incrementally add an additional set of fifteen features at each step, continuing this iterative process until all features



(a) (b)

Figure 6.2: Uplift models with the top features

are integrated into the model. UMODL-FS allows us to continue selecting top features, as long as it gives them an importance score greater than zero. Once a score of zero is reached, no additional features are selected, thus establishing a threshold for feature selection.

Results Figure 6.2 shows the performance obtained with the two model approach and X-learner respectively. Each of these two models are used with a random forest of 10 trees. UMODL-FS shows a good performance selecting the top features leading to performant uplift models. In addition, UMODL-FS automatically determines the features to eliminate without user intervention. In contrast, the other feature selection methods cannot automatically determine an appropriate cut-off score for a feature to be discarded or considered.

6.4 Kuplift Library

This section gives a brief introduction to Kuplift [2] and how to use it. Kuplift is a Python library that implements our parameter-free Bayesian methods. It implements uplift Bayesian decision trees, random forests, discretization and feature selection. It provides a standard interface (like scikit-learn [41] and causalml [15]).

UB-DT and UB-RF

```
# Uplift Bayesian Decision Tree
from kuplift import BayesianDecisionTree
```

[2] https://github.com/UData-Orange/kuplift

```
T=BayesianDecisionTree(df_train,treatment_column,outcome_col)
T.fit()
preds=T.predict(df_test)
# To visualize a tree in a text form
print(T.export_tree())

# Uplift Bayesian Random Forest
from kuplift import BayesianRandomForest
T=BayesianRandomForest(df_train,treatment_column,outcome_col)
T.fit()
preds=T.predict(df_test)
```

UMODL discretization

```
from kuplift import UnivariateEncoding
ue = UnivariateEncoding()
ue.fit(df_train,treatment_column, outcome_col)
df_test=ue.transform(df_test)
```

UMODL-FS

```
from kuplift.FeatureSelection import FeatureSelection
fs = FeatureSelection()
important_vars = fs.filter(data, treatment_column, outcome_col)
```

6.5 Conclusion

In this chapter, we first discussed in Section 6.2 important steps required to perform uplift modeling on telecommunications data. These steps include defining the outcome, assigning the treatment, preparing the data, learning the uplift model, and finally evaluating the model. We highlight that the uplift modeling process is iterative and requires ongoing monitoring and refinement to achieve the greatest possible benefit from a marketing campaign.

Then in Section 6.3, we conducted a series of experiments using real telecom data from the French company Orange. First, we compared our proposed uplift modeling methods, UB-DT and UB-RF, with state-of-the-art approaches. The results show the efficiency of our methods. We then implemented variable transformation and feature selection as preprocessing steps for the traditional approaches to improve their performance. We explained how variable transformation inherently involves feature selection. Based on our experimental results, these two preprocessing steps significantly improved the effectiveness of the uplift algorithms.

We observed that variable transformation was particularly useful when the base learner was a logistic regression model. However, feature selection showed higher efficiency when the base learner was a random forest algorithm. In our final experiment, we highlighted the advantages of a parameter-free feature selection approach. This method automatically identifies the cutoff number of informative features in the data, eliminating the need for users to manually set this parameter.

Our conclusion suggests that it is advantageous to use parameter-free approaches when developing an uplift model. These techniques eliminate the need to select a specific base learner, set parameters for a specific algorithm, perform extensive preprocessing, or determine a cutoff number of features for a feature selection approach. This strategy simplifies the modeling process and reduces the potential for error.

Finally, in Section 6.4, we introduced Kuplift, our Python package which implements all the algorithms mentioned in this thesis.

CHAPTER 7

Conclusions and Perspectives

Contents

7.1 Conclusion

This thesis contributes to the field of uplift modeling. It was conducted in collaboration with the French telecommunications company *Orange*. In particular, this thesis addresses three main challenges:

1. The parameterization problem for existing uplift modeling algorithms.
2. Data bias in uplift modeling.
3. The high dimensionality problem in uplift modeling.

We address these challenges by first proposing in **Chapter 3** the UMODL approach. UMODL aims to identify the model that is most likely given the data. This means finding the model that maximizes the posterior probability $P(M|D)$. By applying Bayes' theorem, we find that maximizing the posterior probability is similar to maximizing the product of the prior probability and the likelihood. Thus, UMODL specifies a space of models and a prior distribution. From this model space, a Bayesian optimal evaluation criterion is defined, which is determined by taking the negative logarithm of the stated posterior probability. A search algorithm is then used to find the model with the optimal criterion. This approach is completely user parameter free and can be applied to a variety of model types.

For an uplift discretization model (see **Section 3.2.1**), the prior distribution is defined by the number of intervals, the bounds of those intervals, the presence or absence of a treatment effect in each interval, and the class frequencies per treatment in each interval (or per interval). The prior distribution is defined by assuming the independence of the distributions across intervals and by assuming a uniform distribution for each parameter. We have shown that the UMODL

discretization, being a density estimation approach, is a good univariate uplift estimator. Finding a set of intervals with different treatment effects is equivalent to estimating the treatment effect for each instance in those intervals. We have conducted an experimental protocol to assess UMODL as an uplift estimator through discretization. We have defined different synthetic uplift patterns and generated accordingly several datasets with several data sizes. The use of synthetic data gave us the advantage to know the true uplift value and thus be able to compare the estimated uplift value by our approach and the true one. By observing the RMSE of the predicted uplift values and the number of found intervals by data size, we were able to infer the following characteristics: 1. UMODL is a good uplift estimator through discretization. 2. UMODL does not overfit 3. It needs sufficient number of instances to give prominence to a model with more intervals

We proceeded to show the application of the UMODL discretization technique to categorical variables. Essentially, the discretization attempts to create a set of intervals that can partition a continuous variable into distinct categories while preserving the maximum amount of information from the original continuous attribute. One obvious method for applying the UMODL discretization strategy to a categorical variable is to transform the categorical values into numerical values using traditional label encoding. In the same chapter, **Section 3.4**, we explain why traditional label encoding is inadequate, and then propose a supervised uplift-based label encoding. The proposed encoding ranks the categorical values of a variable according to their respective uplift values. This technique efficiently groups instances with similar uplift values, as explained in **Section 3.4.2**.

In the last part of Chapter 3, we introduced UMODL-FS, a feature selection method that was shown experimentally to be efficient at eliminating noise variables and to find the set of variables that lead to uplift models with the best qini. The UMODL-FS approach computes a divergence measure between the treatment and control distributions for each interval found in a variable. The sum of the divergences over the intervals becomes the score of the variable. Additionally to selecting the most informative set of features, UMODL-FS allows us to continue selecting top features, as long as it gives them an importance score greater than zero. Once a score of zero is reached, no additional features are selected, thus establishing a threshold for feature selection. However, determining an appropriate cut-off score for feature selection cannot be done for other state-of-the art feature selection approaches. We conducted an experimental protocol on real and synthetic datasets, where the idea was to gradually add noisy features and build several uplift models, each with a different feature selection method as a preprocessing step. Experiments show that UMODL-FS removes irrelevant features and

clearly outperforms state of the art methods by providing uplift models with the highest and most stable qini.

Following the same Bayesian approach, in **Chapter 4** we propose a new user-parameter-free uplift Bayesian decision tree approach, which we call UB-DT. Unlike conventional decision tree learning approaches, we transform the uplift decision tree learning problem into an optimization problem, where the goal is to find the uplift tree that is most likely given the data. UB-DT consists of two components: a global evaluation criterion for a binary uplift decision tree and a search algorithm to find the optimal tree. A global evaluation criterion evaluates an entire tree model, taking into account all splits in the tree at once. Following the same logic, we have defined the parameters and the evaluation criterion for an uplift decision tree. We have defined the uplift tree model by:

- its structure (a set of internal and leaf nodes),
- the distribution of instances in internal nodes. This is described by the segmentation variable for each node and the way the instances are divided into two child nodes,
- the distribution of instances in leaf nodes. Here, each leaf node could either have a treatment effect, which is described by the distribution of instances per treatment, or it could have no treatment effect, in which case it's represented by the distribution of instances.

These parameters were formally defined as well as their hierarchy, which describes the dependencies between the parameters. The hierarchy of the uplift tree model is described from the root node to its children and recursively to the leaves. We also assume independence of the distribution of outcome values between child nodes and a uniform distribution for each parameter. Again, the global evaluation criterion is defined as the negative logarithm of the posterior probability of an uplift tree model given the data. A search algorithm is then used to find the optimal tree model. The extension of the uplift decision tree search algorithm to random forests is also presented in **Section 4.2.5**. We evaluated UB-DT versus state-the-art tree-based approaches on 2 synthetic patterns. We generated several datasets according to these patterns with several different numbers of instances ranging from 100 to 100,000 instances. For each dataset, models are learnt using 10-fold cross validation and evaluated using the RMSE. With UB-DT, RMSE decreases and converges to zero when the data size increases for both synthetic patterns. Subsequently, we evaluated UB-DT and UB-RF against state-of-the-art uplift algorithms on real and synthetic datasets widely used in the uplift

modeling community. The results show that our proposed approaches remain competitive when compared to existing state-of-the-art algorithms.

In **Chapter 5**, we address the problem of data bias. More particularly, we define the non-random assignment bias (NRA) in the context of uplift modeling. The NRA bias occurs when the treatment and control groups do not have the same distributions of the samples. We designed an experimental protocol to simulate the NRA bias in uplift datasets. The goal of the protocol was to generate the NRA bias in the data and study its effects on state-of-the-art uplift methods, as well as our proposed decision tree and random forest approaches, UB-DT and UB-RF. The results of our experiments showed that the models with the highest resistance to the NRA bias are mainly the meta-learners and ***our newly proposed UB-RF method***. Our UB-DT approach, although being a single tree, showed competitive performance versus the NRA bias. Another conclusion was that the class-transformation approach is particularly sensitive to the NRA bias. In the second contribution of this chapter, we proposed to use a reweighting method based on the propensity scores to weight individuals in the treatment group based on their corresponding weights in the control group, thereby making the biased population more similar to the unbiased population. We tested this approach with the class transformation approach, that was greatly improved with the reweighting method.

In **Chapter 6** we performed additional evaluations of our proposed approaches on real telecom data. We first illustrate the steps involved in an uplift modeling process as practiced in telecom companies. Following this, we run a series of experiments to compare UB-DT and UB-RF approaches against state-of-the-art methods using real telecom data provided by the French company *Orange*. We highlighted the importance of parameter-free approaches to liberate us from choosing parameters and/or base learners for meta-learners. We then investigated the effects of preprocessing uplift data via feature selection (using UMODL-FS) and/or variable transformation (using UMODL discretization) on the resultant uplift models. Our analysis showed that when logistic regression is used as the base learner, variable transformation is significantly more beneficial than feature selection. Conversely, when random forests are used as the base learner for meta-learners, feature selection becomes the more favorable option. In all our experiments (except the tests with ED-RF approach), preprocessing the data with feature selection or variable transformation was found to lead to better uplift models than only using the original data without preprocessing.

7.2 Future directions

In this section we give several perspectives for this thesis. We begin by presenting two axes of perspectives: one that focuses on future directions for our Bayesian uplift approach, and the second on exploring perspectives for uplift modeling. Fig. 7.1 shows a visual illustration of these two axes. Finally, we present a third axis about exploring data bias in the context of uplift modeling.

7.2.1 Extension of our Bayesian approach

Several extensions to our Bayesian approach can be proposed. First, the proposed criterions for UMODL and UB-DT were designed for continuous attributes. For categorical variables, in Section 3.4 we proposed an uplift-based encoding for categorical data. We used this encoding to efficiently convert categorical variables into numerical variables and thus apply the UMODL approach.

However, a new UMODL criterion for categorical variables could be designed. Given a categorical variable, this criterion would aim to group together values that exhibit similar behavior, i.e., have the same uplift density. The MODL approach for categorical variables [9] could be considered as a first reference. This new criterion can also be used for uplift feature selection to give more reliable importance scores for categorical variables.

Subsequently, the UMODL criterion for continuous variables that we presented in Chapter 3 could be integrated with the UMODL criterion for categorical variables to extend the uplift decision tree approach. Thus, a new UB-DT criterion could be proposed that makes appropriate splits for both continuous and categorical variables.

Concerning our proposed uplift decision tree approach, we emphasize that improving the search algorithm design and implementation is crucial to make the developed algorithms more tractable. In addition, our proposed uplift decision tree search algorithm can be improved by using a post-optimization algorithm to prevent our search algorithm from falling into local minima [71].

We also note that for simplicity, our proposed UB-DT criterion and search algorithm were designed for binary trees. An extended criterion to *General Trees*, where each node can have many children, can be defined. This will give the UB-DT approach the freedom to do binary splitting and/or multiple splittings depending on the value of its criterion. This may allow uplift Bayesian decision trees to model more complex patterns.

Finally, the Bayesian approach we present in this thesis is a general approach that can also be applied to a variety of models such as Selective Naive Bayes [10], k-nearest-neightbours [23]. The difference for each type of model is how we de-

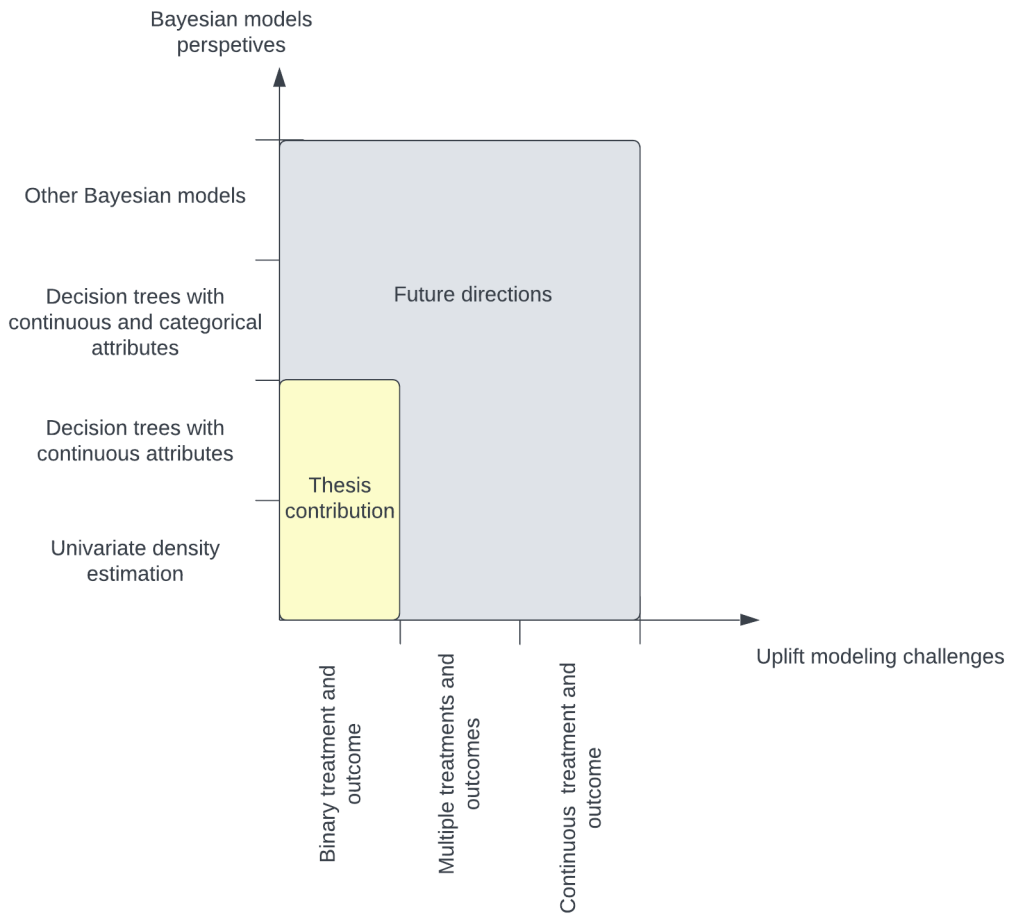


Figure 7.1: Future directions

termine the prior distribution and the model space. As discussed in Chapter 3 and Chapter 4, in general, a prior distribution is determined by exploiting the hierarchical structure of its parameters (each model type has its own set of parameters). This hierarchy indicates the dependencies between parameters and implies that the parameters must be chosen in a certain order when defining the prior distribution.

7.2.2 Future directions for the uplift modeling problem

So far, our Bayesian approach, as well as most of the literature on uplift modeling, has been designed for the case of binary treatment and binary outcome. However, there are applications with multiple treatments, such as when we try to find the optimal marketing campaign among several types of campaigns. Examples of

multiple treatments in marketing include choosing the optimal treatment for each customer such as offering a discount on a customer's monthly bill or offering extra data for free or offering free access to a premium service. To our knowledge, the problem of multiple treatments has not been studied much in the literature. Some studies include tree-based approaches for multiple treatments [53, 63, 26].Our Bayesian criterion for discretization and uplift decision trees can also be extended to the case of multiple treatments.

In addition, the continuous treatment problem is very interesting and very needed in various domains such as medical and marketing domains. Examples of continuous treatments are the drug dosage to be given to a patient in the medical field and the length of SMS sent in the marketing field. Modeling continuous variables using the MODL approach was studied in [35] and can be used as a basis of a new uplift criterion for continous treatments and outcomes.

Another type of an uplift modeling problem is the estimation of individual uplift based on sequence data [20]. Essentially, this involves determining the uplift for each individual, taking into account a sequence of their behaviors. An example of sequence data: a customer initially accepted an Internet offer, then upgraded to a premium package, and then subscribed to a movie platform.

7.2.3 Data bias in uplift modeling

A third axis of perspective is data bias. In this thesis, we have studied a particular type of bias called the non-random assignment bias. In Section 2.5, we have also introduced other types of bias, such as the non-response bias and the deployment bias. These types of biases can be studied in the future. Experimental protocols can be designed to simulate them in the datasets, similar to what we did in Chapter 5. As an example, one potential experimental protocol could be to simulate the deployment bias within the data and then examine its effect on the uplift modeling approaches. This could be accomplished by designing experiments in which we first apply the uplift approach to training data and then test it on data from an identical distribution. Gradually, we would introduce variation into the test data, moving it away from the distribution of the training set. In this way, we can observe the ability of different uplift methods to maintain their generalization and the degree to which they can resist these changes.

APPENDIX A

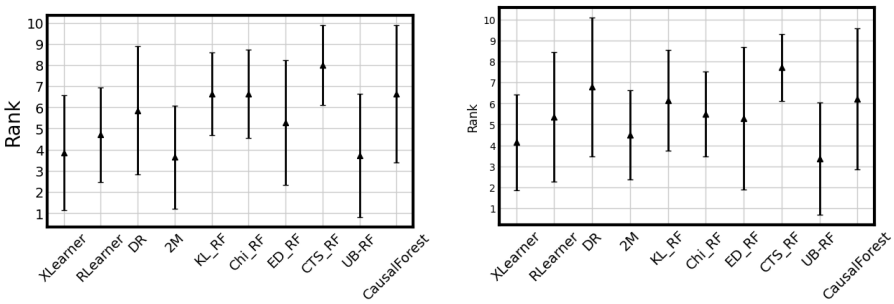
Appendix

A.1 Experimental results with 50 trees

In Chapter 4, we conduct an experimental protocol to compare the performance of the UB-RF approach with state-of-the-art uplift modeling approaches on several datasets. For all approaches, we use 10 trees to construct the uplift models. In this appendix, we present supplementary results in which each of the algorithms is built using 50 trees. The corresponding average qini values are shown in Table A.1.

Dataset	XLearner	RLearner	DR	2M	KL-RF	Chi-RF	ED-RF	CTS-RF	UB-RF	CausalForest
Hillstrom-m	1.1(2.6)	0.6(2.0)	1.1(1.8)	0.3(2.7)	-1.0(1.5)	-0.3(1.6)	-0.4(1.8)	-0.1(1.5)	**1.5(1.4**	-0.2(2.1)
Hillstrom-w	4.2(0.9)	6.1(1.6)	6.0(1.7)	4.7(1.5)	4.3(1.5)	4.7(1.3)	4.4(1.2)	4.5(1.4)	**6.5(0.9**	4.0(1.6)
Hillstrom-mw	2.9(2.3)	**3.8(2.6**	3.4(2.7)	2.8(1.9)	0.4(1.1)	0.7(1.0)	0.6(1.8)	1.1(1.1)	3.0(1.7)	0.5(1.0)
Gerber-n	**6.1(0.6**	1.9(0.6)	0.6(1.0)	5.7(0.6)	5.3(1.0)	5.4(1.0)	5.7(0.6)	4.2(0.9)	2.8(0.8)	4.5(0.9)
Gerber-s	**5.4(0.8**	1.7(0.7)	1.1(0.9)	4.9(1.0)	4.9(0.7)	5.0(1.0)	5.0(0.7)	4.5(0.6)	2.1(0.8)	4.7(0.8)
Criteo-c	19.6(2.3)	19.3(1.0)	16.6(6.0)	18.4(1.3)	19.6(1.9)	19.1(1.8)	**22.0(1.6**	10.0(1.9)	20.8(0.9)	15.3(1.9)
Criteo-v	3.1(0.7)	5.0(0.5)	-3.4(3.5)	2.6(0.7)	5.9(0.5)	5.1(0.6)	**6.5(0.5**	2.9(0.8)	6.1(0.6)	1.6(0.4)
Megafon	**18.8(0.6**	2.6(0.5)	2.3(0.4)	18.3(0.7)	16.5(0.6)	15.9(0.4)	17.2(0.5)	13.8(0.7)	14.3(0.8)	14.0(0.5)
Bank-tel	-4.8(8.2)	0.9(5.2)	-2.7(13.1)	16.4(9.1)	-13.9(5.4)	-9.1(8.1)	-16.6(4.2)	-20.5(3.9)	26.0(5.9)	**38.5(8.2**
Bank-cell	11.8(4.6)	19.9(5.1)	5.7(11.2)	27.5(3.4)	0.6(3.1)	0.9(2.3)	-0.6(3.1)	-1.7(2.8)	**49.0(2.7**	30.9(2.3)
Bank-tel-cell	10.3(4.2)	17.4(8.3)	3.7(10.1)	27.6(3.9)	1.5(3.0)	1.3(3.2)	-2.5(3.7)	-0.6(1.9)	**48.6(1.5**	32.0(0.9)
Information	13.0(3.2)	10.1(2.8)	3.3(2.2)	11.8(4.6)	12.9(3.2)	13.1(3.3)	13.3(3.0)	12.4(3.6)	13.6(3.3)	**14.1(2.4**
Starbucks	17.9(4.6)	**22.5(3.8**	22.4(3.9)	17.6(3.3)	17.3(3.8)	16.9(5.7)	17.1(5.3)	16.9(3.8)	20.2(3.3)	13.8(4.3)
RHC	33.0(2.8)	29.3(4.1)	28.2(5.0)	**36.8(2.9**	34.1(4.8)	34.3(5.1)	34.5(4.7)	32.6(4.6)	30.6(4.4)	30.4(3.7)

Table A.1: Average qini values and standard deviation (multiplied by 100) across datasets and uplift approaches. In bold, the best value for each dataset. Each approach was learnt with 50 trees.



(a) 10 trees (b) 50 trees

Figure A.1: Overall average ranking of the uplift approaches

We show in Fig. A.1b the overall average ranking of the uplift approaches with 50 trees. Compared to the average ranking of the uplift approaches with 10 trees (cf. Fig. A.1a), the UB-RF has a slight improvement in ranking, but for the rest of the uplift approaches no significant ranking difference can be observed.

To facilitate the comparison of the performance of different methods on 50 trees (see Table 4.3) and 10 trees (see Table A.1), we show the delta values in Table A.2. They represent the difference between the qini values in both tables, associated with each uplift approach and each respective dataset. A positive delta value means an improvement in performance when increasing the number of trees from 10 to 50.

When evaluating Table A.2, we note that the performance of UB-RF, along with other forest-based methods (KL-RF, Chi-RF, ED-RF, Causal Forest), shows an improvement as the number of trees increases. Particularly noteable are the positive delta values observed in the Causal Forest approach for all datasets. However, it is to be noted that the initial performance of the Causal Forest approach with 10 trees was poor. The performance of metalearners using xgboost as a base learner decreases as the number of trees is increased. When increasing the number of trees, the performance of UB-RF shows an improvement and at the same time maintains its position among the best uplift approaches. With 50 trees, it has the best average ranking among all the uplift approaches.

Dataset	XLearner	RLearner	DR	2M	KL-RF	Chi-RF	ED-RF	CTS-RF	UB-RF	CausalForest
Hillstrom-m	0.8	0.4	0.1	-0.4	-1.0	0.6	-0.5	0.2	-0.3	-0.2
Hillstrom-w	-2.0	-0.1	0.0	-0.2	-1.9	-2.3	0.7	0.2	-0.2	2.0
Hillstrom-mw	-0.8	0.0	-0.4	-0.2	-2.6	-2.1	0.6	0.7	-0.1	-0.6
Gerber-n	2.4	0.0	0.3	2.6	3.5	3.3	1.4	1.0	0.1	2.1
Gerber-s	3.0	0.0	0.7	2.7	3.6	3.6	1.5	1.2	0.3	1.9
Criteo-c	-2.7	0.0	-3.5	-1.1	5.0	6.7	1.6	0.7	2.1	4.0
Criteo-v	2.8	-0.3	-6.7	-1.3	0.5	0.3	1.2	0.5	0.4	1.4
Megafon	0.6	0.0	0.1	1.7	5.3	4.9	6.4	4.6	1.5	4.3
Bank-tel	-19.3	-1.5	-18.6	-4.7	1.6	-3.0	-2.2	-2.1	-0.7	12.7
Bank-cell	-7.0	-5.0	-15.0	-3.5	0.2	-0.6	2.3	-0.5	3.5	10.2
Bank-tel-cell	-5.9	-5.8	-14.2	-2.9	0.1	1.7	0.4	1.0	2.5	11.6
Information	-1.9	0.3	-0.9	-1.9	3.3	3.4	2.2	0.9	1.6	3.3
Starbucks	-4.4	0.0	-0.1	-5.1	-5.1	-4.5	2.2	2.7	0.0	4.0
RHC	0.4	-2.3	-1.7	2.1	5.4	5.0	1.9	1.9	4.1	5.9
Mean	-2.4	-1.0	-4.3	-0.9	1.3	1.2	1.4	0.9	1.1	4.5

Table A.2: The delta values (the difference) between qini values when the model is trained with 50 trees and qini values when the model is trained with 10 trees. The last row shows the average of the delta values.

Bibliography

[1] Salem Alelyani, Jiliang Tang, and Huan Liu. Feature selection for clustering: A review. *Data Clustering*, pages 29–60, 2018.

[2] Joshua D. Angrist. Treatment effect. In Steven N. Durlauf and Lawrence E. Blume, editors, *Microeconometrics*, pages 329–338, London, 2010. Palgrave Macmillan UK.

[3] Onur Atan, James Jordon, and Mihaela Van der Schaar. Deep-treat: Learning optimal personalized treatments from observational data using neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[4] Susan Athey and Guido Imbens. Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, 113(27):7353–7360, 2016.

[5] Susan Athey and Guido W Imbens. Machine learning methods for estimating heterogeneous causal effects. *stat*, 1050(5):1–26, 2015.

[6] RICHARD Bellman. Dynamic programming, princeton univ. *Press Princeton, New Jersey*, 1957.

[7] Artem Betlei, E. Diemert, and Massih-Reza Amini. Uplift prediction with dependent feature representation in imbalanced treatment and control conditions. In *International Conference on Neural Information Processing*, 2018.

[8] Marc Boullé. MODL: A bayes optimal discretization method for continuous attributes. *Mach. Learn.*, 65(1):131–165, 2006.

[9] M. Boullé. A Bayes optimal approach for partitioning the values of categorical attributes. *Journal of Machine Learning Research*, 6:1431–1452, 2005.

[10] M. Boullé. Compression-based averaging of selective naive Bayes classifiers. *Journal of Machine Learning Research*, 8:1659–1685, 2007.

[11] M. Boullé. *Recherche d'une représentation des données efficace pour la fouille des grandes bases de données*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, 2007.

[12] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[13] Leo Breiman, J. H. Friedman, Richard A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.

[14] Francesco Calabrese, Laura Ferrari, and Vincent D Blondel. Urban sensing using mobile phone network data: a survey of research. *Acm computing surveys (csur)*, 47(2):1–20, 2014.

[15] Huigang Chen, Totte Harinen, Jeong-Yoon Lee, Mike Yung, and Zhenyu Zhao. Causalml: Python package for causal machine learning. *arXiv preprint arXiv:2002.11631*, 2020.

[16] Alfred F. Connors et al. The effectiveness of right heart catheterization in the initial care of critically ill patients. support investigators. *JAMA*, 276 11:889–97, 1996.

[17] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30, 2006.

[18] Floris Devriendt, Jente Van Belle, Tias Guns, and Wouter Verbeke. Learning to rank for uplift modeling. *IEEE Transactions on Knowledge and Data Engineering*, 34(10):4888–4904, 2022.

[19] Eustache Diemert, Artem Betlei, Christophe Renaudin, and Massih-Reza Amini. A Large Scale Benchmark for Uplift Modeling. In *KDD*, London, United Kingdom, 2018.

[20] E. Egho, D. Gay, M. Boullé, N. Voisine, and F. Clérot. A user parameter-free approach for mining robust sequential classification rules. *Knowledge and Information Systems*, 52(1):53–81, 2017.

[21] Elias Egho, Dominique Gay, Marc Boullé, Nicolas Voisine, and Fabrice Clérot. A user parameter-free approach for mining robust sequential classification rules. *Knowledge and Information Systems*, 52:53–81, 2017.

[22] Mohamed K El Mahrsi, Romain Guigourès, Fabrice Rossi, and Marc Boullé. Co-clustering network-constrained trajectory data. *Advances in Knowledge Discovery and Management: Volume 5*, pages 19–32, 2016.

[23] Sylvain Ferrandiz and Marc Boullé. Bayesian instance selection for the nearest neighbor rule. *Machine learning*, 81:229–256, 2010.

[24] João Gama, Indrundefined Žliobaitundefined, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Comput. Surv.*, 46(4), March 2014.

[25] Alan S. Gerber, Donald P. Green, and Christopher W. Larimer. Social pressure and voter turnout: Evidence from a large-scale field experiment. *American Political Science Review*, 2008.

[26] Robin Marco Gubela and Stefan Lessmann. Interpretable multiple treatment revenue uplift modeling. *CoRR*, abs/2101.03336, 2021.

[27] Leo Guelman, Montserrat Guillén, and Ana Maria Pérez-Marín. Uplift random forests. *Cybernetics and Systems*, 46:230 – 248, 2015.

[28] Leo Guelman, Montserrat Guillén, and Ana M. Pérez-Marín. A decision support framework to implement optimal personalized marketing interventions. *Decision Support Systems*, 72:24–32, 2015.

[29] Pierre Gutierrez and Jean-Yves Gérardy. Causal inference and uplift modelling: A review of the literature. In *PAPIs*, 2016.

[30] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[31] Behram Hansotia and Brad Rukstales. Incremental value modeling. *Journal of Interactive Marketing*, 16:35–46, 2002.

[32] Jennifer L. Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011.

[33] Günter J. Hitsch and S. Misra. Heterogeneous treatment effects and optimal targeting policy evaluation. *Randomized Social Experiments eJournal*, 2018.

[34] Jinping Hu. Customer feature selection from high-dimensional bank direct marketing data for uplift modeling. *Journal of Marketing Analytics*, pages 1–12, 2022.

[35] C. Hue and M. Boullé. A new probabilistic approach in rank regression with optimal bayesian partitioning. *Journal of Machine Learning Research*, pages 2727–2754, 2007.

[36] Daniel Jacob. Cate meets ml. *Digital Finance*, 3(2):99–148, 2021.

[37] KJ Jager, C Zoccali, A Macleod, and FW Dekker. Confounding: what it is and how to deal with it. *Kidney international*, 73(3):256–260, 2008.

[38] Maciej Jaskowski and Szymon Jaroszewicz. Uplift modeling for clinical trial data. In *ICML Workshop On Clinical Data Analysis*, 2012.

[39] Edward H. Kennedy. Optimal doubly robust estimation of heterogeneous causal effects, 2020.

[40] Wouter M. Kouw and Marco Loog. An introduction to domain adaptation and transfer learning, 2019.

[41] Oliver Kramer and Oliver Kramer. Scikit-learn. *Machine learning for evolution strategies*, pages 45–53, 2016.

[42] Sören R. Künzel, Jasjeet S. Sekhon, Peter J. Bickel, and Bin Yu. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences*, 116(10):4156–4165, Feb 2019.

[43] The Jackson Laboratory. What is personalized medicine? https://www.jax.org/personalized-medicine/precision-medicine-and-you/what-is-precision-medicine.

[44] Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*, volume 793. John Wiley & Sons, 2019.

[45] Huan Liu and Rudy Setiono. Feature selection via discretization. *IEEE Trans. Knowl. Data Eng.*, 9(4):642–645, 1997.

[46] Victor SY Lo. The true lift model: a novel data mining approach to response modeling in database marketing. *ACM SIGKDD Explorations Newsletter*, 4(2):78–86, 2002.

[47] Christos Louizos, Uri Shalit, Joris M Mooij, David Sontag, Richard Zemel, and Max Welling. Causal effect inference with deep latent-variable models. *Advances in neural information processing systems*, 30, 2017.

[48] Jose G. Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V. Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521 – 530, 2012.

[49] Sérgio Moro, Paulo Cortez, and Paulo Rita. A data-driven approach to predict the success of bank telemarketing. *Decis. Support Syst.*, 62:22–31, 2014.

[50] Houssam Nassif, Finn Kuusisto, Elizabeth S Burnside, and Jude W Shavlik. Uplift modeling with roc: An srl case study. In *ILP (late breaking papers)*, pages 40–45. Citeseer, 2013.

[51] GE Naumov. Np-completeness of problems of construction of optimal decision trees. In *Soviet Physics Doklady*, volume 36, page 270, 1991.

[52] X Nie and S Wager. Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika*, 108(2):299–319, 09 2020.

[53] Diego Olaya, Kristof Coussement, and Wouter Verbeke. A survey and benchmarking study of multitreatment uplift modeling. *Data Min. Knowl. Discov.*, 34(2):273–308, 2020.

[54] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[55] Mohamad Amin Pourhoseingholi, Ahmad Reza Baghestani, and Mohsen Vahedi. How to control confounding effects by statistical analysis. *Gastroenterology and hepatology from bed to bench*, 5(2):79, 2012.

[56] Nicholas Radcliffe. Using control groups to target on predicted lift: Building and assessing uplift model. *Direct Marketing Analytics Journal*, pages 14–21, 2007.

[57] Nicholas J Radcliffe and Patrick D Surry. Real-world uplift modelling with significance-based uplift trees. *White Paper TR-2011-1, Stochastic Solutions*, pages 1–33, 2012.

[58] Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

[59] Paul R. Rosenbaum and Donald B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 04 1983.

[60] Donald B. Rubin. Estimating causal effects from large data sets using propensity scores. *Annals of internal medicine*, 127(8_Part_2):757–763, 1997.

[61] Donald B. Rubin. Using propensity scores to help design observational studies: Application to the tobacco litigation. *Health Services and Outcomes Research Methodology*, 2(3):169–188, Dec 2001.

[62] Piotr Rzepakowski and Szymon Jaroszewicz. Decision trees for uplift modeling. In *2010 IEEE International Conference on Data Mining*, pages 441–450, 2010.

[63] Piotr Rzepakowski and Szymon Jaroszewicz. Decision trees for uplift modeling with single and multiple treatments. *Knowl. Inf. Syst.*, 32(2):303–327, 2012.

[64] Uri Shalit, Fredrik D Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *International Conference on Machine Learning*, pages 3076–3085. PMLR, 2017.

[65] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55, 2001.

[66] Sadia Sharmin, Mohammad Shoyaib, Amin Ahsan Ali, Muhammad Asif Hossain Khan, and Oksam Chae. Simultaneous feature selection and discretization based on mutual information. *Pattern Recognit.*, 91:162–174, 2019.

[67] Michal Soltys, Szymon Jaroszewicz, and Piotr Rzepakowski. Ensemble methods for uplift modeling. *Data Mining and Knowledge Discovery*, 29:1531 – 1559, 2014.

[68] Patrick D Surry and Nicholas J Radcliffe. Quality measures for uplift models. *KDD2011*, 2011.

[69] Giordano Tamburrelli and Alessandro Margara. Towards automated a/b testing. In *Search-Based Software Engineering: 6th International Symposium, SSBSE 2014, Fortaleza, Brazil, August 26-29, 2014. Proceedings 6*, pages 184–198. Springer, 2014.

[70] Joaquin Vanschoren. Meta-learning. *Automated machine learning: methods, systems, challenges*, pages 35–61, 2019.

[71] N. Voisine, M. Boullé, and C. Hue. A bayes evaluation criterion for decision trees. *Advances in Knowledge Discovery and Management (AKDM-1)*, 292:21–38, 2010.

[72] Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113(523):1228–1242, 2018.

[73] Chris S. Wallace and Jon D Patrick. Coding decision trees. *Machine Learning*, 11:7–22, 1993.

[74] Jinsung Yoon, James Jordon, and Mihaela Van Der Schaar. Ganite: Estimation of individualized treatment effects using generative adversarial nets. In *International conference on learning representations*, 2018.

[75] Łukasz Zaniewicz and Szymon Jaroszewicz. L_p support vector machines for uplift modeling. *Knowledge and Information Systems*, 53(1):269–296, 2017.

[76] Łukasz Zaniewicz and Szymon Jaroszewicz. Support vector machines for uplift modeling. In *2013 IEEE 13th International Conference on Data Mining Workshops*, pages 131–138, 2013.

[77] Weijia Zhang, Jiuyong Li, and Lin Liu. A unified survey of treatment effect heterogeneity modelling and uplift modelling. *ACM Computing Surveys (CSUR)*, 54(8):1–36, 2021.

[78] Weijia Zhang, Lin Liu, and Jiuyong Li. Treatment effect estimation with disentangled latent factors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10923–10930, 2021.

[79] Yan Zhao, Xiao Fang, and David Simchi-Levi. Uplift modeling with multiple treatments and general response types, 2017.

[80] Zheng Zhao and Huan Liu. Semi-supervised feature selection via spectral analysis. In *Proceedings of the 2007 SIAM international conference on data mining*, pages 641–646. SIAM, 2007.

[81] Zhenyu Zhao, Yumin Zhang, Totte Harinen, and Mike Yung. Feature selection methods for uplift modeling. *CoRR*, abs/2005.03447, 2020.

[82] Marc-André Zöller and Marco F Huber. Benchmark and survey of automated machine learning frameworks. *Journal of artificial intelligence research*, 70:409–472, 2021.