Université de Montréal

Apprentissage basé sur le Qini pour la prédiction de l'effet causal conditionnel

par

Mouloud-Beallah Belbahri

Département de mathématiques et de statistique
Faculté des arts et des sciences

Thèse présentée à la Faculté des études supérieures et postdoctorales
en vue de l'obtention du grade de
Philosophiæ Doctor (Ph.D.)
en Statistique

août 2021

Université de Montréal

Faculté des études supérieures et postdoctorales

Cette thèse intitulée

Apprentissage basé sur le Qini pour la prédiction de l'effet causal conditionnel

présentée par

Mouloud-Beallah Belbahri

a été évaluée par un jury composé des personnes suivantes :

Martin Bilodeau
(président-rapporteur)

Alejandro Murua
(directeur de recherche)

Vahid Partovi Nia
(co-directeur)

Pierre Duchesne
(membre du jury)

Thierry Duchesne
(examinateur externe)

Sean Horan
(représentant du doyen de la FESP)

Thèse acceptée le :
21 juin 2021

Résumé

Les modèles *uplift* (levier en français) traitent de l'inférence de cause à effet pour un facteur spécifique, comme une intervention de marketing. En pratique, ces modèles sont construits sur des données individuelles issues d'expériences randomisées. Un groupe traitement comprend des individus qui font l'objet d'une action; un groupe témoin sert de comparaison. La modélisation *uplift* est utilisée pour ordonner les individus par rapport à la valeur d'un effet causal, par exemple, positif, neutre ou négatif.

Dans un premier temps, nous proposons une nouvelle façon d'effectuer la sélection de modèles pour la régression *uplift*. Notre méthodologie est basée sur la maximisation du coefficient Qini. Étant donné que la sélection du modèle correspond à la sélection des variables, la tâche est difficile si elle est effectuée de manière directe lorsque le nombre de variables à prendre en compte est grand. Pour rechercher de manière réaliste un bon modèle, nous avons conçu une méthode de recherche basée sur une exploration efficace de l'espace des coefficients de régression combinée à une pénalisation de type lasso de la log-vraisemblance. Il n'y a pas d'expression analytique explicite pour la surface Qini, donc la dévoiler n'est pas facile. Notre idée est de découvrir progressivement la surface Qini comparable à l'optimisation sans dérivée. Le but est de trouver un maximum local raisonnable du Qini en explorant la surface près des valeurs optimales des coefficients pénalisés. Nous partageons ouvertement nos codes à travers la librairie R **tools4uplift**. Bien qu'il existe des méthodes de calcul disponibles pour la modélisation *uplift*, la plupart d'entre elles excluent les modèles de régression statistique. Notre librairie entend combler cette lacune. Cette librairie comprend des outils pour: i) la discrétisation, ii) la visualisation, iii) la sélection de variables, iv) l'estimation des paramètres et v) la validation du modèle. Cette librairie permet aux praticiens d'utiliser nos méthodes avec aise et de se référer aux articles méthodologiques afin de lire les détails.

L'*uplift* est un cas particulier d'inférence causale. L'inférence causale essaie de répondre à des questions telle que « Quel serait le résultat si nous donnions à ce patient un traitement A au lieu du traitement B? ». La réponse à cette question est ensuite utilisée comme prédiction pour un nouveau patient. Dans la deuxième partie de la thèse, c'est sur la prédiction que nous avons davantage insisté. La plupart des approches existantes sont des adaptations de forêts aléatoires pour le cas de l'*uplift*. Plusieurs critères de segmentation ont été proposés dans la littérature, tous reposant sur la maximisation de l'hétérogénéité. Cependant, dans la pratique, ces approches sont sujettes au sur-ajustement. Nous apportons une nouvelle vision pour améliorer la prédiction de l'*uplift*. Nous proposons une nouvelle fonction de perte définie en tirant parti d'un lien avec l'interprétation bayésienne du risque relatif. Notre solution est développée pour une architecture de réseau de neurones jumeaux spécifique permettant d'optimiser conjointement les probabilités marginales de succès pour les individus traités et non-traités. Nous montrons que ce modèle est une généralisation du modèle d'interaction logistique de l'*uplift*. Nous modifions également l'algorithme de descente de gradient stochastique pour permettre des solutions parcimonieuses structurées. Cela aide dans une large mesure à ajuster nos modèles *uplift*. Nous partageons ouvertement nos codes Python pour les praticiens désireux d'utiliser nos algorithmes.

Nous avons eu la rare opportunité de collaborer avec l'industrie afin d'avoir accès à des données provenant de campagnes de marketing à grande échelle favorables à l'application de nos méthodes. Nous montrons empiriquement que nos méthodes sont compétitives avec l'état de l'art sur les données réelles ainsi qu'à travers plusieurs scénarios de simulations.

Mots-clés : Descente de gradient; discrétisation; effets hétérogènes du traitement; fonction de perte; inférence causale; optimisation sans dérivée; régression logistique; régularisation; réseau de neurones artificiels; sélection de variables.

Abstract

Uplift models deal with cause-and-effect inference for a specific factor, such as a marketing intervention. In practice, these models are built on individual data from randomized experiments. A targeted group contains individuals who are subject to an action; a control group serves for comparison. Uplift modeling is used to order the individuals with respect to the value of a causal effect, e.g., positive, neutral, or negative.

First, we propose a new way to perform model selection in uplift regression models. Our methodology is based on the maximization of the Qini coefficient. Because model selection corresponds to variable selection, the task is haunting and intractable if done in a straightforward manner when the number of variables to consider is large. To realistically search for a good model, we conceived a searching method based on an efficient exploration of the regression coefficients space combined with a lasso penalization of the log-likelihood. There is no explicit analytical expression for the Qini surface, so unveiling it is not easy. Our idea is to gradually uncover the Qini surface in a manner inspired by surface response designs. The goal is to find a reasonable local maximum of the Qini by exploring the surface near optimal values of the penalized coefficients. We openly share our codes through the R Package **tools4uplift**. Though there are some computational methods available for uplift modeling, most of them exclude statistical regression models. Our package intends to fill this gap. This package comprises tools for: i) quantization, ii) visualization, iii) variable selection, iv) parameters estimation and v) model validation. This library allows practitioners to use our methods with ease and to refer to methodological papers in order to read the details.

Uplift is a particular case of causal inference. Causal inference tries to answer questions such as "What would be the result if we gave this patient treatment A instead of treatment B ?" . The answer to this question is then used as a prediction for a new patient. In the second part of the thesis, it is on the prediction that we have placed more emphasis. Most

existing approaches are adaptations of random forests for the uplift case. Several split criteria have been proposed in the literature, all relying on maximizing heterogeneity. However, in practice, these approaches are prone to overfitting. In this work, we bring a new vision to uplift modeling. We propose a new loss function defined by leveraging a connection with the Bayesian interpretation of the relative risk. Our solution is developed for a specific twin neural network architecture allowing to jointly optimize the marginal probabilities of success for treated and control individuals. We show that this model is a generalization of the uplift logistic interaction model. We modify the stochastic gradient descent algorithm to allow for structured sparse solutions. This helps fitting our uplift models to a great extent. We openly share our Python codes for practitioners wishing to use our algorithms.

We had the rare opportunity to collaborate with industry to get access to data from large-scale marketing campaigns favorable to the application of our methods. We show empirically that our methods are competitive with the state of the art on real data and through several simulation setting scenarios.

Keywords: Artificial neural network; causal inference; derivative-free optimization; feature selection; gradient descent; heterogeneous treatment effects; logistic regression; loss function; quantization; regularization.

Table des matières

Liste des tableaux

Liste des figures

Remerciements

À ma famille.

Introduction générale

Les modèles *uplift* (levier en français) permettent de quantifier et de prédire l'impact d'une action (ou traitement) sur le comportement d'un individu. Par exemple, en marketing quantitatif, la modélisation *uplift* a pour objectif de détecter les groupes de consommateurs sensibles à une offre commerciale. Plus spécifiquement, l'*uplift* aide à identifier les groupes de personnes étant les plus susceptibles de répondre positivement à une sollicitation marketing. De ce fait, cette stratégie permet de réduire le nombre de messages commerciaux émis, et donc le coût de la publicité.

Au lieu de décrire ce qui se passe (analyse descriptive) ou ce qui va se passer (analyse prédictive), la modélisation *uplift* vise à identifier ce qui *aurait pu* se passer, c'est-à-dire ce qui aurait été observé si le traitement n'avait pas été mis en place, ou bien la réponse que chaque observation aurait manifesté si elle avait été attribuée à un traitement particulier. Ainsi, la modélisation *uplift* est un cas particulier de l'inférence causale avec pour but d'identifier le changement de comportement induit par un traitement. Il existe plusieurs applications dans lesquels les traitements sont utilisés pour identifier le changement de comportement. Nous avons eu la rare opportunité d'accéder aux données réelles de campagnes de marketing à grande échelle, ce qui a permis de guider notre recherche.

De manière générale, nous nous intéressons dans cette thèse à la prédiction de l'effet causal moyen pour différents sous-groupes d'une population d'intérêt à l'aide de données provenant d'expériences randomisées. Nous étudions et discutons les approches de chacune des étapes majeures visant l'inférence causale dans le cadre du paradigme contrefactuel : la sélection de variables, la construction d'un modèle *uplift*, l'estimation des paramètres, les mesures de la qualité d'ajustement et l'évaluation de la performance prédictive. Ces étapes nécessitent toutes des approches différentes de la modélisation statistique traditionnelle. Après un bref survol de l'histoire de la causalité et de la statistique dans le chapitre 1, le chapitre 2 présente

une revue de littérature et introduit la problématique étudiée dans cette thèse. Le reste de la thèse peut être divisé en deux grandes parties.

La première partie, constituée des chapitres 3 et 4, porte sur l'adaptation de la régression logistique pour l'inférence causale. Nous commençons par la proposition d'une nouvelle façon d'estimer les paramètres (et de sélectionner les variables/modèles), basée sur le coefficient Qini, une statistique d'ajustement spécifique aux modèles *uplift*. Le chapitre 3 présente en détail la méthodologie ainsi que l'application aux données réelles ayant motivé cette première direction de recherche. Ce chapitre contient un article écrit en langue anglaise à paraître dans la revue *The Annals of Applied Statistics*. Nous développons dans le chapitre 4 plusieurs outils permettant l'analyse exploratoire des données propres au cadre *uplift*. Nous les présentons à l'aide d'une application pratique des techniques et des développements méthodologiques qui y sont associés. Ce chapitre présente un article en langue anglaise que nous avons récemment soumis à la revue *Journal of Statistical Software*, ainsi que la librairie R **tools4uplift** publiée sur CRAN[1]. Cette librairie comprend des outils pour : i) la discrétisation; ii) la visualisation; iii) la sélection de variables/modèles; iv) l'estimation des paramètres; et v) la validation de modèles *uplift*.

La deuxième partie, constituée des chapitres 5 et 6, aborde l'utilisation des réseaux de neurones pour améliorer la qualité de prédiction des modèles. Le chapitre 5 présente ces derniers et donne un aperçu de certaines des étapes conceptuelles qui ont conduit à leur compréhension mathématique actuelle (en particulier aux théorèmes d'approximation universels). Ce chapitre précise également la notation et le langage utilisé en apprentissage automatique (ou *machine learning* en anglais). Le chapitre 6 présente un article en langue anglaise qui sera prochainement soumis à une revue d'apprentissage statistique. Nous y décrivons notre approche méthodologique et, notamment, l'utilisation des réseaux de neurones jumeaux pour l'inférence causale et la prédiction de l'*uplift*.

Nous concluons cette thèse par l'ouverture d'une discussion mettant en perspective les travaux de recherche présentés dans les chapitres précédents et abordant des directions futures.

[1] `https://cran.r-project.org/web/packages/tools4uplift/index.html`

Chapitre 1

Histoire et inférence causale

Le but de ce court chapitre est de renvoyer vers une bibliographie très succinte le lecteur désireux d'avoir quelques clés sur l'histoire de la causalité et de l'inférence causale. La revue des principales contributions méthodologiques et la problématique étudiée dans cette thèse sont présentées au chapitre 2.

1.1. Introduction

La causalité a, depuis l'antiquité, fasciné les philosophes et les savants. Omniprésente dans notre quête de compréhension du monde, elle constitue un terrain toujours très actif, source de débats, travaux et analyses : « Il n'existe à ce jour aucune analyse communément admise des conditions nécessaires et suffisantes pour considérer que le facteur A est une cause du facteur B » (Max Kitsler, préface de Drouet [2012]). Les difficultés opérationnelles liées au concept de causalité ont graduellement mené les chercheurs dans les sciences appliquées à introduire des notions probabilistes (corrélation, facteurs de risques, etc.) dont les succès sont indéniables dans plusieurs disciplines (épidémiologie, économie, etc.).

Dans l'analyse statistique classique sur des données d'observation (à différencier des études randomisées), le chercheur

- fait des inférences sur l'existence et le type d'association entre variables,
- évalue les probabilités d'événements d'intérêt,
- met à jour ces estimations à la lumière de nouvelles informations.

Cette approche traditionnelle par conditionnement probabiliste se heurte à un obstacle majeur : elle ne peut rien faire de plus que révéler une association éventuelle entre variables. L'inférence causale franchit une étape supplémentaire (*ladder of causation*, ou échelle de

causalité en français [Pearl et Mackenzie, 2018]) : observation, manipulation, contrefactuels. Elle est à contraster avec le (à dissocier du) conditionnement. Elle vise à

- (observation) s'interroger sur le(s) processus de génération de données,
- (manipulation et contrefactuels) évaluer et prédire les effets d'interventions sur les variables explicatives, effectives ou non : « Un contrefactuel est un énoncé qui envisage une possibilité non advenue, qui n'a pas eu lieu » [Drouet, 2015],
- identifier les causes (et leur pertinence) d'événements.

1.2. Causalité et philosophie

Bien qu'il ait toujours été dans la nature humaine de croire qu'il n'y a jamais « de fumée sans feu », l'acception philosophique du concept de causalité a évolué au fil du temps en Occident, tout en demeurant essentiellement de nature métaphysique, de l'antiquité grecque (Platon, Aristote) jusqu'aux penseurs de la Renaissance. Tandis que pour Aristote, la causalité est d'essence multiple (*matérielle, formelle, efficace et finale*), Leibniz, Descartes, Schopenhauer (et bien d'autres) y voient un moyen pour la raison d'affirmer que Dieu est la cause ultime de toute chose (*principe de raison suffisante*).

Ce n'est que vers la fin de la Renaissance que le savant ne cherche plus à se poser des questions sur les causes (les antécédents), mais plutôt à découvrir les lois de la nature. Ceci est à rapprocher, d'une certaine manière, à la cause « efficace » d'Aristote. Le promoteur moderne le plus vigoureux de l'abandon de toute idée de causalité est sans doute Bertrand Russel, arguant du fait que les lois de la physique étant temporellement symétriques, il n'y a pas un *avant* et un *après* un événement, et que la science serait bien servie d'abandonner ce concept devenu obsolète : c'est « une vieille lune, qui ne pouvait pas résister à la critique initiée par D. Hume » [Besneux, 2017].

On s'accorde généralement à dire que la période moderne des débats sur la relation entre causalité et association prend naissance chez le philosophe empiriste écossais David Hume [Hume, 1739, 1748]. Selon Hume, lorsque nous disons de deux types d'objets (d'événements) que « la cause A entraîne l'effet B » (la prise d'aspirine soulage les maux de tête), nous voulons dire que

(1) les causes sont « constamment conjointes » avec les effets,

(2) les effets sont consécutifs (suivent les causes et non l'inverse),

(3) il y a une « connexion nécessaire » entre les causes et les effets. A chaque fois qu'une cause se produit, un effet doit suivre. [Drouet, 2012].

Pour Hume, contrairement aux idées de contiguïté (1) et de succession (2), l'idée de connexion nécessaire (3) est subjective. Elle découle de l'expérience, de l'acte de contempler des objets ou des événements que nous avons vécus comme étant constamment conjoints et se succédant dans un certain ordre, plutôt que de toute propriété observable dans les objets ou les événements eux-mêmes. Cette idée est à la base du problème classique de l'induction.

« Quand donc nous disons qu'un objet est en connexion avec un autre, nous voulons seulement dire que ces objets ont acquis une connexion dans notre pensée et qu'ils font surgir cette inférence qui fait de chacun la preuve de l'existence de l'autre. »

« Après la constante conjonction de deux objets, - chaleur et flamme, par exemple, ou poids et solidité, - nous sommes déterminés par la seule accoutumance à attendre l'un quand paraît l'autre. Cette hypothèse semble même la seule qui explique la difficulté suivante : pourquoi tirons-nous de mille cas une inférence que nous sommes incapables de tirer d'un seul cas, qui ne diffère à aucun égard des précédents? [...] Toutes les inférences tirées de l'expérience sont donc des effets de l'accoutumance et non des effets du raisonnement. » [Hume, 1739].

Brady [2008] « présente les quatre approches observées à travers le temps pour rendre compte d'une causalité » (traduction libre)[1] :

- *conjonction constante :* « Conjonction constante des causes et des effets requis par l'approche néo-huméene » (selon Hume).
- *approche contrefactuelle :* « Aucun effet lorsque la cause est absente dans le monde le plus similaire où la cause est présente comme l'exige l'approche contrefactuelle ».
- *cause manipulée :* « Un effet après la manipulation d'une cause ».
- *mécanismes :* « Activités et processus reliant les causes et les effets requis par l'approche mécanisme. » [Baripedia, 2014].

Plus près de nos préoccupations et objectifs dans cette thèse, une des analyses contemporaines de l'approche contrefactuelle ayant gagné en popularité est celle de David Lewis (quoique des questionnements et débats sont toujours très présents, mais ce n'est pas le lieu d'en discuter ici). En voici un résumé succinct, en ses mots : [la causalité est] « quelque

[1]Nous renvoyons à la bibliographie le lecteur désireux de consulter la source originelle.

chose qui fait une différence, et la différence qu'elle fait doit être une différence par rapport à ce qui se serait passé sans elle. » [Lewis, 1974].

1.3. Causalité et statistique

1.3.1. Les pères de l'inférence statistique moderne : Galton, Pearson et Fisher

Il n'est sans doute pas exagéré de dire que la longue recherche de Francis Galton à expliquer mathématiquement *la théorie de l'évolution* (qu'il finira par estimer non nécessaire à sa propre théorie de l'hérédité) de son demi-cousin Darwin (1859) a été le point de départ de l'inférence statistique moderne. Bien sûr, il y avait des travaux précurseurs : la loi faible des grands nombres de Jacques Bernoulli, la théorie analytique de Laplace ou la méthode des moindres carrés (Laplace, Legendre, Gauss) par exemple. Le contexte était mûr : besoins croissants de prise en compte de l'aléatoire dans des domaines multiples, développement et diffusion d'outils mathématiques adéquats, émergence de statisticiens mathématiciens, etc.

« Galton avait conçu des [...] couples de quantités comme un véritable objet statistique multivarié [...], et examiné à la fois de façon marginale et conditionnelle de n'importe quel point de vue [...]. C'était une perspective statistique radicalement nouvelle, et cela nous a donné un nouveau type de question à poser, et une nouvelle façon de penser l'association statistique : et, plus fondamentalement, une nouvelle façon de penser l'inférence. » [Stigler, 2010]. Eût-il lu les travaux de Gregor Mendel, n'aurait-il pas aussi été à l'origine de l'inférence causale moderne également ? « ... [Galton] a été sur le point de trouver le bon cadre et comment le diagramme de causalité permet de facilement se concentrer sur sa supposition erronée : la transmission du hasard d'une génération à l'autre. [...] ayant découvert la beauté de la corrélation, il en est venu à croire que la causalité n'était plus nécessaire. » [Pearl et Mackenzie, 2018].

Selon l'historien des statistiques Stephen Stigler, après Galton, Karl Pearson (disciple de Galton) et Ronald Fisher ont été les principaux architectes du nouvel édifice statistique, plus à même d'affronter les questions et défis d'un univers scientifique autrement plus vaste et complexe que celui de Laplace et Gauss : « Il y avait d'autres personnes impliquées, bien sûr : Edgeworth, Yule, Gosset et Neyman viennent à l'esprit. Mais, dans une mesure remarquable, ce sont les idées et la direction de Galton, Pearson et Fisher qui ont tracé le chemin à suivre pour le demi-siècle suivant et au-delà. Ils ont tous les trois joué des rôles

complémentaires dans cette transformation. Les brillantes intuitions de Galton des années 1880 ont changé notre façon de penser les données multivariées. Pearson a été l'architecte qui, à partir de la vision de Galton, a élaboré un plan grandiose, mettant lui-même en œuvre une part considérable de ce plan. Fisher a ensuite ajouté une dimension statistique et mathématique plus approfondies qui ont permis d'assoir la statistique comme une nouvelle discipline. Sans l'apport de chacun des trois, l'histoire des statistiques, voire de la science du 20ème siècle, seraient très différentes. » [Stigler, 2012]. Aussi (sur le défi posé par Pearson à Fisher) : « Le défi lancé par Pearson en 1916 avait porté des fruits remarquables - la théorie moderne de l'estimation [...] le cadre s'est plus élargi après son adoption par Neyman et Egon Pearson en 1928. [...] Sans la question de Pearson, il semble extrêmement improbable que Fisher ait vu comment sa surprenante découverte de l'exhaustivité était la clé de toute une théorie de l'estimation, et d'un cadre général permettant d'aborder les questions statistiques de manière plus générale. » [Stigler, 2006].

1.3.2. Bayes, Laplace et l'inférence bayésienne

Le théorème de Bayes (aussi appelé probabilité inverse) est publié à titre posthume en 1763 : *An Essay towards solving a Problem in the Doctrine of Chance* (Un essai pour résoudre un problème dans la doctrine du hasard). Il a été redécouvert et initié par Laplace en 1774 (*Mémoire sur la Probabilité des Causes par les Évènements*) :

« Si un événement peut être produit par un nombre de n causes différentes, les probabilités de l'existence de ces causes prises de l'événement, font entre elles comme les probabilités de l'événement prises de ces causes, et la probabilité de l'existence de chacune d'elles, est égale à la probabilité de l'événement prise de cette cause, divisée par la somme de toutes les probabilités de l'événement prises de chacune de ces causes. »

Laplace utilisera son théorème dans divers domaines : estimation, métrologie (science de la mesure) : « Mais il restait à déterminer la probabilité des erreurs que cette correction laisse encore à craindre : c'est ce que la méthode que je viens d'exposer fait connaître. »

Les statisticiens du début du 20ème siècle, tout comme leurs prédécesseurs depuis la mort de Laplace, étaient conscients de l'inadéquation de l'approche par la probabilité inverse : « ... qui s'appuyait sur une interprétation épistémique de la probabilité. L'une des raisons majeures à cela était le recours au principe d'indifférence pour utiliser un a priori uniforme en

conjonction avec le théorème de Bayes. [...] le principe d'indifférence peut souvent conduire à des absurdités. En dépit des problèmes associés à la probabilité inverse, il a continué à être utilisé par Edgeworth, Karl Pearson, Student, Bowley et d'autres. Même Fisher a fait référence à la probabilité inverse dans son tout premier article. » [Gorroochurn, 2016].

Fisher, puis Neyman et Egon Pearson, ont alors jeté les bases de l'analyse statistique fréquentiste (test de significativité sur l'hypothèse nulle, l'hypothèse alternative). L'information objective tirée d'échantillons aléatoires (sous certaines conditions de régularité) est à même de résoudre le problème de l'induction.

L'approche bayésienne n'a pas disparu pour autant. Parmi ses portes-parole les plus proéminents, citons Jeffreys, Ramsey, de Finetti (La Prévision : ses lois logiques, ses sources subjectives, 1937), Savage (Fondements de la statistique (Foundations of Statistics), 1954), Robbins, Jaynes (La théorie des probabilités : la logique de la science (Probability theory: the logic of science), 2003), etc. [Bacci et Chiandotto, 2019].

« ... au milieu des années 1920, Fisher a proposé des théories fréquentistes sur l'estimation et les tests de signification. Il en a résulté un déclin de la probabilité inverse. Néanmoins, malgré son déclin, elle n'est jamais morte. Ainsi, des mathématiciens comme Jeffreys ont continué à préconiser la probabilité inverse comme méthode d'inférence. Finetti a donné une nouvelle justification à la méthode bayésienne grâce à des travaux brillants, aboutissant à l'un des plus beaux théorèmes de la statistique, le théorème de représentation de Finetti. [...] ; Frank Ramsey, qui a jeté les bases du premier travail sérieux de la théorie subjective des probabilités en 1926. La théorie de Ramsey a été qualifiée à juste titre par Bernardo et Smith, ainsi que par de nombreux autres chercheurs, de *point de repère révolutionnaire dans l'histoire des idées...* » [Gorroochurn, 2016].

1.3.3. L'inférence causale

Tout étudiant dans un premier cours de statistique apprend que *corrélation ne signifie pas causalité*. Cette devise, déjà énoncée par les anciens philosophes, semble avoir commencé à prendre son sens statistique actuel avec l'étude des données de dimensions de crânes humains, non corrélées quand observées par sexe, mais corrélées lorsqu'elles sont agrégées [Pearl et Mackenzie, 2018]. Cette découverte, à la grande surprise de son auteur Pearson qui,

rappelons-le, partageait le septicisme de Hume sur la rationalité du raisonnement inductif, l'a poussé à introduire le concept de corrélation fallacieuse (*spurious correlation*) :

« Cette corrélation peut être qualifiée de fallacieuse, mais comme il est presque impossible de garantir l'homogénéité absolue de toute communauté, nos résultats de corrélation sont toujours sujets à une erreur, dont l'ampleur ne peut être prédite. Pour ceux qui persistent à considérer toute corrélation comme une cause et un effet, le fait qu'une corrélation puisse être produite entre deux caractères A et B tout à fait non corrélés en prenant un mélange artificiel de deux races étroitement alliées, doit plutôt être un choc. » (Pearson et al. (1899) cités par Stigler [2016]).

Une réponse partielle à ce problème a été fournie par Fisher avec les expériences randomisées. Ceci est assurément un cadre restreint à une question embrassant l'ensemble de l'investigation scientifique. Le statisticien y dissocie l'association entre variables de la causalité mais n'identifie pas l'effet causal et n'est pas à l'abri d'effets indésirables (biais de sélection, par exemple).

A la même période le généticien Sewall Wright a introduit son *genetic path modelling* (modélisation du parcours génétique), ancêtre de la modélisation par équations structurelles [Wright, 1920]. La tentative de Wright, quasiment morte-née, car en avance sur son temps « ... était un tour de force [...]. C'est assurément un jalon dans l'histoire de la causalité. » [Pearl et Mackenzie, 2018].

L'approche probabiliste de la causalité a pris son essor durant la seconde moitié du 20ème siècle. Elle a eu pour effet de jouer « un rôle moteur pour le renouveau de la philosophie de la causalité, et les probabilités jouent un rôle central dans le cadre de théories de la causalité développées depuis. » [Drouet, 2012].

Le débat reste à ce jour intense dans le milieu de l'apprentissage statistique [Schölkopf *et al.*, 2021]. Pour Holland [1986], en filiation avec Rubin, les expériences statistiques nous renseignent sur la causalité. Il résume sa pensée dans trois idées fortes : «

(1) L'analyse de la causalité devrait commencer par l'étude des effets des causes plutôt que par l'approche traditionnelle d'essayer de définir la cause d'un effet donné.

(2) Les effets des causes sont toujours relatifs à d'autres causes (c'est-à-dire qu'il faut deux causes pour définir un effet).

(3) Tout ne peut pas être une cause ; en particulier, les attributs des unités ne sont jamais des causes. »

D'autres, plus sceptiques, avancent que « ce n'est certainement pas un recul scientifique - bien au contraire - de reconnaître qu'il y a sans doute des questions auxquelles des données empiriques ne peuvent permettre de répondre de manière non ambiguë. » [Lecoutre, 2004].

1.3.4. En guise de conclusion

L'inférence causale est un processus complexe. Partant du principe (constat?) que les événements observés sont causés par certains éléments, elle veut évaluer l'effet d'une intervention (manipulation) éventuelle (potentielle) d'une variable sur un résultat (et de contrôle pour les variables confondantes). L'observation directe de cause à effet étant rare (lointaine dans le temps, complexité des facteurs, association, etc.), il est nécessaire de faire une inférence, d'utiliser une approche « quasi-expérimentale : quelle est la relation causale qui nous intéresse ? quelle expérimentation nous permettrait, idéalement, de saisir l'effet causal ? quelle stratégie d'identification de l'effet causal allons-nous employer ? quelle est notre mode d'inférence statistique (population, échantillon, incertitude) ? » [Baripedia, 2014]. Ces difficultés font que l'inférence causale est sujette à incertitude.

Diverses approches coexistent aujourd'hui. Citons par exemple les équations structurelles de modélisation [Dufournet, 2017], ou le paradigme graphique [Wright, 1921, Pearl, 2009, Talbot, 2015]. Dans cette thèse, nous nous concentrons sur le paradigme contrefactuel, mis au point par Neyman [1923] dans le cas des études d'expériences randomisées et généralisé ensuite par Rubin [1974] pour les données d'observation. On parle du modèle de résultats potentiels (*potential outcomes*) de Neyman-Rubin [Rosenbaum et Rubin, 1983, Holland, 1986].

Bibliographie

Silvia Bacci et Bruno Chiandotto : *Introduction to statistical decision theory: Utility theory and causal analysis*. CRC Press, 2019.

Baripedia : L'inférence causale. `https://baripedia.org/wiki/L%E2%80%99inf%C3%A9rence_causale`, 2014. Dernière consultation: le 30 janvier 2021.

Jean-Michel Besneux : *Usages de la causalité dans l'argumentation*. Thèse de doctorat, Linguistique. Normandie Université, 2017.

Henry E Brady : *Causation and explanation in social science*. Oxford Handbooks Online, 2008.

Isabelle Drouet : *Causes, probabilités, inférences*. Vuibert. Collection « Philosophie des sciences », 2012.

Isabelle Drouet : La difficulté de comprendre les causes. `https://www.franceculture.fr/emissions/continent-sciences/la-difficulte-de-comprendre-les-causes`, 2015. Continent Science, France Culture.

Marine Dufournet : *Quantification du biais de sélection en sécurité routière: apport de l'inférence causale*. Thèse de doctorat, Santé publique et épidémiologie. Université de Lyon, 2017.

Prakash Gorroochurn : *Classic topics on the history of modern mathematical statistics: From Laplace to more recent times*. John Wiley & Sons, 2016.

Paul W Holland : Statistics and causal inference. *Journal of the American Statistical Association*, 81(396):945–960, 1986.

David Hume : *Traité de la nature humaine. Livre 1 : l'entendement*. Flammarion, Paris, 1739.

David Hume : *Enquête sur l'entendement humain*. 1748.

Bruno Lecoutre : Expérimentation, inférence statistique et analyse causale. *Intellectica*, 38(1):193–245, 2004.

David Lewis : Causation. *The Journal of Philosophy*, 70(17):556–567, 1974.

Jerzy S Neyman : On the application of probability theory to agricultural experiments. *Annals of Agricultural Sciences*, 10:1–51, 1923.

Judea Pearl : *Causality: Models, Reasoning, and Inference*. New York: Cambridge University Press, 2nd édition, 2009.

Judea Pearl et Dana Mackenzie : *The book of why: the new science of cause and effect*. Basic Books, 2018.

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

Donald B Rubin : Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5):688, 1974.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal et Yoshua Bengio : Toward causal representation learning. *Proceedings of the IEEE*, 2021.

Stephen M Stigler : How Ronald Fisher became a mathematical statistician. *Mathématiques et sciences humaines. Mathematics and Social Sciences*, (176):23–30, 2006.

Stephen M Stigler : Darwin, Galton and the statistical enlightenment. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 173(3):469–482, 2010.

Stephen M Stigler : Studies in the history of probability and statistics, L: Karl Pearson and the rule of three. *Biometrika*, 99(1):1–14, 2012.

Stephen M Stigler : *The seven pillars of statistical wisdom*. Harvard University Press, 2016.

Denis Talbot : *Identification de modèles appropriés pour l'inférence causale à partir de données d'observation*. Thèse de doctorat, Mathématiques. Université du Québec à Montréal, 2015.

Sewall Wright : The relative importance of heredity and environment in determining the piebald pattern of guinea-pigs. *Proceedings of the National Academy of Sciences of the United States of America*, 6(6):320–332, 1920.

Sewall Wright : Correlation and causation. *Journal of Agricultural Research*, 20:557–585, 1921.

Chapitre 2

La modélisation *uplift*

Le but de ce chapitre est d'introduire la problématique étudiée dans cette thèse par une revue des écrits scientifiques portant sur la modélisation *uplift*. La sélection des méthodes présentées est un instantané au moment de la rédaction. Une communauté active développe constamment de nouvelles méthodes pour l'*uplift*.

2.1. Introduction

La modélisation *uplift* est une approche émergente d'apprentissage statistique pour estimer l'effet du traitement au niveau d'un individu ou d'un sous-groupe. Elle peut être utilisée pour optimiser les performances d'interventions telles que les campagnes marketing. En pratique, la modélisation *uplift* est utilisée pour prédire quels individus sont le plus susceptibles de bénéficier d'un traitement, puis hiérarchiser la promotion de l'expérience préférée auprès de ces individus. Ce type d'informations est utile pour concevoir et offrir une expérience personnalisée afin d'améliorer l'expérience utilisateur, la satisfaction et l'engagement. Bien que le terme *uplift* ait été introduit dans le contexte du marketing, la méthodologie statistique associée est totalement liée à celle de l'inférence causale. Les exemples suivants soulignent la diversité des domaines d'application des modèles *uplift* [Michel *et al.*, 2019] :

- Le marketing direct tente de convaincre les clients d'acheter un produit ou un service.
- Les campagnes de prévention de l'attrition (ou *churn* en anglais) renforcent ou fidélisent les clients.
- Certains traitements médicaux sont appliqués afin d'aider les patients à se remettre d'une maladie ou à soulager la douleur.
- Les engrais sont utilisés pour augmenter les rendements en agriculture.

- La maintenance préventive est utilisée pour éviter le dysfonctionnement d'une machine.
- Les forces de police sont utilisées de manière préventive pour éviter les crimes, en particulier les cambriolages.

Certains de ces traitements, en particulier lorsqu'ils tentent d'influencer le comportement humain, peuvent être caractérisés comme des *coups de pouce* [Thaler et Sunstein, 2009], c'est-à-dire des traitements qui stimulent doucement les individus à leur avantage sans les priver de leur liberté de décision. L'un des objectifs des méthodes présentées dans cette thèse est de rendre l'effet de tels coups de pouce à la fois mesurable et prévisible [Michel *et al.*, 2019].

2.2. Le paradigme contrefactuel

L'estimation de l'*uplift* est un cas particulier de l'estimation de l'effet causal du traitement pour lequel le cadre statistique le plus utilisé est celui des contrefactuels. Le paradigme contrefactuel, également appelé paradigme des résultats potentiels, a d'abord été développé par Neyman [1923] pour étudier des expériences randomisées. Une généralisation permettant d'étudier les liens causaux avec des données d'observation a ensuite été réalisée par Rubin [1974].

Le plus souvent, l'ampleur de l'effet que le traitement exerce sur le comportement est seulement supposée mais pas exactement connue à l'avance. Cependant, une estimation ultérieure de l'effet est possible dans les cas suivants :

(1) Si des facteurs d'influence supplémentaires peuvent être exclus (par exemple, au moyen d'un plan expérimental), le comportement avant et après le traitement peut être comparé.

(2) S'il existe un groupe d'observations structurellement identiques non exposé au traitement (c'est-à-dire le groupe contrôle), le comportement du groupe d'observations qui a reçu le traitement peut être comparé au comportement du groupe contrôle. Ceci est considéré comme l'approche standard dans cette thèse.

2.2.1. Notation

Soient Y la variable dépendante correspondant au résultat étudié et T la variable indicatrice de traitement. Dans le cas que nous étudions, la variable T est binaire et indique si une unité est exposée au traitement ($T = 1$) ou au contrôle ($T = 0$). On note $X_1, \ldots, X_p$ les variables aléatoires explicatives. On note également par l'indice $i = 1, \ldots, n$ les unités échantillonnées et on utilise les lettres minuscules y, t et x pour représenter les réalisations des variables aléatoires Y, T et X. Enfin, on notera par $\boldsymbol{X} = (X_1, \ldots, X_p)$ le vecteur de variables aléatoires explicatives et $\mathbf{x} = (x_1, \ldots, x_p)$, le vecteur de ses réalisations.

L'inférence *associative* consiste en l'exploration d'associations entre Y et un sous-ensemble de $\boldsymbol{X}$. On distingue Y de $\boldsymbol{X}$ en appelant le dernier *attributs* des unités de la population U, mais, logiquement Y et $\boldsymbol{X}$ sont toutes des variables observées en étudiant une unité expérimentale de U. Les *paramètres associatifs* sont déterminés par la distribution conjointe de Y et $\boldsymbol{X}$. Par exemple, la distribution conditionnelle de Y sachant X_1 décrit comment la distribution des valeurs de Y change sur U lorsque X_1 varie. Un paramètre associatif typique est la régression de Y sur X_1, c'est-à-dire l'espérance conditionnelle $\mathbb{E}[Y \mid X_1 = x_1]$. L'inférence associative consiste à faire des inférences statistiques, telles que l'estimation, le test d'hypothèses statistiques, la mise à jour des distributions postérieures, etc. Ces inférences concernent les paramètres associatifs qui relient Y à $\boldsymbol{X}$ sur la base des données recueillies à partir des unités expérimentales en U.

En plus de l'inférence associative, les unités en U pour l'inférence causale sont les objets d'intérêt sur lesquels les causes peuvent agir. Les termes *cause*, *action* et *traitement* sont interchangeables. Dans le cadre d'expériences randomisées, la valeur de T est contrôlée pour chaque unité, les unités sont assignées au hasard au traitement ou au contrôle. C'est la première différence entre l'inférence associative et causale en ce sens que les valeurs de $\boldsymbol{X}$ ne sont pas attribuées au hasard. Ces attributs indiquent une caractéristique de l'unité alors que T indique une exposition au traitement ou au contrôle. Une autre différence importante entre les deux types d'inférence est le rôle du temps. Dans l'inférence associative, le rôle du temps n'est qu'une partie de la définition de la population d'unités faisant l'objet de l'étude. Au contraire, dans l'inférence causale, le rôle du temps est associé à l'exposition à une cause. Cette exposition doit se produire à un moment précis ou dans une période de temps. Certaines variables sont mesurées avant l'exposition (*pré-exposition*) et d'autres variables

sont mesurées après l'exposition (*post-exposition*). La variable réponse Y est censée mesurer l'effet d'une cause. Ainsi, Y doit être une variable post-exposition. En d'autres termes, la valeur de Y est le *résultat potentiel* associé à une cause particulière, $T = 1$ ou $T = 0$. L'originalité ici est de considérer que chacun a virtuellement deux résultats potentiels, ou contrefactuels : i) le résultat potentiel correspondant à l'exposition, noté $Y(1)$; et ii) le résultat potentiel correspondant à aucune exposition, noté $Y(0)$. Le défi en matière de modélisation est que bien que chaque unité soit associée à deux résultats potentiels, un seul d'entre eux peut être réalisé comme résultat observé. Cela rend donc impossible les observations directes de l'autre condition (la condition contrefactuelle) et donc des effets causaux au niveau individuel. Dans la littérature, ce dilemme est connu sous l'appellation du problème fondamental de l'inférence causale [Holland, 1986].

2.2.2. Suppositions classiques

Pour que de telles quantités aient un sens, il est nécessaire de faire quelques suppositions, classiques dans le monde de l'inférence causale [Rubin, 1974, Rosenbaum et Rubin, 1983]. La première est que chaque unité a des probabilités non nulles d'être exposée ou non exposée. Formellement, il y a une *positivité* conditionnelle (ou *overlap* en anglais) si

$$0 < \Pr(T = 1 \mid \boldsymbol{X} = \mathbf{x}) < 1, \ \forall \ \mathbf{x}. \tag{2.2.1}$$

Dans la littérature, la quantité $\Pr(T = 1 \mid \boldsymbol{X} = \mathbf{x})$ représente le score de propension (ou *propensity score* en anglais). Notons-le par $e(\mathbf{x})$.

On suppose également que l'issue factuelle, c'est-à-dire l'issue réellement observée, est la même que l'issue potentielle correspondant au niveau d'exposition observé. Formellement, on dit qu'il y a *cohérence* (ou *consistency* en anglais) si

$$Y = TY(1) + (1 - T)Y(0). \tag{2.2.2}$$

Bien que la cohérence semble évidente, c'est une supposition, et non un fait. Nous pouvons nous attendre à ce que la supposition de cohérence se maintienne dans le cas d'une intervention bien définie, c'est-à-dire que le traitement est aléatoire, non un attribut de l'unité [Holland, 1986], et qu'il n'y a pas d'interférence causale, c'est-à-dire que le résultat potentiel d'une unité n'est pas affecté par le fait qu'une autre unité ait été traitée. En effet, il est également courant de faire la supposition que le niveau d'exposition reçu par une

unité n'affecte pas le résultat des autres unités et qu'il n'existe pas de versions multiples de chaque niveau d'exposition. La structure ne serait pas adéquate lorsque, par exemple, la réponse de l'unité i au traitement t dépend du traitement donné à l'unité j. La supposition selon laquelle il y a une valeur unique $Y_i(t)$ correspondant à l'unité i et au traitement t a été appelée *supposition de valeur de traitement unitaire stable* (*stable unit-treatment value assumption* en anglais ou SUTVA) [Rosenbaum et Rubin, 1983]. Cette supposition exclut des cas importants. Par exemple, une campagne de vaccination a un impact sur la propagation de la maladie et peut donc indirectement protéger des personnes qui n'ont pas été vaccinées. Le traitement a donc une valeur ou une utilité différente pour une unité donnée en fonction du traitement reçu par les autres unités [Talbot, 2015].

Lorsque les données proviennent d'une expérience randomisée, la variable aléatoire T est indépendante de toutes les caractéristiques de pré-exposition, c'est-à-dire,

$$Y(0), Y(1), \boldsymbol{X} \perp\!\!\!\perp T, \tag{2.2.3}$$

où $\perp\!\!\!\perp$ dénote l'indépendance statistique. En inférence causale, (2.2.3) est connue sous la *supposition forte d'ignorabilité*. Par contre, lorsque l'allocation du traitement n'est pas aléatoire, différents facteurs peuvent influencer à la fois le niveau d'exposition des sujets et leur réponse potentielle. Il est donc commun de considérer la supposition plus faible d'indépendance conditionnelle de variables aléatoires, souvent désignée par la *supposition faible d'ignorabilité* (ou *unconfoundedness* en anglais) dans la littérature, c'est-à-dire,

$$Y(0), Y(1) \perp\!\!\!\perp T \mid \boldsymbol{X}. \tag{2.2.4}$$

Dans ce cas, on dira qu'il y a *absence de confusion* pour la relation causale entre T et Y. Les suppositions énoncées ci-dessus sont typiques dans tous travaux liés à l'inférence causale parce qu'elles permettent l'estimation sans biais de l'effet causal moyen du traitement (voir Proposition 2.2.1).

2.2.3. Définition de l'effet causal

L'effet du traitement individuel (*Individual Treatment Effect* ou ITE en anglais) est défini comme suit :

$$\tau_i = Y_i(1) - Y_i(0). \tag{2.2.5}$$

Maintenant, bien que l'ITE lui-même soit fondamentalement non-observable [Holland, 1986], nous pouvons (peut-être remarquablement) utiliser des expériences aléatoires pour apprendre certaines propriétés de l'ITE. En particulier, de grandes expériences randomisées nous permettent de récupérer l'effet moyen du traitement (*Average Treatment Effect* ou ATE en anglais). En effet, l'estimation d'intérêt la plus courante est l'effet du traitement s'il était étendu à l'ensemble de la population, soit formellement :

$$\tau = \mathbb{E}[Y_i(1) - Y_i(0)]. \tag{2.2.6}$$

En l'absence de facteurs de confusion, l'estimation de l'ATE par différence de moyennes entre les groupes de traitement et de contrôle est sans biais. Supposons que nous disposons d'un échantillon de n observations indépendantes et identiquement distribuées (iid) $(Y_i, T_i, \boldsymbol{X}_i)$ provenant d'une expérience randomisée. L'estimateur par différence de moyennes est défini par

$$\hat{\tau} = \frac{1}{n_1} \sum_{\{i:T_i=1\}} Y_i - \frac{1}{n_0} \sum_{\{i:T_i=0\}} Y_i, \tag{2.2.7}$$

où $n_t = |\{i : T_i = t\}|$, $n_t > 0$ et $n_0 + n_1 = n$.

Proposition 2.2.1. *Assumons que les suppositions* (2.2.2) *et* (2.2.3) *soient vérifiées. Alors* $\hat{\tau}$ *est sans biais pour* τ*.*

Démonstration. En notant que pour $t \in \{0,1\}$,

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{n_t} \sum_{\{i:T_i=t\}} Y_i\right] &= \mathbb{E}[Y_i \mid T_i = t] \text{ (iid)} \\
&= \mathbb{E}[Y_i(t) \mid T_i = t] \text{ (par la supposition (2.2.3))} \\
&= \mathbb{E}[Y_i(t)] \text{ (par la supposition (2.2.2))}
\end{aligned}$$

nous avons $\mathbb{E}[\hat{\tau}] = \mathbb{E}[Y_i(1)] - \mathbb{E}[Y_i(0)] = \tau$. □

En pratique, l'effet du traitement est susceptible de dépendre des caractéristiques des personnes. Par exemple, les jeunes chômeurs peuvent davantage bénéficier d'un programme de formation que des chômeurs plus âgés (ou inversement) [Givord, 2014]. Ainsi, il peut être pertinent d'estimer des effets moyens, à caractéristiques observables données (l'âge par exemple). Ceci permet de mesurer l'hétérogénéité de l'effet du traitement, soit l'objectif principal de cette thèse. En pratique, cela revient à estimer des effets causaux au niveau

individuel, ou plus spécifiquement, des effets moyens conditionnels aux caractéristiques individuelles observables, soit l'effet causal conditionnel (*Conditional Averaged Treatment Effect* ou CATE en anglais). Le CATE est défini comme suit :

$$\tau(\mathbf{x}) = \mathbb{E}[Y_i(1) - Y_i(0) \mid \boldsymbol{X}_i = \mathbf{x}]. \tag{2.2.8}$$

Dans la littérature, les termes ITE, CATE et *uplift* réfèrent souvent à la même quantité, soit le CATE. En effet, l'*uplift* est un cas particulier du CATE lorsque la variable dépendante Y est binaire $0-1$. Ainsi, l'*uplift* est défini comme l'effet du traitement moyen conditionnel dans différentes sous-populations en fonction des valeurs possibles des covariables, soit :

$$u(\mathbf{x}) = \Pr(Y_i = 1 \mid \boldsymbol{X}_i = \mathbf{x}, T_i = 1) - \Pr(Y_i = 1 \mid \boldsymbol{X}_i = \mathbf{x}, T_i = 0), \tag{2.2.9}$$

où (2.2.8) se simplifie à (2.2.9) lorsqu'il y a absence de confusion et que la variable réponse Y est binaire $0-1$ (voir Proposition 2.2.2).

Proposition 2.2.2. *Soient Y la variable dépendante binaire $0-1$ correspondant au résultat étudié et T la variable aléatoire binaire indicatrice de l'allocation du traitement. Soit $\boldsymbol{X}$ le vecteur de variables aléatoires explicatives. Assumons que les suppositions* (2.2.2) *et* (2.2.4) *soient vérifiées. Alors*

$$\tau(\mathbf{x}) = u(\mathbf{x}).$$

Démonstration.

$$\begin{aligned}
\tau(\mathbf{x}) &= \mathbb{E}[Y_i(1) - Y_i(0) \mid \boldsymbol{X}_i = \mathbf{x}] \\
&= \mathbb{E}[Y_i(1) \mid \boldsymbol{X}_i = \mathbf{x}] - \mathbb{E}[Y_i(0) \mid \boldsymbol{X}_i = \mathbf{x}] \text{ (par linéarité de l'espérance)} \\
&= \mathbb{E}[Y_i(1) \mid \boldsymbol{X}_i = \mathbf{x}, T_i = 1] - \mathbb{E}[Y_i(0) \mid \boldsymbol{X}_i = \mathbf{x}, T_i = 0] \text{ (par la supposition (2.2.4))} \\
&= \mathbb{E}[Y_i \mid \boldsymbol{X}_i = \mathbf{x}, T_i = 1] - \mathbb{E}[Y_i \mid \boldsymbol{X}_i = \mathbf{x}, T_i = 0] \text{ (par la supposition (2.2.2))} \\
&= \Pr(Y_i = 1 \mid \boldsymbol{X}_i = \mathbf{x}, T_i = 1) - \Pr(Y_i = 1 \mid \boldsymbol{X}_i = \mathbf{x}, T_i = 0) \text{ (car } Y \text{ est binaire } 0-1) \\
&= u(\mathbf{x})
\end{aligned}$$

□

Il existe différentes méthodes pour estimer les effets causaux conditionnels. Le processus d'estimation de ces effets est également appelé modélisation des effets hétérogènes, modélisation des effets du traitement individuel ou bien modélisation *uplift*. En pratique, le but de la modélisation *uplift* est de séparer les individus en sous-groupes hétérogènes ordonnés en terme d'*uplift* afin de prédire le changement de comportement de futurs individus induit par le traitement. Pour la suite, nous assumons que toutes les suppositions classiques énoncées dans la section 2.2.2 sont vérifiées et que nous disposons d'un échantillon de n observations iid provenant d'une expérience randomisée.

Remarque 2.2.3. *Bien qu'il soit possible de définir d'autres quantités causales, par exemple le risque relatif causal $Y_i(1)/Y_i(0)$, il est plus commun de travailler avec l'effet causal défini par la différence entre les résultats potentiels. Nous avons présenté les effets causaux d'intérêts les plus populaires et les plus cohérents avec le sujet de la thèse.*

2.3. Estimation indirecte

Par rapport à l'inférence associative, la difficulté de l'inférence causale est que la véritable variable à expliquer n'est pas définie pour une observation individuelle. En effet, il est commun de mesurer l'impact d'un traitement par une comparaison de groupes d'observations structurellement identiques qui ont (groupe traitement) ou n'ont pas (groupe contrôle) reçu le traitement. Par conséquent, les méthodes classiques ne sont pas directement applicables, mais elles présentent la base. Les approches de modélisation présentées dans les sous-sections suivantes ne fournissent pas une estimation directe de l'*uplift*, mais $\Pr(Y_i = 1 \mid \boldsymbol{X}_i = \mathbf{x}, T_i = 1)$ et $\Pr(Y_i = 1 \mid \boldsymbol{X}_i = \mathbf{x}, T_i = 0)$ sont plutôt modélisées séparément. Leur différence est ensuite utilisée comme une estimation de $u(\mathbf{x})$. Il est procédé de la même façon pour l'estimateur utilisant la variable réponse transformée de la Section 2.3.3 où il s'agit d'abord d'estimer une probabilité, de calculer ensuite une fonction de la probabilité estimée afin d'obtenir l'*uplift*.

2.3.1. Deux modèles indépendants

L'approche la plus intuitive consiste à créer deux modèles indépendants. Cela consiste à ajuster deux modèles de probabilité conditionnelle séparés : un pour les individus traités et un autre pour les individus non traités.

Dans Hansotia and Rukstales [2001], la méthode proposée est de construire une régression logistique pour chaque groupe. Évidemment, il est également possible d'utiliser des arbres de classification, des réseaux de neurones ou toute autre méthode permettant la modélisation d'une probabilité conditionnelle. Formellement, pour le cas de Hansotia and Rukstales [2001], soient

$$\Pr(Y_i = 1 \mid \mathbf{x}_i, T_i = 1, \beta_o^{(1)}, \boldsymbol{\beta}^{(1)}) = \left(1 + \exp\{-(\beta_o^{(1)} + \mathbf{x}_i^\top \boldsymbol{\beta}^{(1)})\}\right)^{-1}$$

et

$$\Pr(Y_i = 1 \mid \mathbf{x}_i, T_i = 0, \beta_o^{(0)}, \boldsymbol{\beta}^{(0)}) = \left(1 + \exp\{-(\beta_o^{(0)} + \mathbf{x}_i^\top \boldsymbol{\beta}^{(0)})\}\right)^{-1},$$

les deux modèles indépendants où $(\beta_o^{(t)}$,$\boldsymbol{\beta}^{(t)})$ pour $t \in \{0,1\}$ représentent les paramètres de la régression logistique pour les groupes contrôle ($t = 0$) et traitement ($t = 1$), et l'exposant $^\top$ indique la transposition.

Chaque observation est ensuite évaluée par les deux modèles et l'*uplift* est estimée comme la différence entre ces deux modèles. La différence de ces prédictions représente la valeur ajoutée du traitement ou l'impact net du traitement. Ainsi, pour un futur individu (associé à un vecteur de covariables $\mathbf{x}_{n+1}$), l'*uplift* prédit est simplement

$$\hat{u}(\mathbf{x}_{n+1}) = \left(1 + \exp\{-(\hat{\beta}_o^{(1)} + \mathbf{x}_{n+1}^\top \hat{\boldsymbol{\beta}}^{(1)})\}\right)^{-1} - \left(1 + \exp\{-(\hat{\beta}_o^{(0)} + \mathbf{x}_{n+1}^\top \hat{\boldsymbol{\beta}}^{(0)})\}\right)^{-1},$$

où $(\hat{\beta}_o^{(t)}$,$\hat{\boldsymbol{\beta}}^{(t)})$ pour $t \in \{0,1\}$ sont estimés par maximum de vraisemblance.

L'atout de cette technique est sa simplicité. Le modèle construit sur le groupe des individus traités représente l'impact du traitement sur les observations et attribue des scores plus élevés à ceux qui semblent répondre positivement au traitement. En revanche, le modèle construit sur les observations non traitées représente un bruit aléatoire. Les observations répondant sans traitement obtiennent des scores plus élevés que celles qui ne réagissent pas. Cependant, il ne fonctionne pas bien en pratique [Radcliffe and Surry, 2011]. Une explication plausible est que les deux modèles se concentrent sur la prédiction d'une probabilité, et les informations sur l'autre traitement ne sont jamais fournies à l'autre modèle. De plus, le calcul de la différence entre deux modèles ne signifie pas automatiquement que cette différence sera également maximisée ou minimisée afin de créer des groupes hétérogènes en ce qui concerne l'*uplift,* surtout si le bruit aléatoire est beaucoup plus important que l'effet du traitement. Dans ce cas, le modèle pour le groupe d'individus traités sera plus susceptible de prédire le bruit aléatoire au lieu de l'*uplift.* De plus, les différences entre les distributions de covariables

dans les deux groupes peuvent entraîner un biais dans l'estimation de l'effet du traitement. Enfin, les variances des modèles indépendants s'additionnent et, par conséquent, conduisent à une variance plus élevée de l'*uplift.*

2.3.2. L'estimateur du modèle d'interaction

L'utilisation d'un modèle unique corrige quelques inconvénients de l'estimateur à deux modèles. La méthode décrite dans cette section peut être considérée comme une estimation indirecte de l'*uplift* utilisant un modèle unique. Dans le modèle de Lo [2002], l'assignation de traitement est utilisée comme une caractéristique (silencieuse) du modèle, ajoutant des termes d'interaction explicites entre chaque covariable et la variable indicatrice de traitement afin d'ajuster une régression logistique. Les paramètres des termes d'interactions mesurent l'effet supplémentaire de chaque covariable en raison du traitement. Formellement, le modèle est défini comme suit,

$$\Pr(Y_i = 1 \mid \mathbf{x}_i, t_i, \theta_o, \boldsymbol{\theta}) = \Big(1 + \exp\{-(\theta_o + \mathbf{x}_i^\top \boldsymbol{\beta} + \gamma t_i + t_i \mathbf{x}_i^\top \boldsymbol{\delta})\}\Big)^{-1},$$

où $\boldsymbol{\theta} = (\boldsymbol{\beta}, \gamma, \boldsymbol{\delta})$. Ici, θ_o désigne l'ordonnée à l'origine, γ désigne l'effet du traitement, $\boldsymbol{\beta}$ est le vecteur des effets principaux et $\boldsymbol{\delta}$ est le vecteur des effets d'interaction.

Notons les estimations par maximum de vraisemblance de $(\theta_o, \boldsymbol{\theta})$ par $(\hat{\theta}_o, \hat{\boldsymbol{\theta}})$, avec $\hat{\boldsymbol{\theta}} = (\hat{\boldsymbol{\beta}}, \hat{\gamma}, \hat{\boldsymbol{\delta}})$. L'*uplift* prédit associé au vecteur de covariables $\mathbf{x}_{n+1}$ d'un futur individu est donné par

$$\hat{u}(\mathbf{x}_{n+1}) = \Big(1 + \exp\{-(\hat{\theta}_o + \mathbf{x}_{n+1}^\top \hat{\boldsymbol{\beta}} + \hat{\gamma} + \mathbf{x}_{n+1}^\top \hat{\boldsymbol{\delta}})\}\Big)^{-1} - \Big(1 + \exp\{-(\hat{\theta}_o + \mathbf{x}_{n+1}^\top \hat{\boldsymbol{\beta}})\}\Big)^{-1}.$$

Pour le modèle de Lo [2002], la variance par rapport à deux modèles indépendants est réduite et l'homogénéité des observations aux propriétés similaires est conservée indépendamment du fait qu'elles aient été traitées ou non [Michel *et al.*, 2019]. Les paramètres peuvent être estimés facilement et leur interprétation est inchangée par rapport à une régression logistique classique. Cependant, il y a encore un inconvénient : la solution n'est pas optimisée pour la recherche de groupes hétérogènes en fonction de l'*uplift.* En outre, il n'est pas évident de savoir comment régler un tel modèle pour corriger cela.

2.3.3. La variable réponse transformée

La méthode de la variable réponse transformée a été introduite par Jaskowski and Jaroszewicz [2012]. La méthode consiste à créer la variable aléatoire suivante,

$$W_i = T_i Y_i + (1 - T_i)(1 - Y_i), \tag{2.3.1}$$

pour toutes les observations $i = 1, \ldots, n$. Étant donné que la variable réponse Y est binaire $0 - 1$, la nouvelle variable W s'écrit :

$$W = \begin{cases} 1, & \text{si } T = 1, Y = 1 \\ 1, & \text{si } T = 0, Y = 0 \\ 0 & \text{sinon.} \end{cases}$$

L'interprétation de la variable réponse transformée est intéressante. Lorsque W vaut 1, cela indique une réaction dans le groupe de traitement ou une non-réaction dans le groupe contrôle, c'est-à-dire une observation qui permet d'augmenter l'*uplift*. Dans l'hypothèse où les groupes contrôle et traitement sont équilibrés sur tous les profils d'individus, c'est-à-dire $e(\mathbf{x}) = 1/2$ pour tout $\mathbf{x}$, il est facile de montrer que

$$u(\mathbf{x}) = 2\Pr(W = 1 \mid \boldsymbol{X} = \mathbf{x}) - 1.$$

Par conséquent, le problème de la modélisation *uplift* est réduit à la modélisation d'une variable réponse binaire W. Ainsi, toute technique de classification binaire peut être appliquée. Une généralisation à l'assignation de traitement déséquilibrée et aux configurations de régression peut être empruntée à Athey and Imbens [2015] qui proposent d'estimer l'*uplift* en utilisant la variable réponse transformée suivante,

$$Z_i = \frac{Y_i T_i}{e(\mathbf{x}_i)} - \frac{Y_i(1 - T_i)}{1 - e(\mathbf{x}_i)}, \tag{2.3.2}$$

pour toutes les observations $i = 1, \ldots, n$. Cette variable transformée a la propriété clé que, sous la supposition faible d'ignorabilité (2.2.4), son espérance conditionnelle à $\boldsymbol{X} = \mathbf{x}$ est égale à l'*uplift*. Cette propriété implique que tout estimateur cohérent de $\mathbb{E}[Z \mid \boldsymbol{X} = \mathbf{x}]$ est également un estimateur cohérent de $u(\mathbf{x})$ [Gutierrez and Gérardy, 2017]. Par contre, la même critique que pour l'estimateur du modèle d'interaction est applicable ici, puisque la méthode n'est pas optimisée pour modéliser l'*uplift* directement, mais elle modélise plutôt des taux de réaction à l'aide d'une variable auxiliaire [Michel *et al.*, 2019].

Il existe d'autres méthodes d'estimation indirecte de l'*uplift* dans la littérature. La méthode de Tian *et al.* [2014] est une variante du modèle de Lo [2002]. Dans cette approche, au lieu de transformer la variable réponse, la caractéristique clé est une transformation appropriée des variables explicatives. Dans le contexte de la régression logistique avec $e(\mathbf{x}) = 1/2$, il est possible de montrer l'équivalence entre la méthode de transformation de covariables et celle utilisant la variable réponse transformée W [Guelman et al., 2015]. D'autres méthodes suggèrent des transformations de la variable réponse afin d'adapter au cas *uplift* les algorithmes de séparateurs à vaste marge (*support vector machine* ou SVM en anglais). Citons par exemple les travaux de Imai *et al.* [2013], Zaniewicz and Jaroszewicz [2013] ou Kuusisto et al. [2014]. Imai *et al.* [2013] déclarent que selon leur expérience, les prédictions de leur modèle pourraient ne pas être interprétables comme des probabilités (c'est-à-dire, > 1 ou < 0) et, par conséquent, les différences non plus, mais l'ordre des observations par les *uplift* prédits donnerait tout de même des résultats très utiles. Nous ne présentons pas ces méthodes mais nous renvoyons le lecteur désireux d'avoir plus d'informations vers les références ou revues sur le sujet [Devriendt et al., 2018, Michel *et al.*, 2019].

2.4. Estimation directe

Comme cela a été exposé ci-dessus, deux modèles indépendants n'optimisent pas nécessairement le modèle par rapport à la cible souhaitée, c'est-à-dire l'*uplift*. En conséquence, les groupes traités et non-traités doivent être considérés simultanément dans un seul modèle. Pour cette tâche, cependant, les méthodes existantes doivent être ajustées, car elles ne sont pas conçues de manière appropriée. Afin de déduire l'*uplift* de manière directe, il est donc nécessaire de modifier les modèles existants. La plupart des recherches actives actuelles vont dans ce sens. Ces méthodes sont principalement l'adaptation de deux types de modèles non paramétriques : a) les arbres de décisions [Hansotia and Rukstales, 2002, Radcliffe and Surry, 2011, Guelman et al., 2012, Sołtys et al., 2015, Rzepakowski and Jaroszewicz, 2010]; et b) la méthode des k plus proches voisins [Alemi et al., 2009, Su et al., 2012]. Les méthodes les plus populaires dans la littérature restent des adaptations des arbres de classification et de régression (ou CART) [Breiman et al., 1984]. Dans cette section, nous nous concentrons sur ces méthodes et discutons de la tâche principale de la génération d'arbres de décision : le critère de segmentation.

2.4.1. Les arbres *uplift*

Contrairement à d'autres techniques de modélisation, l'ajustement d'un arbre de décision permet à chaque itération de partitionner l'échantillon de manière unique. Cela signifie que chaque segmentation peut être immédiatement vérifiée par rapport à l'impact du traitement. Étant donné que le but de la modélisation *uplift* est de trouver une partition en sous-groupes de la population, il semble naturel d'utiliser un arbre de décision comme méthode de choix. Une fois l'arbre construit, chaque feuille contient une prédiction de l'*uplift*, c'est-à-dire la différence de proportions de réponses positives entre les groupes traitement et contrôle dans la feuille. Cette simple modification permet donc d'estimer l'*uplift* directement. Ensuite, sur la base de ces valeurs, le comportement des observations vis-à-vis du traitement peut être prédit.

L'inconvénient de cette méthode est assez évident. En effet, dans l'algorithme CART d'origine, lorsque la variable dépendante est binaire, les critères de segmentation les plus largement adoptés sont l'entropie de Shannon, l'indice de diversité de Gini et leurs variantes. Ainsi, la recherche des prédicteurs et de leurs points de segmentation est optimisée en fonction de la réponse Y et non de l'*uplift*. Comme le montre souvent la pratique, la probabilité de réponse et l'*uplift* ont des facteurs assez différents en termes de prédicteurs et, par conséquent, cette méthode n'est pas optimisée pour la question à l'étude [Radcliffe and Surry, 2011]. Dans les cas extrêmes, tous les *uplift* dans les feuilles pourraient être les mêmes (mais pas les probabilités de réponse positive). Cette méthode souffre donc du même problème que les méthodes indirectes.

Lors de la construction d'un arbre de décision, le critère de segmentation déterminant les nœuds et les feuilles doit être optimisé pour produire différents *uplift*, ce qui est effectué par les méthodes suivantes.

2.4.2. Critères de segmentation

Dans Radcliffe and Surry [1999, 2011] ou Hansotia and Rukstales [2002, 2001], des critères de segmentation modifiés qui convenaient à l'objectif *uplift* ont été étudiés. La méthode la plus intuitive proposée par Hansotia and Rukstales [2002] utilise la différence absolue des *uplift* observés, c'est-à-dire,

$$\Delta = |u_l - u_r|, \tag{2.4.1}$$

où u_l, u_r représentent les *uplift* observés respectivement dans les nœuds enfants gauche et droit. Cependant, elle souffre du problème de surestimation de certaines segmentations car le nombre d'observations dans les feuilles résultantes n'est pas pris en compte dans le critère (2.4.1). Il est possible d'utiliser la différence de taille des nœuds comme une sorte de terme de pénalité pour ajuster la différence brute Δ [Radcliffe and Surry, 2011]. Si les deux tailles de nœuds enfants sont n_l et n_r, une forme pénalisée pourrait être

$$\Delta / \Big(\frac{n_l + n_r}{2 \min\{n_l, n_r\}} \Big)^k, \tag{2.4.2}$$

pour un hyper-paramètre donné $k > 0$, qui est défini de manière heuristique. Ce critère se réduit à Δ si $n_l = n_r$.

D'autres critères proposés dans la littérature sont basés sur la statistique du χ^2 [Su et al., 2009, Radcliffe and Surry, 2011], qui est généralement une fonction de Δ^2. Les critères introduits ci-dessus reposent sur la maximisation de l'hétérogénéité des effets du traitement (Δ) et évaluent les qualités de tous les points de segmentation potentiels. Le modèle de partitionnement binaire récursif effectue ensuite la segmentation au point ayant le score le plus élevé.

Dans une autre direction, Rzepakowski and Jaroszewicz [2010] proposent des arbres de décision *uplift* basés sur une mesure de dissimilarité entre deux distributions de probabilités. L'approche est basée sur l'idée de comparer les distributions de réponses dans les groupes traitement et contrôle. Le critère de segmentation compare cette mesure avant et après la segmentation. Avant d'introduire ce critère, afin de clarifier la notation, nous présentons brièvement une mesure de «distance» entre distributions finies.

En théorie des probabilités et en théorie de l'information, la divergence de Kullback-Leibler (ou entropie relative) est une mesure de dissimilarité entre deux distributions de probabilités [Kullback et Leibler, 1951]. Pour deux distributions de probabilités discrètes $P = (p_1,...,p_S)$ et $Q = (q_1,...,q_S)$, la divergence de Kullback–Leibler de P par rapport à Q est définie par

$$D_{\mathrm{KL}}(P||Q) = \sum_{s=1}^{S} p_s \log \Big(\frac{p_s}{q_s} \Big). \tag{2.4.3}$$

Par définition, $D_{\mathrm{KL}}(P||Q) \geq 0$ et $P = Q$ si et seulement si $D_{\mathrm{KL}}(P||Q) = 0$. L'idée est donc de prendre P comme la distribution de la variable réponse Y dans le groupe traitement ($T = 1$) et Q comme la distribution de Y dans le groupe contrôle. Ici, dans le cas où Y est binaire, $S = 2$. Maintenant, pour n'importe quel nœud, supposons qu'il existe une segmentation possible ω au point $X = x$ qui crée deux nœuds enfants $l = (X < x)$ et $r = (X \geq x)$, désignant respectivement les nœuds gauche et droit. En outre, soit n le nombre total de sujets dans le nœud parent. Ainsi, conditionnellement à la segmentation ω, la divergence peut être exprimée comme la divergence KL au sein de chaque nœud enfant, pondérée par la proportion d'observations dans chaque nœud, c'est-à-dire,

$$D_{\mathrm{KL}}(P||Q \mid \omega) = \frac{1}{n} \sum_{j \in \{l,r\}} n_j D_{\mathrm{KL}}(P||Q \mid j). \tag{2.4.4}$$

Le critère de segmentation $D_{\mathrm{KL}_{\mathrm{gain}}}$ de Rzepakowski and Jaroszewicz [2010] est défini comme l'augmentation en terme d'entropie relative à partir d'une segmentation, par rapport à l'entropie relative dans le nœud parent, c'est-à-dire,

$$D_{\mathrm{KL}_{\mathrm{gain}}}(\omega) = D_{\mathrm{KL}}(P||Q \mid \omega) - D_{\mathrm{KL}}(P||Q). \tag{2.4.5}$$

En fin de compte, $D_{\mathrm{KL}_{\mathrm{gain}}}(\omega)$ est calculé pour toutes les segmentations pertinentes ω, et celle avec un $D_{\mathrm{KL}_{\mathrm{gain}}}(\omega)$ maximal est choisie. Le même principe est répliqué pour d'autres mesures de «distance» [Rzepakowski and Jaroszewicz, 2010], soit les divergences de type euclidien, χ^2 et L_1 définies respectivement par

$$D_{\mathrm{E}}(P||Q) = \sum_{s=1}^{S} (p_s - q_s)^2, \tag{2.4.6}$$

$$D_{\chi^2}(P||Q) = \sum_{s=1}^{S} \frac{(p_s - q_s)^2}{q_s}, \tag{2.4.7}$$

$$D_{L_1}(P||Q) = \sum_{s=1}^{S} |p_s - q_s|. \tag{2.4.8}$$

2.4.3. Les forêts aléatoires

Les critères de segmentation présentés ci-dessus forment la base des arbres *uplift.* En pratique, l'utilisation d'un seul arbre permet l'interprétation et la visualisation du modèle, mais ne garantit pas la meilleure performance du point de vue prédiction. Ainsi, il est commun de construire des modèles plus sophistiqués, en combinant plusieurs modèles de base différents, ajustés avec des conditions de départ légèrement différentes. Cette technique

est appelée modélisation d'ensemble et la méthode la plus populaire dans le cas *uplift* est celle des forêts aléatoires [Breiman, 2001].

Utilisant le critère de segmentation d'entropie relative (2.4.5), Guelman et al. [2012] généralisent les arbres de décisions *uplift* de Rzepakowski and Jaroszewicz [2010] et présentent le cadre des *forêts aléatoires uplift*. Formellement, une forêt aléatoire RF est composée de $M > 1$ arbres. En plus des hyper-paramètres propres aux arbres de décisions, chacun des arbres est ajusté en utilisant à la fois un sous-échantillon des observations et un sous-échantillon des variables explicatives, tirés aléatoirement. Ceci permet de dé-corréler les arbres en question [Murua, 2002]. Chaque arbre m prédit l'*uplift* associé au vecteur de covariables $\mathbf{x}_{n+1}$ d'un futur individu par l'*uplift* observé dans la feuille correspondant à cet individu, $\hat{u}_m(\mathbf{x}_{n+1})$. L'idée est alors de combiner ces M prédictions en une seule en utilisant une fonction d'agrégation, telle que la moyenne, c'est-à-dire,

$$\hat{u}(\mathbf{x}_{n+1}) = \frac{1}{M} \sum_{m=1}^{M} \hat{u}_m(\mathbf{x}_{n+1}). \tag{2.4.9}$$

En raison des propriétés de réduction de la variance des fonctions d'agrégation (et en particulier de la moyenne), le modèle RF devrait avoir un meilleur pouvoir prédictif qu'un modèle composé d'un seul arbre. Suivant le même cadre, les *forêts d'inférence conditionnelle causales* (*causal conditional inference forests* ou CCIF en anglais) tentent d'améliorer les performances de généralisation des forêts aléatoires *uplift* [Guelman et al., 2015]. L'algorithme sépare la sélection de variables de la procédure de segmentation. Pour chaque nœud terminal de l'arbre, un test statistique de l'hypothèse nulle globale d'absence d'effet d'interaction entre le traitement et une des variables explicatives est effectué. Lorsque l'hypothèse nulle globale n'est pas rejetée, le processus de segmentation est arrêté à ce nœud. Sinon, la variable avec la plus petite valeur-p est sélectionnée. Il est important de noter que cette méthode considère un critère d'arrêt supplémentaire aux critères habituels. En effet, suivant le cadre de CART dans la construction d'un arbre, un nœud terminal est déclaré lorsque l'une des règles d'arrêt suivantes est satisfaite : le nœud devient pur dans le sens où toutes les valeurs des covariables sont les mêmes; le nombre total d'observations dans le nœud est inférieur à un seuil pré-spécifié; la profondeur du nœud est égale à une profondeur d'arbre maximale prédéfinie; le nombre d'observations du groupe traitement ou le nombre d'observations du groupe contrôle est inférieur à un seuil pré-spécifié.

Il existe d'autres travaux reposant sur les forêts aléatoires *uplift* dans la littérature. Par exemple, un cas particulier de forêts aléatoires généralisées de Athey et al. [2019] considère le cas du partitionnement récursif pour les effets causaux hétérogènes ou *arbres causaux* [Athey et Imbens, 2016, Wager and Athey, 2018]. Dans ce cas, l'originalité vient de l'estimation *honnête*. Un modèle est dit *honnête* s'il n'utilise pas les mêmes informations pour la partition de l'espace des covariables (c'est-à-dire le critère de segmentation) et pour l'estimation des effets du traitement dans les feuilles de l'arbre. Cela se traduit par la division de l'échantillon en deux parties, une pour la construction de l'arbre et une seconde pour l'estimation. Cela a pour objectif de réduire le sur-ajustement (qui se produit lorsqu'un modèle décrit le bruit dans les données au lieu de la véritable relation sous-jacente).

Il y a encore quelques inconvénients aux méthodes de forêts aléatoires *uplift*. D'abord, il faut noter que même si les modèles uniques sont facilement interprétables, l'interprétation est perdue lors de la combinaison de plusieurs modèles, puisque typiquement $M > 10$ (souvent $M = 100$). Il est également connu que les mesures d'importance des variables [Breiman, 2001] (ou *variable importance* en anglais) sont biaisées lorsque les variables prédictives sont fortement corrélées, ce qui conduit à préférer artificiellement les variables prédictives sous-optimales [Strobl *et al.*, 2007]. Dans le cas de l'*uplift*, il faut être un peu plus prudent avec le sur-ajustement, en particulier en ce qui concerne la taille des feuilles, car suffisamment d'observations doivent être présentes pour être en mesure de calculer un *uplift* valide. Cela est particulièrement vrai dans les cas où le groupe contrôle est beaucoup plus petit que le groupe traitement, ou vice-versa. Les segmentations sont susceptibles d'être placées à côté de valeurs extrêmes (des covariables) car les valeurs extrêmes de chaque groupe (traitement ou contrôle) peuvent influencer le choix de segmentation. De plus, les segmentations successives ont tendance à regrouper des valeurs extrêmes similaires, introduisant plus de variance dans la prédiction [Zhao et al., 2017]. Il faut donc un réglage fin des différents hyper-paramètres de la forêt afin d'obtenir des modèles satisfaisants. Alternativement, certains modèles utilisent la variable réponse transformée afin de prédire l'*uplift* dans chaque feuille [Athey and Imbens, 2015]. Cependant, cet estimateur souffre d'une variance plus élevée que l'estimateur utilisant la différence de proportions de réponses positives dans chaque groupe [Powers et al., 2018]. Malgré cela, la littérature suggère que les méthodes d'ensemble basées sur les arbres soient à la pointe de la technologie pour la modélisation *uplift* [Sołtys et al., 2015].

2.5. Évaluation de modèles *uplift*

Une grande variété de mesures de qualité sont utilisées pour les applications de modélisation traditionnelles, allant des mesures nominales non paramétriques sur des tables de contingence $2 \times k$ (matrice de confusion, χ^2, gain d'information), en passant par des mesures ordinales (rangs) sur des tableaux de contingence ou résultats triés (coefficient de Gini, statistique de Kolmogorov-Smirnov), aux statistiques paramétriques telles que le R^2, les mesures de divergence et de maximum de vraisemblance.

Dans le cas de l'inférence causale, les mesures paramétriques ponctuelles ne peuvent pas être appliquées car le véritable effet causal du traitement au niveau individuel est inconnu. Cependant, on peut évaluer les performances d'un modèle en comparant des groupes d'observations. Pour les modèles *uplift*, ceci est réalisé à l'aide de visualisations adaptées et à travers le coefficient Qini [Radcliffe, 2007], une métrique basée sur le rang qui peut être vue comme une généralisation du coefficient de Gini [Gini, 1997] aux modèles *uplift*. Les méthodes de rangs sont prometteuses comme passerelle vers des méthodes paramétriques pour l'évaluation des performances d'un modèle (en utilisant le rang induit par le modèle).

2.5.1. Visualisation

Typiquement, l'évaluation visuelle d'un modèle *uplift* se base sur les prédictions ordonnées. Formellement, pour un modèle donné, notons $\hat{u}_{(1)} \geq \hat{u}_{(2)} \geq ... \geq \hat{u}_{(n)}$ les *uplift* prédits et ordonnés pour n observations. Soit $\phi \in [0,1]$ une proportion donnée et posons

$$N_\phi = \{i : \hat{u}_i \geq \hat{u}_{(\lceil \phi n \rceil)}\} \subset \{1, \ldots, n\}, \tag{2.5.1}$$

comme le sous-ensemble d'individus avec les $\phi n \times 100\%$ *uplift* prédits les plus élevés. Ici, $\lceil s \rceil$ désigne le plus petit entier supérieur ou égal à $s \in \mathbb{R}$. Puisque N_ϕ est une fonction des prédictions, N_ϕ est une fonction du modèle ajusté.

Une visualisation commune dans l'évaluation des modèles *uplift* requiert la construction d'un graphique à barres. Pour ce faire, le domaine de $\phi \in [0,1]$ est partitionné en J intervalles, ou $J+1$ points $0 = \phi_1 < \phi_2 < ... < \phi_{J+1} = 1$. En pratique, l'échantillon est divisé en quintiles ($J = 5$) ou déciles ($J = 10$). Soit B_k le $k$ième intervalle $(\phi_k, \phi_{k+1}] \subseteq (0,1]$, $k = 1, \ldots, J$.

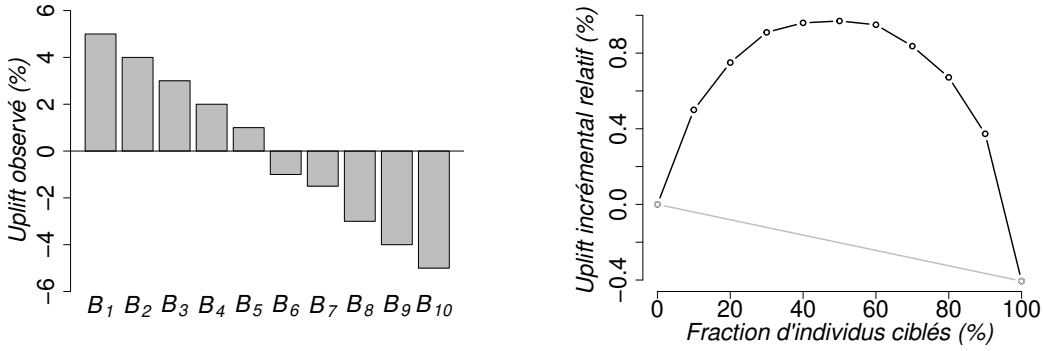

Fig. 2.1. Une présentation typique des résultats d'un modèle prédictif montrant les *uplift* observés pour chaque décile induit par le modèle (à gauche) et la courbe Qini (à droite). La ligne droite grise correspond à une stratégie qui cible les individus au hasard.

Posons $\bar{u}_k$ l'*uplift* observé dans l'intervalle B_k, $k \in \{1, ..., J\}$, soit,

$$\bar{u}_k = \sum_{i \in B_k} y_i t_i / \sum_{i \in B_k} t_i - \sum_{i \in B_k} y_i (1 - t_i) / \sum_{i \in B_k} (1 - t_i). \tag{2.5.2}$$

La Figure 2.1 de gauche illustre un graphique à barres des *uplift* observés associés à chacun des $J = 10$ intervalles. Le graphique à barres *uplift* permet de visualiser l'*uplift* mais ne fournit aucune statistique permettant la comparaison de différents modèles. Néanmoins, cela donne un premier moyen de présenter et d'évaluer la qualité d'ajustement d'un modèle. La motivation à envisager des méthodes de rangs vient de la croyance qu'un bon modèle devrait être capable de sélectionner en premier les individus ayant le plus grand *uplift* et devrait induire une disposition décroissante des *uplift* observés.

Il est possible de procéder de la même façon pour la méthode de visualisation la plus commune de la modélisation *uplift* : la *courbe Qini* [Radcliffe, 2007]. Elle permet la comparaison de plusieurs modèles (c'est-à-dire, en traçant plusieurs courbes sur le même graphique). Pour la tracer, nous devons définir la fonction associée. Formellement, en fonction de la fraction de population traitée ϕ, définissons l'*uplift* incrémental relatif par

$$g(\phi) = \left(\sum_{i \in N_\phi} y_i t_i - \sum_{i \in N_\phi} y_i (1 - t_i) \left\{ \sum_{i \in N_\phi} t_i / \sum_{i \in N_\phi} (1 - t_i) \right\} \right) / \sum_{i=1}^{n} t_i, \tag{2.5.3}$$

où $\sum_{i \in N_\phi} (1 - t_i) \neq 0$, avec $g(0) = 0$ et $g(1)$ équivaut à l'*uplift* global. La courbe Qini est construite en traçant $g(\phi)$ en fonction de $\phi \in [0,1]$. En pratique, le même partitionnement du domaine que pour le graphique à barres est utilisé. Ceci est illustré dans la Figure 2.1

de droite. La courbe Qini peut être interprétée comme suit. L'axe des abscisses représente la fraction d'individus ciblés (ordonnés par ordre décroissant des prédictions du modèle) et l'axe des ordonnées montre le nombre incrémentiel de réponses positives par rapport au nombre total d'individus ciblés. La ligne droite entre les points (0,0) et $(1, g(1))$ représente un *benchmark* permettant de comparer la qualité de prédiction du modèle à une stratégie qui cible les individus au hasard. En d'autres termes, lorsque la stratégie consiste à traiter des individus au hasard, si une proportion ϕ de la population est traitée, nous nous attendons à observer un *uplift* égal à ϕ fois l'*uplift* global.

2.5.2. Statistiques de qualité d'ajustement

Bien qu'il soit possible de sélectionner le modèle optimal parmi plusieurs à travers les visualisations présentées ci-dessus, il est souvent préférable de définir des statistiques de qualité d'ajustement. Ceci est d'autant plus vrai lorsqu'il faut choisir parmi un grand nombre de modèles possibles. Par définition, $u(\mathbf{x}) \in [-1; 1]$. Dans certaines applications, l'objectif est de prédire pour quels individus nous espérons que le traitement soit bénéfique. Ainsi, il est assez logique de définir une statistique de qualité d'ajustement basée sur une stratégie voulant traiter les individus pour lesquels l'*uplift* prédit $\hat{u}(\mathbf{x})$ est positif [Zhao et al., 2017]. Notons cette stratégie par $d(\mathbf{x}) = \mathbb{1}(\hat{u}(\mathbf{x}) > 0)$. La statistique associée est connue sous le terme de *valeur* (ou *value* en anglais) d'une stratégie de traitement. On la définit par :

$$v = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(d(\mathbf{x}_i) = t_i) y_i. \tag{2.5.4}$$

En d'autres termes, la valeur est le résultat attendu d'un individu lorsque tous les individus sont traités selon la stratégie $d(\mathbf{x})$. Pour les applications dont les valeurs les plus élevées sont souhaitables (par exemple, durée de vie, taux de clics, taux de ventes), la stratégie qui maximise la valeur est optimale. Dans ce cas, tout ce qui nous intéresse est la décision de traitement (et non l'*uplift* lui-même). Cependant, un inconvénient de cette métrique est qu'elle n'utilise qu'une partie des données. En effet, elle n'inclut pas les individus dont les traitements reçus ne correspondent pas à $d(\mathbf{x})$.

Dans la littérature, la méthode d'évaluation de modèles *uplift* la plus populaire est basée sur la courbe Qini et se fait habituellement par le calcul du *coefficient Qini* [Radcliffe, 2007]. Le coefficient Qini est un indice unique de la performance du modèle. Il est défini comme

l'aire entre la courbe Qini et la ligne droite (voir la Figure 2.1 de droite), soit

$$q = \int_0^1 Q(\phi)\ \mathrm{d}\phi = \int_0^1 \{g(\phi) - \phi\ g(1)\}\ \mathrm{d}\phi, \tag{2.5.5}$$

où $Q(\phi) = g(\phi) - \phi\ g(1)$. Cette aire peut être approximée numériquement en utilisant une méthode de Newton-Cotes comme la règle trapézoïdale :

$$\hat{q} = \frac{1}{2}\sum_{j=1}^{K}(\phi_{j+1} - \phi_j)\{Q(\phi_{j+1}) + Q(\phi_j)\}. \tag{2.5.6}$$

En général, lors de la comparaison de plusieurs modèles, le modèle préféré est celui avec le coefficient Qini maximal [Radcliffe, 2007].

Dans d'autres applications, il se peut que nous ne soyons pas intéressés par une estimation de la performance prédictive du modèle sur l'ensemble de la population cible, mais uniquement sur une fraction ϕ prédéterminée de cette population. C'est généralement le cas, par exemple, dans les applications de marketing direct, où une entreprise peut cibler au plus 20% d'une population donnée [Guelman et al., 2015]. Le *top uplift* pour ϕ est obtenu en calculant l'*uplift* observé dans la fraction ϕ des individus avec les *uplift* prédits les plus élevés. C'est-à-dire,

$$\tau_\phi = \sum_{i\in N_\phi} y_i t_i / \sum_{i\in N_\phi} t_i - \sum_{i\in N_\phi} y_i(1-t_i) / \sum_{i\in N_\phi}(1-t_i). \tag{2.5.7}$$

Ces statistiques d'ajustement démontrent bien la particularité des modèles *uplift* dans le cadre de l'inférence causale. De par les applications liées à la modélisation *uplift*, l'objectif est double : estimer/prédire l'*uplift* à travers un modèle statistique et sélectionner le modèle ayant le meilleur Qini, ce qui permet de traiter de futurs individus selon l'ordre induit par les prédictions de ce modèle.

Remarque 2.5.1. *Dans le contexte de la sélection de modèles prédictifs, en pratique, étant donné L modèles et/ou hyper-paramètres associés, nous ajustons L estimateurs* $\{\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_L\}$. *L'objectif est de maximiser les performances de prédiction attendues en utilisant des statistiques de qualité d'ajustement. Il existe plusieurs façons d'évaluer les performances de prédiction. Cependant, le fractionnement des données en* données d'entraînement *et* données de validation *est le plus largement utilisé dans la pratique [Arlot et al., 2010]. Cette procédure est connue sous le nom de validation croisée. Avant d'ajuster les modèles, les observations sont réparties aléatoirement en échantillon d'apprentissage* $\mathcal{T}$ *et en échantillon de validation* $\mathcal{V}$. *Tous les modèles sont ajustés sur* $\mathcal{T}$, *mais évalués*

et comparés en terme de performance de prédiction sur $\mathcal{V}$. Il existe plusieurs méthodes permettant la validation croisée. Nous renvoyons le lecteur vers le manuel de Hastie et al. *[2009] pour une bibliographie approfondie sur l'apprentissage statistique et, entre autres, la validation croisée.*

2.6. Discussion

Nous avons présenté dans ce chapitre les méthodes actuelles permettant l'estimation/prédiction de l'*uplift*. Dans chacun des chapitres suivants, nous consacrons aussi une section à la bibliographie spécifique au sujet abordé. De manière générale, nous pouvons diviser les méthodes présentées en deux catégories : i) les méthodes indirectes qui reposent surtout sur des modèles paramétriques; ii) les méthodes directes qui reposent sur des modèles non paramétriques.

Du point de vue de la complexité, les modèles paramétriques sont plus simples que les modèles non paramétriques tels que les arbres de régression, car pour les modèles paramétriques, le nombre de paramètres est maintenu petit et fixe. Pour de nombreux analystes, bien que la prédiction soit la cible principale, d'un point de vue pratique, l'interprétation du modèle est très importante. Savoir quelles variables et la manière dont ces variables distinguent les sous-groupes d'individus est un important objectif dans la prédiction de l'*uplift*. Pour ces raisons, nous nous concentrons d'abord sur les modèles paramétriques.

La méthode d'interaction de la section 2.3.2 représente une amélioration par rapport à la méthode utilisant deux modèles indépendants, en ce sens qu'elle fournit un moyen formel d'effectuer un test de signification des paramètres d'interaction entre le traitement et les covariables. Cependant, elle souffre de problèmes de sur-ajustement lors de l'inclusion de tous les effets d'interaction avec un espace de covariables de grande dimension. De plus, la solution n'est pas optimisée pour la recherche de groupes hétérogènes en fonction de l'*uplift*. Afin de corriger cela, nous développons une nouvelle méthode permettant l'estimation des paramètres (et la sélection de variables) spécifique à la modélisation de l'*uplift*. Nous montrons qu'un modèle optimisé pour le coefficient Qini (2.5.6) agit comme un facteur de régularisation pour l'*uplift*, tout comme un modèle de vraisemblance pénalisé le fait pour la régression. Il en résulte des modèles interprétables avec des variables explicatives

pertinentes peu nombreuses. Nous développons notre méthodologie pour la régression logistique parce que l'interprétation des rapports des cotes est bien connue. Cependant, notre procédure d'estimation peut être facilement généralisée à d'autres modèles paramétriques. La méthodologie associée est présentée dans le chapitre 3.

Dans le chapitre 4, nous poursuivons dans la même direction en proposant une nouvelle méthode permettant la modélisation non-linéaire de l'*uplift*. En effet, lorsqu'une variable discrète est incluse dans un modèle, elle permet de mesurer une relation non-linéaire avec la variable dépendante. Nous introduisons des méthodes de discrétisation (univariée et bivariée) de variables continues permettant de capturer la relation avec l'*uplift* directement. Dans la foulée, ceci nous amène à développer les outils associés à l'analyse exploratoire des données dans un contexte d'inférence causale. En effet, bien qu'il existe des méthodes de calcul disponibles pour la modélisation *uplift*, la plupart d'entre elles excluent les modèles de régression statistique. Nous développons donc une librairie R **tools4uplift** dans l'objectif de combler cette lacune. Cette librairie comprend des outils pour : i) la discrétisation; ii) la visualisation; iii) la sélection de variables; iv) l'estimation des paramètres; et v) la validation du modèle. Nous les présentons à l'aide d'une application pratique des techniques et des développements méthodologiques qui y sont associés.

Le point commun des méthodes non paramétriques d'estimation de l'uplift présentées dans la section 2.4 est le critère de segmentation. En effet, en plus d'estimer l'*uplift* par l'*uplift* observé dans chaque feuille d'un arbre, l'ajustement du modèle est basé sur l'optimisation d'un critère propre à l'*uplift*. Dans le chapitre 3, nous employons une approche similaire pour les modèles paramétriques. En effet, nous montrons empiriquement qu'estimer les paramètres en maximisant la vraisemblance ne permet pas forcément de maximiser les statistiques de qualité d'ajustement des modèles *uplift*, à savoir le coefficient Qini. De plus, mettant à profit la flexibilité des réseaux de neurones, nous généralisons cette idée pour construire des modèles prédictifs plus performants. Nous proposons une nouvelle méthode basée sur une fonction de perte qui permet de mieux capturer l'*uplift*. Nous généralisons l'utilisation de réseaux de neurones jumeaux [Bromley et al., 1994] pour l'inférence causale, ce qui permet une estimation directe de l'*uplift*. Nous présentons cette dernière méthode dans le chapitre 6 de la thèse.

Bibliographie

Farrokh Alemi, Harold Erdman, Igor Griva et Charles H Evans : Improved statistical methods are needed to advance personalized medicine. *The Open Translational Medicine Journal*, 1:16, 2009.

Sylvain Arlot, Alain Celisse *et al.* : A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79, 2010.

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

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

Susan Athey, Julie Tibshirani, Stefan Wager *et al.* : Generalized random forests. *The Annals of Statistics*, 47(2):1148–1178, 2019.

Leo Breiman : Random forests. *Machine Learning*, 45(1):5–32, 2001.

Leo Breiman, Jerome Friedman, Charles J Stone et Richard A Olshen : *Classification and Regression Trees*. CRC press, 1984.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger et Roopak Shah : Signature verification using a “siamese" time delay neural network. *In Advances in Neural Information Processing Systems*, pages 737–744, 1994.

Floris Devriendt, Darie Moldovan et Wouter Verbeke : A literature survey and experimental evaluation of the state-of-the-art in uplift modeling: A stepping stone toward the development of prescriptive analytics. *Big Data*, 6(1):13–41, 2018.

Corrado Gini : Concentration and dependency ratios. *Rivista di politica economica*, 87:769–792, 1997.

Pauline Givord : Méthodes économétriques pour l'évaluation de politiques publiques. *Économie et prévision*, (1):1–28, 2014.

Leo Guelman *et al.* : *Optimal personalized treatment learning models with insurance applications*. Thèse de doctorat, Universitat de Barcelona, 2015.

Leo Guelman, Montserrat Guillén et Ana M Pérez-Marín : Random forests for uplift modeling: an insurance customer retention case. *In Modeling and Simulation in Engineering, Economics and Management*, pages 123–133. Springer, 2012.

Pierre Gutierrez et Jean-Yves Gérardy : Causal inference and uplift modelling: A review of the literature. *In International Conference on Predictive Applications and APIs*, pages 1–13, 2017.

Behram Hansotia et Bradley Rukstales : Direct marketing for multichannel retailers: Issues, challenges and solutions. *Journal of Database Marketing and Customer Strategy Management*, 9(3):259–266, 2001.

Behram Hansotia et Bradley Rukstales : Incremental value modeling. *Journal of Interactive Marketing*, 16(3):35, 2002.

Trevor Hastie, Robert Tibshirani et Jerome Friedman : *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

Paul W Holland : Statistics and causal inference. *Journal of the American Statistical Association*, 81(396):945–960, 1986.

Kosuke Imai, Marc Ratkovic *et al.* : Estimating treatment effect heterogeneity in randomized program evaluation. *The Annals of Applied Statistics*, 7(1):443–470, 2013.

Maciej Jaskowski et Szymon Jaroszewicz : Uplift modeling for clinical trial data. *In ICML Workshop on Clinical Data Analysis*, 2012.

Solomon Kullback et Richard A Leibler : On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

Finn Kuusisto, Vitor Santos Costa, Houssam Nassif, Elizabeth Burnside, David Page et Jude Shavlik : Support vector machines for differential prediction. *In Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 50–65. Springer, 2014.

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.

René Michel, Igor Schnakenburg et Tobias Von Martens : *Targeting Uplift: An Introduction to Net Scores*. Springer Nature, 2019.

Alejandro Murua : Upper bounds for error rates of linear combinations of classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):591–602, 2002.

Jerzy S Neyman : On the application of probability theory to agricultural experiments. *Annals of Agricultural Sciences*, 10:1–51, 1923.

Scott Powers, Junyang Qian, Kenneth Jung, Alejandro Schuler, Nigam H Shah, Trevor Hastie et Robert Tibshirani : Some methods for heterogeneous treatment effect estimation in high dimensions. *Statistics in Medicine*, 37(11):1767–1787, 2018.

Nicholas J Radcliffe et Patrick D Surry : Real-world uplift modelling with significance-based uplift trees. *White Paper TR-2011-1, Stochastic Solutions*, 2011.

NJ Radcliffe : Using control groups to target on predicted lift: Building and assessing uplift models. *Direct Market J Direct Market Assoc Anal Council*, 1:14–21, 2007.

NJ Radcliffe et PD Surry : Differential response analysis: Modeling true response by isolating the effect of a single action. *Credit Scoring and Credit Control VI. Edinburgh, Scotland*, 1999.

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

Donald B Rubin : Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5):688, 1974.

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

Michał Sołtys, Szymon Jaroszewicz et Piotr Rzepakowski : Ensemble methods for uplift modeling. *Data Mining and Knowledge Discovery*, 29(6):1531–1559, 2015.

Carolin Strobl, Anne-Laure Boulesteix, Achim Zeileis et Torsten Hothorn : Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, 8(1):25, 2007.

Xiaogang Su, Joseph Kang, Juanjuan Fan, Richard A Levine et Xin Yan : Facilitating score and causal inference trees for large observational studies. *Journal of Machine Learning Research*, 13(Oct):2955–2994, 2012.

Xiaogang Su, Chih-Ling Tsai, Hansheng Wang, David M Nickerson et Bogong Li : Subgroup analysis via recursive partitioning. *Journal of Machine Learning Research*, 10 (Feb):141–158, 2009.

Denis Talbot : *Identification de modèles appropriés pour l'inférence causale à partir de données d'observation*. Thèse de doctorat, Mathématiques. Université du Québec à Montréal, 2015.

Richard H Thaler et Cass R Sunstein : *Nudge: Improving decisions about health, wealth, and happiness*. Penguin, 2009.

Lu Tian, Ash A Alizadeh, Andrew J Gentles et Robert Tibshirani : A simple method for estimating interactions between a treatment and a large number of covariates. *Journal of the American Statistical Association*, 109(508):1517–1532, 2014.

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

Lukasz Zaniewicz et Szymon Jaroszewicz : Support vector machines for uplift modeling. *In 2013 IEEE 13th International Conference on Data Mining Workshops*, pages 131–138. IEEE, 2013.

Yan Zhao, Xiao Fang et David Simchi-Levi : A practically competitive and provably consistent algorithm for uplift modeling. *In 2017 IEEE International Conference on Data Mining (ICDM)*, pages 1171–1176. IEEE, 2017a.

Yan Zhao, Xiao Fang et David Simchi-Levi : Uplift modeling with multiple treatments and general response types. *In Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 588–596. SIAM, 2017b.

Chapitre 3

Régression *uplift* basée sur le Qini

Premier article.

Qini-based Uplift Regression

par

Mouloud Belbahri[1], Alejandro Murua[1], Olivier Gandouet[2], and Vahid Partovi Nia[3]

([1]) Université de Montréal
([2]) TD Assurance
([3]) École Polytechnique de Montréal

Cet article a été officiellement accepté par la revue *The Annals of Applied Statistics*

Mes contributions et le rôle des coauteurs. Dans cet article, j'ai

- proposé la méthode de sélection de variables basée sur le coefficient Qini;

- développé la relation mathématique permettant l'interprétation de la dérivée de la fonction de la courbe Qini;
- rédigé le code R permettant la mise en oeuvre des méthodes étudiées;
- rédigé l'article ainsi que les réponses aux arbitres du journal;
- élaboré et mis en oeuvre les simulations inspirées des données réelles;
- analysé les données d'entreprise et interprété les résultats.

En tant que directeur principal, Alejandro Murua a participé à la rédaction de l'article, au développement méthodologique ainsi qu'à l'analyse des données réelles. En tant que co-directeur, Vahid Partovi Nia a effectué une relecture apportant une contribution significative au niveau du contenu intellectuel. En tant que directeur industriel, Olivier Gandouet a agi en tant que personne ressource apportant des critiques pertinentes permettant une étude approfondie des données réelles.

RÉSUMÉ. Les modèles *uplift* apportent une solution au problème d'isoler l'effet causal d'une campagne marketing. Pour réduire le taux de désabonnement des clients, des modèles *uplift* sont utilisés pour identifier les clients susceptibles de répondre positivement à une activité de rétention *uniquement* s'ils sont ciblés, et pour éviter de gaspiller des ressources sur des clients très susceptibles de passer à une autre entreprise. Nous introduisons un modèle de régression *uplift* basé sur le Qini pour analyser une campagne marketing de fidélisation de clients d'une grande compagnie d'assurance. Notre approche est basée sur des modèles de régression logistique. Nous montrons qu'un modèle *uplift* optimisé pour le Qini agit comme un facteur de régularisation pour l'*uplift*, tout comme un modèle de vraisemblance pénalisé le fait pour la régression. Il en résulte des modèles interprétables avec peu de variables explicatives pertinentes. Nos résultats montrent que l'estimation des paramètres basée sur le Qini améliore considérablement l'ajustement des modèles *uplift*.

Mots clés : corrélation de Kendall, inférence causale, lasso, optimisation sans dérivée, régression logistique

Abstract. Uplift models provide a solution to the problem of isolating the marketing effect of a campaign. For customer churn reduction, uplift models are used to identify the customers who are likely to respond positively to a retention activity *only* if targeted, and to avoid wasting resources on customers that are very likely to switch to another company. In practice, the uplift models performance is measured by the Qini coefficient. We introduce a Qini-based uplift regression model to analyze a large insurance company's retention marketing campaign. Our approach is based on logistic regression models. We show that a Qini-optimized uplift model acts as a regularizing factor for uplift, much as a penalized likelihood model does for regression. This results in interpretable models with few relevant explanatory variables. Our results show that Qini-based parameter estimation significantly improves the Qini prediction performance of uplift models.

Keywords: casual inference, Kendall's correlation, lasso, logistic regression, derivative-free optimization

3.1. Introduction

This work proposes a methodology that identifies characteristics associated with a home insurance policy that can be used to infer the link between marketing intervention and policy renewal rate. Using the resulting statistical model, the goal is to predict which customers the company should focus on, in order to deploy future retention campaigns.

A subscription-based company loses its customers when they stop doing business with their service. Also known as customer attrition, customer churn can be a drag on the business growth. It is less expensive to retain existing customers than to acquire new customers, so businesses put effort into marketing strategies to reduce customer attrition. Customer loyalty, on the other hand, is usually more profitable because the company has already earned the trust and loyalty of existing customers. Businesses mostly have a defined strategy for mitigating customer churn. Organizations are able to determine their success rate in customer loyalty and identify improvement strategies using available data and learning about churn.

With the increasing amount of data available, a company tries to find the causal effects of customer churn. The term *causal*, as in causal study, refers to a study that tries to discover a cause-effect relationship. The statement A causes B means that changing the value of A will change the distribution of B. When A causes B, A and B will be associated but the converse is not, in general, true, since association does not necessarily imply causation.

There exists two frameworks for discussing causation [Pearl, 2009]. We will consider the statistical framework for causal inference formally introduced by Rubin [1974], which uses the notation of counterfactual random variables. This framework is also associated with the potential outcome framework [Neyman, 1923], also known as the Rubin causal model [Holland, 1986]. Suppose a company decides to deploy a marketing campaign, and that customers are randomly divided into two groups. The first group is targeted with a marketing initiative (treatment group), and the second group serves as control (or baseline). A potential outcome is the theoretical response each customer would have manifested, had it been assigned to a particular group. Under randomization, association and causation coincide and these outcomes are independent of the assignment other customers receive. In practice, potential outcomes for an individual cannot be observed. Each customer is only assigned to either treatment or control, making direct observations in the other condition (called the counterfactual condition) and the observed individual treatment effects, impossible [Holland, 1986].

In marketing, it is common to compare the targeted customers' mean response rate (treatment group) with the non-targeted customers' mean response rate (control group). A campaign is considered successful if it succeeds in increasing the mean response rate of the treated group relative to the mean response rate of the control group. The difference in mean response rates is the increase due to the campaign. To further increase the returns of future direct marketing campaigns, a predictive response model can be developed. Response models [Smith and Swinyard, 1982, Hanssens et al., 2003, Coussement et al., 2015] of client behavior are used to predict the probability that a client responds to a marketing campaign (e.g. renews subscription). Marketing campaigns using response models concentrate on clients with high probability of positive response. However, this strategy does not necessarily cause the renewal. In other words, the customers could renew their subscription without marketing effort. Therefore, it is important to extract the cause of the renewal, and isolate the effect of marketing.

Data from one of the leader north-American insurers is at our disposal to evaluate the performance of the methodology introduced in this work. This company is interested in designing retention strategies to minimize its policyholders' attrition rate. For that purpose,

Tab. 3.1. Renewal rate by group for $n = 20{,}997$ home insurance policies.

	Control	Called	Overall
Renewed policies	2,253	18,018	20,271
Cancelled policies	72	654	726
Renewal rate	96.90%	96.50%	96.54%

during three months, an experimental loyalty campaign was implemented, from which policies coming up for renewal were randomly allocated into one of the following two groups: a treatment group, and a control group. Policyholders under the treatment group received an outbound courtesy call made by one of the company's licensed insurance advisors, with the objective to reinforce the customers confidence in the company, to review their coverage and address any questions they might have about their policy. No retention efforts were applied to the control group. The goal of the study is develop models that will be used to identify which clients are likely to benefit from a call at renewal, that is, clients that are likely to renew their policy *only* if they are called by an advisor during their renewal period. Also, clients that should not be targeted could

- renew their policy on their own,
- cancel their policy whether they receive a call or not,
- cancel their policy *only* if they receive a call.

Table 3.1 shows the marketing campaign retention results. The observed difference in retention rates between the treated group and the control group is small, but there is some evidence of a slightly negative impact of the outbound call. Even if the difference is slightly negative, it may be the case that the campaign had positive retention effects on some subgroup of customers, but they were offset by negative effects on other subgroups. For example, one possible reason could be that some customers are already dissatisfied with their insurance policies and have already decided to change them before receiving the call.

In a randomized experiment, researchers often focus on the estimation of average treatment effects. However, there might be a proportion of the customers that responds favorably to the marketing campaign, and another proportion that does not. A decision at the individual level based on average treatment effects would require adjustments because of the heterogeneity in responses that can be originated by many factors. Table 3.2 describes some

of the $p = 97$ available explanatory variables in the dataset, in addition to the treatment (Called or Control) and outcome (Renewed or Cancelled the policy) variables.

The so-called uplift model [Radcliffe and Surry, 1999, Hand and Yu, 2001, Lo, 2002] provides a solution to the problem of isolating the marketing effect. Instead of modeling the class probabilities, uplift attempts to model the difference between conditional class probabilities in the treatment (e.g., a marketing campaign) and control groups. Uplift modeling aims at identifying groups on which a predetermined action will have the most positive effect.

Assessing model performance is complex for uplift modeling, as the actual value of the response, that is, the *true* uplift, is unknown at the individual subject level. To overcome this limitation, one can assess model performance by comparing groups of observations. This is done through the Qini coefficient [Radcliffe, 2007], which plays a similar role as the Gini coefficient [Gini, 1997] in Economics. The Qini coefficient is a single statistic drawn form the Qini curve. This latter object is a generalization of the Lorenz curve [Lorenz, 1905] traditionally used in direct marketing for response models.

As in all regression-based modeling, an issue in uplift modeling is the ease of interpretation of the results. The model becomes harder to interpret when the number of potential explanatory variables, that is the *dimension* of the explanatory variables increases. When the variable dimension is small, knowledge-based approaches to select the optimal set of variables can be effectively applied. When the number of potentially important variables is too large, it becomes too time-consuming to apply a manual variable selection process. In this case one may consider using automatic subset selection tools. Variable selection is an important step. It reduces the dimension of the model, avoids overfitting, and improves model stability and accuracy [Guyon and Elisseeff, 2003]. Well-known variable selection techniques such as forward, backward, stepwise [Montgomery et al., 2012], stagewise [Hastie et al., 2007], lasso [Tibshirani, 1996], and LARS [Efron et al., 2004], among others, are not designed for uplift models. One might need to adapt them to perform variable selection in this context.

We propose a new way to perform model selection in uplift regression models. Our methodology is based on the maximization of a modified version of the Qini coefficient, *the adjusted Qini*, that we introduced in Section 3.2.1. Because model selection corresponds to variable selection, the task is haunting and intractable if done in a straightforward manner

Tab. 3.2. Descriptive statistics of some available variables for $n = 20{,}997$ home insurance policies. Because of randomization, the treatment and control group means associated with each available predictor are not significantly different. For privacy concerns, we hide some values with $*$.

	Control	Called	Diff Mean	Diff SD	Domain
Sample size	18,672	2,325	-	-	-
Credit Score	756.93	756.92	-0.00	1.46	$\mathbb{R}^+$
Age (Years)	44.97	45.26	0.30	0.25	$\mathbb{R}^+$
Gender					
Male	0.59	0.60	0.01	0.01	$\{0, 1\}$
Marital Status					
Divorced	0.02	0.02	0.00	0.00	$\{0, 1\}$
Married	0.69	0.69	0.00	0.01	$\{0, 1\}$
Single	0.23	0.23	0.00	0.01	$\{0, 1\}$
Seniority (Years)	9.57	9.73	0.16	0.16	$\mathbb{R}^+$
Policy Premimm ($)					
New Premium	$*$	$*$	-7.44	16.14	$\mathbb{R}^+$
Old Premium	$*$	$*$	-5.18	15.12	$\mathbb{R}^+$
Territory					
Rural	0.06	0.06	0.00	0.01	$\{0, 1\}$
Products					
Auto and Home	0.86	0.85	-0.01	0.01	$\{0, 1\}$
Auto Policies Count	1.05	1.04	-0.01	0.01	$\mathbb{N}$
Mortgage Count	0.66	0.67	0.01	0.01	$\mathbb{N}$
Residences Count	1.07	1.08	0.01	0.01	$\mathbb{N}$
Endorsement Count	1.98	2.00	0.02	0.03	$\mathbb{N}$
Extra Options					
Option 1	0.20	0.20	0.00	0.01	$\{0, 1\}$
Option 2	0.73	0.73	0.00	0.01	$\{0, 1\}$
Option 3	0.71	0.72	0.01	0.01	$\{0, 1\}$

when the number of variables to consider is large, e.g. $p \approx 100$, like in the case of the insurance data. To realistically search for a good model, we conceived a searching method based on an efficient exploration of the regression coefficients space combined with a lasso penalization of the log-likelihood. There is no explicit analytical expression for the adjusted Qini surface (nor for the Qini curve), so unveiling it is not easy. Our idea is to gradually uncover the adjusted Qini surface in a manner inspired by surface response designs. The goal is to find the global maximum or a reasonable local maximum of the adjusted Qini by exploring the surface near optimal values of the coefficients. These coefficient values are given by maximizing the lasso penalized log-likelihood. The exploration is done using Latin hypercube sampling structures [McKay et al., 2000] centered in a sequence of penalized estimates of the coefficients.

The rest of the paper is organized as follows. We first present the current uplift models in Section 3.2 and Section 3.3 introduces the notation and details of Qini-based uplift regression. Sections 3.4 and 3.5 present the computational results of the proposed methodology on synthetic and real datasets. Final remarks and conclusion are given in Section 3.6.

3.2. Uplift modeling

Let Y be the 0-1 binary response variable, T the 0-1 treatment indicator variable and $X_1, \ldots, X_p$ the explanatory variables (predictors). The binary variable T indicates if a unit is exposed to treatment ($T = 1$) or control ($T = 0$). Suppose that n independent units are observed $\{(y_i, \mathbf{x}_i, t_i)\}_{i=1}^{n}$, where $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$ are realisations of the predictors random variables. For $i = 1, \ldots, n$, an uplift model estimates

$$u(\mathbf{x}_i) = \Pr(Y_i = 1 \mid \mathbf{x}_i, T_i = 1) - \Pr(Y_i = 1 \mid \mathbf{x}_i, T_i = 0), \tag{3.2.1}$$

where the notation $\Pr(Y_i = y_i \mid \mathbf{x}_i, T_i = t_i)$ stands for the corresponding conditional probability. Uplift modeling was formally introduced in Radcliffe and Surry [1999] under the appellation of *differential response modeling* where a thorough motivation and several practical cases promoted uplift modeling in comparison with common regression or basic tree-based methods that were used to predict the probability of success for the treatment group. They showed that conventional models, which were referred to as *response models*, did not target the people who were the most positively influenced by the treatment. In [Radcliffe and Surry, 1999] and [Hansotia and Rukstales, 2002], the methods introduced are tree-based

algorithms similar to CART [Breiman et al., 1984], but using modified split criteria that suited the uplift purpose. The method proposed by Hansotia and Rukstales [2002] uses the uplift's absolute difference $\Delta = |u_l - u_r|$, where u_l, u_r are the observed uplifts in the left and right child nodes, respectively. It is also possible to use the difference in node sizes as some sort of penalty term to adjust the differences in uplift [Radcliffe and Surry, 2011]. Other split criteria proposed in the literature are based on the χ^2 statistic [Su et al., 2009, Radcliffe and Surry, 2011], which is usually a function of Δ^2. All these splitting criteria rely on maximizing heterogeneity in treatment effects (Δ).

3.2.1. Adjusted Qini

Evaluating uplift models requires the construction of the *Qini curve* and the computation of the *Qini coefficient* [Radcliffe, 2007]. The motivation to consider the Qini curve comes from the fact that a good model should be able to select individuals with highest uplift first. More explicitly, for a given model, let $\hat{u}_{(1)} \geq \hat{u}_{(2)} \geq ... \geq \hat{u}_{(n)}$ be the sorted predicted uplifts. Let $\phi \in [0,1]$ be a given proportion and let $N_\phi = \{i : \hat{u}_i \geq \hat{u}_{(\lceil \phi n \rceil)}\} \subset \{1, \ldots, n\}$ be the subset of individuals with the $\phi n \times 100\%$ highest predicted uplifts $\hat{u}_i$ (here $\lceil s \rceil$ denotes the smallest integer larger or equal to $s \in \mathbb{R}$). Because N_ϕ is a function of the predicted uplifts, N_ϕ is a function of the fitted model. For a parametric model such as (3.3.2), N_ϕ is a function of the model's parameters estimates, and should be denoted $N_\phi(\hat{\boldsymbol{\theta}})$. To simplify the notation, we prefer to omit this specification.

As a function of the fraction of population targeted ϕ, the relative incremental uplift is defined as

$$g(\phi) = \left(\sum_{i \in N_\phi} y_i t_i - \sum_{i \in N_\phi} y_i (1 - t_i) \left\{ \sum_{i \in N_\phi} t_i / \sum_{i \in N_\phi} (1 - t_i) \right\} \right) / \sum_{i=1}^{n} t_i, \tag{3.2.2}$$

where $\sum_{i \in N_\phi} (1 - t_i) \neq 0$, with $g(0) = 0$. The incremental uplift has been normalized by the number of subjects treated in N_ϕ. Note that $g(1)$ is the overall sample observed uplift, that is

$$g(1) = \left(\sum_{i=1}^{n} y_i t_i / \sum_{i=1}^{n} t_i \right) - \left(\sum_{i=1}^{n} y_i (1 - t_i) / \sum_{i=1}^{n} (1 - t_i) \right).$$

The Qini curve is constructed by plotting $g(\phi)$ as a function of $\phi \in [0,1]$. This is illustrated in Figure 3.1. The curve can be interpreted as follows. The x-axis represents the fraction of targeted individuals and the y-axis shows the incremental number of positive responses

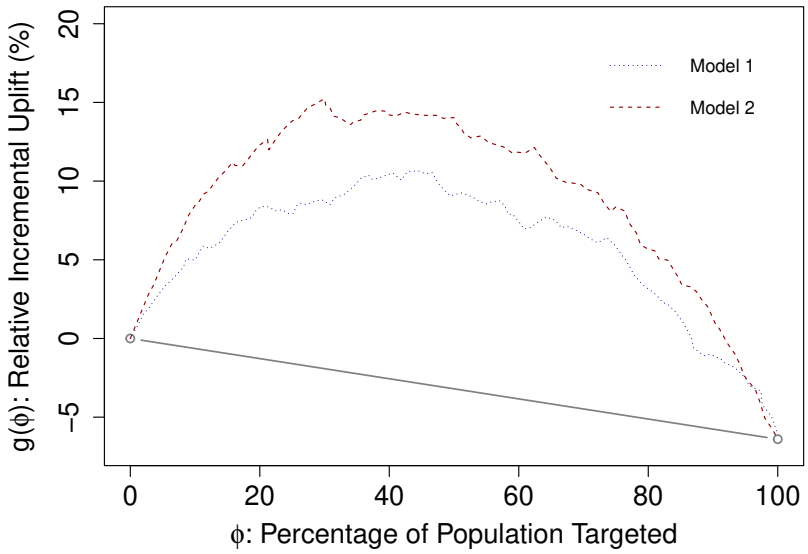

Fig. 3.1. Example of Qini curves corresponding to two different uplift models compared to a random targeting strategy.

relative to the total number of targeted individuals. The straight line between the points (0,0) and $(1, g(1))$ in Figure 3.1 represents a benchmark to compare the performance of the model to a strategy that would randomly target subjects. In other words, when the strategy is to treat individuals randomly, if a proportion ϕ of the population is treated, we expect to observe an uplift equal to ϕ times the global uplift. The Qini coefficient q is a single index of model performance. It is defined as the area between the Qini curve and the straight line

$$q = \int_0^1 Q(\phi)\ \mathrm{d}\phi = \int_0^1 \{g(\phi) - \phi\ g(1)\}\ \mathrm{d}\phi, \tag{3.2.3}$$

where $Q(\phi) = g(\phi) - \phi\ g(1)$. This area can be numerically approximated using a Riemann method such as the trapezoid rule formula: the domain of $\phi \in [0,1]$ is partitioned into J panels, or $J+1$ grid points $0 = \phi_1 < \phi_2 < ... < \phi_{J+1} = 1$, to approximate the Qini coefficient q (3.2.3) by its empirical estimation

$$\hat{q} = \frac{1}{2}\sum_{j=1}^{J}(\phi_{j+1} - \phi_j)\{Q(\phi_{j+1}) + Q(\phi_j)\} \times 100\%. \tag{3.2.4}$$

In general, when comparing several models, the preferred model is the one with the maximum Qini coefficient [Radcliffe, 2007].

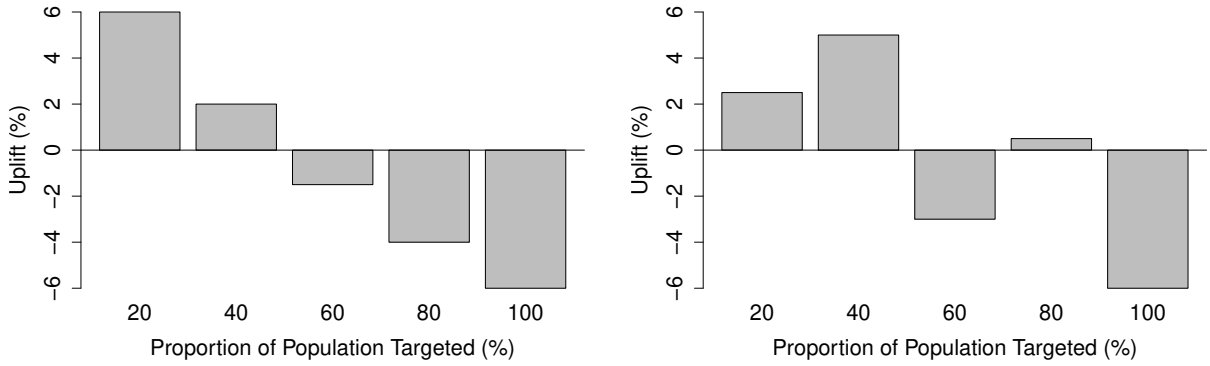

Fig. 3.2. Theoretical predicted uplift barplots with 5 panels corresponding to two different models. A good model should order the observed uplift from highest to lowest. The Kendall's uplift rank correlation is $\rho = 1$ for the left barplot and $\rho = 0.6$ for the right barplot.

Another visualization associated with uplift model validation is based on the observed uplifts in each of the J bins used to compute the Qini coefficient: a good model should induce a decreasing disposition of the observed uplifts in these bins. Figure 3.2 illustrates good and bad uplift models as barplots of observed uplifts associated with each of the J bins. A decreasing disposition of the uplift values in the J bins is an important property of an uplift model. To measure the degree to which a model does this correctly, we suggest the use of the Kendall rank correlation coefficient [Kendall, 1938]. The goal is to find a model that maximizes the correlation between the predicted uplift and the observed uplift. Let B_k denote the kth bin $(\phi_k, \phi_{k+1}] \subseteq (0,1]$, $k = 1, \ldots, J$. The Kendall's uplift rank correlation is defined as

$$\rho = \frac{2}{J(J-1)} \sum_{i<j} \text{sign}(\bar{\hat{u}}_i - \bar{\hat{u}}_j) \ \text{sign}(\bar{u}_i - \bar{u}_j), \tag{3.2.5}$$

where $\bar{\hat{u}}_k$ is the average predicted uplift in bin B_k, $k \in 1,...,J$, and $\bar{u}_k$ is the observed uplift in the same bin B_k, that is,

$$\bar{u}_k = \sum_{i \in B_k} y_i t_i / \sum_{i \in B_k} t_i - \sum_{i \in B_k} y_i (1 - t_i) / \sum_{i \in B_k} (1 - t_i). \tag{3.2.6}$$

From a business point of view, this statistic and the associated barplot are easier to interpret than the Qini coefficient and the Qini curve. However, we do not advise the use of ρ alone for model selection. If two models have the same $\hat{q}$, the preferred one should be the one with the highest ρ. But, when two models' $\hat{q}$ differ, it is not clear that the preferred

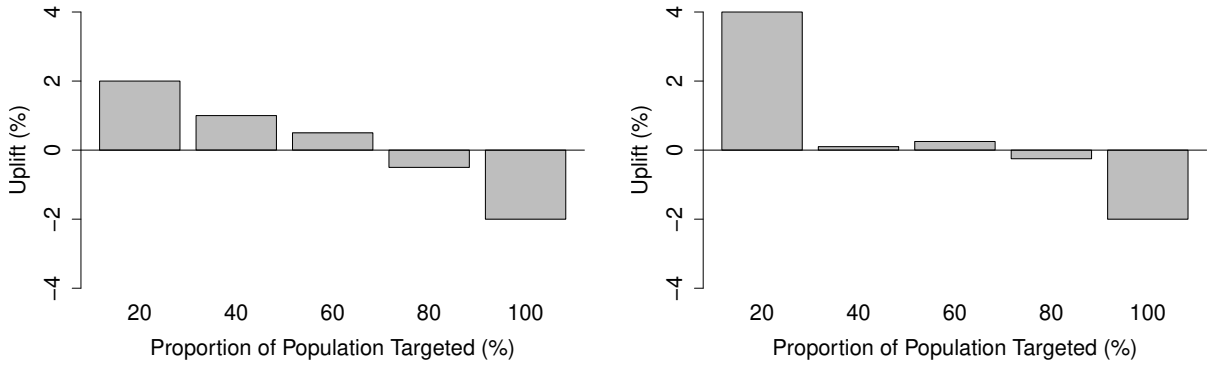

Fig. 3.3. Theoretical predicted uplift barplots with 5 panels corresponding to two different models. The left panel model has a much smaller value of $\hat{q}$ than the one on the right panel. However, $\rho = 1$ for the left panel and $\rho = 0.8$ for the right panel.

one should be the one with the highest ρ. In Figure 3.3, we show an example of two models where the one in the left has a perfect Kendall's uplift rank correlation ($\rho = 1$) but with a Qini coefficient much smaller than the model on the right panel. In this scenario, the model with $\rho = 0.8$ may be preferred since its Qini coefficient is much higher.

We propose an appropriate combination of (3.2.4) and (3.2.5): the *adjusted Qini coefficient* which is given by

$$\hat{q}_{\text{adj}} = \rho \max\{0, \hat{q}\}. \tag{3.2.7}$$

Maximizing the adjusted Qini coefficient maximizes the Qini coefficient and simultaneously promotes grouping the individuals in decreasing uplift bins, which in turn result in concave Qini curves. In fact, the observed uplift $\bar{u}_k$ in the kth bin B_k can be interpreted as the Qini curve's slope between two points $\Big(\phi_k, g(\phi_k)\Big)$ and $\Big(\phi_{k+1}, g(\phi_{k+1})\Big)$, $k \in 1, \ldots, J$. This is shown in Section 3.7.1 of the appendix. For the moment, suppose that $\bar{u}_k \approx \nabla g_k \stackrel{\text{def}}{=} \{g(\phi_{k+1}) - g(\phi_k)\}/(\phi_{k+1} - \phi_k)$. That is, $\bar{u}_k$ approximates the derivative of the Qini curve at ϕ_k. Maximizing the Kendall's upift rank correlation pushes the sequence $\{\bar{u}_k\}$ to be decreasing. In other words, it pushes $\text{sign}(\bar{u}_k - \bar{u}_{k+1})$ to be positive. Note that $\bar{u}_k - \bar{u}_{k+1} \approx \nabla g_k - \nabla g_{k+1}$, which can be seen as an estimate of the second derivative of the Qini curve. Then, maximizing the Kendall's uplift rank correlation is pushing for concavity in the Qini curve.

A note on the estimation of the Qini curve. The number of bins J may be seen as a hyper-parameter. Its choice will certainly affect the computation of the adjusted Qini coefficient. In practice, the sample is divided into quintiles ($J = 5$) or deciles ($J = 10$). In order to have a hint on what adequate values for J are, suppose that the relative incremental uplift function $g(\phi)$ is twice-differentiable, with bounded second derivative. Consider the trapezoid rule approximation to the integral q based on J bins. Let us assume that the bin sizes are proportional to $1/J$. It is well-known that under these assumptions the error of the approximation is order $\mathcal{O}(1/J^2)$. Since $g(\cdot)$ is unknown, one needs to estimate it with data. Let $\hat{g}_j$ be the estimate of $g(\phi_j)$, $j = 1, \ldots, J$. Suppose that $\hat{g}_j$ is obtained as a mean of n/J random variables observed in the j-th bin. We suppose that these random variables are independent and identically distributed with mean $g(\phi_j)$ and a certain finite variance. The weak law of large numbers says that $\hat{g}_j$ converges to $g(\phi_j)$, and the error in this approximation is of order $\mathcal{O}(J/\sqrt{n})$. It turns out that we need J to minimize $\kappa_1/J^2 + \kappa_2 J/\sqrt{n}$, where κ_1, κ_2 are constants. The solution is $J = \mathcal{O}(n^{1/6})$. So, for example, if $n \approx 1000$, then the optimal $J \approx 3$. Hence, the usual values of $J = 5$ and $J = 10$ seem reasonable to estimate the Qini [Radcliffe, 2007].

3.2.2. Brief overview of previous work on uplift modeling

The intuitive approach to uplift modeling is to build two separated classification models. Hansotia and Rukstales [2001] used the *two-model* approach which consists in direct subtraction of models for the treated and untreated groups. The asset of this technique is its simplicity. However, in many cases this approach performs poorly [Radcliffe and Surry, 2011]. Both models focus on predicting the class probabilities instead of making the best effort to predict the uplift, i.e., the difference between two probabilities. General discussions following differential response modeling and the two-model approach appeared in Hansotia and Rukstales [2002] where the technique known as *incremental value modeling* was introduced. This uses the difference in response rates in the two groups (treatment and control) as the split criterion of a regression tree. Also, Lo [2002] introduced the *true lift modeling* using a single standard logistic regression model which explicitly added interaction terms between each explanatory variable and the treatment indicator. The interaction terms measure the additional effect of each explanatory variable because of treatment. The model

yields an indirect estimation of the causal effect by subtracting the corresponding prediction probabilities, which are obtained by respectively setting the treatment indicator variable to treated and control in the fitted model. The disadvantage with this solution is that it is not optimized with respect to the goodness-of-fit measures designed for uplift. Instead, the parameters are estimated with respect to the likelihood. Our results show that estimating the regression parameters by maximizing the adjusted Qini significantly improves the Qini prediction performance of uplift models.

Most current approaches that directly model the uplift causal effect are adaptations of classification and regression trees [Breiman et al., 1984]. Rzepakowski and Jaroszewicz [2010] propose a tree-based method based on generalizing classical tree-building split criteria and pruning methods. The approach is based on the idea of comparing the distributions of outcomes in treatment and control groups, using a divergence statistic, such as the Kullback-Leibler divergence or a modified Euclidean distance [Rzepakowski and Jaroszewicz, 2012, Guelman et al., 2012, Rzepakowski and Jaroszewicz, 2010]. Another non-parametric method is discussed in [Alemi et al., 2009, Su et al., 2012]. Therein the uplift is estimated from the nearest neighbors containing at least one treated and one control observation. This method quickly becomes computationally expensive when dealing with large datasets, because the entire dataset has to be stored in order to predict the uplift for new observations. For a more detailed overview of the uplift modeling literature, the reader is referred to the works of Kane et al. [2014], Gutierrez and Gérardy [2017] and Devriendt et al. [2018].

From a complexity point of view, parametric models are simpler than non-parametric ones such as regression trees, because for parametric models, the number of parameters is kept small and fixed. Although, for many analysts prediction is the main target, from a business point of view, model interpretation is very important. Knowing which variables and how these variables discriminate between groups of clients is one of the main goals of uplift modeling for marketing. For these reasons, in this work we focus on parametric models. We develop our methodology for the logistic regression since interpretation of the odds ratios is well-known. However, our estimation procedure can be easily generalized to other parametric models.

3.3. Qini-based logistic regression for uplift

Logistic regression is a well-known parametric model for binary response variables. Given a p-dimensional predictor vector $\mathbf{x}_i$, $i \in \{1,\ldots,n\}$, logistic intercept $\theta_o \in \mathbb{R}$, and logistic regression coefficients $\boldsymbol{\beta} \in \mathbb{R}^{\mathrm{p}}$, the model is

$$p_i = p_i(\theta_o, \boldsymbol{\beta}) = \Pr(Y_i = 1 \mid \mathbf{x}_i, \theta_o, \boldsymbol{\beta}) = \Big(1 + \exp\{-(\theta_o + \mathbf{x}_i^\top \boldsymbol{\beta})\}\Big)^{-1}$$

or, equivalently, $\mathrm{logit}(p_i) = \theta_o + \mathbf{x}_i^\top \boldsymbol{\beta}$, where $\mathrm{logit}(p_i) = \log\{p_i/(1-p_i)\}$. Throughout the paper, the superscript $^\top$ stands for the transpose of a column vector or matrix. In the uplift context, one needs to add explicit interaction terms between each explanatory variable and the treatment indicator. Let γ denote the treatment effect, $\boldsymbol{\beta}$, the vector of main effects, $\boldsymbol{\delta}$, the vector of interactions effects, and θ_o, the intercept. The model is

$$p_i(\theta_o, \boldsymbol{\theta}) = \Pr(Y_i = 1 \mid \mathbf{x}_i, t_i, \theta_o, \boldsymbol{\theta}) = \Big(1 + \exp\{-(\theta_o + \gamma t_i + \mathbf{x}_i^\top [\boldsymbol{\beta} + t_i \boldsymbol{\delta}])\}\Big)^{-1}, \qquad (3.3.1)$$

where $\boldsymbol{\theta} = (\boldsymbol{\beta}, \gamma, \boldsymbol{\delta})$, denotes all model parameters except for the intercept θ_o. The likelihood function associated with the uplift model is

$$\mathcal{L}(\theta_o, \boldsymbol{\theta}) = \prod_{i=1}^{n} p_i(\theta_o, \boldsymbol{\theta})^{y_i} \{1 - p_i(\theta_o, \boldsymbol{\theta})\}^{(1-y_i)}, \qquad (3.3.2)$$

where $\{y_i : i = 1,\ldots,n\}$ are the observed response variables. The maximum likelihood estimates of $(\theta_o, \boldsymbol{\theta})$ will be denoted by $(\hat{\theta}_o, \hat{\boldsymbol{\theta}})$, with $\hat{\boldsymbol{\theta}} = (\hat{\boldsymbol{\beta}}, \hat{\gamma}, \hat{\boldsymbol{\delta}})$. The predicted uplift associated with the covariates vector $\mathbf{x}_{n+1}$ of a future individual is estimated by

$$\hat{u}(\mathbf{x}_{n+1}) = \Big(1 + \exp\{-(\hat{\theta}_o + \hat{\gamma} + \mathbf{x}_{n+1}^\top [\hat{\boldsymbol{\beta}} + \hat{\boldsymbol{\delta}}])\}\Big)^{-1} - \Big(1 + \exp\{-(\hat{\theta}_o + \mathbf{x}_{n+1}^\top \hat{\boldsymbol{\beta}})\}\Big)^{-1}.$$

We propose to select a regression model that maximizes the adjusted Qini coefficient. To realistically search for a good model, we conceived a searching method based on Latin hypercube sampling of the regression coefficients space combined with a lasso penalization of the log-likelihood. The procedure is explained in the following sections.

3.3.1. Estimation of the Qini maximizer

Because the adjusted Qini function is not straightforward to optimize with respect to the parameters, one needs to explore the parameter space in order to find the maximum of the adjusted Qini.

Latin hypercube sampling (LHS) is a statistical method for quasi-random sampling based on a multivariate probability law inspired by the Monte Carlo method [McKay et al., 2000]. The method performs the sampling by ensuring that each sample is positioned in a space Ω of dimension p as the only sample in each hyperplane of dimension $p-1$ aligned with the coordinates that define its position. Each sample is therefore positioned according to the position of previously positioned samples to ensure that they do not have any common coordinates in the Ω space. When sampling a function of p variables, the range of each variable is divided into M equally probable intervals. M sample points are then placed to satisfy the Latin hypercube requirements; this forces the number of divisions, M, to be equal for each variable. Also this sampling scheme does not require more samples for more dimensions (variables); this independence is one of the main advantages of this sampling scheme. We use LHS to find the coefficient parameters that maximize the adjusted Qini. The procedure to search for the Qini maximizer is explained next. It is based on the lasso penalized likelihood and several LHS structures.

3.3.1.1. *Penalized log-likelihood*

In the context of linear regression, the effectiveness of penalization has been amply supported practically and theoretically in several studies. In order to decrease the mean squared error of least squares estimates, ridge regression [Hoerl et Kennard, 1970] has been proposed as a trade-off between bias and variance. This technique adds an L_2-norm penalization term to the least squares loss. The *lasso* (least absolute shrinkage and selection operator) penalization technique [Tibshirani, 1996] uses an L_1-norm penalization which sets some of the regression coefficients to zero (sparse selection) while shrinking the rest. The elastic net penalization technique [Zou et Hastie, 2005] linearly combines the L_1 and L_2-norms to provide better prediction in the presence of collinearity. Other penalization techniques such as *scad* [Fan and Li, 2001] and bridge regression [Frank and Friedman, 1993], offer interesting theoretical properties, including consistency.

Here, we focus on sparse estimation of the coefficients. That is, the selection of a small subset of features to predict the response. This is often achieved with a L_1-norm penalization. Given $\lambda \in \mathbb{R}^+$, in the context of linear regression, the *lasso* penalization [Tibshirani, 1996] finds the estimate of the coefficients $\hat{\boldsymbol{\beta}}(\lambda)$ that maximizes the penalized log-likelihood, say

$\ell(\beta) + \lambda \sum_{j=1}^{p} |\beta_j|$. Setting the penalization constant $\lambda = 0$ returns the least squares estimates which performs no shrinking and no selection. For $\lambda > 0$, the regression coefficients $\hat{\boldsymbol{\beta}}(\lambda)$ are shrunk towards zero, and some of them are set to zero (sparse selection). Friedman et al. [2007] proposed a fast pathwise coordinate descent method to find $\hat{\boldsymbol{\beta}}(\lambda)$, using the current estimates as warm starts. In practice, the value of λ is unknown. Cross-validation is often used to search for a good value of the penalization constant. The least angle regression (or LARS algorithm) efficiently computes a path of values of $\hat{\boldsymbol{\beta}}(\lambda)$ over a sequence of values of $\lambda = \lambda_1 < \cdots < \lambda_j < \cdots < \lambda_{\min(n,p)}$, for which the parameter dimension changes [Efron et al., 2004]. The entire sequence of steps in the LARS algorithm with $p < n$ variables requires $\mathcal{O}(p^3 + np^2)$ computations, which is the cost of a single least squares fit on p variables. Extensions to generalized linear models with nonlinear loss functions require some form of approximation. In particular, for the logistic regression case, which is our model of interest, Friedman et al. [2010] extend the pathwise coordinate descent algorithm [Friedman et al., 2007] by first, approximating the log-likelihood (quadratic Taylor expansion about current estimates), and then using coordinate descent to solve the penalized weighted least-squares problem. The algorithm computes the path of solutions for a decreasing sequence of values for $\lambda = \lambda_{\min(n,p)} > \cdots > \lambda_j > \cdots > \lambda_1$, starting at the smallest value for which the entire vector $\hat{\boldsymbol{\beta}} = 0$. The algorithm works on large datasets, and is publicly available through the R Package **glmnet** [Friedman et al., 2009], which we use in this work. In what follows, we will refer to the sequence of regularizing constant values given by **glmnet** as the *logistic-lasso sequence*.

3.3.1.2. *Qini-optimized uplift regression*

Recall the uplift model likelihood given in (3.3.2). The vector of parameters $\boldsymbol{\theta} = (\boldsymbol{\beta}, \gamma, \boldsymbol{\delta})$ is a $p' = (2p + 1)$-dimensional vector. Because of the considerations mentioned in the previous sections, in order to select an appropriate sparse model for uplift, we adapt the lasso algorithm to explore a relatively small set of reasonable models, so as to avoid an exhaustive model search. The penalized uplift model log-likelihood is given by

$$\ell(\theta_o, \boldsymbol{\theta} \mid \lambda) = \sum_{i=1}^{n} \Big(y_i \log\{p_i(\theta_o, \boldsymbol{\theta})\} + (1 - y_i) \log\{1 - p_i(\theta_o, \boldsymbol{\theta})\} \Big) + \lambda \|\boldsymbol{\theta}\|_1, \tag{3.3.3}$$

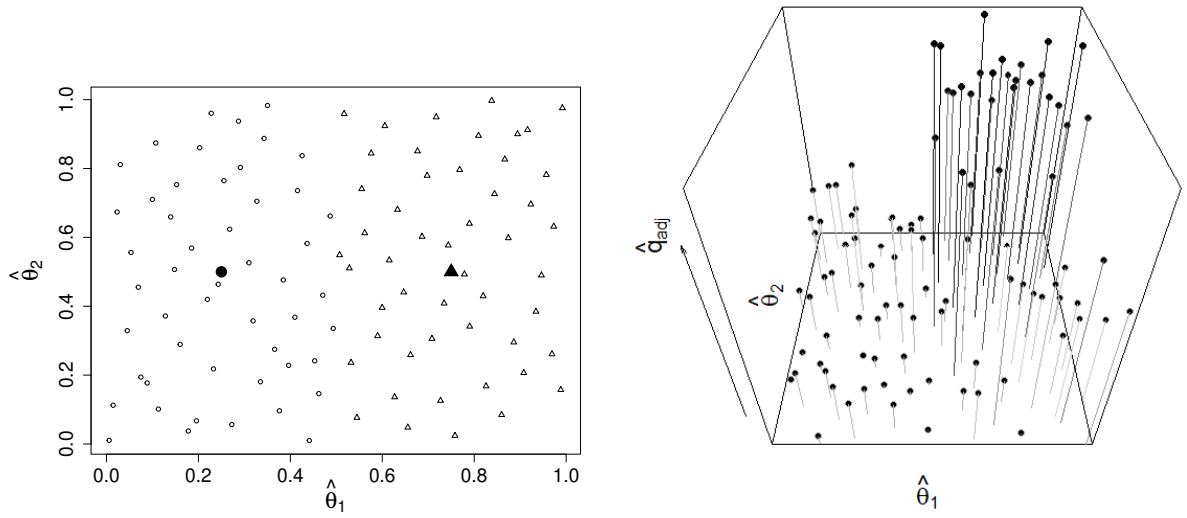

Fig. 3.4. Example of a 2-dimensional Latin hypercube sampling for $\boldsymbol{\theta} = (\theta_1, \theta_2)$. For the regularization constants λ_1,λ_2, there are two penalized estimates $\hat{\boldsymbol{\theta}}(\hat{\lambda}_j)$, $j = 1,2$ (solid symbols in the left panel). The idea is to sample points (outlined symbols in the left panel) centered at $\hat{\boldsymbol{\theta}}(\hat{\lambda}_j)$, $j = 1,2$ and then to compute $\hat{q}_{\text{adj}}$ for all these points (right panel).

where $p_i(\theta_o, \boldsymbol{\theta})$ is as in (3.3.1), and $\|\cdot\|_1$ stands for the L_1-norm. For any given λ, the parameters that maximize the penalized log-likelihood (3.3.3) are denoted by

$$\left(\hat{\theta}_o(\lambda), \hat{\boldsymbol{\theta}}(\lambda)\right) = \underset{\theta_o, \boldsymbol{\theta}}{\operatorname{argmax}} \ \ell(\theta_o, \boldsymbol{\theta} \mid \lambda). \tag{3.3.4}$$

Applying the pathwise coordinate descent algorithm to the uplift model, we get a sequence of critical penalization values $\lambda_1 < \cdots < \lambda_{\min\{n,p'\}}$ and corresponding model parameters $\{(\hat{\theta}_o(\lambda_j), \hat{\boldsymbol{\theta}}(\lambda_j))\}_{j=1}^{\min\{n,p'\}}$ associated with different model dimensions $m \in \{1, \ldots, p'\}$.

3.3.1.3. *The LHS search*

For each λ_j, $j = 1,...,\min\{n,p'\}$, we generate a LHS comprising L points $\{\hat{\boldsymbol{\theta}}(\hat{\lambda}_j)_l\}_{l=1}^{L}$ in the neighborhood of $\hat{\boldsymbol{\theta}}(\hat{\lambda}_j)$, and evaluate the adjusted Qini on each of these points. The optimal coefficients are estimated as those coefficients among the $(\min\{n,p'\} \times L)$ LHS points that maximize the adjusted Qini. Figure 3.4 illustrates the procedure.

3.3.1.4. *A simpler estimate of the Qini-based uplift regression parameters*

We also consider a simpler two-stage procedure to find a good uplift model. This one is based only on the penalized log-likelihood and does not require the posterior LHS-based

search for the optimal coefficients. Let $\hat{q}_{\text{adj}}(\lambda)$ be the adjusted Qini coefficient associated with the model with parameters $(\hat{\theta}_o(\lambda), \hat{\boldsymbol{\theta}}(\lambda))$. The first stage of the procedure solves

$$\hat{\lambda} = \text{argmax } \Big(\hat{q}_{\text{adj}}(\lambda_j) : j = 1, \ldots, \min\{n,p'\}\Big), \tag{3.3.5}$$

where as before, the sequence $\lambda_1 < \cdots < \lambda_{\min\{n,p'\}}$ is the logistic-lasso sequence. On the second-stage, a reduced model that only include those explanatory variables associated with non-zero entries of the estimated parameter $\hat{\boldsymbol{\theta}}(\hat{\lambda})$ is fitted without penalization, that is, with λ set to zero. The parameters are estimated with maximum likelihood. This yields the selected model. In our simulations, this model performs well. It also serves to show that the value of the penalization parameter $\hat{\lambda} > 0$ that maximizes the Qini or adjusted Qini, is not necessarily the same as the one that maximizes the penalized log-likelihood.

3.4. Simulations

We conduct a simulation study to examine the performance of Qini-based uplift regression. More specifically, we compare the different proposed parameter estimation methods by varying both the complexity of the data, and the number of predictors in the model. In order to create realistic scenarios, we based our artificial data generation on the home insurance policy data described in the introduction. We take advantage of the opportunity to have real data in order to generate realistic scenarios. We proceed as follows. First, we fit a non-parametric model on a random sample $\mathcal{D}$ of the home insurance policy data. Based on the resulting model, we can extract the probabilities

$$p_1(\mathbf{x}) = \Pr(Y = 1 \mid \mathbf{x}, T = 1), \text{ and } p_0(\mathbf{x}) = \Pr(Y = 1 \mid \mathbf{x}, T = 0),$$

for any given value $\mathbf{x}$. Then, we use these probabilities to generate synthetic data. We start by creating a bootstrap sample $\mathcal{S}$ of size $n_{\mathcal{S}}$ from $\mathcal{D}$. For each observation $\mathbf{x}_i \in \mathcal{S}$, we generate a random vector $\tilde{y}_i = (\tilde{y}_{i0}, \tilde{y}_{i1})$, where $\tilde{y}_{i0}$ is the binary outcome of a Bernoulli trial with success probability $p_0(\mathbf{x}_i)$, and $\tilde{y}_{i1}$ is the binary outcome of a Bernoulli trial with success probability $p_1(\mathbf{x}_i)$, $i = 1, \ldots, n_{\mathcal{S}}$. The augmented synthetic dataset $\{(\mathbf{x}_i, t_i, \tilde{y}_i)\}_{i=1}^{n_{\mathcal{S}}}$, which we are going to denote again by $\mathcal{S}$, is the data of interest in the simulation. For each simulated dataset, we implement the following models:

(a) a multivariate logistic regression without penalization as in (3.3.2). This is the baseline model, and we will refer to it as *Baseline*.

(b) our Qini-based uplift regression model that uses several LHS structures to search for the optimal parameters (see Section 3.3.1.2). We denote this model by *Q+LHS*.

(c) our Qini-based uplift regression model that uses the simpler estimate of the regression parameters as explained in Section 3.3.1.4. We denote this model by *Q+lasso*.

Note that our *Q+LHS* method is a derivative free optimization procedure. Another derivative free optimization method is the well-known Nelder-Mead method [Nelder and Mead, 1965]. In order to obtain benchmarks for the LHS search, we implement the following Nelder-Mead Qini-based uplift regression models:

(d) *Base+NM*, which initializes the Nelder-Mead algorithm with the maximum likelihood estimates (the *Baseline* model solution) and which searches for coefficients that maximize the adjusted Qini coefficient.

(e) *Q+NM*, which initializes the Nelder-Mead algorithm with coefficients from the lasso-sequence (the first-stage of *Q+lasso*) and which searches for coefficients that maximize the adjusted Qini coefficient.

Data generation. As discussed in Section 3.2.2, several tree-based methods have been suggested in the uplift literature. Here, we use the uplift random forest [Guelman et al., 2012] as the data generating process. We chose this method due to its simplicity, and because it is readily available in R through the Package **uplift** [Guelman, 2014]. Algorithm 3.1 describes the associated methodology.

Algorithm 3.1 Uplift Random Forest [Guelman et al., 2012]

1: $B \leftarrow$ number of bootstrap samples
2: **for** $b = 1$ to B **do**
3: Draw a bootstrap sample of size $n_{\mathcal{S}}$ with replacement from the data
4: Fit an uplift decision tree T_b to the bootstrap data
5: Output the ensemble of uplift trees T_b; $b = \{1,2,...,B\}$ and the predicted probabilities $\Pr(Y = 1 \mid \mathbf{x}, T = 1)$ and $\Pr(Y = 1 \mid \mathbf{x}, T = 0)$ obtained by averaging the predictions of the individual trees in the ensemble

In our simulations, we vary two parameters: the depth of the trees used to fit the uplift random forests, and the number of variables k considered when fitting the uplift logistic models. Algorithm 3.2 details the procedure.

We define 21 scenarios by varying two parameters: (i) the depth of the uplift random forest trees used to generate the synthetic data is either 1, 2 or 3, and (ii) the number of total covariates k considered to build the forest model, $k \in \{10, 20, 30, 50, 75, 90, 97\}$. For scenarios 1-7, the depth is 1, and we vary k; for scenarios 8-14, the depth is 2; and for scenarios 15-21, the depth is 3. Each scenario was replicated 100 times.

Algorithm 3.2 Simulations

1: $M \leftarrow$ number of simulations
2: **for** $m = 1$ to M **do**
3: Draw a stratified sample $\mathcal{D}_m$ of size $n_{\mathcal{S}}$ without replacement from the data
4: Fit an uplift random forest of a given tree depth to the sampled data $\mathcal{D}_m$
5: Generate a complete (i.e., including the binary responses) synthetic data $\mathcal{S}_m$ with data $\mathcal{D}_m$
6: Fit an uplift random forest of the same given tree depth to the synthetic data $\mathcal{S}_m$ using all p predictors
7: **for** each k **do**
8: Sample $k \leq p$ random predictors for modeling
9: Fit the different uplift logistic models with k predictors on $\mathcal{S}_m$
10: Output the average and standard errors of the metrics for each method

The sample means of $\hat{q}$ (3.2.3), and $\hat{q}_{\text{adj}}$ (3.2.7) and their corresponding standard errors are reported in Tables 3.3, and 3.4, respectively. Since the conclusions are similar for the three tree depths, we report only the results associated with depth 3, that is, for the most complex model. For each comparison group, we also report the corresponding performance of an uplift random forest (*RF*) fitted to the synthetic data using all available predictors (i.e., $k = 97$).

In Table 3.3, we compare the performance of the models according to the Qini coefficient $\hat{q}$. We observe that performing variable selection driven by the adjusted Qini coefficient (*Q+lasso*) significantly improves the performance of the baseline model. As expected, the models using a LHS-driven optimization perform better. The performance of *Q+lasso* is

Tab. 3.3. Qini coefficient ($\hat{q}$) averaged over 100 simulations. Standard-errors are reported in parenthesis. The *RF* model (with $k = 97$ and depth= 3) performance is 1.60 (0.039). $n = 5000$ observations.

k	Baseline	Q+lasso	Q+LHS	Base+NM	Q+NM
10	0.51 (0.023)	0.56 (0.020)	0.76 (0.021)	0.64 (0.019)	0.68 (0.021)
20	0.74 (0.020)	0.83 (0.022)	1.03 (0.024)	0.93 (0.025)	0.95 (0.020)
30	0.94 (0.023)	1.01 (0.023)	1.20 (0.023)	1.04 (0.024)	1.13 (0.026)
50	1.09 (0.039)	1.19 (0.025)	1.48 (0.026)	1.23 (0.031)	1.35 (0.029)
75	0.97 (0.073)	1.37 (0.036)	1.48 (0.033)	1.35 (0.059)	1.49 (0.056)
90	1.00 (0.084)	1.36 (0.045)	1.54 (0.036)	1.40 (0.071)	1.47 (0.064)
97	0.83 (0.083)	1.36 (0.055)	1.59 (0.037)	1.31 (0.088)	1.41 (0.079)

Tab. 3.4. Adjusted Qini coefficient ($\hat{q}_{\text{adj}}$) averaged over 100 simulations. Standard-errors are reported in parenthesis. The *RF* model (with $k = 97$ and depth= 3) performance is 1.40 (0.048). $n = 5000$ observations.

k	Baseline	Q+lasso	Q+LHS	Base+NM	Q+NM
10	0.36 (0.027)	0.39 (0.026)	0.72 (0.024)	0.54 (0.025)	0.61 (0.027)
20	0.58 (0.025)	0.68 (0.027)	1.02 (0.025)	0.83 (0.025)	0.91 (0.027)
30	0.83 (0.026)	0.92 (0.029)	1.20 (0.023)	1.03 (0.029)	1.13 (0.029)
50	0.97 (0.041)	1.12 (0.026)	1.48 (0.026)	1.23 (0.033)	1.35 (0.028)
75	0.90 (0.069)	1.32 (0.039)	1.48 (0.033)	1.35 (0.061)	1.49 (0.058)
90	0.94 (0.081)	1.28 (0.049)	1.54 (0.036)	1.37 (0.071)	1.46 (0.069)
97	0.79 (0.085)	1.30 (0.063)	1.59 (0.037)	1.23 (0.089)	1.32 (0.081)

similar to the *Base+NM* performance, and is slightly lower than *Q+LHS* and *Q+NM*. Using the lasso-sequence in order to initialize the posterior searches, that is using *Q+LHS* or *Q+NM*, improves the performance of the *Q+lasso* and *Base+NM* models. However, for the *Q+NM* solution, the standard error of the Qini coefficient increases with k. It is almost twice the standard errors from *Q+LHS* for $k \geq 75$. Using all predictors ($k = 97$) enables the *Q+LHS* models to achieve the same performance as the *RF* model.

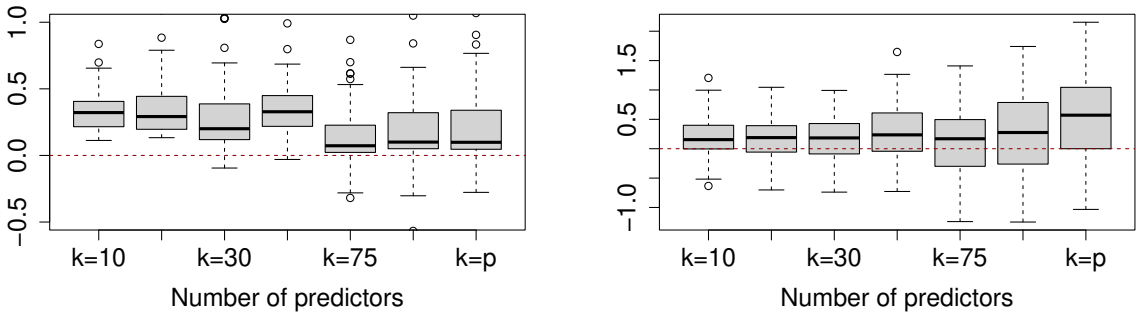

Fig. 3.5. Comparison between *Q+LHS* and *Q+lasso* (left panel) and *Q+LHS* and *Base+NM* (right panel). Boxplots of the differences in terms of $\hat{q}_{\text{adj}}$ as a function of the number of predictors used in the models over the 100 simulations with $n = 5000$ observations. The black lines represent the differences' medians.

In Table 3.4, we compare the main statistic of interest, that is, the adjusted Qini coefficient. These results corroborate the findings from the previous table. Guiding the variable selection by this statistic leads to significant improvements from the results of the baseline model. Similarly, estimating the parameters with a derivative-free maximization of the adjusted Qini coefficient improves the performance of the models in comparison to maximum likelihood estimation. The best results are obtained with models that make use of the lasso-sequence in order to explore the space of the parameters (*Q+LHS* and *Q+NM*). As in Table 3.3, the *Q+LHS* models give the best results. Moreover, when using all available predictors, the *Q+LHS* models outperform both the *RF* and the *Q+NM* models.

The difference in performance between *Q+LHS* and *Q+lasso* is significant. The left panel of Figure 3.5 display boxplots of the differences in performance between these two models in each simulation. It is clear that *Q+LHS* performs much better than *Q+lasso* most of the time. The relative performance of *Q+lasso* improves slightly when the number of predictors approaches the total number of predictors available ($p = 97$). The same pattern is observed in the difference of performance between *Q+LHS* and *Base+NM* (see right panel of Figure 3.5). This confirms the importance of the use of the lasso-sequence in order to estimate the model's parameters.

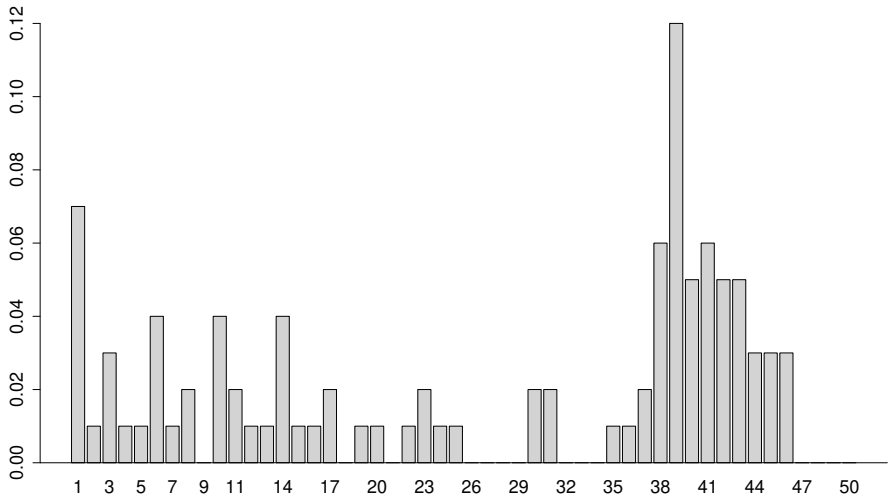

Fig. 3.6. Barplot of the distribution of the *Q+lasso* rankings associated with $\tilde{\lambda}$.

Choosing an appropriate sparse model. Next, we compare the models selected by the Qini-based uplift regression and the classical *lasso* approach where the penalization constant is chosen by cross-validation on the log-likelihood. Consider the values of the logistic-lasso sequence $\lambda_1 < \cdots < \lambda_{\min\{n,p'\}}$ sorted according to the results of *Q+lasso*. That is, consider the permutation $(\pi_1,\pi_2,\ldots,\pi_{\min\{n,p'\}})$ of $(1,2,\ldots,\min\{n,p'\})$ so that $\lambda_{\pi_{\min\{n,p'\}}} \preceq \cdots \preceq \lambda_{\pi_1}$, where the relation $\lambda_{\pi_i} \preceq \lambda_{\pi_j}$ means that $\hat{q}_{\text{adj}}(\lambda_{\pi_i}) \leq \hat{q}_{\text{adj}}(\lambda_{\pi_j})$. We look at the value of $\tilde{\lambda} \in \{\lambda_1,\ldots,\lambda_{\min\{n,p'\}}\}$ that is chosen by cross-validation of the log-likelihood, and report its ranking based on the sorted Q+lasso sequence $\lambda_{\pi_{\min\{n,p'\}}} \preceq \cdots \preceq \lambda_{\pi_1}$. Comparing the two models is equivalent to check when lasso finds the "best" λ, that is, when λ_{π_1} is equal to $\tilde{\lambda}$. We repeated the simulation 100 times, each time using $n = 5000$ observations randomly selected from the full data set. The barplot in Figure 3.6 shows that only 7% of the time $\tilde{\lambda}$ also maximizes $\hat{q}_{\text{adj}}$. Observe that 4% of the time $\tilde{\lambda}$ is positioned 10th in the ranking, and 12% of the time, it is positioned 39th. These results clearly show that choosing the penalization constant by cross-validation of the log-likelihood does not solve the problem of maximizing the adjusted Qini coefficient, and therefore, is not necessarily appropriate for uplift models.

3.5. Insurance data analysis

Recall the insurance data introduced in Section 3.1. The insurance company is interested in designing retention strategies to minimize its policyholders' attrition rate. An experimental loyalty campaign was implemented, from which policies coming up for renewal were randomly allocated into one of the following two groups: treatment group, and control group. The goal of this section is to analyze the marketing campaign results so as to identify both the set of persuadable clients, and the set of clients that should not be disturbed.

3.5.1. Parameter estimation

We fit the Qini-based uplift regression *Q+LHS* to the data using the methodology described in the previous sections. For comparison purposes, we also considered the model *Q+lasso*. Although, we are interested in interpretable parametric models, we also fit an uplift random forest (*RF*) as a benchmark for our comparison.

In order to choose the optimal value from the logistic-lasso sequence of penalization constant values $\{\lambda_1, \ldots, \lambda_{\min\{n,p'\}}\}$, we use a 5-fold cross-validation on the adjusted Qini statistics. We compare the resulting models with the one yielded by applying the classical lasso approach, that is, with the model associated with the value of the penalization constant that maximizes the cross-validated log-likelihood. We will refer to this latter model as *MLE+lasso*. The two-stage approach was used in all the cases. The first stage estimates the best λ in the logistic-lasso sequence by cross-validation. The second stage fits the non penalized logistic regression model with the subset of selected variables.

For the *Q+LHS* model, for each λ_j, we perform a LHS search to directly maximize the adjusted Qini coefficient (see Appendix 3.7.2 for more details). In this case, applying the LHS search leads to the selection of the model associated with the penalization constant $\approx 3 \times 10^{-5}$, while for the *MLE+lasso* logistic regression, it is $\approx 10^{-3}$. The number of selected variables are, respectively, 163 and 53 out of a total of 195 main and interaction effect terms. If we use the simple search method described in Section 3.3.1.4, which was denoted by *Q+lasso* in the previous section, the optimal value of the penalization constant is $\approx 4 \times 10^{-5}$. In this case, the number of selected variables is 156.

In order to have a fair comparison, we followed a process similar to the one applied to the *Q+LHS* model to fit the uplift *RF* model. The accuracy of a random forest can be sensitive

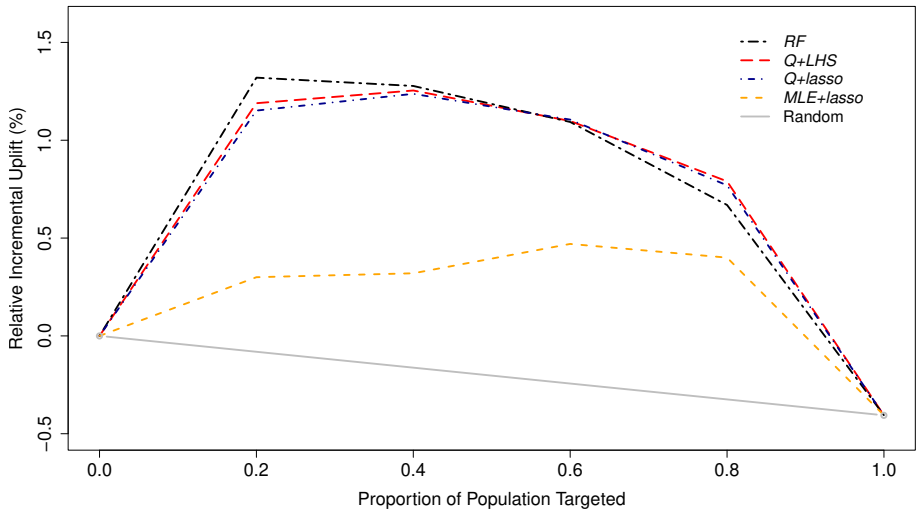

Fig. 3.7. Performance of the final models based on the Qini curves.

to several training hyper-parameters: number of trees (from 10 to 200, with increments of 10 trees), maximum depth on each tree (from 1 to 10), minimum number of observations per node (either 100, 200 or 500), and split criterion, either Euclidean distance or Kullback-Leibler divergence; see Guelman et al. [2012] for more details on the split criteria. The optimal *RF* hyper-parameters were those that maximized the adjusted Qini coefficient with a 5-fold cross-validation over the grid given by the possible values of the hyper-parameters. Hence, the chosen *RF* was composed of 100 trees of maximum depth 3, with a minimum of 200 observations per node, with trees splitted according to the Kullback-Leibler criterion. The final *RF* was fitted using all available data.

Figures 3.7 and 3.8 show the performance of the models in terms of the Qini curve and the uplift barplot (Kendall's rank correlation), respectively.

As expected, the Qini-based uplift regression models outperform the classic lasso approach, that is, the *MLE+lasso* model, both in terms of overall adjusted Qini coefficient (see Table 3.5) and in terms of sorting the individuals in decreasing order of uplift (see Figure 3.8). Surprisingly, the non-penalized model, that is, the model whose parameters are estimated by maximizing the non-penalized likelihood, yields $\hat{q}_{\text{adj}} = 0.85$ (with 195 non-zero coefficients) which is larger than $\hat{q}_{\text{adj}}$ for *MLE+lasso* ($\hat{q}_{\text{adj}} = 0.39$) (see Table 3.5). However, we see an increase in terms of $\hat{q}_{\text{adj}}$ for *Q+lasso* ($\hat{q}_{\text{adj}} = 1.02$). Therefore, for this dataset, eliminating $\approx 20\%$ of the coefficients yielded a better uplift estimation in terms of adjusted

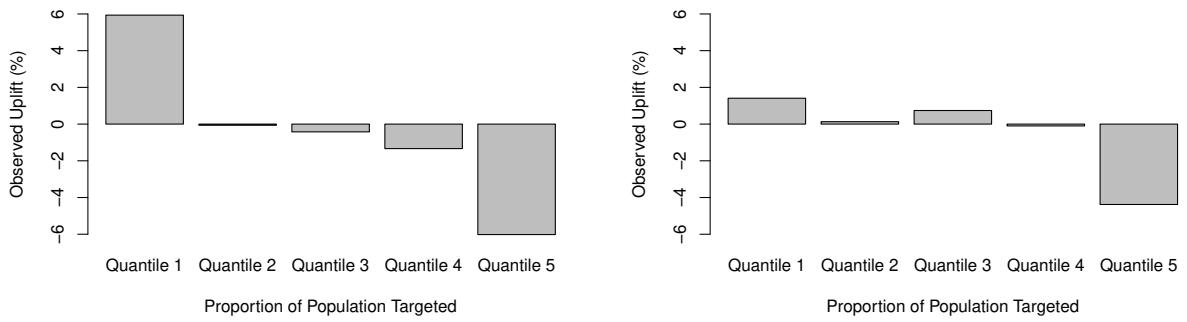

Fig. 3.8. Performance of the Q+LHS and MLE+lasso models based on uplift Kendall's correlations. The left barplot corresponds to Q+LHS ($\rho = 1$) and the right barplot to to the MLE+lasso model ($\rho = 0.8$). The observed uplift was computed as in (3.2.6).

Qini. Moreover, the performance of the Qini-based uplift regression is slightly lower, but comparable to the one of the *RF* model ($\hat{q}_{\text{adj}} = 1.07$), even though the *RF* is more complex. Indeed, with 100 trees of depth 3, the *RF* model can both model non-linearity and interactions between covariates, which makes interpretation of the final *RF* hard. However, the in-sample performance is similar to our models. This is interesting because it shows that it is possible to get powerful models without loosing interpretation when estimating the parameters with the adjusted Qini function.

Based on the final models, we can identify both (i) the group of clients at the top 20% of predicted uplifts, that is, the clients to pursue in the marketing, and (ii) the group of clients at the bottom 20% of predicted uplifts, that is, the clients not to disturb with any marketing. The group of clients at the top 20% of predicted uplifts provides very strong return on investment cases when applied to retention activities. For example, by only targeting the persuadable customers in an outbound marketing campaign, the contact costs and hence the return per unit spend can be dramatically improved [Radcliffe and Surry, 2011]. We observe from Table 3.5 that the *RF* model finds the highest top 20% uplift group which presents an uplift of 6.41%, while the *Q+LHS* model finds the lowest bottom 20% uplift group which shows an uplift of approximately -6%. Note that the overall uplift is approximately -0.5%. The performance of the *Q+lasso* model is slightly lower than the one of the *Q+LHS* model.

Tab. 3.5. Uplift results of the top and bottom 20% observed uplift groups for the models *MLE, RF, Q+LHS, Q+lasso,* and *MLE+lasso.* The adjusted Qini coefficient $\hat{q}_{\text{adj}}$ is given for each model.

Method	$\hat{q}_{\text{adj}}$	**Top** 20%	**Bottom** 20%
MLE	0.85	4.98%	−5.87%
RF	1.07	6.41%	−5.36%
Q+LHS	1.03	5.94%	−6.02%
Q+lasso	1.02	5.76%	−5.93%
MLE+lasso	0.39	1.41%	−4.38%

3.5.2. Model interpretation

Because the *Q+LHS* model is a logistic model, we can interpret the results through its coefficients. The usual approach is that of the odds ratios. For a specific variable, the odds ratio is computed by fixing the other covariates at fixed values, such as their mean, which is what we have done here. Since the company is not interested in all the variables included in the model, we will analyze a subset with relevant interpretation for the business. In addition, for confidentiality reasons, we do not show the analysis of variables related to the insurance premium. The following variables are chosen by our model: client's *credit score, age, gender* and *marital status* (single or not); client's *products:* whether it is a single line (home) or a multi-line (automobile and home) account; client's number of *automobile policies, mortgages* and *residences*; and whether the client has *extra options* (additional endorsements) in his/her account. For a model with p variables, the odds ratio $\text{OR}_{X_j}(t)$ associated with one-unit change for a specific variable X_j is given by

$$\frac{\Pr(Y=1 \mid X_j = x_j+1, T=t)/\Pr(Y=0 \mid X_j = x_j+1, T=t)}{\Pr(Y=1 \mid X_j = x_j, T=t)/\Pr(Y=0 \mid X_j = x_j, T=t)}$$
$$= \frac{\exp(\hat{\beta}_j(x_j+1) + \hat{\delta}_j t\ (x_j+1))}{\exp(\hat{\beta}_j x_j + \hat{\delta}_j t\ x_j)} = \exp(\hat{\beta}_j)\ \exp(\hat{\delta}_j t)\ ,$$

where T is the treatment indicator, and where for a binary variable, such as *extra options*, x_j is set to 0 in the above expression. When the company does not call a client ($T=0$), the odds ratio is $\text{OR}_{X_j}(0) = \exp(\hat{\beta}_j)$ and when the company calls a client ($T=1$), the odds ratio is $\text{OR}_{X_j}(1) = \exp(\hat{\beta}_j)\ \exp(\hat{\delta}_j) = \text{OR}_{X_j}(0)\ \exp(\hat{\delta}_j)$. Table 3.6 gives the estimated odds ratios

Tab. 3.6. Odds ratios and 95% confidence intervals estimated by the Qini-based uplift regression model (*Q+LHS*) for some of the selected variables. The symbol * indicates significant coefficients; the symbol † indicates significant interaction terms.

	$\exp(\hat{\beta}_j)$	**CI** (95%)	$\exp(\hat{\beta}_j + \hat{\delta}_j)$	**CI** (95%)
Credit Score	0.998	(0.994, 1.003)	1.001	(0.999, 1.001)
Age (Years)	0.995	(0.969, 1.023)	0.998	(0.989, 1.006)
Gender				
Male	1.297	(0.778, 2.163)	0.962	(0.814, 1.135)
Marital Status				
Single†	2.759	(0.956, 7.963)	0.697	(0.452, 1.077)
Products				
Auto and Home	1.619	(0.586, 4.472)	*1.418	(1.017, 1.977)
Auto Policies Count	2.106	(0.653, 6.790)	*1.996	(1.368, 2.918)
Mortgage Count	1.381	(0.789, 2.418)	*1.366	(1.137, 1.642)
Residences Count†	0.505	(0.172, 1.489)	*1.788	(1.122, 2.848)
Extra Options†	*0.350	(0.141, 0.874)	1.276	(0.949, 1.715)

$\mathrm{OR}_{X_j}(0)$ and $\mathrm{OR}_{X_j}(1)$ with 95% confidence intervals. We can see, for example, that when the company does not call a client who has *extra options* in his/her policy, his/her odds ratio of renewing the policy is 0.35 while when the company calls that same client, the odds ratio becomes 1.28. It is worth noting that the interaction terms between the treatment indicator and the variables *Single*, *Residences Count* and *Extra Options* are significant.

Next, we use the *Q+LHS* model predictions to describe in more detail the two extreme groups found by the model (top 20% and bottom 20% predicted uplifts). This provides the insurance company with typical profiles of clients that are persuadables (top 20%), and clients that should not be targeted (bottom 20%). Table 3.7 shows descriptive statistics of some selected predictors for both groups. A MANOVA comprising only these two groups for the selected variables, followed by ANOVA tables involving individual selected variables separately, show that all mean differences were statistically significant (p-value < 0.0005). For example, there are 30% more single clients in the do-not-disturb group than in the

Tab. 3.7. Profiles of the persuadables and do not disturb groups predicted by the Qini-based uplift regression model (*Q+LHS*) for some of the selected variables. Means and standard deviations (in parenthesis) associated with each group are given for continuous variables. Only frequencies are shown for categorical variables. Standard-errors for the predicted uplift are shown in parenthesis. Note that all group means are significantly different from 0 (p-value < 0.0005).

	Persuadables	**Do Not Disturb**
Number of observations	4199	4200
Observed Uplift	5.94%	−6.02%
Predicted Uplift	4.94% (0.10%)	−5.45% (0.07%)
Credit Score	771 (60)	736 (77)
Age (Years)	46.1 (11.6)	41.4 (11.8)
Gender		
Male	50%	61%
Marital Status		
Single	13%	43%
Products		
Auto and Home	83%	66%
Auto Policies Count	1.04 (0.64)	0.76 (0.63)
Mortgage Count	0.63 (0.63)	0.51 (0.55)
Residences Count	1.15 (0.41)	1.02 (0.19)
Extra Options	87%	41%

persuadable group. Also, 87% of persuadable clients have additional coverage in their policy while less than half of the do-not-disturb clients opt for extra coverage options.

Looking at the average profiles of *persuadable* and *do not disturb* clients, we can say that a *persuadable* client has a higher credit score and is slightly older than a client that should not be targeted. A *persuadable* client is less likely to be single and more likely to hold both company insurance products (i.e., home and auto policies). Also, this type of client holds

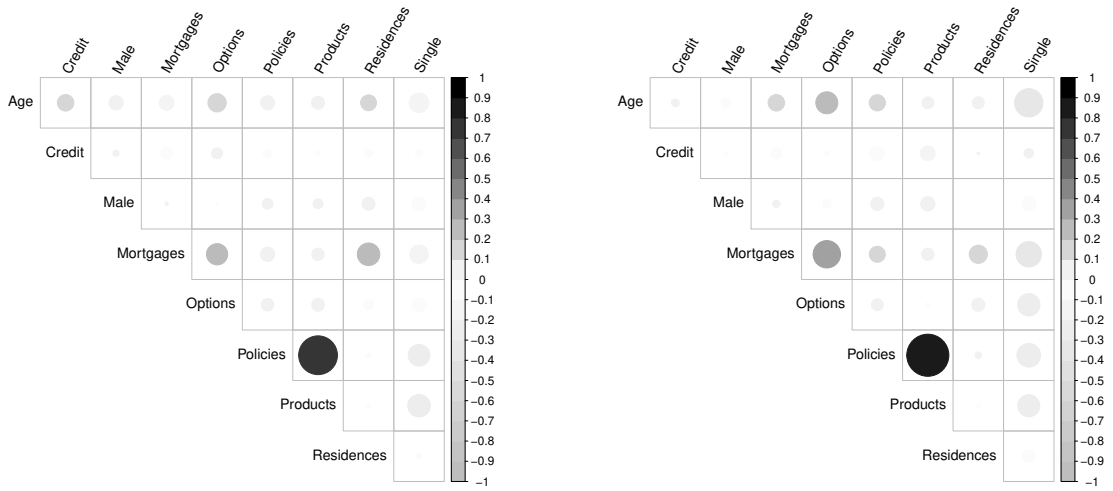

Fig. 3.9. Correlations of selected variables of interest for Persuadable (left panel) and Do-not-disturb (right panel) clients. Positive correlations are displayed in black and negative correlations in grey color. Color intensity and the size of the circle are proportional to the correlation coefficients.

more auto policies in his/her account, more mortgages, more residences in his/her name and is more likely to have extra coverage options. Thus, it seems that a persuadable client is a customer with many products to insure. The correlation matrices associated with these two groups are displayed in image format in Figure 3.9. There are some obvious patterns that distinguish the two groups. For example, *credit score* is slightly correlated with *client age* for persuadable clients, but not for do-not-disturb clients. Client *age* is negatively correlated with *marital status* for do-not-disturb clients, but only slighlty correlated for persuadables. Indeed, there are several differences in the *marital status* correlations in both groups. Also, the number of *mortgages* and *residences* are more correlated for persuadables than do-not-disturb, and the number of *mortgages* and whether or not a client has *extra options* are more correlated for do-not-disturb than persuadables.

The differences between these two groups can also be observed through the odds ratios. For any specific variable X_j which takes average values $x_j^{(\mathrm{p})}$ (for persuadable clients), and $x_j^{(\mathrm{d})}$ (for do-not-disturb clients), consider the odds ratio $\mathrm{OR}_{X_j}^{(\mathrm{group})}(t)$ between persuadable

and do-not-disturb clients

$$\frac{\Pr(Y=1 \mid X_j = x_j^{(\mathrm{p})}, T=t)/\Pr(Y=0 \mid X_j = x_j^{(\mathrm{p})}, T=t)}{\Pr(Y=1 \mid X_j = x_j^{(\mathrm{d})}, T=t)/\Pr(Y=0 \mid X_j = x_j^{(\mathrm{d})}, T=t)}$$
$$= \Big(\exp(\hat{\beta}_j)\,\exp(\hat{\delta}_j t)\Big)^{x_j^{(\mathrm{p})}-x_j^{(\mathrm{d})}} = \Big(\mathrm{OR}_{X_j}(t)\Big)^{x_j^{(\mathrm{p})}-x_j^{(\mathrm{d})}}\,, \tag{3.5.1}$$

where T is the treatment indicator. Table 3.8 shows these odds ratios for the two values of $T \in \{0,1\}$. For example, if we only consider *extra options,* when the insurance company calls a client (i.e., $T = 1$), the odds ratio between a persuadable client (Extra Options= 87%) and a do-not-disturb client (Extra Options= 41%) is about 1.12 with a 95% confidence interval of [0.98; 1.28]. On the other hand, when the company does not call a customer (i.e., $T = 0$), the odds ratio becomes 0.62 with a 95% confidence interval of [0.41; 0.94]. These results are quite logical in the sense that the odds of renewing the insurance policy are higher for the persuadable clients if the company calls, while the same odds are higher for the do-not-disturb clients if the company does not call.

Overall, we observe that when calling a client, the odds of renewing the insurance policy of persuadable clients are almost twice (1.81) the odds of do-not-disturb clients with a 95% confidence interval of [1.43; 2.29]. Conversely, when the company does not call a client, the odds of renewing the insurance policy of persuadable clients are half (0.52) the odds of do-not-disturb clients with a 95% confidence interval of [0.26; 1.03]. Hence, based on our model, by calling identified persuadable clients and not calling identified do-not-disturb clients in future marketing campaigns should result in increased retention rates for the company.

3.5.3. Uplift prediction

The main objective in analyzing the insurance data is to estimate the parameters of the parametric uplift model which maximizes the Qini. Based on these estimates, we were able to provide useful insights to the company. In order to prevent overfitting, we made use of 5-fold cross validation in the fitting process. However, since uplift models can also be used for predicting future clients behaviour, it is important to evaluate out-of-sample performance. In Table 3.5, we showed the in-sample performance. Since we do not have a test sample, in order to evaluate the out-of-sample performance, we proceeded in the following way. We ran 30 experiments. For each experiment, we reserved 25% randomly drawn observations for out-of-sample performance (test-set). We used the remaining observations

Tab. 3.8. Odds ratios $\mathrm{OR}_{X_j}^{(\mathrm{group})}(t)$ of the persuadable compared to the do not disturb clients (Eq. 3.5.1) and 95% confidence intervals estimated by the Qini-based uplift regression model (*Q+LHS*) for some of the selected variables. The $\Delta = x_j^{(\mathrm{p})} - x_j^{(\mathrm{d})}$ column represents the difference of group means from Table 3.7.

	Δ	**Control**	**CI** (95%)	**Called**	**CI** (95%)
Overall	-	0.52	(0.26, 1.03)	1.81	(1.43, 2.29)
Credit Score	35	0.95	(0.82, 1.09)	1.02	(0.98, 1.07)
Age (Years)	4.7	0.98	(0.86, 1.11)	0.99	(0.95, 1.03)
Gender					
Male	−11%	0.97	(0.92, 1.03)	1.00	(0.99, 1.02)
Marital Status					
Single	−30%	0.73	(0.54, 1.01)	1.11	(0.98, 1.27)
Products					
Auto and Home	17%	1.09	(0.91, 1.29)	1.06	(1.00, 1.12)
Auto Policies Count	0.28	1.23	(0.89, 1.71)	1.21	(1.09, 1.35)
Mortgage Count	0.12	1.04	(0.97, 1.11)	1.04	(1.02, 1.06)
Residences Count	0.13	0.92	(0.80, 1.05)	1.08	(1.02, 1.15)
Extra Options	46%	0.62	(0.41, 0.94)	1.12	(0.98, 1.28)

to fit the models. These observations were further randomly divided into training-set, which comprised 2/3 of the remaining observations, and validation-set. We use the training data to fit the models and compute the test-set performance through the adjusted Qini coefficient to measure performance from a predictive point of view. In order to mitigate the overfitting, the model parameters and/or coefficients are chosen so as to find the best fit for the validation-set (cross-validation). For each experiment, the training-set size was 10,394 observations, the validation-set size was 5,354 observations, and the test-set size was 5,249 observations. To fit the *RF*s models, we searched for the hyper-parameters that maximize the adjusted Qini coefficient following the same procedure that was applied in the first part of the insurance data analysis. The average results are displayed in Table 3.9.

Tab. 3.9. Out-of-sample performance when models are trained with cross-validation. The adjusted Qini coefficients are averaged over 30 experiments. Standard-errors are shown in parenthesis.

Method	**training-set**	**validation-set**	**test-set**
RF	**0.896** (**0.031**)	0.152 (0.037)	0.071 (0.018)
Q+LHS	0.885 (0.032)	**0.859** (**0.051**)	**0.556** (**0.024**)
Q+lasso	0.618 (0.041)	0.450 (0.030)	0.127 (0.017)
MLE+lasso	0.303 (0.037)	0.057 (0.028)	0.049 (0.009)

Based on these experiments, we see that the *RF* model shows the highest performance in the training-set. However, there are strong signs of overfitting. The *Q+LHS* model clearly outperform the other methods and gives the best results in terms of prediction. We are not surprised by the performance of the *RF* model because we had experimented with these *RF* models in the past, and we have not been able to get better predictive performance in other marketing campaign initiatives.

Next, we explain the performance of each predictive model in terms of clients retained *thanks* to the call. We hope this gives the reader a better idea about the test-set $\hat{q}_{\text{adj}}$ values presented in Table 3.9. To do so, a comparison of targeting strategies based on the models is presented in Table 3.10. We normalize the results to 10,000 potential customers.

As shown in Table 3.1, overall, the marketing campaign had a negative effect, that is, an uplift of -0.41%. This means that the proportion of renewals in the treatment group is lower than the proportion of renewals in the control group. This implies that based on 10,000 clients, we would expect a difference of 41 renewals between the following strategies: contacting all clients to convince them to stay, or not contacting any client at all. Now, using the models' results on the test-set, we can compute this difference as a function of the number of clients contacted. It is common to present these numbers by quantiles of predicted uplifts (according to each model, much like uplift barplots, or Qini curves). We present the results in Table 3.10. We refer to this difference as the *expected retained customers*.

Let us look at the result associated with the *Q+LHS* model when 2,000 clients are contacted. By this, we mean the results of the marketing campaign had the company contacted the 2,000 clients with the highest predicted uplifts (according to the *Q+LHS* model). By

Tab. 3.10. Expected retained customers as a function of the number of targeted customers following the predictive models from Table 3.9. The numbers are averaged and rounded to the nearest unit.

	Clients contacted				
Method	2,000	4,000	6,000	8,000	10,000
RF	10	2	0	−23	−41
Q+LHS	55	57	42	11	−41
Q+lasso	8	19	11	−10	−41
MLE+lasso	6	−10	−2	−23	−41
Random	−8	−16	−24	−32	−41

contacting these clients we would expect a gain of 55 more clients renewing their policy with respect to the number of clients renewing their policy had these clients not been contacted at all. Similarly, if the 4,000 customers with the highest predicted uplifts were contacted, the company would retain 57 more customers. This means that we expect that the call does not have such a large impact for customer 2,001 to customer 4,000 when compared with only calling the first 2,000 clients. In terms of financial impact, depending on the cost of the call and the strategy adopted by a business, it is possible to prefer to contact only the first 2,000 customers of the ordered list. Also, not contacting customers 2,001 to 4,000 saves the costs associated with each extra call.

Note that if the *RF* model is used instead of the *Q+LHS* model, the company would lose many clients. Indeed, if we call the 2,000 customers with the highest predicted uplifts according to the *RF* model, we would expect a gain of only 10 more customers renewing their policy with respect to the number of clients renewing their policy had these clients not been contacted at all. If we increase the number of contacted clients to the 4,000 clients with highest uplifts, the expected number of retained customers would decrease to 2. Furthermore, for the remaining clients in the list, the effect of the call would become negative (with respect to the RF model), and therefore, there would be more chance of retaining customers if they were not called at all.

Finally, if the company decides to follow the *Q+lasso* model's predictions, 19 clients would be retained by contacting the top 4,000 of the list (in comparison to not contacting

them). If we compare this strategy to the strategy of calling 4,000 customers at random, that is, without relying on any uplift model, we would expect to drive away 16 customers because of the call. So, it is better to follow the *Q+LHS* model strategy than targeting clients at random.

3.6. Conclusion

Our goal was to analyze the data of a marketing campaign conducted by an insurance company to retain customers at the end of their contract. A random group of policyholders received an outbound courtesy call made by one of the company's licensed insurance advisors, with the objective to reinforce the customers confidence in the company, to review their coverage and address any questions they might have about their renewal. In the database at our disposal, an independent group of clients was observed and serves as control. In order to evaluate the causal effect of the courtesy call on the renewal or cancellation of the insurance policy of its clients, an uplift model needed to be applied.

We have developed a methodology for estimating parameters of a logistic regression in the context of uplift models. This is based on a new statistic specially conceived to evaluate uplift models. The statistic, the adjusted Qini, is based on the Qini coefficient. It takes into account the correlation between the observed uplift and the predicted uplift by a model. Maximizing the adjusted Qini to choose an adequate model for uplift acts as a regularizing factor to select parsimonious models, much as lasso does for regression models.

Since the Qini is a difficult statistic to compute, maximizing the adjusted Qini directly is not an easy task. Instead, we proposed to use lasso-type likelihood penalization to search the space of appropriate uplift models, so as to only consider relevant variables for uplift. Since the usual lasso is not designed for uplift models, we adapted it, by selecting the value $\hat{\lambda}$ of the lasso penalization constant that maximizes the adjusted Qini. At first, this ensures that the selected variables (i.e., those associated with non-zero regression coefficients) are important variables for estimating uplift. Then, in a second step, we estimate the parameters that maximize the adjusted Qini by searching a Latin hypercube sampling (LHS) surface around the lasso estimates. A variant of this procedure consists of estimating the parameters as those that maximize the likelihood associated with the model selected by $\hat{\lambda}$, using only the selected variables.

Experimental evaluation showed that for the first stage of the Qini optimized uplift regression, choosing the penalization constant from the logistic-lasso sequence by maximizing the adjusted Qini dramatically improves the performance of uplift models. This is the *Q+lasso* model. In addition, using a LHS search on the second stage leads to a direct maximization of the adjusted Qini coefficient, and to a further boost in the performance of the model. The resulting model is the *Q+LHS* model. In addition, our empirical studies clearly show that the performance of a Qini-based regression model is much better than the performance of the usual lasso penalized logistic regression model.

Concerning the particular marketing data available to us from the insurance company, we selected two final models and compared them to the usual lasso regression approach as well as the uplift random forest. The results show that our method clearly surpasses the usual approach in terms of performance. We argue that this is due to the Qini-based methods performing variable selection explicitly built for optimizing uplift. Although, even if overall, the marketing campaign of the insurance company did not appear to be successful, the uplift models with the selection of the right variables identify a group of customers for which the campaign worked very well. Indeed, the results show that a persuadable client is a customer with many products to insure. Also, notice there is a subgroup of clients for whom the call had a negative impact. This can be explained by the fact that some customers are already dissatisfied with their insurance policies and have already decided to change them before receiving the call. This call can also trigger a behavior that encourages customers to look for better rates. For future campaigns, the company can target only those customers for whom the courtesy call will be useful and remove and investigate more the clients for whom the marketing campaign had a negative effect.

3.7. Appendix

3.7.1. The observed uplfit can be seen as the slope of the Qini curve

This can be shown as follows. The slope is defined as

$$\nabla g_k = \frac{g(\phi_{k+1}) - g(\phi_k)}{\phi_{k+1} - \phi_k}$$

By construction, the denominator is simplified to $\phi_{k+1} - \phi_k = 1/J$. Now, because of randomization, we have the following approximations,

$$\sum_{i \in N_{\phi_k}} t_i / \sum_{i \in N_{\phi_k}} (1 - t_i) \approx \sum_{i=1}^{n} t_i / \sum_{i=1}^{n} (1 - t_i),$$

and,

$$\sum_{i \in B_k} t_i / \sum_{i=1}^{n} t_i \approx \sum_{i \in B_k} (1 - t_i) / \sum_{i=1}^{n} (1 - t_i) \approx 1/J.$$

Also, we have

$$\sum_{i \in N_{\phi_{k+1}}} y_i t_i - \sum_{i \in N_{\phi_k}} y_i t_i = \sum_{i \in B_k} y_i t_i,$$

so the numerator simplifies as follows

$$\begin{aligned}
g(\phi_{k+1}) - g(\phi_k) &\approx \left(\sum_{i \in B_k} y_i t_i - \left\{ \sum_{i=1}^{n} t_i / \sum_{i=1}^{n} (1 - t_i) \right\} \sum_{i \in B_k} y_i (1 - t_i) \right) / \sum_{i=1}^{n} t_i \\
&= \sum_{i \in B_k} y_i t_i / \sum_{i=1}^{n} t_i - \sum_{i \in B_k} y_i (1 - t_i) / \sum_{i=1}^{n} (1 - t_i) \\
&= \left\{ \sum_{i \in B_k} t_i / \sum_{i=1}^{n} t_i \right\} \left\{ \sum_{i \in B_k} y_i t_i / \sum_{i \in B_k} t_i \right\} - \\
&\quad \left\{ \sum_{i \in B_k} (1 - t_i) / \sum_{i=1}^{n} (1 - t_i) \right\} \left\{ \sum_{i \in B_k} y_i (1 - t_i) / \sum_{i \in B_k} (1 - t_i) \right\} \\
&\approx (1/J) \left\{ \sum_{i \in B_k} y_i t_i / \sum_{i \in B_k} t_i - \sum_{i \in B_k} y_i (1 - t_i) / \sum_{i \in B_k} (1 - t_i) \right\} \\
&= \bar{u}_k / J
\end{aligned}$$

Therefore,

$$\nabla g_k \approx \bar{u}_k.$$

Maximizing the adjusted Qini coefficient maximizes the Qini coefficient and simultaneously promotes grouping the individuals in decreasing uplift bins, which in turn results in concave Qini curves. Suppose that for a given model, $\rho = 1$. In this case, we have

$$\begin{aligned}
\rho = 1 &\iff \bar{u}_1 \geq \bar{u}_2 \geq \ldots \geq \bar{u}_J \\
&\iff \bar{u}_k - \bar{u}_{k+1} \geq 0 && \text{for all } k \in 1, \ldots, J-1 \\
&\iff \nabla g_k - \nabla g_{k+1} \geq 0 && \text{for all } k \in 1, \ldots, J-1
\end{aligned}$$

which implies concavity of the model's Qini curve.

3.7.2. K-fold cross validation

In the following, we give details about our implementation of the *Q+LHS* K-fold cross-validation used in the *parameter estimation* part of Section 3.5. This implementation is available in the R Package **tools4uplift** [Belbahri *et al.*, 2020].

- Using all data, obtain the lasso sequence $\{\lambda_j\}_{j=1}^{J}$, for example by a 3-fold cross-validation with the `glmnet.cv` function available in R through the Package **glmnet**
- Divide the data into K disjoint equal size parts D_k, $k = 1, \ldots, K$.
- For each $k \in \{1, \ldots, K\}$ do:
  - Set the kth fold D_k as the validation set, and the rest of the observations D_{-k} as the training set.
  - For each $j \in \{1, \ldots, J\}$ do:
    - Fit the penalized model with $\lambda = \lambda_j$ on the training set D_{-k} to get $\hat{\boldsymbol{\theta}}_{-k}(\lambda_j)$
    - Generate L LHS points in the neighbourhood of $\hat{\boldsymbol{\theta}}_{-k}(\lambda_j)$, $\{\hat{\boldsymbol{\theta}}_{-k}^{(l)}(\lambda_j)\}_{l=1}^{L}$
    - For each $l \in \{1, \ldots, L\}$ do:
      - Using the training set D_{-k}, compute $\hat{q}_{\text{adj},-k}^{(l)}(\lambda_j)$ associated with $\hat{\boldsymbol{\theta}}_{-k}^{(l)}(\lambda_j)$
    - Keep the coefficients associated with the maximum in the set $\{\hat{q}_{\text{adj},-k}^{(l)}(\lambda_j)\}_{l=1}^{L}$, which we denote by $\tilde{\boldsymbol{\theta}}_{-k}(\lambda_j)$
    - Using the kth fold D_k as the validation set, compute $\hat{q}_{\text{adj},k}(\lambda_j)$ associated with $\tilde{\boldsymbol{\theta}}_{-k}(\lambda_j)$
  - Return the lasso-LHS sequence $\{\tilde{\boldsymbol{\theta}}_{-k}(\lambda_j)\}_{j=1}^{J}$ and associated $\{\hat{q}_{\text{adj},k}(\lambda_j)\}_{j=1}^{J}$
- For $j = 1, \ldots, J$, compute the K-fold cross-validated average adjusted Qini coefficients, that is

$$\bar{\hat{q}}_{\text{adj}}(\lambda_j) = \frac{1}{K} \sum_{k=1}^{K} \hat{q}_{\text{adj},k}(\lambda_j)$$

- Return $\hat{\lambda} = \arg\max_{\lambda_j} \bar{\hat{q}}_{\text{adj}}(\lambda_j)$, the value of λ_j that maximizes the K-fold cross-validated average adjusted Qini, and the K associated coefficients

$$\Big(\tilde{\boldsymbol{\theta}}_{-1}(\hat{\lambda}), \ldots, \tilde{\boldsymbol{\theta}}_{-K}(\hat{\lambda})\Big).$$

Once the procedure is complete, in a second stage, we fit a model using all the data by either

(i) calling again the LHS search starting with the coefficient vector $\hat{\boldsymbol{\theta}}(\hat{\lambda})$, that is, the parameters estimated by the penalized likelihood when $\lambda = \hat{\lambda}$; or

(ii) performing K different LHS searches starting from each of the K associated coefficients $\left(\tilde{\boldsymbol{\theta}}_{-1}(\hat{\lambda}), \ldots, \tilde{\boldsymbol{\theta}}_{-K}(\hat{\lambda})\right)$.

Acknowledgements

Mouloud Belbahri and Alejandro Murua were partially funded by The Natural Sciences and Engineering Research Council of Canada grant 2019-05444. Vahid Partovi Nia was supported by the Natural Sciences and Engineering Research Council of Canada grant 418034-2012. The authors thank the Annals of Applied Statistics editor, associate editor, and reviewers for their valuable comments and suggestions that helped improve the manuscript.

Bibliography

Farrokh Alemi, Harold Erdman, Igor Griva, and Charles H Evans. Improved statistical methods are needed to advance personalized medicine. *The Open Translational Medicine Journal*, 1:16, 2009.

Mouloud Belbahri, Olivier Gandouet, Alejandro Murua, and Vahid Partovi Nia. *tools4uplift: Tools for Uplift Modeling*, 2020. URL `https://CRAN.R-project.org/package=tools4uplift`. R package version 1.0.0.

Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and Regression Trees*. CRC press, 1984.

Kristof Coussement, Paul Harrigan, and Dries F Benoit. Improving direct mail targeting through customer response modeling. *Expert Systems with Applications*, 42(22):8403–8412, 2015.

Floris Devriendt, Darie Moldovan, and Wouter Verbeke. A literature survey and experimental evaluation of the state-of-the-art in uplift modeling: A stepping stone toward the development of prescriptive analytics. *Big Data*, 6(1):13–41, 2018.

Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.

Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.

LLdiko E Frank and Jerome H Friedman. A statistical view of some chemometrics regression tools. *Technometrics*, 35(2):109–135, 1993.

Jerome Friedman, Trevor Hastie, Holger Höfling, Robert Tibshirani, et al. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

Jerome Friedman, Trevor Hastie, and Rob Tibshirani. *glmnet: Lasso and elastic-net regularized generalized linear models*, 2009. URL `https://CRAN.R-project.org/package=glmnet`. R package version 4.1.

Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1, 2010.

Corrado Gini. Concentration and dependency ratios. *Rivista di politica economica*, 87: 769–792, 1997.

Leo Guelman. *uplift: Uplift Modeling*, 2014. URL `https://CRAN.R-project.org/package=uplift`. R package version 0.3.5.

Leo Guelman, Montserrat Guillén, and Ana M Pérez-Marín. Random forests for uplift modeling: an insurance customer retention case. In *Modeling and Simulation in Engineering, Economics and Management*, pages 123–133. Springer, 2012.

Pierre Gutierrez and Jean-Yves Gérardy. Causal inference and uplift modelling: A review of the literature. In *International Conference on Predictive Applications and APIs*, pages 1–13, 2017.

Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3(Mar):1157–1182, 2003.

David J Hand and Keming Yu. Idiot's bayes not so stupid after all. *International Statistical Review*, 69(3):385–398, 2001.

Behram Hansotia and Bradley Rukstales. Direct marketing for multichannel retailers: Issues, challenges and solutions. *Journal of Database Marketing and Customer Strategy Management*, 9(3):259–266, 2001.

Behram Hansotia and Bradley Rukstales. Incremental value modeling. *Journal of Interactive Marketing*, 16(3):35, 2002.

Dominique M Hanssens, Leonard J Parsons, and Randall L Schultz. *Market response models: Econometric and time series analysis*, volume 12. Springer Science & Business Media, 2003.

Trevor Hastie, Jonathan Taylor, Robert Tibshirani, Guenther Walther, et al. Forward stagewise regression and the monotone lasso. *Electronic Journal of Statistics*, 1:1–29, 2007.

Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Paul W Holland. Statistics and causal inference. *Journal of the American Statistical Association*, 81(396):945–960, 1986.

Kathleen Kane, Victor SY Lo, and Jane Zheng. Mining for the truly responsive customers and prospects using true-lift modeling: Comparison of new and existing methods. *Journal of Marketing Analytics*, 2(4):218–238, 2014.

Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

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.

Max O Lorenz. Methods of measuring the concentration of wealth. *Publications of the American Statistical Association*, 9(70):209–219, 1905.

Michael D McKay, Richard J Beckman, and William J Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61, 2000.

Douglas C Montgomery, Elizabeth A Peck, and G Geoffrey Vining. *Introduction to linear regression analysis*, volume 821. John Wiley & Sons, 2012.

John A Nelder and Roger Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

Jerzy S Neyman. On the application of probability theory to agricultural experiments. *Annals of Agricultural Sciences*, 10:1–51, 1923.

J Pearl. Causal inference in statistics: An overview. *Statistics Surveys*, 3:96–146, 2009.

Nicholas J Radcliffe and Patrick D Surry. Real-world uplift modelling with significance-based uplift trees. *White Paper TR-2011-1, Stochastic Solutions*, 2011.

NJ Radcliffe. Using control groups to target on predicted lift: Building and assessing uplift models. *Direct Market J Direct Market Assoc Anal Council*, 1:14–21, 2007.

NJ Radcliffe and PD Surry. Differential response analysis: Modeling true response by isolating the effect of a single action. *Credit Scoring and Credit Control VI. Edinburgh, Scotland*, 1999.

Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5):688, 1974.

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

Piotr Rzepakowski and Szymon Jaroszewicz. Decision trees for uplift modeling with single and multiple treatments. *Knowledge and Information Systems*, 32(2):303–327, 2012.

Robert E Smith and William R Swinyard. Information response models: An integrated approach. *Journal of Marketing*, 46(1):81–93, 1982.

Xiaogang Su, Chih-Ling Tsai, Hansheng Wang, David M Nickerson, and Bogong Li. Subgroup analysis via recursive partitioning. *Journal of Machine Learning Research*, 10(Feb): 141–158, 2009.

Xiaogang Su, Joseph Kang, Juanjuan Fan, Richard A Levine, and Xin Yan. Facilitating score and causal inference trees for large observational studies. *Journal of Machine Learning Research*, 13(Oct):2955–2994, 2012.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, 67(2):301–320, 2005.

Chapitre 4

Régression *uplift* : La librairie R tools4uplift

Deuxième article.

Uplift Regression: The R Package tools4uplift

par

Mouloud Belbahri[1], Alejandro Murua[1], Olivier Gandouet[2], and Vahid Partovi Nia[3]

([1]) Université de Montréal
([2]) TD Assurance
([3]) École Polytechnique de Montréal

Cet article a été soumis à la revue *Journal of Statistical Software*

Mes contributions et le rôle des coauteurs. Dans cet article, j'ai

- proposé toutes les méthodes originales incluses dans la librairie R **tools4uplift**;
- développé les algorithmes pour la discrétisation de variables explicatives continues;
- rédigé le code R ainsi que la documentation associée permettant la publication de la librairie **tools4uplift**;
- rédigé l'article;
- analysé les données publiques et interprété les résultats.

En tant que directeur principal, Alejandro Murua a participé à la rédaction de l'article et au développement méthodologique. En tant que co-directeur, Vahid Partovi Nia a contribué dans la rédaction de l'article et a donné des conseils de programmation précieux permettant le développement de la librairie **tools4uplift**. En tant que directeur industriel, Olivier Gandouet a contribué dans le développement méthodologique et a supervisé l'implémentation des fonctions de la librairie **tools4uplift**.

RÉSUMÉ. La modélisation *uplift* vise à prédire l'effet causal d'une action telle qu'une campagne marketing sur un individu particulier. Un groupe ciblé comprend des individus qui font l'objet d'une action; un groupe témoin sert de comparaison. La modélisation *uplift* est utilisée pour ordonner les individus par rapport à la valeur d'un effet causal, par exemple, positif, neutre ou négatif. Bien qu'il existe des méthodes de calcul disponibles pour la modélisation *uplift*, la plupart d'entre elles excluent les modèles de régression statistique. La librairie R **tools4uplift** entend combler cette lacune. Cette librairie comprend des outils pour: i) la discrétisation; ii) la visualisation; iii) la sélection de variables; iv) l'estimation des paramètres; et v) la validation du modèle.

Mots clés : régression logistique, optimisation, discrétisation, visualisation, sélection de variables, R

ABSTRACT. Uplift modeling aims at predicting the causal effect of an action such as a marketing campaign on a particular individual. A targeted group contains individuals who are subject to an action; a control group serves for comparison. Uplift modeling is used to order the individuals with respect to the value of a causal effect, e.g., positive, neutral, or negative. Though there are some computational methods available for uplift modeling, most of them exclude statistical regression models. The R Package **tools4uplift** intends to fill this gap. This package comprises tools for: i) quantization; ii) visualization; iii) feature selection; iv) parameters estimation; and v) model validation.

Keywords: logistic regression, optimization, quantization, visualization, feature selection, R

4.1. Introduction

The term causal study refers to a study that tries to discover a cause-effect relationship. If there is a causal relationship between two events, the events are highly dependent. However, the converse might not be true, since association is not necessarily causation. If a randomized experiment study is performed to isolate the causal effect, association and causation coincide.

The statistical framework for causal inference was formally introduced by Rubin [1974]. This framework is also associated with the potential outcome framework of Neyman [1923], also known as the Rubin causal model [Holland, 1986]. A potential outcome is the theoretical response each unit would have manifested, had it been assigned to a particular treatment. Under randomization, these outcomes are independent of the assignment other observations receive. In practice, potential outcomes for an individual cannot be observed. A single unit is only assigned to either treatment or control, making direct observations in the other condition (called the *counterfactual* condition) and the observed individual causal effects, impossible. This is well-known as the fundamental problem of causal inference [Holland, 1986]. Often, in a randomized experiment, researchers focus on the estimation of average treatment effects and the effect of the treatment is determined from this estimate. However, there might be a proportion of the population that may respond favorably to the treatment, and another proportion that may not, depending on whether or not individual treatment effects vary widely in the population. A decision based on an average treatment effect for a new arriving individual would require a baseline adjustment because of the heterogeneity in treatment response originated by many characteristics.

In marketing, *response models* [Hanssens et al., 2003] of client behavior are based on historical data. They are used to predict the probability that a client responds to a marketing campaign, e.g., the client buys a product. Marketing campaigns using response models concentrate on clients associated with a high probability of a positive response. However, this strategy does not ensure a purchase. On the other hand, customers may buy the product without any marketing effort. Therefore, it is important to extract the cause of the purchase and to isolate the effect of marketing. *Uplift models* [Radcliffe and Surry, 1999, Hansotia and Rukstales, 2001, Lo, 2002] provide a solution to the problem of isolating the marketing effect. Instead of modeling the different response or class probabilities, *uplift* attempts to model the difference between conditional response probabilities in the treatment and control groups. Uplift modeling aims at identifying groups of individuals on which a predetermined action will have the most positive effect.

In the R Package **tools4uplift** presented here [Belbahri *et al.*, 2020], we make available to practitioners a combination of tools for uplift modeling, including some novel techniques introduced in this paper. Our package comprises tools for: i) quantization; ii) visualization; iii) feature selection; iv) parameters estimation; and v) model validation, alongside their associated functions. We hope that the package will enable practitioners to save time and effort when analyzing their uplift data.

The methods implemented in the R Package **tools4uplift** are related to, but distinct from the ones implemented in the R Package **uplift** [Guelman, 2014]. The functions included in **uplift** are designed for building and testing the uplift models proposed by Guelman et al. [2015]. It focuses on the adaptation of non-parametric machine learning classifiers such as random forests and k-nearest neighbours. The R Package **tools4uplift** offers a complementary set of functions targeting uplift regression models. It focuses on building regression models adapted for uplift [Belbahri et al., 2021]; it proposes methods for quantization and visualization of continuous variables; and it introduces a method to perform automatic variable selection in uplift regression models. Finally, the R Package **tools4uplift** also includes model validation functions.

The remaining of the paper is organized as follows. Section 4.2 introduces the notation, and discusses the general uplift modeling methodology, alongside its statistical background and its implementation in R. In Section 4.3, we present a quantization method designed

for uplift models. Section 4.4 discusses variable selection and the implementation in R of the uplift regression model of interest. Section 4.5 shows an application of the proposed methodology to real data using **tools4uplift** and some final remarks are given in Section 4.6.

4.2. Uplift models

In marketing, we are interested in the conditional probability that a client buys a product given that he was targeted by a marketing campaign (the treatment group). We also want to measure the conditional probability that a client buys the product given that he was not targeted (the control group). Uplift attempts to model the difference between conditional class probabilities in the treatment and control groups. The variable of interest has two possible outcomes: whether or not the purchase is made.

The logistic regression model is a widely used statistical model that uses a logistic function to model a binary dependent variable. It is easy to implement and has an elegant interpretation, thanks, in particular, to the odds ratio. The odds ratio is the ratio that compares the change in odds of buying a product for two different sets of values of the factors in the model, e.g., change in age, gender, etc. The logistic regression model is in part more popular than other binary-outcome models because odds ratios are readily available.

A customer base is a historical list of clients to whom a business sold products and services. This list can be segmented along two dimensions in function of the response value (yes or no), and the associated treatment (yes or no), given rise to the following groups [Kane et al., 2014]:

(1) the "persuadables" who respond to the marketing action because they are targeted,
(2) the "sure" individuals who respond whether or not they are targeted,
(3) the "lost" individuals who do not respond, regardless of whether or not they are targeted, and
(4) the "do not disturb" individuals who are less likely to respond, just because they are targeted.

In general, the interesting customers from a marketing point of view are the "persuadables" and the "do not disturb". The persuadables provide incremental responses whereas the "do not disturb" individuals should not be disturbed because the marketing campaign has a negative effect on them. Uplift modeling attempts to separate customers into the

four groups described above. The intuitive approach is to build two classification models. Recall that the uplift is the difference between two conditional probabilities. Hansotia and Rukstales [2001] proposed an indirect method to estimate the uplift based on a two-model approach. This consists of fitting two separated conditional probability models: one for the treated individuals, and another for the untreated individuals. The uplift is estimated as the difference between these two conditional probability models. The asset of this technique is its simplicity. However, both models focus on predicting only a one-class probability instead of making an effort to predict the uplift. Any conventional statistical or algorithmic binary-outcome classification method may serve to fit these models. In order to improve the accuracy of the two-model approach, Lo [2002] proposed an interaction model. Interactions may arise when considering the relationship among three or more variables, and describes a situation in which the simultaneous influence of two variables on a third is not additive. The methodology is based on adding explicit interaction terms between each covariate and the treatment indicator using a standard logistic regression. The parameters of the interaction terms measure the additional effect of each covariate because of the treatment. As in the two-model approach, an indirect estimation of the uplift is achieved by subtracting the predicted probabilities associated with the control group from the probabilities associated with the treatment group.

Other approaches to uplift modeling try to directly model the difference in conditional success probabilities between the treatment and control groups. Most current active research is in this direction. Such methods are mainly adaptation of three types of machine learning algorithms: a) decision tree learners (Rzepakowski and Jaroszewicz [2010], Radcliffe and Surry [2011], Guelman et al. [2015], Sołtys et al. [2015] or Zhao et al. [2017]), b) regression models adapted to the uplift (Radcliffe [2007], Jaskowski and Jaroszewicz [2012] or Belbahri et al. [2021]) and c) support vector machines for uplift (Zaniewicz and Jaroszewicz [2013], Kuusisto et al. [2014] or Zaniewicz and Jaroszewicz [2017]).

To formalize the problem, let Y be the $0-1$ binary response variable, T the $0-1$ treatment indicator variable and $X_1, \ldots, X_p$ the explanatory variables (predictors). The binary variable T indicates if a unit is exposed to treatment ($T = 1$) or control ($T = 0$). Suppose that n independent units are observed $\{(y_i, \mathbf{x}_i, t_i)\}_{i=1}^{n}$, where $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$ are realisations of the predictors random variables. Denote the potential outcomes $Y_i(0), Y_i(1)$ under control

and treatment by $\{Y_i \mid T_i = 0\}$ and $\{Y_i \mid T_i = 1\}$ respectively. The uplift model estimates

$$u(\mathbf{x}_i) = \Pr(Y_i = 1 \mid \mathbf{x}_i, T_i = 1) - \Pr(Y_i = 1 \mid \mathbf{x}_i, T_i = 0). \tag{4.2.1}$$

4.2.1. The two-model estimator

The *two-model* estimator [Hansotia and Rukstales, 2001] consists in the subtraction of logistic regression models for the treated and untreated populations. Let

$$\Pr(Y_i = 1 \mid \mathbf{x}_i, T_i = 1, \beta_o^{(1)}, \boldsymbol{\beta}^{(1)}) = \left(1 + \exp\{-(\beta_o^{(1)} + \mathbf{x}_i^\top \boldsymbol{\beta}^{(1)})\}\right)^{-1}$$

and

$$\Pr(Y_i = 1 \mid \mathbf{x}_i, T_i = 0, \beta_o^{(0)}, \boldsymbol{\beta}^{(0)}) = \left(1 + \exp\{-(\beta_o^{(0)} + \mathbf{x}_i^\top \boldsymbol{\beta}^{(0)})\}\right)^{-1},$$

where $(\beta_o^{(t)}, \boldsymbol{\beta}^{(t)})$ for $t = \{0,1\}$ are the logistic regression parameters for control ($t = 0$) and treatment ($t = 1$) groups, and the superscript $^\top$ denote transposition. The two-model estimator predicts the uplift associated with a covariate vector $\mathbf{x}_{n+1}$ for a future individual as

$$\hat{u}(\mathbf{x}_{n+1}) = \left(1 + \exp\{-(\hat{\beta}_o^{(1)} + \mathbf{x}_{n+1}^\top \hat{\boldsymbol{\beta}}^{(1)})\}\right)^{-1} - \left(1 + \exp\{-(\hat{\beta}_o^{(0)} + \mathbf{x}_{n+1}^\top \hat{\boldsymbol{\beta}}^{(0)})\}\right)^{-1},$$

where $(\hat{\beta}_o^{(t)}, \hat{\boldsymbol{\beta}}^{(t)})$ for $t = \{0,1\}$ are the maximum likelihood estimates for each group. The R Package **tools4uplift** provides a straightforward implementation of this model with the function `DualUplift()`. The arguments are

```
DualUplift(data, treat, outcome, predictors)
```

where `data`, `treat` and `outcome` are necessary arguments in order to fit the two-model estimator with respect to `predictors`. The data frame `data` must contain the treatment, outcome and predictors variables. The names of these variables are used as the arguments of the `DualUplift()` function. Then, in order to predict the uplift for a new observation `newdata`, the output of the `DualUplift()` function needs to be passed as the `object` argument of the `predict()` function

```
predict(object, newdata, ...)
```

where `...` represents additional arguments that can be passed to the `predict.glm` function for each sub-model. The advantage of the two-model estimator is that it is easy to understand. Each sub-model can have its own interpretation. The model built on the treated group represents the impact of targeting individuals and assigns higher scores to those who seem to respond to the marketing campaign. On the other hand, the model built on the control group represents random noise. The differences between both sub-models represent the causal impact of the marketing campaign. In practice, the two-model estimator is a natural baseline model.

4.2.2. The interaction model estimator

The *interaction model* [Lo, 2002] uses a standard logistic regression with first order interactions terms. Let

$$\log\left(\frac{\Pr(Y_i = 1 \mid \mathbf{x}_i, t_i, \beta_o, \boldsymbol{\beta}, \gamma, \boldsymbol{\delta})}{1 - \Pr(Y_i = 1 \mid \mathbf{x}_i, t_i, \beta_o, \boldsymbol{\beta}, \gamma, \boldsymbol{\delta})}\right) = \beta_o + \mathbf{x}_i^\top \boldsymbol{\beta} + \gamma t_i + t_i \mathbf{x}_i^\top \boldsymbol{\delta}$$

or equivalently

$$\Pr(Y_i = 1 \mid \mathbf{x}_i, t_i, \beta_o, \boldsymbol{\beta}, \gamma, \boldsymbol{\delta}) = \Big(1 + \exp\{-(\beta_o + \mathbf{x}_i^\top \boldsymbol{\beta} + \gamma t_i + t_i \mathbf{x}_i^\top \boldsymbol{\delta})\}\Big)^{-1},$$

where $(\beta_o, \boldsymbol{\beta}, \gamma, \boldsymbol{\delta})$ are the logistic regression parameters. The predicted uplift associated with the covariate vector $\mathbf{x}_{n+1}$ of a future individual is estimated by

$$\hat{u}(\mathbf{x}_{n+1}) = \Big(1 + \exp\{-(\hat{\beta}_o + \mathbf{x}_{n+1}^\top \hat{\boldsymbol{\beta}} + \hat{\gamma} + \mathbf{x}_{n+1}^\top \hat{\boldsymbol{\delta}})\}\Big)^{-1} - \Big(1 + \exp\{-(\hat{\beta}_o + \mathbf{x}_{n+1}^\top \hat{\boldsymbol{\beta}})\}\Big)^{-1},$$

where $(\hat{\beta}_o, \hat{\boldsymbol{\beta}}, \hat{\gamma}, \hat{\boldsymbol{\delta}})$ are the maximum likelihood estimates. The implementation of the interaction model estimator in R follows the same logic as the one of the two-model in Section 4.2.1. The function `InterUplift()` has the following arguments

```
InterUplift(data, treat, outcome, predictors, input = c("all", "best"))
```

where the arguments `(data, treat, outcome, predictors)` have the same role as in the `DualUplift()` function. The argument `input = c("all", "best")` is important because it specifies which model to use. If this argument is set to `"all"`, the function `InterUplift()` uses the list of predictors given in the argument `predictors` to create the interaction terms

between the `treat` variable and the `predictors`, so as to fit the interaction model. The option `input = "best"` stands for "best features". In this case, `InterUplift()` uses the list of the selected main variables and interaction terms provided by the method `BestFeatures()` described later in Section 4.4 which performs variable selection for uplift. The output of `BestFeatures()` is exactly the list of the selected main variables and interaction terms for the interaction model. Then, in order to predict the uplift for a new observation `newdata`, the output of the `InterUplift()` function and the treatment variable name need to be passed as the `object` and `treat` arguments of the `predict()` function

```
predict(object, newdata, treat, ...)
```

where `...` represents additional arguments that can be passed to the `predict.glm` function for the interaction model. The main advantage of the interaction model estimator is that it is a single logistic regression model. Therefore, interpretation of the coefficients and odds ratios is straightforward.

4.3. Quantization

Data manipulation is an important aspect of statistical analysis. Feature engineering, exploration of missing values patterns, outliers detection and descriptive statistics are useful to get insight about the collected data to formalize the research question and must be performed before fitting any model. Quantization transforms a continuous variable into a categorical variable. Quantization of continuous variables into bins is extremely useful when trying to model non-linearity in the data. Alternatives consist of finding a good transformation such as splines. Existing algorithms for optimal partitioning of a continuous variable are suitable to response modeling but not to uplift modeling [Garcia et al., 2013]. In practice, when exploring uplift data, the partition is performed with two options: equal length intervals and equal frequency intervals. For example, the bins are based on the deciles of the variable in the `niv()` function from **uplift** R Package. Here, we suggest a univariate supervised quantization tree-based algorithm for optimal partitioning similar to CART [Breiman et al., 1984] with a modified splitting criterion based on hypothesis testing for uplift. The same idea is extended to bivariate quantization in order to look for for

potential interactions. Interactions may arise when considering the relationship among three or more variables, and describes a situation in which the simultaneous influence of two variables on a third is not additive. We build a non-parametric supervised quantization algorithm guided by the observed uplift, where the two-dimensional feature space is divided in rectangles. In addition, the R Package **tools4uplift** provides visualization tools for both quantization methods: uplift barplots for the univariate case, and heatmaps for the bivariate case.

4.3.1. Univariate quantization

Recursive partitioning provides an ideal method for supervised quantization of continuous variables. We follow the CART [Breiman et al., 1984] framework. Two main goals of recursive partitioning are to find the best cut points and to find the finite number of regions that are better adapted to the learning task. Therefore, quantization requires the development of the following two subjects. First, *splitting criterion* is the criterion made for choosing the best cut points in order to split a set of distinct numeric values into intervals. Second, *stopping criterion* is a criterion for stopping the quantization process in order to yield the finite number of intervals.

Suppose that n independent units are observed $(y_i, \mathbf{x}_i, t_i)$, $i = 1, \ldots, n$. The number of treated units is $n_t = \sum_{i=1}^{n} t_i$ and the number of control units is $n_c = \sum_{i=1}^{n}(1 - t_i)$. The objective is to quantize a continuous explanatory variable X in order to find out whether there exists subgroups of individuals in which the treatment shows heterogeneous effects, and if so, how the treatment effect varies accross them. Formally, the goal is to split the sample Ω (or root node) into two child nodes Ω_{left} and Ω_{right} based on X in a way that

$$u_l \neq u_r, \tag{4.3.1}$$

with respect to a certain criterion, where u_l and u_r are the two uplifts in the left and right child nodes respectively. We want to build a statistical test. The idea is to test different split values and the ones that are significant (with a p-value smaller than a pre-specified threshold) are eligible to be chosen. For instance, one can choose the split with smallest p-value. The procedure is then repeated recursively into each child node until the stopping rules are satisfied.

For this section, assume that we are given a specific split point x. Observations that satisfy the condition $\{X < x\}$ go to the left child node (Ω_{left}) and observations that do not satisfy the condition go to the right child node (Ω_{right}).

The uplift in Equation (4.2.1) can be used in order to reorganize Equation (4.3.1) in terms of the following hypothesis

$$\begin{cases} \mathrm{H}_0 : p_{lt} - p_{lc} = p_{rt} - p_{rc} \\ \mathrm{H}_1 : p_{lt} - p_{lc} \neq p_{rt} - p_{rc} \end{cases},$$

where $_l$ and $_r$ subscripts refer to *left* and *right* child nodes respectively, $_t$ and $_c$ refer to treatment and control groups respectively, and

$$p_{lt} = \Pr(Y = 1 \mid \Omega_{\text{left}}, T = 1),\ p_{lc} = \Pr(Y = 1 \mid \Omega_{\text{left}}, T = 0),$$
$$p_{rt} = \Pr(Y = 1 \mid \Omega_{\text{right}}, T = 1),\ p_{rc} = \Pr(Y = 1 \mid \Omega_{\text{right}}, T = 0),$$

with $n_{lt}, n_{lc}, n_{rt}, n_{rc}$ the associated sample sizes and $n_{lt} + n_{lc} + n_{rt} + n_{rc} = n$. With the assumption of randomization, treatment and control groups are independent. But, within each group, the assignment to left or right nodes of each observation depends on the split point. Before we go into the details, we need to define two last quantities for the root node Ω,

$$p_t = \Pr(Y = 1 \mid \Omega, T = 1),\ p_c = \Pr(Y = 1 \mid \Omega, T = 0).$$

Table 4.1 gives us a possible direction on how we can build the statistical test for uplift. Conditional on a split point, the treatment group split can be represented in a 2×2 contingency table. The same development applies for the control group. Let us first focus on the treatment group.

	Left Node ($X < x$)	Right Node ($X \geq x$)	Total
Responder ($Y = 1$)	z_t	$p_t n_t - z_t$	$p_t n_t$
Non-responder ($Y = 0$)	$n_{lt} - z_t$	$n_{rt} - (p_t n_t - z_t)$	$(1 - p_t) n_t$
Total	n_{lt}	n_{rt}	n_t

Tab. 4.1. Conditional on a given split, the observations of the treatment group can be represented into a 2×2 contingency table with the variables *node assignment* and *response*.

We are interested in the number of responder units (i.e. $Y = 1$) in the left and right nodes, that is, z_t and $p_t n_t - z_t$ in Table 4.1, conditional on a given split. The total number of units that go to the left node n_{lt} is necessarily random and unknown prior to the split. Once the split is made, we can determine n_{lt} and calculate the distribution of responders in the left and right nodes for the given value of n_{lt}.

Now, for a given split, the number of responders units out of the n_{lt} units that are assigned to the left node follows a Binomial distribution $\mathcal{B}(n_{lt}; p_{lt})$. Similarly, the number of responder units out of the n_{rt} units that are assigned to the right node follows a Binomial distribution $\mathcal{B}(n_{rt}; p_{rt})$. Their odds ratio is given by

$$\omega_t = \frac{\omega_{lt}}{\omega_{rt}} = \frac{p_{lt}/(1 - p_{lt})}{p_{rt}/(1 - p_{rt})},$$

where ω_{lt} and ω_{rt} are the odds for the left node and right node groups respectively (for treatment observations). The sampling distribution of responder units assigned to the left node Z_t conditional upon the split is Fisher's noncentral hypergeometric distribution [Fog, 2008]. Its parameters are: $n_{lt},\ n_{rt} \in \mathbb{N}$; $n_t = n_{lt} + n_{rt}$; $p_t n_t \in \mathbb{N}$ and $0 \leq p_t n_t < n_t$; and $\omega_t \in \mathbb{R}_+$.

If $\omega_t = 1$, it simplifies to the (central) hypergeometric distribution. An implementation for R is available as the package named **BiasedUrn** [Fog, 2015] and includes univariate and multivariate probability mass functions, distribution functions, quantiles, random variable generating functions, mean and variance.

The hypothesis can be rearranged such as

$$\begin{cases} \mathrm{H}_0 : p_{lt} - p_{rt} = p_{lc} - p_{rc} \\ \mathrm{H}_1 : p_{lt} - p_{rt} \neq p_{lc} - p_{rc} \end{cases}.$$

For the left-hand-side of H_0, we consider the following estimators based on Table 4.1

$$\hat{p}_{lt} = \frac{z_t}{n_{lt}},$$
$$\hat{p}_{rt} = \frac{p_t n_t - z_t}{n_{rt}}.$$

Using the noncentral hypergeometric distribution properties, we can compute

$$\mathbb{E}[\hat{p}_{lt} - \hat{p}_{rt}] = \frac{n_t \mathbb{E}[Z_t]}{n_{lt} n_{rt}} - \frac{p_t n_t}{n_{rt}},$$
$$\mathbb{V}[\hat{p}_{lt} - \hat{p}_{rt}] = \frac{n_t^2 \mathbb{V}[Z_t]}{n_{lt}^2 n_{rt}^2}.$$

where $\mathbb{E}[.]$ stands for the mathematical expectation and $\mathbb{V}[.]$ stands for variance. We can compute $\mathbb{E}[Z_t]$ and $\mathbb{V}[Z_t]$ using the following functions from the package **BiasedUrn**

```
meanFNCHypergeo(m1, m2, n, odds, precision=1E-7)
varFNCHypergeo(m1, m2, n, odds, precision=1E-7)
```

where `m1` is n_{lt} and `m2` is n_{rt}. The argument `n` represents the total number of responder units in the root node $p_t n_t$ and the `odds` argument is ω_t. Since ω_t is an unknown parameter, we also estimate it using values from Table 4.1 such as

$$\hat{\omega}_t = \frac{z_t/(n_{lt} - z_t)}{(p_t n_t - z_t)/\{n_{rt} - (p_t n_t - z_t)\}}.$$

The same development applies to the control group where we only need to replace the subscript $_t$ by $_c$. Therefore, we define the statistic associated with the uplift test

$$\begin{cases} \mathrm{H}_0 : (p_{lt} - p_{rt}) - (p_{lc} - p_{rc}) = 0 \\ \mathrm{H}_1 : (p_{lt} - p_{rt}) - (p_{lc} - p_{rc}) \neq 0 \end{cases}$$

based on the asymptotic pivotal quantity

$$z_{\text{obs}} = \frac{[(\hat{p}_{lt} - \hat{p}_{rt}) - (\hat{p}_{lc} - \hat{p}_{rc})] - \mathbb{E}[(\hat{p}_{lt} - \hat{p}_{rt}) - (\hat{p}_{lc} - \hat{p}_{rc})]}{\sqrt{\mathbb{V}[(\hat{p}_{lt} - \hat{p}_{rt}) - (\hat{p}_{lc} - \hat{p}_{rc})]}} \tag{4.3.2}$$

where, by linearity of the mathematical expectation,

$$\mathbb{E}[(\hat{p}_{lt} - \hat{p}_{rt}) - (\hat{p}_{lc} - \hat{p}_{rc})] = \frac{n_t \mathbb{E}[Z_t]}{n_{lt} n_{rt}} - \frac{p_t n_t}{n_{rt}} - \frac{n_c \mathbb{E}[Z_c]}{n_{lc} n_{rc}} + \frac{p_c n_c}{n_{rc}},$$

and because of the assumption of independence between treatment and control groups,

$$\mathbb{V}[(\hat{p}_{lt} - \hat{p}_{rt}) - (\hat{p}_{lc} - \hat{p}_{rc})] = \frac{n_t^2 \mathbb{V}[Z_t]}{n_{lt}^2 n_{rt}^2} + \frac{n_c^2 \mathbb{V}[Z_c]}{n_{lc}^2 n_{rc}^2}.$$

By the Central Limit Theorem, the statistic given by the right-hand-side of Equation (4.3.2) is asymptotically normally distributed under the null hypothesis; therefore the test

rejects H_0 at a level α when

$$| z_{\text{obs}} | > z_{\frac{\alpha}{2}} \tag{4.3.3}$$

where z_α denotes the upper-tail α-percentile of the standard normal distribution. We will use $| z_{\text{obs}} |$ as the splitting statistic.

Several split points x can satisfy this inequality for a fixed α. The best split can be defined as the one that yields the maximum $| z_{\text{obs}} | > z_{\frac{\alpha}{2}}$ among all permissible splits. Once the best split is chosen, the observations in the parent node are then split according to it. The same procedure is applied to split both child nodes. Recursively doing so results in quantization of the continuous variable X. The natural stopping criterion is met when no more splits are significant.

The function that performs the optimal partitioning is called `BinUplift()`. Its arguments are

```
BinUplift(data, treat, outcome, x, n.split = 10, alpha = 0.05,
          n.min = 30)
```

where `data, treat, outcome` are the arguments for the data, treatment indicator and outcome variable of interest. The `x` argument is the name of the explanatory variable to quantize by trying `n.split` equidistant values in the range of the variable. The arguments `alpha` and `n.min` control the performance of the statistical test: `alpha` is the significance level of the test; `n.min` is the minimum number of observations in each group (treatment or control) required to consider a split. The function returns a vector of split points for variables that are successfully quantized. If it is not possible to quantize the variable at a level `alpha`, the function returns a message indicating that no split was possible at the given significance level.

Remark 4.3.1. *If X is a nominal explanatory variable with K different categories, one can transform it into an ordinal variable sorted from the lowest to the highest observed uplift categories. Using the ranking of these categories, one can consider $K - 1$ possible splits to test. This idea is useful in practice when a nominal variable has a large number of categories [Su et al., 2009].*

4.3.2. Uplift heatmap

Suppose that we want to quantize simultaneously two continuous explanatory variables X_1 and X_2 so as to construct a single categorical interaction variable $X_{1,2}$. The idea is to partition the plane into disjoint rectangles S based on their associated observed uplifts

$$u_S = \sum_{i \in S} y_i t_i / \sum_{i \in S} t_i - \sum_{i \in S} y_i (1 - t_i) / \sum_{i \in S} (1 - t_i).$$

Algorithm 4.1 Uplift Bivariate Quantizaion

1: X_1, $X_2 \leftarrow$ two continuous explanatory variables
2: $b > 1 \leftarrow$ number of intervals each variable will be cut into
3: Find the minimum and the maximum values of X_1 and X_2.
4: Divide the feature space $\{X_{1,\min}, X_{1,\max}\} \times \{X_{2,\min}, X_{2,\max}\}$ into b^2 rectangles.
5: Compute the observed uplift in each rectangle.
6: Predict the individual uplift of each observation by the observed uplift of its rectangle u_S.
7: Output a new categorical variable $X_{1,2}$ where the categories are the sorted (from the highest to the lowest) predicted uplift values.

The method we propose works as described in Algorithm 4.1. Note that the parameter b can be set to the optimizer of a cross-validation criterion based on an uplift goodness-of-fit measure. The function that creates the heatmap and the associated bivariate quantization is called `BinUplift2d()`. Its arguments are

```
BinUplift2d(data, var1, var2, treat, outcome, valid = NULL,
            n.split = 10, n.min = 30, plotit = TRUE, nb.col = 20)
```

where `data` is a data frame containing the variables of interest `var1, var2`. The argument `n.split` corresponds to the parameter b of Algorithm 4.1. For visualization purposes, the argument `plotit` is set by default to `TRUE`. The function returns a heatmap of observed uplifts per rectangle containing a minimum of `n.min` observations per treatment and control groups. `BinUplift2d()` also returns an augmented dataset (and an augmented validation

set if a `valid` dataset is provided to the function) with a new variable `Uplift_var1_var2`, representing the observed uplift within each of the `n.split` × `n.split` rectangles.

4.4. Qini-based uplift

Typically, model validation is accomplished by choosing an appropriate loss function to define the lack of fit between the predicted and the actual values of the response variable at the individual observational units. Assessing model performance is more complex for uplift modeling, as the actual value of the response, that is, the *true* uplift, is unknown at the individual subject level. However, one can assess model performance by comparing groups of observations. For uplift models, this is achieved with the Qini coefficient [Radcliffe, 2007].

4.4.1. The adjusted Qini

Most often used in economics, the Gini coefficient [Gini, 1997] aims at measuring the model's goodness-of-fit and is one of the measures used in direct marketing for traditional response models. One way of computing the Gini coefficient is to first draw a Lorenz curve [Lorenz, 1905]. The plot depicting the Lorenz curve illustrates the goodness-of-fit of a response model. The predicted scores of the targeted observations are sorted in decreasing order. The horizontal axis represents the observed cumulative percentages associated to the sorted predicted scores with respect to the whole targeted sample. The vertical axis, the Lorenz curve, depicts the ratio of the cumulative response lift associated with each cumulative percentage to the total number of responses. The Gini coefficient is a single index of model performance based on the Lorenz curve. Radcliffe [2007] proposes a straightforward extension of the Lorenz curve and the Gini coefficient for uplift modeling: the *Qini curve* and the *Qini coefficient*. Basically, the Qini curve is a Lorenz curve where the predictive scores are replaced by the predicted uplifts. The intuition is that a good model should be able to select individuals with positive uplift first. More explicitly, for a given model, let $\hat{u}_{(1)} \geq \hat{u}_{(2)} \geq ... \geq \hat{u}_{(n)}$ be the sorted predicted uplifts. Let $\phi \in [0,1]$ be a given proportion and let $N_\phi = \{i : \hat{u}_i \geq \hat{u}_{(\lceil \phi n \rceil)}\} \subset \{1, \ldots, n\}$ be the subset of individuals with the $\phi n \times 100\%$ highest predicted uplifts $\hat{u}_i$. As a function of the fraction of population targeted ϕ, the

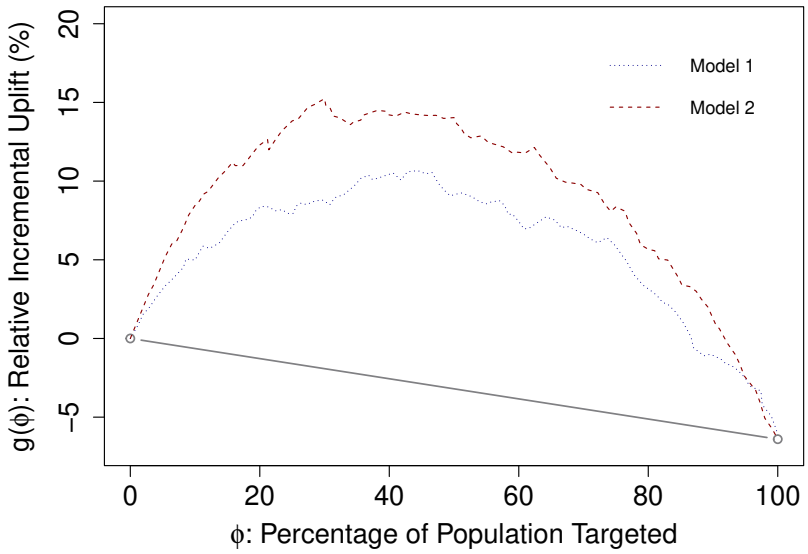

Fig. 4.1. Example of Qini curves corresponding to two different uplift models. The straight gray line corresponds to a random targeting strategy.

relative incremental uplift is defined as

$$g(\phi) = \left(\sum_{i \in N_\phi} y_i t_i - \sum_{i \in N_\phi} y_i (1 - t_i) \left\{ \sum_{i \in N_\phi} t_i / \sum_{i \in N_\phi} (1 - t_i) \right\} \right) / \sum_{i=1}^{n} t_i, \tag{4.4.1}$$

where $\sum_{i \in N_\phi} (1 - t_i) \neq 0$, with $g(0) = 0$. The incremental uplift has been normalized by the number of subjects treated in N_ϕ. Note that $g(1) = \bar{u}$ where $\bar{u}$ is the overall observed uplift $\bar{u} = \sum_{i=1}^{n} y_i t_i / \sum_{i=1}^{n} t_i - \sum_{i=1}^{n} y_i (1 - t_i) / \sum_{i=1}^{n} (1 - t_i)$.

The Qini curve is constructed by plotting $g(\phi)$ as a function of $\phi \in [0,1]$. This is illustrated in Figure 4.1. The figure can be interpreted as follows: the x-axis represents the fraction of targeted individuals and the y-axis shows the incremental number of positive responses relative to the total number of targeted individuals. The straight line between the points (0,0) and $(1, \bar{u})$ in Figure 4.1 represents a benchmark to compare the performance of the model to a strategy that would randomly target subjects. The Qini coefficient q is a single index of model performance. It is defined as the area under the Qini curve. This area can be approximated using a Riemann sum such as the trapezoid formula: the domain of $\phi \in [0,1]$ is partitioned into J panels, or $J + 1$ grid points $0 = \phi_1 < \phi_2 < ... < \phi_{J+1} = 1$, to compute

the empirical estimation of the Qini coefficient $\hat{q}$ as

$$\hat{q} = \int_0^1 Q(\phi)\mathrm{d}\phi \approx \frac{1}{2}\sum_{j=1}^{J}(\phi_{j+1} - \phi_j)\{Q(\phi_{j+1}) + Q(\phi_j)\} \times 100\%, \tag{4.4.2}$$

where $Q(\phi) = g(\phi) - \phi\ \bar{u}$. In general, when comparing several models, the preferred model is the one with maximum Qini coefficient. To be able to compute the Qini coefficient, we first need to find the coordinates of the Qini curve. This is achieved using the `PerformanceUplift()` function

```
PerformanceUplift(data, treat, outcome, prediction, nb.group = 10)
```

where `data, treat, outcome` are the necessary arguments in order to fit an uplift model and `prediction` is the predicted uplift value for the `data`. The uplift values could be the output of `predict()`, or any other statistical method that gives an uplift prediction. The `nb.group` argument represents the J panels used in order to construct the Qini curve and compute the Qini coefficient. The number of panels is usually $J \geq 2$ and, depending on the available data points, could be as large as the user would like. In practice, the results are presented with 5 or 10 groups. In order to display the Qini curve, we use the `plot(x, ...)` function where `x` is an object of class `PerformanceUplift`. Finally, adding a second curve on the same figure in order to compare two models is done using the `lines(x, ...)` function.

The results from the `PerformanceUplift()` function can also be used to draw a barplot representing the observed uplift between two grid points j and $j+1$, $j \in \{0,...,J\}$, as a function of the predicted uplift by the model, as shown in Figure 4.2. This is done with the `barplot(x, ...)` function. A decreasing disposition of the uplift values in the J bins is an important property of an uplift model. To measure the degree to which a model does this correctly, the use of the Kendall rank correlation [Kendall, 1938] between the predicted uplift and the observed uplift has been suggested in Belbahri et al. [2021]. The Kendall's uplift rank correlation is defined as

$$\rho = \frac{2}{J(J-1)}\sum_{i<j}\mathrm{sign}(\bar{\hat{u}}_i - \bar{\hat{u}}_j)\ \mathrm{sign}(\bar{u}_i - \bar{u}_j), \tag{4.4.3}$$

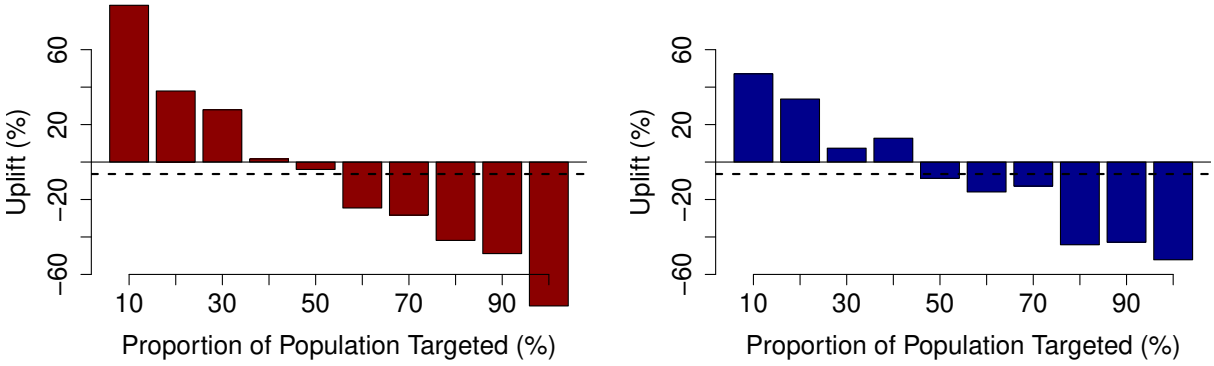

Fig. 4.2. Theoretical predicted uplift barplots with 10 panels corresponding to two different models. A good model should order the observed uplift from highest to lowest. The Kendall's uplift rank correlation is $\rho = 1$ for the left barplot and $\rho = 0.87$ for the right barplot. Dashed lines represent the overall observed uplift.

where $\bar{\hat{u}}_k$ is the average predicted uplift in the kth bin, $k \in \{1,...,J\}$, and $\bar{u}_k$ is the observed uplift in the same bin. Then, by combining (4.4.2) and (4.4.3), the *adjusted Qini coefficient* is defined as

$$\hat{q}_{\text{adj}} = \rho \max\{0, \hat{q}\}. \tag{4.4.4}$$

Maximizing the adjusted Qini coefficient maximizes the area under the Qini curve and encourages grouping the individuals in decreasing uplift bins [Belbahri et al., 2021]. The `QiniArea(x, adj=TRUE)` function uses `x`, the output of `PerformanceUplift()` as an input in order to compute the adjusted Qini coefficient.

4.4.2. Qini-based variable selection

Model selection refers to selecting the right (or best) model according to a given criterion. It is usually accomplished by selecting a subset of the variables available in a given dataset. Model selection is useful because it reduces the dimension of the model, avoids overfitting, and improves model stability and accuracy. When the input space dimension is small, knowledge-based approaches to identify a good set of variables can easily be performed and is sometimes preferable. In other situations, we may have a large number of potentially important variables and it soon becomes a time consuming effort to follow a manual variable selection process. In this case, we may consider using automatic subset selection tools.

Popular linear variable selection techniques are forward, backward, stepwise [Montgomery et al., 2012], and stage-wise selection [Hastie et al., 2007], as well as more recent techniques such as lasso [Tibshirani, 1996], and LARS [Efron et al., 2004], among others. However, these techniques have not been designed for uplift models, so they need to be adapted. In this work, we have chosen to adapt lasso because of its popularity and success in selecting variables when dealing with complex and high-dimensional models. We suggest a two-stage approach. Our adapted lasso algorithm chooses the regularization hyper-parameter, that is, the penalty parameter, in adequacy with uplift models performance measures, i.e., by maximizing the adjusted Qini coefficient $\hat{q}_{\text{adj}}$.

Consider the interaction model of Section 4.2.2. Let $\lambda > 0$ be the penalty constant. For any given λ, let $(\hat{\beta}_o(\lambda), \hat{\boldsymbol{\beta}}(\lambda), \hat{\gamma}(\lambda), \hat{\boldsymbol{\delta}}(\lambda))$ be the value of the parameters that maximizes the penalized log-likelihood

$$\ell(\beta_o, \boldsymbol{\beta}, \gamma, \boldsymbol{\delta} \mid \lambda) = \sum_{i=1}^{n} \left\{ y_i \log\left(\frac{p_i}{1-p_i}\right) + \log(1-p_i) \right\} + \lambda \left\{ |\gamma| + \sum_{j=1}^{p} \Big(|\beta_j| + |\delta_j| \Big) \right\}, \quad (4.4.5)$$

where

$$p_i = \Pr(Y_i = 1 \mid \mathbf{x}_i, t_i, \beta_o, \boldsymbol{\beta}, \gamma, \boldsymbol{\delta}) = \Big(1 + \exp\{-(\beta_o + \mathbf{x}_i^\top \boldsymbol{\beta} + \gamma t_i + t_i \mathbf{x}_i^\top \boldsymbol{\delta})\}\Big)^{-1}.$$

Let $\hat{q}_{\text{adj}}(\lambda)$ be associated with the model with parameters $(\hat{\beta}_o(\lambda), \hat{\boldsymbol{\beta}}(\lambda), \hat{\gamma}(\lambda), \hat{\boldsymbol{\delta}}(\lambda))$. Our lasso procedure solves

$$(\hat{\beta}_o(\hat{\lambda}), \hat{\boldsymbol{\beta}}(\hat{\lambda}), \hat{\gamma}(\hat{\lambda}), \hat{\boldsymbol{\delta}}(\hat{\lambda})) = \underset{\lambda}{\text{argmax}} \ \hat{q}_{\text{adj}}(\lambda). \quad (4.4.6)$$

Using the `glmnet()` function from the **glmnet** R Package in order to generate the regularization path [Friedman et al., 2010], we defined a new function `LassoPath()` that is callable directly from the R Package **tools4uplift**. This function is used inside the function `BestFeatures()` which returns the variables and interaction terms that maximize the Qini coefficient. The arguments of the function are

```
BestFeatures(data, treat, outcome, predictors, nb.group = 10, ...)
```

where `data, treat, outcome` and `predictors` are defined as above. The argument `nb.group` is the number of panels J used to compute the Qini coefficient. The `...` default arguments can be passed to the function. For example, if `validation` is set to `TRUE`,

the function performs cross-validation. By default, the validation set is fixed to a randomly chosen 30% of the data, `p = 0.3`. The function returns a vector of names of the selected features. The output of the function can be used directly in the `InterUplift()` function in order to fit the second stage of the modeling process. In this case, the second stage of the modeling process estimates the coefficients of the selected variables by maximizing the non-penalized likelihood.

4.4.3. Qini-based uplift regression

The interaction model introduced in Section 4.2.2 is not optimized with respect to the goodness-of-fit measures designed for uplift. Instead, the parameters are estimated with respect to the likelihood. The methodology introduced in Belbahri et al. [2021] was specifically conceived for parameter estimation in the uplift regression context. This methodology is based on a derivative-free optimization of the $\hat{q}_{\text{adj}}$ and imposes sparsity. Empirical results show that estimating the regression parameters by maximizing the adjusted Qini significantly improves the uplift models prediction performance.

Recall the uplift model penalized log-likelihood given in (4.4.5). In the same spirit as for the Qini-based variable selection, applying the pathwise coordinate descent algorithm [Friedman et al., 2007] to the uplift model gives a sequence of critical regularization values $\lambda_1 < \cdots < \lambda_{\min\{n,2p+1\}}$ and corresponding model parameters $\{(\hat{\beta}_o(\lambda_j), \hat{\boldsymbol{\beta}}(\lambda_j), \hat{\gamma}(\lambda_j), \hat{\boldsymbol{\delta}}(\lambda_j))\}_{j=1}^{\min\{n,2p+1\}}$ associated with different model dimensions. Once again, this is achieved using our `LassoPath()` function. Now, because the adjusted Qini function is not straightforward to optimize with respect to the parameters, one needs to explore the parameters space in order to find the maximum. Belbahri et al. [2021] use Latin hypercube sampling (LHS) to find the coefficient parameters that maximize the adjusted Qini. LHS is a statistical method for quasi-random sampling based on a multivariate probability law inspired by the Monte Carlo method [McKay et al., 2000]. For each λ_j, $j = 1,...,\min\{n,2p+1\}$, using the `improvedLHS()` function from the **lhs** R Package [Carnell, 2019], we generate a LHS comprising L points in the neighborhood of $(\hat{\beta}_o(\lambda_j), \hat{\boldsymbol{\beta}}(\lambda_j), \hat{\gamma}(\lambda_j), \hat{\boldsymbol{\delta}}(\lambda_j))$, and evaluate the adjusted Qini on each of these points. The optimal coefficients are estimated as those coefficients among the $(\min\{n,2p+1\} \times L)$ LHS points that maximize the adjusted Qini (please refer to Belbahri et al. [2021] for more

details). Our implementation of the Qini-based uplift regression follows the same logic as the one of the interaction model in Section 4.2.2. The function `qLHS()` has the following arguments

```
qLHS(data, treat, outcome, predictors, lhs_points = 50, lhs_range = 1,
    adjusted = TRUE, nb.group = 10, ...)
```

where `lhs_points` is the number of points L to generate in the neighborhood of each penalized estimate $(\hat{\beta}_o(\lambda_j), \hat{\boldsymbol{\beta}}(\lambda_j), \hat{\gamma}(\lambda_j), \hat{\boldsymbol{\delta}}(\lambda_j))$ and `lhs_range` controls the size of the neighborhood. The remaining arguments are related to the Qini coefficient and to the `PerformanceUplift()` function. The function returns an object of class `InterUplift`.

4.5. Application

In this section, we analyze a publicly available dataset from a marketing campaign [Hillstrom, 2008] using the R Package **tools4uplift**. The data contain records of 64,000 customers who last purchased a product within twelve months. The individuals were randomly assigned to three groups; two groups were targeted by two different e-mail campaigns and one group served as control. The treatment assignment was performed in a randomized experiment fashion: a third of the individuals were randomly chosen to receive an e-mail campaign featuring men merchandise, another third were randomly chosen to receive an e-mail campaign featuring women merchandise, and the last third, the control group, did not receive any form of initiative. The results were tracked during a period of two weeks following the e-mail campaign. Some questions can be answered with an uplift model: What is the incremental response of customers targeted by any of two campaigns? Is there a way to optimally select the subset of customers that should be targeted? Conversely, is there a subset of customers that should be removed from future campaigns? The historical customer attributes available include `recency` which indicates the number of months since the last purchase; `history` which is the amount in dollars spent in the past year; two binary variables indicating if the customer purchased `men` merchandise or `women` merchandise in the past year; the `zip_code` of the customer categorized as urban, suburban or rural; an indicator variable `newbie` indicating if the customer is a new customer in the past twelve months; and the `channel` from

which the customer purchased in the past year, i.e., by phone, web or both. For variable selection purposes, we augment the data with iid covariates for which the observations are sampled from a standard Gaussian distribution $\mathcal{N}(0,1)$. The treatment allocation variable included in the dataset is `segment`. In this application, we only focus on the target variable `visit` which is a binary variable indicating whether or not the customer visited the website. Moreover, to simplify the analysis, we restrict the treatment data to the treatment group `treat = 1` that received e-mail on women merchandise, and to the control group `treat = 0` that received no e-mail. The overall observed uplift for this marketing campaign is 4.5%.

4.5.1. Baseline model

First, we use the function `SplitUplift()` in order to split the dataset into training and validation datasets with respect to the overall uplift. It is important to partition the data into subsets that keep the same distribution of treated versus nontreated and responders versus nonresponders. This is achieved by specifying the stratification variables in the argument `group = c("treat", "visit")`.

```
R>set.seed(1988)
R>split.data1 <- SplitUplift(data = data1, p = 0.7,
+                            group = c("treat", "visit"))
R>train <- split.data1[[1]]
R>valid <- split.data1[[2]]
```

Using the two-model estimator of Section 4.2.1 we fit a baseline model for comparison purposes. We fit the two-model estimator using the following code

```
R># baseline model on train set: fitting the two-model estimator
R>predictors <- colnames(train[, -c(10,11)])
R>base.tm <- DualUplift(train, "treat", "visit", predictors)
```

The function returns an object of class `DualUplift`. Its first element is the baseline model fitted for nontreated individuals and the second is the baseline model fitted for treated individuals. Using the validation set, the function `predict()` predicts the uplift.

```
R># predict the uplift on the validation set
R>base.tm.valid <- valid
R>base.tm.valid$pred <- predict(base.tm, base.tm.valid)
```

Finally, to evaluate the quality of the baseline model, we plot the Qini curve and the uplift barplot and we compute the adjusted Qini coefficient with `QiniArea()`. We use `nb.group = 5` to evaluate all models.

```
R># evaluate the model's performance
R>base.tm.perf <- PerformanceUplift(base.tm.valid,
+                                    "treat",
+                                    "visit",
+                                    "pred",
+                                    nb.group = 5)
R>plot(base.tm.perf, type = 'b', lwd = 2, col= 'blue4',
+      cex.axis = 1.5, cex.lab = 1.5)

R>barplot(base.tm.perf, col = 'blue4',
+         cex.axis = 1.5, cex.names = 1.5, cex.lab = 1.5)
R>abline(h = 4.5, lwd = 2, lty = 2)
R>round(QiniArea(base.tm.perf, adjusted = TRUE), 2)
[1] 0.84
```

As we can see in the R output above, the adjusted Qini coefficient associated with the baseline model is $\hat{q}_{\text{adj}} = 0.84$. Figure 4.3 shows the performance of the baseline model using the functions `plot()` and `barplot()`. Since the interaction model in Section 4.2.2 adds an interaction term between all predictors and the treatment variable, the resulting estimation

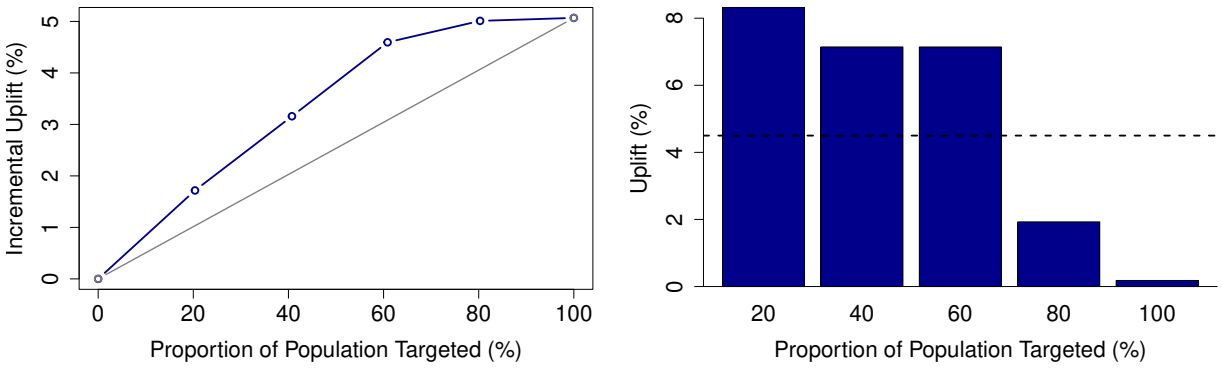

Fig. 4.3. Performance of the baseline model of Section 4.2.1 on a validation set. On the left panel, we see that the Qini coefficient is positive and outperforms random targeting ($\hat{q}_{\text{adj}} = 0.84$). On the right panel, we observe that the baseline model sorts well the individuals to target, but there is room for improvement for the first groups. A good model should order the observed uplift from highest to lowest (see Figure 4.2). The object `PerformanceUplift` is visualized using the `plot()` command (left panel) and the `barplot()` command (right panel).

is equivalent to the one of the two-model estimator. Therefore, we do not present the results here. However, for the rest of the analysis, we will use the interaction model estimator `InterUplift()` for feature selection and parameters estimation using `BestFeatures()` and `qLHS()` functions. In theses cases, we hope the results will improve compared to the baseline model.

4.5.2. Univariate quantization

The dataset contains two continuous variables, `recency` and `history`. We want to quantize both variables using the function `BinUplift()`.

```
R>bin.recency <- BinUplift(data = train,
+                           treat = "treat",
+                           outcome = "visit",
+                           x = "recency",
```

```
+                          n.split = 100,
+                          alpha = 0.05)
R>bin.recency
[1] "oups..no significant split"
```

For a significance level of $\alpha = 0.05$, the decision tree does not find any significant partition of the data with respect to the `recency` variable. Hence, one can either keep the variable as continuous in the models or increase the level of significance α. For $\alpha = 0.10$, there is indeed a significant split, Figure 4.4 displays the associated barplots on training and validation datasets.

```
R># change the level of signification from 5% to 10%
R>bin.recency <- BinUplift(data = train,
+                          treat = "treat",
+                          outcome = "visit",
+                          x = "recency",
+                          n.split = 100,
+                          alpha = 0.10)
[1] "The variable recency has been cut at:"
[1] 12
R># try with 10% for history
R>bin.history <- BinUplift(data = train,
+                          treat = "treat",
+                          outcome = "visit",
+                          x = "history",
+                          n.split = 100,
+                          alpha = 0.10)
R>bin.history
[1] "oups..no significant split"
```

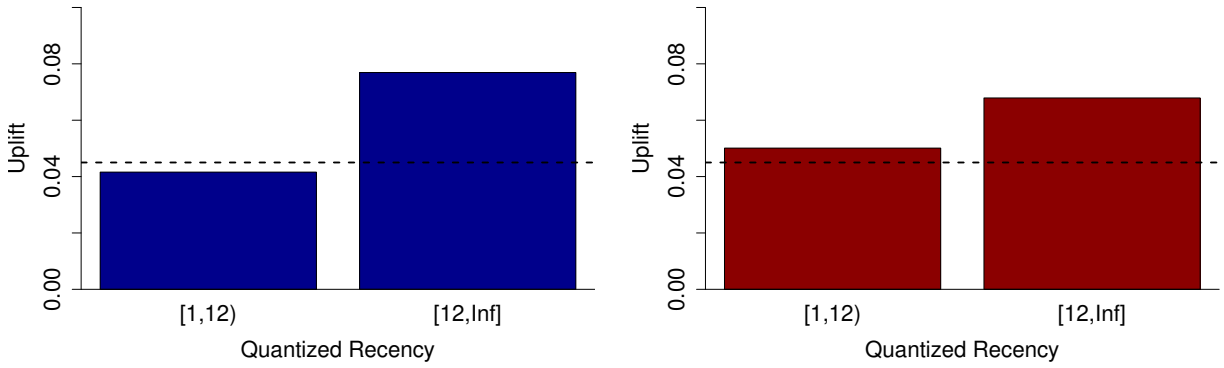

Fig. 4.4. Univariate quantization for `recency` variable with respect to the observed uplift. The variable was quantized using the training dataset observations only (left panel) and the optimal solution gives two groups with significantly ($\alpha = 0.10$) different positive uplift values. The quantization generalizes well for the validation dataset (right panel).

Since there are no significant splits with $\alpha = 0.10$ for variable `history`, we will use the continuous (original) version for the rest of the analysis.

4.5.3. Uplift heatmap

Searching for a possible interaction between `recency` and `history` with respect to the uplift, we use the function `BinUplift2d()` in order to visualize the interaction in a heatmap and create a new categorical variable based on Algorithm 4.1 of Section 4.3.

The following code returns an augmented dataset with a new variable `Uplift_history_recency`, representing the observed uplift within each of the `n.split` $\times$ `n.split` rectangles.

```
R>heatmap <- BinUplift2d(train, "history", "recency", "treat", "visit",
+                        n.split = 3, plotit = TRUE)
```

The function also returns the associated heatmap displayed in Figure 4.5. This visualization suggests an interaction between `recency`, `history` and the uplift. Therefore, one can include an interaction term in the uplift models.

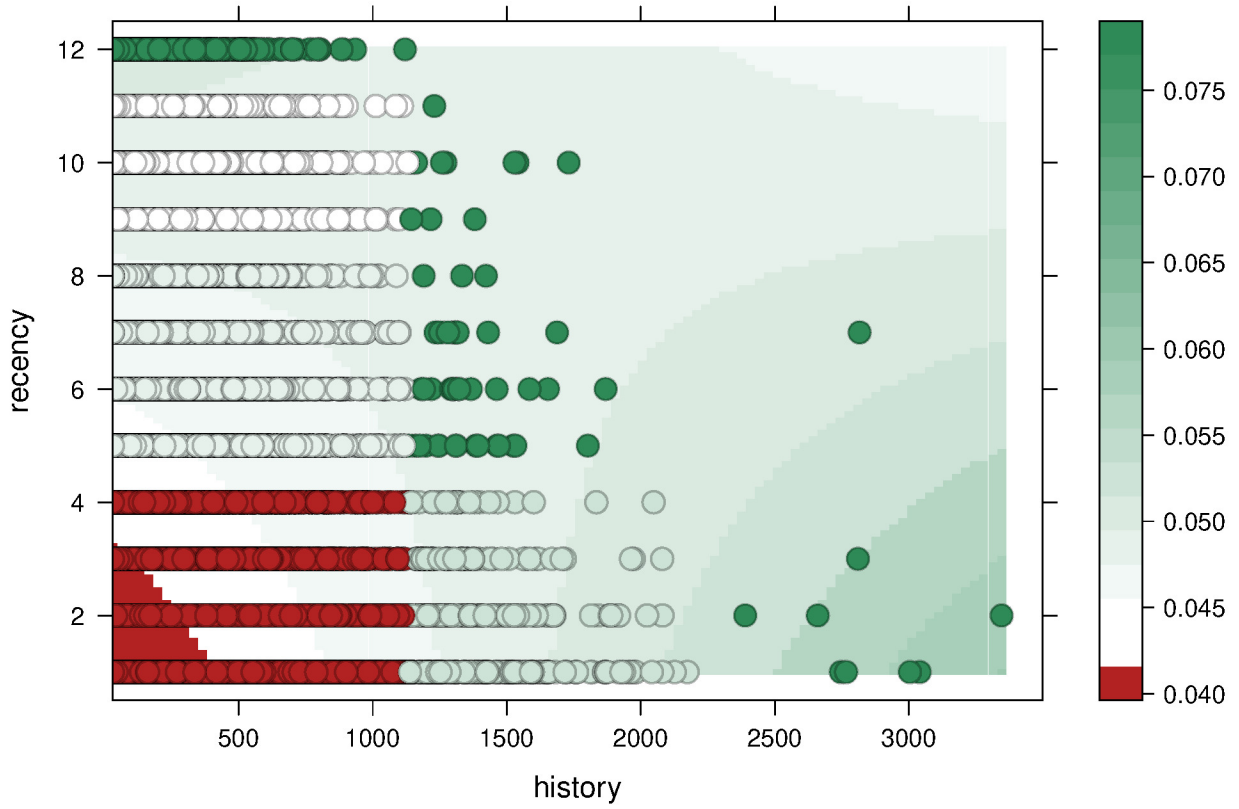

Fig. 4.5. Bivariate quantization with respect to the observed uplift. By default, the `BinUplift2d()` command returns the associated heatmap. The heatmap is based on $b^2 = 9$ rectangles. Note that for customers that spent less than $ 1,000 in the past year, we see a clear difference in terms of uplift as a function of the number of months since last purchase. On the other hand, the observed uplift seems to dependent less on the recency of the last purchase for customers that spent more than $ 1,000. The heatmap colors are based on the rainbow palette with the red color representing the lowest uplift (less than the average) and the green color representing the highest uplift (higher than the average).

4.5.4. Model selection and comparison

The objective of this section is to improve the fitting of the baseline model by including quantized variables and interactions, by performing variable selection and by searching for the optimal parameters with the Qini-based uplift regression. This is achieved using the `BestFeatures()`, `InterUplift()` and `qLHS()` methods.

We compare several models that differ in the number and type of explanatory variables. For example, we compare the fittings with the quantized version of the `recency` variable against models fitted with the original variables. In order to create the quantized version of `recency`, it suffices to use the `predict()` function as follows:

```
R># create categorical variable cat_recency in train and valid datasets
R>train$recency_cat <- predict(bin.recency, train$recency)
R>valid$recency_cat <- predict(bin.recency, valid$recency)
```

where `bin.recency` is an object of type `BinUplift` and the second argument is the original version of the `recency` variable.

Another model is fitted using the `Uplift_history_recency` variable created with the bivariate quantization function `BinUplift2d()`. The following code implements the Qini-based uplift regression model with quantized `recency` and `Uplift_history_recency`. This model yields the best performance. This is seen in Figure 4.6.

```
R># qLHS with quantized recency and interaction
R>predictors <- colnames(train[, -c(1, 10, 11)])
R>qlhs.quant.int.model <- qLHS(train, "treat", "visit",
+                              predictors = predictors,
+                              equal.intervals = TRUE,
+                              nb.group = 5,
+                              lhs_points = 50,
+                              lhs_range = 0.05,
+                              validation=FALSE)

R># standardize the covariates from the validation set
R>qlhs.quant.int.model.valid <- cbind(valid[,c(10, 11)],
+                                      scale(valid[,-c(10,11)]))
R># predict the uplift on the validation set
R>qlhs.quant.int.model.valid$pred <- predict(qlhs.quant.int.model,
```

```
+                                              qlhs.quant.int.model.valid,
+                                              "treat")


R># evaluate the model's performance
R>qlhs.quant.int.model.valid.perf <- PerformanceUplift(
+                                        qlhs.quant.int.model.valid,
+                                        "treat",
+                                        "visit",
+                                        "pred",
+                                        equal.intervals = TRUE,
+                                        nb.group = 5)
R>plot(qlhs.quant.int.model.valid.perf, ylim=c(0,6), col='red4',
+      lty=6, type='l', lwd=2, cex.axis = 1.5, cex.lab = 1.5)
R>barplot(qlhs.quant.int.model.valid.perf, col = 'red4',
+         cex.axis = 1.5, cex.names = 1.5, cex.lab = 1.5)
R>abline(h = 4.5, lwd = 2, lty = 2)
R>round(QiniArea(qlhs.quant.int.model.valid.perf, adjusted=TRUE), 2)
[1] 0.96
```

The R Package **tools4uplift** makes it easy and fast to implement different models with variable selection, with both continuous and categorical variables. Table 4.2 displays the adjusted Qini coefficients associated with different models, evaluated on the validation set. The first column specifies which variables are included in the model. We compare the following methods: `DualUplift()`; `InterUplift()` with automatic variable selection, i.e. using `BestFeatures()` and `qLHS()`. The first line presents the results when the original version of `recency` is used. The second line presents the results when the quantized version of `recency` instead. The third line models use the original version of `recency` but add the quantized version of the interaction between `history` and `recency`. Finally, line four models replace `recency` with its quantized version and add the quantized `Uplift_history_recency` interaction term.

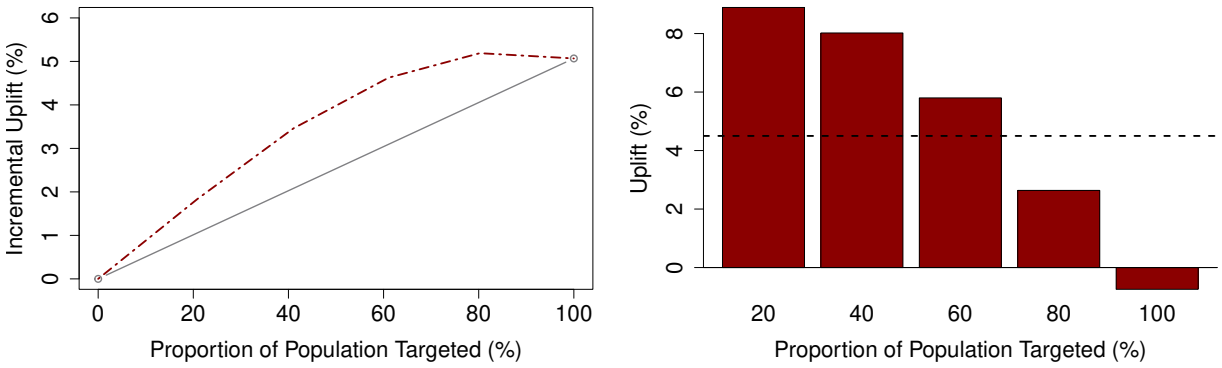

Fig. 4.6. Performance of the best interaction model. The model includes the quantized version of the `recency` variable and the interaction variable `Uplift_history_recency`. The parameters are estimated by maximizing the adjusted Qini coefficient on the training dataset using the `qLHS()` method. The validation adjusted Qini coefficient is $\hat{q}_{\text{adj}} = 0.96$.

Covariates \Method	`DualUplift()`	`BestFeatures()`	`qLHS()`
Original	0.84	0.89	0.92
Original + `BinUplift()`	0.73	0.89	0.91
Original + `BinUplift2d()`	0.69	0.87	0.92
Original + `BinUplift()` + `BinUplift2d()`	0.86	0.89	0.96

Tab. 4.2. Comparison of models performances on a validation set, based on the adjusted Qini coefficient $\hat{q}_{\text{adj}}$. The non linearity introduced by the quantization of `recency` does not seem to help the model. However, when both quantized `recency` and `Uplift_history_recency` are included, `DualUplift()` achieves its highest performance ($\hat{q}_{\text{adj}} = 0.86$). Moreover, guiding variable selection by the Qini coefficient with `BestFeatures()` always improves upon the performance of the baseline model. Finally, estimating the parameters using the `qLHS()` method gives the best results in all scenarios.

4.6. Summary

We presented the methodology associated with the new R Package **tools4uplift** together with an application to a real world marketing campaign dataset, as an illustration of how the

Function	Description
`BestFeatures()`	Qini-based variable selection
`BinUplift()`	Univariate quantization
`BinUplift2d()`	Bivariate quantization
`DualUplift()`	Two-model estimator
`InterUplift()`	Interaction estimator
`LassoPath()`	*lasso* path for the penalized logistic regression
`PerformanceUplift()`	Performance of an uplift model
`QiniArea()`	(adjusted) Qini coefficient
`qLHS()`	Qini-based uplift regression
`SplitUplift()`	Split data with respect to the sample uplift distribution
`UpliftPerCat()`	Uplift barplot for categorical variables

Tab. 4.3. Summary of the functions available in the R Package **tools4uplift**

package could be used to analyse uplift data. The functions presented in this work are summarized in Table 4.3. The purpose of **tools4uplift** is to give practitioners the necessary tools to get some insight about the uplift signal in the context of a randomized experiment. This work deals with five crucial steps in statistical modeling: i) quantization, ii) visualization, iii) variable selection, iv) parameter estimation and v) model validation. All the available functions in the package are thoroughly described and accompanied by a motivating example. The use of **tools4uplift** will enable practitioners to save time and effort when analyzing their uplift data.

Computational details

The results in this paper were obtained using R3.4.4 with the Packages **tools4uplift** [Belbahri *et al.*, 2020], **mvtnorm** [Genz et al., 2018] and **dummies** [Brown, 2012]. R itself and all packages used are available from CRAN at `http://CRAN.R-project.org/`.

Acknowledgments

Mouloud Belbahri was supported in part by the MITACS acceleration program in the context of a research internship[1]. Alejandro Murua was supported in part by the Natural Sciences and Engineering Research Council of Canada through grant number 327689-06. Vahid Partovi Nia was supported by the Natural Sciences and Engineering Research Council of Canada discovery grant 418034-2012.

Bibliography

Mouloud Belbahri, Olivier Gandouet, Alejandro Murua, and Vahid Partovi Nia. *tools4uplift: Tools for Uplift Modeling*, 2020. URL `https://CRAN.R-project.org/package=tools4uplift`. R package version 1.0.0.

Mouloud Belbahri, Alejandro Murua, Olivier Gandouet, and Vahid Partovi Nia. Qini-based uplift regression. *The Annals of Applied Statistics*, 2021. to appear.

Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and Regression Trees*. CRC press, 1984.

Christopher Brown. *dummies: Create dummy/indicator variables flexibly and efficiently*, 2012. URL `https://CRAN.R-project.org/package=dummies`. R package version 1.5.6.

Rob Carnell. *lhs: Latin Hypercube Samples*, 2019. URL `https://CRAN.R-project.org/package=lhs`. R package version 1.0.1.

Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.

Agner Fog. Sampling methods for wallenius' and fisher's noncentral hypergeometric distributions. *Communications in Statistics - Simulation and Computation*, 37(2):241–257, 2008.

Agner Fog. *BiasedUrn: Biased Urn Model Distributions*, 2015. URL `https://CRAN.R-project.org/package=BiasedUrn`. R package version 1.07.

Jerome Friedman, Trevor Hastie, Holger Höfling, Robert Tibshirani, et al. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

[1] `http://www.mitacs.ca/en/programs/accelerate`

Salvador Garcia, Julian Luengo, José Antonio Sáez, Victoria Lopez, and Francisco Herrera. A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, 25(4):734–750, 2013.

Alan Genz, Frank Bretz, Tetsuhisa Miwa, Xuefei Mi, Friedrich Leisch, Fabian Scheipl, and Torsten Hothorn. *mvtnorm: Multivariate Normal and t Distributions*, 2018. URL `https://CRAN.R-project.org/package=mvtnorm`. R package version 1.0-8.

Corrado Gini. Concentration and dependency ratios. *Rivista di politica economica*, 87: 769–792, 1997.

Leo Guelman. *uplift: Uplift Modeling*, 2014. URL `https://CRAN.R-project.org/package=uplift`. R package version 0.3.5.

Leo Guelman et al. *Optimal personalized treatment learning models with insurance applications*. PhD thesis, Universitat de Barcelona, 2015.

Behram Hansotia and Bradley Rukstales. Direct marketing for multichannel retailers: Issues, challenges and solutions. *Journal of Database Marketing and Customer Strategy Management*, 9(3):259–266, 2001.

Dominique M Hanssens, Leonard J Parsons, and Randall L Schultz. *Market response models: Econometric and time series analysis*, volume 12. Springer Science & Business Media, 2003.

Trevor Hastie, Jonathan Taylor, Robert Tibshirani, Guenther Walther, et al. Forward stagewise regression and the monotone lasso. *Electronic Journal of Statistics*, 1:1–29, 2007.

Kevin Hillstrom. The minethatdata e-mail analytics and data mining challenge, 2008. Data retrieved from `https://blog.minethatdata.com/2008/03/minethatdata-e-mail-analytics-and-data.html`.

Paul W Holland. Statistics and causal inference. *Journal of the American Statistical Association*, 81(396):945–960, 1986.

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

Kathleen Kane, Victor SY Lo, and Jane Zheng. Mining for the truly responsive customers and prospects using true-lift modeling: Comparison of new and existing methods. *Journal of Marketing Analytics*, 2(4):218–238, 2014.

Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

Finn Kuusisto, Vitor Santos Costa, Houssam Nassif, Elizabeth Burnside, David Page, and Jude Shavlik. Support vector machines for differential prediction. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 50–65. Springer, 2014.

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.

Max O Lorenz. Methods of measuring the concentration of wealth. *Publications of the American Statistical Association*, 9(70):209–219, 1905.

Michael D McKay, Richard J Beckman, and William J Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61, 2000.

Douglas C Montgomery, Elizabeth A Peck, and G Geoffrey Vining. *Introduction to linear regression analysis*, volume 821. John Wiley & Sons, 2012.

Jerzy S Neyman. On the application of probability theory to agricultural experiments. *Annals of Agricultural Sciences*, 10:1–51, 1923.

Nicholas J Radcliffe and Patrick D Surry. Real-world uplift modelling with significance-based uplift trees. *White Paper TR-2011-1, Stochastic Solutions*, 2011.

NJ Radcliffe. Using control groups to target on predicted lift: Building and assessing uplift models. *Direct Market J Direct Market Assoc Anal Council*, 1:14–21, 2007.

NJ Radcliffe and PD Surry. Differential response analysis: Modeling true response by isolating the effect of a single action. *Credit Scoring and Credit Control VI. Edinburgh, Scotland*, 1999.

Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5):688, 1974.

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

Michał Sołtys, Szymon Jaroszewicz, and Piotr Rzepakowski. Ensemble methods for uplift modeling. *Data Mining and Knowledge Discovery*, 29(6):1531–1559, 2015.

Xiaogang Su, Chih-Ling Tsai, Hansheng Wang, David M Nickerson, and Bogong Li. Subgroup analysis via recursive partitioning. *Journal of Machine Learning Research*, 10(Feb): 141–158, 2009.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

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

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

Yan Zhao, Xiao Fang, and David Simchi-Levi. Uplift modeling with multiple treatments and general response types. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 588–596. SIAM, 2017.

Chapitre 5

Réseaux de neurones

L'objectif de ce chapitre est de donner une brève introduction sur les réseaux de neurones ainsi qu'un aperçu de certaines des étapes conceptuelles qui ont conduit à leur compréhension mathématique actuelle (en particulier aux théorèmes d'approximation universels). Ce chapitre introduit également la notation et le langage utilisé en apprentissage automatique. Pour une bibliographie historique complète et approfondie sur les réseaux de neurones, Schmidhuber [2015] a compilé 888 références. Voir Goodfellow *et al.* [2016] pour une introduction élémentaire bien documentée sur le sujet.

5.1. Introduction

Au milieu des années 1980, l'introduction des réseaux de neurones a marqué un passage de la modélisation prédictive vers l'informatique et l'apprentissage automatique. Un réseau de neurones est un modèle hautement paramétré, inspiré de l'architecture du cerveau humain, qui a été largement promu en tant qu'approximateur universel - une machine qui, avec suffisamment de données, pouvait apprendre toute relation prédictive lisse.

Un réseau de neurones *feed-forward* transforme une entrée $\mathbf{x} \in \mathcal{D} \subset \mathbb{R}^d$ en une sortie $\hat{y} = \text{NN}(\mathbf{x}) \in \mathbb{R}$ (pour une définition formelle, voir par exemple Pinkus [1999] ou Calin [2020]). La Figure 5.1 montre une représentation simple d'un réseau de neurones avec une seule couche cachée. Il y a d prédicteurs, covariables ou *entrées* x_j, m unités *cachées* et une seule unité de *sortie*. Afin de propager les valeurs d'entrée, chaque neurone h_l est connecté à la couche d'entrée via un vecteur de paramètres ou de *poids* $\{\theta_{lj}^{(1)}\}_{j=1}^{d}$ (le $^{(1)}$ se réfère à la connexion entre la première couche C_1 et la couche cachée C_2; $_{lj}$ se réfère à la $j$ième covariable et à la $l$ième unité cachée). La couche cachée possède un nombre de neurones

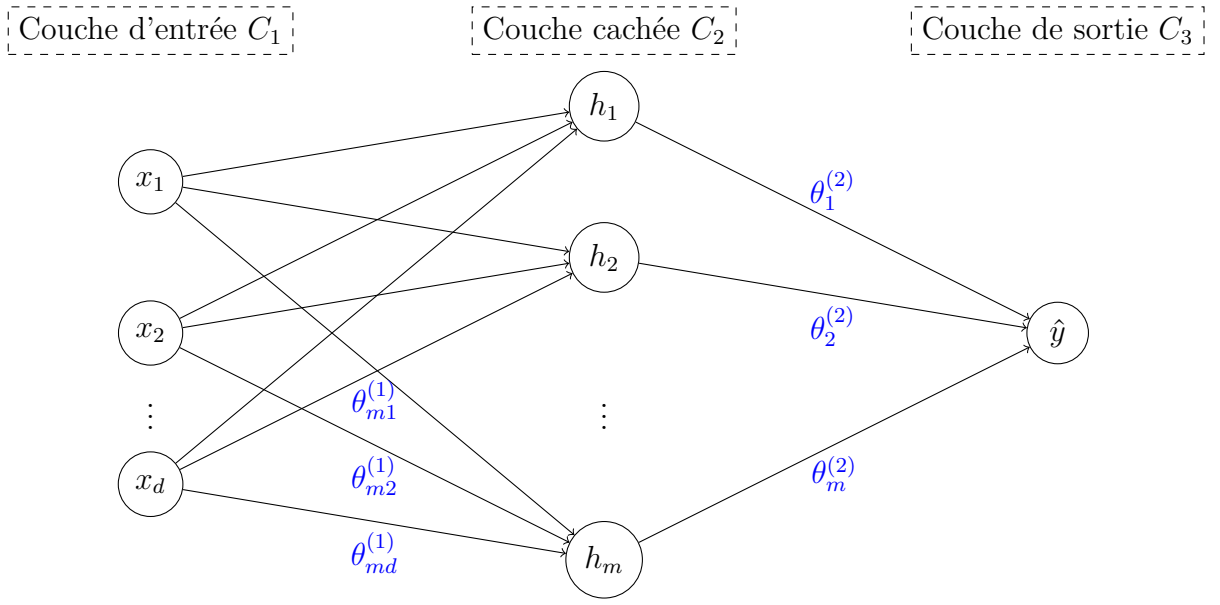

Fig. 5.1. Représentation classique d'un réseau de neurones *feed-forward* avec une seule couche cachée. La couche cachée calcule des transformations des entrées (transformations non linéaires de combinaisons linéaires) qui sont ensuite propagées pour prédire la valeur de la couche de sortie.

arbitraire m, souvent déterminé par la complexité du problème à résoudre. Formellement, posons $s_j^{(1)} = e_j^{(1)} = x_j$ pour $j = 1,...,d$ (c'est-à-dire, les entrées et sorties des neurones de la couche d'entrée C_1 sont simplement les valeurs des covariables). Ensuite, pour $l = 1,...,m$, nous avons les entrées et sorties suivantes :

$$e_l^{(2)} = \theta_{lo}^{(1)} + \sum_{j=1}^{d} \theta_{lj}^{(1)} s_j^{(1)}, \tag{5.1.1}$$

$$s_l^{(2)} = h(e_l^{(2)}) := h_l. \tag{5.1.2}$$

Les termes $\theta_{lo}^{(1)}$ sont appelés *biais* dans le langage de l'apprentissage automatique, et la fonction $h(\cdot)$ est une *non-linéarité*, telle que la fonction logistique (ou sigmoïde), définie par $h(z) = 1/\{1 + \exp(-z)\}$, $z \in \mathbb{R}$. La couche finale ou de sortie C_3 est également connectée aux neurones cachés h_l via le biais $\theta_o^{(2)}$, les poids $\{\theta_l^{(2)}\}_{l=1}^m$ et une fonction de sortie $g(\cdot)$, telle que :

$$e^{(3)} = \theta_o^{(2)} + \sum_{l=1}^{m} \theta_l^{(2)} s_l^{(2)}, \tag{5.1.3}$$

$$s^{(3)} = g(e^{(3)}) := \hat{y}. \tag{5.1.4}$$

Les fonctions $h(\cdot)$ et $g(\cdot)$ sont aussi appelées fonctions *d'activation*. Pour la régression linéaire, $g(\cdot)$ est la fonction identité et, pour une régression logistique (réponse binaire), $g(\cdot)$ est à nouveau la sigmoïde. Noter que, sans la non-linéarité dans la couche cachée, le réseau de neurones se réduirait à un modèle linéaire généralisé [Nelder et Wedderburn, 1972]. Une autre fonction d'activation très populaire aujourd'hui est la fonction unité linéaire rectifiée (ou ReLU pour *Rectified Linear Unit* en anglais), définie par $\text{ReLU}(z) = \max\{0,z\}$, $z \in \mathbb{R}$. En règle générale, les réseaux de neurones sont ajustés par maximum de vraisemblance, généralement avec une variété de formes de régularisation [Goodfellow *et al.*, 2016].

5.2. Théorème d'approximation universel

Avec l'interprétation ci-dessus, un réseau de neurones (NN) est une fonction réelle concrète de d variables réelles, d étant a priori un entier positif arbitraire. Une question naturelle qui se pose depuis longtemps est la suivante : ce NN (avec une architecture appropriée) peut-il approximer des fonctions d'une classe donnée (exemple courant : fonctions continues multivariées définies sur $\mathcal{D} = \mathcal{K}$, un sous-ensemble compact de $\mathbb{R}^d$, mais les espaces fonctionnels plus généraux, tels que les espaces L^p, sont généralement le cadre approprié d'un point de vue mathématique abstrait) [Mhaskar et Hahm, 1996]? En d'autres termes, un NN peut-il générer un sous-ensemble topologiquement dense de la classe de fonctions en question? Par exemple, dans le cas d'un espace vectoriel normé, étant donné $\epsilon > 0$, existe-t-il et, si c'est le cas, peut-on dans les faits construire à un coût raisonnable un NN tel que

$$\|f - \hat{y}\| < \epsilon \ ?$$

Une question connexe, parmi de nombreuses autres que les chercheurs abordent [Fan *et al.*, 2020], est : dans quelle mesure le choix de la norme (topologie) affecte-t-il cette existence et/ou cette construction?

L'interprétation mathématique appropriée d'un NN est basée sur les théorèmes d'approximation universels (*universal approximation theorems* en anglais ou *UAT*). Les UAT sont grosso-modo répartis en deux catégories :

(1) nombre arbitraire de neurones dans le cas de réseaux de neurones *shallow* (SNN, largeur arbitraire). C'est le cas classique. Il remonte aux travaux pionniers des années 1940-1980;

(2) nombre arbitraire de couches avec des réseaux de neurones profonds (DNN, profondeur arbitraire). Dans la pratique, ces derniers se sont avérés plus efficaces [Mhaskar et Poggio, 2019].

Vus ainsi, les UAT sont des représentations de théorèmes d'existence classiques. Ils mettent en oeuvre la construction (c'est-à-dire trouvent les poids appropriés) de l'approximation de grandes classes de fonctions [Paluzo-Hidalgo *et al.*, 2020]. Cependant, ils ne disent pas comment explicitement trouver (avec une formule ou un algorithme) cette approximation (ces poids). Dans ce qui suit, nous donnons un bref résumé de certaines des étapes conceptuelles qui ont conduit à la compréhension mathématique actuelle des UAT.

Le théorème de représentation (ou superposition) de Kolmogorov-Arnold [Kolmogorov, 1957] énonce que toute fonction réelle continue multivariée $f : \mathbb{R}^d \to \mathbb{R}$ peut être représentée par une superposition de fonctions réelles continues à une variable :

$$f(\mathbf{x}) = \sum_{i=0}^{2d} \Phi_i \left(\sum_{j=1}^{d} \phi_{ij}(x_j) \right).$$

Ce théorème donne une réponse partielle au 13ème problème de Hilbert [Akashi, 2001, Khesin et Tabachnikov, 2014]. Une construction est proposée par Polar et Poluektov [2020].

Le théorème de Kolmogorov-Arnold donne une représentation exacte de la fonction $f(\mathbf{x})$. Cette représentation se révèle très complexe (nature fractale) et très lente en terme de calcul numérique, donc inutilisable pour un NN. La condition de précision a dû être relaxée par une condition d'approximation (L^p ou norme uniforme). La solution optimale est recherchée par des méthodes itératives (la descente de gradient, introduite par Jacques Hadamard en 1908, étant la plus répandue). Les travaux pionniers modernes sont dus à Cybenko [1989] pour les fonctions d'activation sigmoïde (largeur arbitraire), Funahashi [1989], Hornik [1991] (il a montré que UAT était potentiellement constructible avec une architecture multicouche du NN plutôt qu'un choix particulier de la fonction d'activation), Barron [1993] (bornes sur les erreurs de fonction lisses), etc ...

Leshno *et al.* [1993] et Pinkus [1999] ont montré que la propriété d'approximation universelle (voir Kratsios [2019] pour une définition) équivaut à une fonction d'activation non polynomiale.

Théorème 5.2.1. *(Cybenko [1989], Hornik [1991], Pinkus [1999]) Soient $d \in \mathbb{N}$, $\mathcal{K} \subset \mathbb{R}^d$ un compact, $f : \mathcal{K} \to \mathbb{R}$ une fonction continue, $\rho : \mathbb{R} \to \mathbb{R}$ une fonction non polynomiale*

continue, et soit $\epsilon > 0$. Alors il existe un réseau de neurones Φ à une couche cachée avec fonction d'activation ρ tel que

$$\|\Phi - f\|_\infty < \epsilon.$$

Après avoir joui d'une popularité considérable pendant un certain nombre d'années, les réseaux de neurones ont été quelque peu mis à l'écart par de nouvelles inventions au milieu des années 1990, telles que le *boosting* [Freund et Schapire, 1995, Freund *et al.*, 1996] et les SVM [Cortes et Vapnik, 1995]. Ils ont réapparu après 2010, avec l'essor des réseaux de neurones profonds (DNN), dont la discipline est aujourd'hui connue sous le nom *d'apprentissage profond* [Goodfellow *et al.*, 2016]. Cet enthousiasme renouvelé est le résultat d'améliorations massives des ressources informatiques, de certaines innovations et des tâches d'apprentissage de niche idéales telles que la classification d'images et de vidéos, et le traitement de la parole et du texte [Efron and Hastie, 2016].

Les DNN ont démontré leurs capacités dans l'approximation des fonctions [Liang et Srikant, 2016, Yarotsky, 2017, Fan *et al.*, 2020] et leurs bonnes performances empiriques expérimentales, mais nous ne savons généralement pas pourquoi. C'est un domaine de recherche très actif [Caterini et Chang, 2018]. Énonçons deux théorèmes récents spécifiques aux DNN et à l'approximation universelle.

Théorème 5.2.2. *(Lu* et al. *[2017]) Soient $f : \mathbb{R}^d \to \mathbb{R}$ Lebesgue-intégrable et $\epsilon > 0$. Alors il existe un réseau* ReLU *de largeur $\leq d + 4$ tel que*

$$\int_{\mathbb{R}^d} |\Phi(\mathbf{x}) - f(\mathbf{x})| \, d\mathbf{x} < \epsilon.$$

Pour énoncer le second théorème, définissons d'abord $\mathrm{NN}_{d,d',m}$, avec $d, d', m \in \mathbb{N}$, comme la classe de fonctions $\mathbb{R}^d \to \mathbb{R}^{d'}$ décrite par les réseaux de neurones avec d neurones dans la couche d'entrée, d' neurones dans la couche de sortie, et un nombre arbitraire de couches cachées, chacune avec m neurones et fonction d'activation ρ. Chaque neurone de la couche de sortie a pour fonction d'activation l'identité.

Théorème 5.2.3. *(Kidger et Lyons [2020]) Soit $\rho : \mathbb{R} \to \mathbb{R}$ une fonction continue non affine qui est continûment différentiable en au moins un point, avec une dérivée non nulle en ce point. Soit $\mathcal{K} \subset \mathbb{R}^d$ un compact. Alors $NN_{d,d',d+d'+2}$ est dense dans $\mathcal{C}\left(\mathcal{K}, \mathbb{R}^{d'}\right)$ par rapport à la norme uniforme.*

Remarque 5.2.4. *Avant les années 1980 [McCulloch et Pitts, 1943, Fukushima, 1975], les architectures NN étaient peu profondes. Une compréhension des SNN donne une bonne appréciation des DNN. Il est toujours possible de créer un SNN pour émuler un DNN. Cependant le nombre de neurones croît de façon exponentielle pour obtenir une performance comparable (en terme d'erreur) [Telgarsky, 2016, Eldan et Shamir, 2016].*

5.3. Ajuster un réseau de neurones

Comme nous l'avons décrit ci-dessus, un réseau de neurones est une fonction hiérarchique complexe NN($\mathbf{x}$,$\boldsymbol{\theta}$) du vecteur de covariables $\mathbf{x}$ et de la collection de poids $\boldsymbol{\theta}$. Pour les choix typiques de fonctions d'activation, la fonction NN($\cdot$) sera différentiable en $\boldsymbol{\theta}$. Les réseaux de neurones peuvent être considérés comme des méthodes de régression élaborées visant uniquement la prédiction, et non l'estimation ou l'explication dans le langage des statistiques. En statistique, on n'étudie pas nécessairement les approximations universelles. Par contre, les théorèmes énoncés ci-dessus démontrent la flexibilité des réseaux de neurones, permettant ainsi d'atteindre un niveau de performance de prédiction très élevé.

Dans la pratique, étant donné un échantillon (ou ensemble d'entraînement dans le langage de l'apprentissage statistique) $\{(\mathbf{x}_i,y_i)\}_{i=1}^n$ et une fonction de perte $\ell : \mathbb{R}^2 \to \mathbb{R}$, on pourrait chercher à résoudre selon des critères familiers

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ell\Big(y_i, \mathrm{NN}(\mathbf{x}_i, \boldsymbol{\theta})\Big) + \lambda \mathcal{R}(\boldsymbol{\theta}) \right\}, \tag{5.3.1}$$

où $\mathcal{R}(\boldsymbol{\theta})$ est une fonction de régularisation des éléments de $\boldsymbol{\theta}$, et $\lambda \geq 0$ est une constante qui contrôle le montant de régularisation. À titre d'exemple, une fonction de régularisation populaire est la fonction quadratique, comme dans la régression Ridge [Hoerl et Kennard, 1970]. Elle tire les poids vers zéro et les biais ne sont généralement pas pénalisés. Les fonctions de régularisation de type lasso [Tibshirani, 1996] sont également populaires, tout comme leurs mélanges (*elastic-net* [Zou et Hastie, 2005] par exemple).

Quelques astuces supplémentaires visant la régularisation ont été nouvellement introduites. En particulier, la méthode connue sous l'appellation de *dropout* en anglais est une forme de régularisation [Srivastava *et al.*, 2014]. Cette méthode a été inspirée par la sélection aléatoire de variables à considérer pour la construction de chaque arbre dans l'ajustement d'une forêt aléatoire. Considérons par exemple le calcul (5.1.1) de l'entrée $e_l^{(2)}$ de la couche

C_2 pour une seule observation pendant la propagation des valeurs vers l'avant. L'idée est de fixer au hasard chacun des d nœuds $s_j^{(1)}$ à zéro avec une probabilité β, et de gonfler les autres d'un facteur $1/(1-\beta)$. Par conséquent, pour cette observation, les nœuds conservés doivent remplacer ceux omis. Cela peut être démontré comme une forme de régularisation Ridge qui, lorsque mise en œuvre correctement, améliore les performances de prédiction. Ses liens avec la régression linéaire Ridge sont mieux décrits dans Wager *et al.* [2013].

Pour une variable dépendante continue, la fonction de perte $\ell(\cdot)$ est généralement définie par l'erreur quadratique moyenne, auquel cas le réseau de neurones équivaudrait à une régression pénalisée, quoique fortement paramétrée. Les fonctions de perte sont généralement convexes par rapport à $\mathrm{NN}(\cdot)$, mais pas par rapport aux éléments de $\boldsymbol{\theta}$. La résolution de (5.3.1) est difficile et, dans les meilleurs des cas, on cherche un bon minimum local. Il faut donc procéder avec des méthodes itératives. La plupart sont basées sur une forme de descente de gradient stochastique [Robbins and Monro, 1951, Bottou, 1991].

D'un point de vue mise en œuvre, la méthode la plus élégante pour l'obtention de la dérivée de premier ordre (de $\ell(\cdot)$ par rapport aux éléments de $\boldsymbol{\theta}$) est la rétropropagation de l'erreur de prédiction (ou *backpropagation* en anglais) [Rumelhart *et al.*, 1986]. L'idée de la rétropropagation est de faire circuler l'information sur la dérivée de la fonction de perte à partir de la couche de sortie, où l'erreur de prédiction est connue, jusqu'à la couche d'entrée. Pour y arriver, il suffit d'exprimer la dérivée de la fonction de perte d'un nœud en fonction de l'information donnée par les couches suivantes. En se servant ensuite de la formule de dérivée en chaîne, il est assez simple de trouver, de façon récursive, les dérivées à chaque nœud. Prenons par exemple la fonction de perte suivante :

$$\ell(y,\hat{y}) = \frac{1}{2}(\hat{y}-y)^2, \tag{5.3.2}$$

où $\hat{y} \in \mathbb{R}$ est la sortie (ou la prédiction) d'un réseau de neurones à $R \geq 2$ couches ($R = 3$ dans la Figure 5.1) pour une observation (ou entrée) arbitraire $\mathbf{x} \in \mathbb{R}^{\mathrm{d}}$. Rappelons que chaque neurone de la couche C_r, $1 \leq r < R$, possède une connexion avec tous les neurones de la couche C_{r+1}. Tel que défini précédemment, chacune de ces connexions est pondérée par $\theta_{lj}^{(r)}$, le poids liant la sortie $s_j^{(r)}$ du $j$ième neurone de la couche C_r à l'entrée $e_l^{(r+1)}$ du $l$ième neurone de la couche C_{r+1}, pour $1 \leq r < R-1$. Notons le poids connectant le $j$ième neurone de la couche C_{R-1} au neurone de la couche de sortie C_R par $\theta_j^{(R-1)}$. La dérivée de

la fonction de perte (5.3.2) par rapport à la prédiction $\hat{y}$ est donnée par

$$\frac{\partial \ell}{\partial \hat{y}} = \hat{y} - y.$$

Supposons que la fonction d'activation de la couche de sortie C_R est la fonction identité. Dans ce cas, la dérivée de la fonction de perte à l'entrée de la couche de sortie est la même que celle à la sortie, c'est-à-dire,

$$\frac{\partial \ell}{\partial e^{(R)}} = \frac{\partial \ell}{\partial s^{(R)}} = \frac{\partial \ell}{\partial \hat{y}}.$$

Maintenant, en remarquant que la fonction de perte est affectée par la sortie d'un neurone de la couche C_r à travers l'ensemble des neurones de la couche C_{r+1}, par la formule de dérivée en chaîne, nous avons :

$$\begin{aligned} \frac{\partial \ell}{\partial s_j^{(r)}} &= \sum_l \theta_{lj}^{(r)} \frac{\partial \ell}{\partial e_l^{(r+1)}}, && \text{si } r < R-1, \\ \frac{\partial \ell}{\partial s_j^{(r)}} &= \theta_j^{(r)} \frac{\partial \ell}{\partial e^{(r+1)}}, && \text{si } r = R-1. \end{aligned}$$

Aussi, la fonction de perte est affectée par l'entrée d'un neurone de la couche C_r seulement au travers de la sortie de ce même neurone. Ainsi, nous avons :

$$\frac{\partial \ell}{\partial e_j^{(r)}} = \frac{\partial \ell}{\partial s_j^{(r)}} \frac{\partial s_j^{(r)}}{\partial e_j^{(r)}}.$$

Prenons maintenant la fonction d'activation sigmoïde pour chaque couche cachée C_r, $1 < r < R$. Puisque la dérivée de la fonction sigmoïde est donnée par $\partial h(z)/\partial z = h(z)\{1-h(z)\}$, la dérivée de la fonction de perte par rapport a l'entrée d'un neurone est

$$\frac{\partial \ell}{\partial e_j^{(r)}} = \frac{\partial \ell}{\partial s_j^{(r)}} s_j^{(r)}\{1 - s_j^{(r)}\}.$$

Cette dernière égalité permet de rétropropager la dérivée de l'erreur. En appliquant la règle de dérivée en chaîne, la dérivée de la fonction de perte par rapport à chacun des poids du réseau de neurones est donnée par :

$$\begin{aligned} \frac{\partial \ell}{\partial \theta_{lj}^{(r)}} &= \frac{\partial \ell}{\partial e_l^{(r+1)}} \frac{\partial e_l^{(r+1)}}{\partial \theta_{lj}^{(r)}} = \frac{\partial \ell}{\partial e_l^{(r+1)}} s_j^{(r)}, && \text{si } r < R-1, \\ \frac{\partial \ell}{\partial \theta_j^{(r)}} &= \frac{\partial \ell}{\partial e^{(r+1)}} \frac{\partial e^{(r+1)}}{\partial \theta_j^{(r)}} = \frac{\partial \ell}{\partial e^{(r+1)}} s_j^{(r)}, && \text{si } r = R-1. \end{aligned}$$

Pour une description plus approfondie de la rétropropagation, nous renvoyons le lecteur vers le manuel de [Bishop, 1995, Chapitre 4]. Plus de détails sur l'algorithme de descente du gradient stochastique sont donnés au chapitre 6 de la thèse.

Aujourd'hui, les NN font partie des outils/méthodes populaires qu'un statisticien utilise pour résoudre un problème de prédiction. Il y a une littérature abondante sur les réseaux de neurones, avec des centaines de livres et des milliers d'articles. Avec la popularité récente de l'apprentissage profond, le nombre de publications connaît un essor considérable. Deux références statistiques pionnières sur les réseaux de neurones sont Bishop [1995] et Ripley [1996]. Le prochain chapitre de la thèse décrit notre utilisation des réseaux de neurones pour l'inférence causale et la prédiction de l'*uplift*.

Bibliographie

Shigeo Akashi : Application of ϵ-entropy theory to Kolmogorov–Arnold representation theorem. *Reports on Mathematical Physics*, 48(1-2):19–26, 2001.

Andrew R Barron : Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945, 1993.

Christopher M Bishop : *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

Léon Bottou : Stochastic gradient learning in neural networks. *Proceedings of Neuro-Nımes*, 91(8):12, 1991.

Ovidiu Calin : *Deep Learning Architectures*. Springer, 2020.

Anthony L Caterini et Dong Eui Chang : *Deep Neural Networks in a Mathematical Framework*. Springer, 2018.

Corinna Cortes et Vladimir Vapnik : Support-vector networks. *Machine Learning*, 20 (3):273–297, 1995.

George Cybenko : Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.

Bradley Efron et Trevor Hastie : *Computer age statistical inference*, volume 5. Cambridge University Press, 2016.

Ronen Eldan et Ohad Shamir : The power of depth for feedforward neural networks. *In Conference on Learning Theory*, pages 907–940, 2016.

Fenglei Fan, Jinjun Xiong et Ge Wang : Universal approximation with quadratic deep networks. *Neural Networks*, 124:383–392, 2020.

Yoav Freund et Robert E Schapire : A desicion-theoretic generalization of on-line learning and an application to boosting. *In European conference on Computational Learning Theory*, pages 23–37. Springer, 1995.

Yoav Freund, Robert E Schapire *et al.* : Experiments with a new boosting algorithm. *In ICML*, volume 96, pages 148–156, 1996.

Kunihiko Fukushima : Cognitron: A self-organizing multilayered neural network. *Biological Cybernetics*, 20(3-4):121–136, 1975.

Ken-Ichi Funahashi : On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2(3):183–192, 1989.

Ian Goodfellow, Yoshua Bengio et Aaron Courville : *Deep learning*. MIT press, 2016.

Arthur E Hoerl et Robert W Kennard : Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Kurt Hornik : Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.

Boris A Khesin et Serge L Tabachnikov : *ARNOLD: Swimming Against the Tide: Swimming Against the Tide*, volume 86. American Mathematical Society, 2014.

Patrick Kidger et Terry Lyons : Universal approximation with deep narrow networks. *In Conference on Learning Theory*, pages 2306–2327, 2020.

Andrei Nikolaevich Kolmogorov : On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. *In Doklady Akademii Nauk*, volume 114, pages 953–956. Russian Academy of Sciences, 1957.

Anastasis Kratsios : The universal approximation property: Characterizations, existence, and a canonical topology for deep-learning. *arXiv preprint arXiv:1910.03344*, 2019.

Moshe Leshno, Vladimir Ya Lin, Allan Pinkus et Shimon Schocken : Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867, 1993.

Shiyu Liang et Rayadurgam Srikant : Why deep neural networks for function approximation? *arXiv preprint arXiv:1610.04161*, 2016.

Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu et Liwei Wang : The expressive power of neural networks: A view from the width. *In Advances in Neural Information Processing Systems*, pages 6231–6239, 2017.

Warren S McCulloch et Walter Pitts : A logical calculus of the ideas immanent in nervous activity. *The bulletin of Mathematical Biophysics*, 5(4):115–133, 1943.

HN Mhaskar et Nahmwoo Hahm : System identification using neural networks. *In Neural Networks for Signal Processing VI. Proceedings of the 1996 IEEE Signal Processing Society Workshop*, pages 82–88. IEEE, 1996.

HN Mhaskar et T Poggio : Function approximation by deep networks. *arXiv preprint arXiv:1905.12882*, 2019.

John Ashworth Nelder et Robert WM Wedderburn : Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384, 1972.

Eduardo Paluzo-Hidalgo, Rocio Gonzalez-Diaz et Miguel A Gutiérrez-Naranjo : Two-hidden-layer feed-forward networks are universal approximators: A constructive approach. *Neural Networks*, 2020.

Allan Pinkus : Approximation theory of the MLP model in neural networks. *Acta Numerica*, 8(1):143–195, 1999.

Andrew Polar et Michael Poluektov : Urysohn operators as adaptive filters. *arXiv preprint arXiv:2001.04652*, 2020.

Brian D Ripley : *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

Herbert Robbins et Sutton Monro : A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

David E Rumelhart, Geoffrey E Hinton et Ronald J Williams : Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

Jürgen Schmidhuber : Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever et Ruslan Salakhutdinov : Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Matus Telgarsky : Benefits of depth in neural networks. *arXiv preprint arXiv:1602.04485*, 2016.

R. Tibshirani : Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

Stefan Wager, Sida Wang et Percy Liang : Dropout training as adaptive regularization. *arXiv preprint arXiv:1307.1493*, 2013.

Dmitry Yarotsky : Error bounds for approximations with deep relu networks. *Neural Networks*, 94:103–114, 2017.

Hui Zou et Trevor Hastie : Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, 67(2):301–320, 2005.

Chapitre 6

Un modèle neuronal pour la prédiction de l'*uplift*

Troisième article.

A Twin Neural Model for Uplift

par

Mouloud Belbahri[1], Alejandro Murua[1], Olivier Gandouet[2], and Vahid Partovi Nia[3]

([1]) Université de Montréal
([2]) TD Assurance
([3]) École Polytechnique de Montréal

Mes contributions et le rôle des coauteurs. Dans cet article, j'ai

- proposé la méthodologie basée sur les réseaux de neurones jumeaux;

- développé la fonction de perte et l'algorithme de descente de gradient;
- rédigé le code Python permettant l'implémentation du modèle neuronal;
- rédigé le code R permettant l'implémentation des modèles *uplift* existants;
- rédigé l'article;
- élaboré et mis en oeuvre les simulations;
- analysé les données réelles.

En tant que directeur principal, Alejandro Murua a participé au développement méthodologique et a effectué une relecture de l'article apportant une contribution significative au niveau du contenu intellectuel. En tant que co-directeur, Vahid Partovi Nia a participé à la rédaction de l'article et au développement méthodologique. En tant que directeur industriel, Olivier Gandouet a contribué dans le développement méthodologique et a supervisé l'implémentation Python du réseau de neurones.

RÉSUMÉ. L'*uplift* est un cas particulier de modélisation de l'effet du traitement conditionnel. Ces modèles traitent de l'inférence de cause à effet pour un facteur spécifique, comme une intervention de marketing ou un traitement médical. En pratique, ces modèles sont construits sur des données individuelles issues d'essais cliniques randomisés où l'objectif est de répartir les participants en groupes hétérogènes en fonction de l'*uplift*. La plupart des approches existantes sont des adaptations de forêts aléatoires pour le cas de l'*uplift*. Plusieurs critères de segmentation ont été proposés dans la littérature, tous reposant sur la maximisation de l'hétérogénéité. Cependant, dans la pratique, ces approches sont sujettes au surajustement. Dans ce travail, nous apportons une nouvelle vision pour la modélisation uplift. Nous proposons une nouvelle fonction de perte définie en tirant parti d'un lien avec l'interprétation bayésienne du risque relatif. Notre solution est développée pour une architecture de réseau de neurones jumeaux spécifique permettant d'optimiser conjointement les probabilités marginales de succès pour les individus traités et non-traités. Nous montrons que ce modèle est une généralisation du modèle d'interaction logistique de l'*uplift*. Nous modifions l'algorithme de descente de gradient stochastique pour permettre des solutions parcimonieuses structurées. Cela aide dans une large mesure à ajuster nos modèles *uplift*. Nous montrons empiriquement que notre méthode est compétitive avec l'état de l'art, sur des données de simulations et sur des données réelles provenant d'expériences randomisées à grande échelle.

Mots clés : inférence causale, effet de traitement hétérogène, fonction de perte, descente de gradient, régularisation

Abstract. Uplift is a particular case of conditional treatment effect modeling. Such models deal with cause-and-effect inference for a specific factor, such as a marketing intervention or a medical treatment. In practice, these models are built on individual data from randomized clinical trials where the goal is to partition the participants into heterogeneous groups depending on the uplift. Most existing approaches are adaptations of random forests for the uplift case. Several split criteria have been proposed in the literature, all relying on maximizing heterogeneity. However, in practice, these approaches are prone to overfitting. In this work, we bring a new vision to uplift modeling. We propose a new loss function defined by leveraging a connection with the Bayesian interpretation of the relative risk. Our solution is developed for a specific twin neural network architecture allowing to jointly optimize the marginal probabilities of success for treated and control individuals. We show that this model is a generalization of the uplift logistic interaction model. We modify the stochastic gradient descent algorithm to allow for structured sparse solutions. This helps training our uplift models to a great extent. We show our proposed method is competitive with the state-of-the-art in simulation setting and on real data from large scale randomized experiments.

Keywords: causal inference, heterogeneous treatment effects, loss function, gradient descent, regularization

6.1. Introduction

Causal inference draws conclusion about cause and effect relationships through empirical observations. The most widely used statistical framework for causal inference is the *counterfactual* framework. The counterfactual paradigm, also called the *potential outcome* paradigm, was first developed by Neyman [1923] to study randomized experiments. A generalization allowing the study of causal links with observational data was subsequently carried out by Rubin [1974]. Although the process of causal inference is usually complex, it is of extreme importance. In the field of health science, causal inference techniques make it possible to assess the effect of a potential intervention on the health of individuals, in particular in the case of randomized clinical trials. In the field of marketing, such models deal with customers behavioral change caused by a specific treatment, such as a marketing intervention, a courtesy call, targeted advertisement. The counterfactual paradigm assumes that for each individual, there are two potential outcomes, or counterfactuals: i) the potential outcome corresponding to the exposure (treatment), and ii) the potential outcome corresponding to

the absence of exposure (control). However, one cannot simultaneously observe treatment and control for a single individual [Holland, 1986].

In causal inference, the most common population-level estimand is the *average treatment effect*. In the absence of *confounders* (i.e., a variable that influences both the treatment and response variables), this is simply the difference of the two averages between the treatment and control groups. Another estimand include the *individual treatment effect,* also called *conditional average treatment effect* in the context of causal inference, or *uplift* in the context of marketing research. There are different methods to estimate individual treatment effects. The process of estimating these effects has different synonyms like heterogeneous effect modeling, individual treatment effect modeling, or uplift modeling.

Uplift modeling is an important research area in marketing field [Radcliffe and Surry, 1999, Hansotia and Rukstales, 2001, Lo, 2002, Radcliffe, 2007]. This modeling framework has also been proposed to allow for prediction of an individual patient's response to a medical treatment [Jaskowski and Jaroszewicz, 2012, Lamont et al., 2018]. Typically, uplift models are developed for randomized experiments, with both the treatment and outcome as binary random variables, where prediction power is the most important issue.

Most common research in the uplift field is based on classification and regression trees (CART) [Breiman et al., 1984]. Unlike other modeling techniques, fitting a decision tree allows each iteration to uniquely partition the sample. This means that each segmentation can be immediately checked against the impact of the treatment. Since the goal of uplift modeling is to find a partition into subgroups of the population, it seems natural to use a decision tree as the method of choice. Then, the idea is to predict the individual uplift by the uplift observed in a terminal node, or by the average when several trees are used, e.g., random forests [Breiman, 2001]. However, this is not enough. Indeed, with CART classic division criteria, the search for predictors and their split points is optimized according to the response variable and not the uplift. As practice often shows, response probability and uplift have quite different factors in terms of predictors [Radcliffe and Surry, 2011]. In extreme cases, all uplifts in the leaves might be the same (but not the positive response probabilities). Therefore, various modifications of the split criteria have been proposed in the literature [Radcliffe and Surry, 1999, Hansotia and Rukstales, 2001, Rzepakowski and Jaroszewicz, 2010].

In the case of parametric modeling approaches, the simplest model that can be used to estimate the uplift is logistic regression, because the response variable is binary [Lo, 2002]. Thus, the underlying optimization problem becomes the maximization of the Binomial likelihood. In this case, the approach does not provide a direct uplift search, but rather the probabilities of positive responses for treated and non-treated are modeled separately. Then, their difference is used as an estimate of the uplift. However, the solution is not optimized for searching for heterogeneous groups depending on the uplift. Hence, maximizing the likelihood is not necessarily the right way to estimate the uplift. Therefore, changes are required in the optimization problem in order to appropriately estimate the uplift.

In this work, we introduce a new uplift loss function defined by leveraging a connection with the Bayesian interpretation of the relative risk, another treatment effect measure specific to the binary case. Defining an appropriate loss function for uplift also allows to use simple models or any off the shelf method for the related optimization problem, including complex models such as neural networks. When prediction becomes more important than estimation, neural networks become more attractive than classical statistical models. There are several reasons why neural networks are suitable tools for uplift: i) they are flexible models and easy to train with current GPU hardware; ii) with a single hidden layer modeling covariates interactions is straightforward [Tsang et al., 2018]; iii) they are guaranteed to approximate a large class of functions [Cybenko, 1989, Hornik, 1991, Pinkus, 1999]; iv) neural networks perform very well on predictive tasks which is the main objective of uplift; v) a simple network architecture ensures model interpretability for further studies.

Our methodology is developed in the context of a specific neural network architecture in which the proposed loss function can be easily optimized. Our solution uses a representation that resembles the twin networks [Bromley et al., 1994] known in the context of deep learning. This representation helps the fitting process to a great extent. We show our model generalizes Lo's uplift logistic interaction model to a 1-hidden layer neural network model. In the logistic regression context, one can use *lasso* [Tibshirani, 1996] to produce a sparse model. As more hidden nodes are added to the neural network, lasso becomes choosing the right number of parameters to include in the network (i.e., pruning). However, the fitting mechanism of statistically-driven methods such as the lasso needs to be adapted for stochastic gradient descent to provide a neural network pruning technique. We propose to use a proximal

version of gradient descent and to introduce a scaling factor for structured pruning, which is common in neural networks [Ramakrishnan et al., 2020]. This allows to select automatically the number of neurons for each hidden layer. Compared to other existing methods, we thus offer a unified principled approach to obtain a sparse solution that provides well-optimized uplift estimates.

The contributions of this paper are the following: i) defining a new loss function derived from an intuitive interpretation of treatment effects estimation; ii) generalizing the uplift logistic interaction model to a 1-hidden layer ReLU neural-network; iii) introducing a new twin neural architecture to predict conditional average treatment effects; iv) guiding model selection (or architecture search) with sparse group lasso regularization; v) establishing empirically the validity of the estimation procedure on both synthetic and real-world datasets.

6.2. Related work

Uplift is a particular case of conditional treatment effect modeling which falls within the potential outcomes framework, also known as the Neyman-Rubin causal model [Rubin, 1974, Rosenbaum et Rubin, 1983, Holland, 1986].

Let T be the binary treatment indicator, and $\boldsymbol{X} = (X_1, \ldots, X_p)$ be the p-dimensional predictors vector. The binary variable T indicates if a unit is exposed to treatment ($T = 1$) or control ($T = 0$). Let $Y(0)$ and $Y(1)$ be the binary potential outcomes under control and treatment respectively. Assume a distribution $(Y(0), Y(1), \boldsymbol{X}, T) \sim \mathcal{P}$ from which n iid samples are given as the training observations $\{(y_i, \mathbf{x}_i, t_i)\}_{i=1}^n$, where $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$ are realisations of the predictors and t_i the realisation of the treatment for observation i. Although each observation i is associated with two potential outcomes, only one of them can be realized as the observed outcome y_i. By Assumption 6.2.1, under the counterfactual consistency, each observation is missing only one potential outcome: the one that corresponds to the absent treatment either $t = 0$ or $t = 1$.

Assumption 6.2.1. *(Consistency) Observed outcome Y is represented using the potential outcomes and treatment assignment indicator as follows:*

$$Y = TY(1) + (1 - T)Y(0).$$

In general, we will assume the following representation of $\mathcal{P}$:

$$\boldsymbol{X} \sim \Lambda$$

$$T \sim \text{Bernoulli}(e(\mathbf{x}))$$

$$Y(t) \sim \text{Bernoulli}(m_{1t}(\mathbf{x}))$$

where Λ is the marginal distribution of $\boldsymbol{X}$ and $e(\cdot)$ is the propensity score (see Definition 6.2.2). The probabilities of positive responses for the potential outcomes under control and treatment are given by the functions $m_{1t}(\cdot) : \mathbb{R}^{\text{p}} \to (0,1)$ for $t = 0$ and $t = 1$ respectively.

Definition 6.2.2. *(Propensity score) For any $\boldsymbol{X} = \mathbf{x}$, the propensity score is defined as:*

$$e(\mathbf{x}) = \Pr(T_i = 1 \mid \boldsymbol{X}_i = \mathbf{x}).$$

Given the notation above, the conditional average treatment effect (CATE) is defined as follows:

$$\text{CATE}(\mathbf{x}) = \mathbb{E}[Y_i(1) - Y_i(0) | \boldsymbol{X}_i = \mathbf{x}] \tag{6.2.1}$$

In order for the CATE to be identifiable, we must make some additional assumptions, standard in the world of causal inference. The propensity score parameter is widely used to estimate treatment effects from observational data [Rosenbaum et Rubin, 1983]. Assumption 6.2.3 states that each individual has non-zero probabilities of being exposed and being unexposed. This is necessary to make the mean quantities meaningful.

Assumption 6.2.3. *(Overlap) For any $\boldsymbol{X} = \mathbf{x}$, the true propensity score is strictly between 0 and 1, i.e., for $\epsilon > 0$,*

$$\epsilon < e(\mathbf{x}) < 1 - \epsilon.$$

For the rest of the paper, we will consider the case of randomized experiments, with $e(\mathbf{x}) = 1/2$, which is common in the uplift literature and is the case of our data. When the data comes from observational studies, combined with the previous assumptions, Assumption 6.2.4 allows the identification of the CATE.

Assumption 6.2.4. *(Unconfoundedness) Potential outcomes $Y(0)$,$Y(1)$ are independent ($\perp\!\!\!\perp$) of the treatment assignment indicator T conditioned on all pre-treatment characteristics $\boldsymbol{X}$, i.e.,*

$$Y(0), Y(1) \perp\!\!\!\perp T|\boldsymbol{X}.$$

For randomized experiments, the random variable T is independent of any pre-treatment characteristics, that is, $Y(0), Y(1), \boldsymbol{X} \perp\!\!\!\perp T$, which is a stronger assumption than the unconfoundedness assumption.

In the literature, the terms ITE, CATE and uplift often refer to the same quantity, namely the CATE. Indeed, the uplift is a special case of the CATE when the dependent variable Y is binary $0-1$. Thus, the uplift is defined as the conditional average treatment effect in different sub-populations according to the possible values of the covariates, namely:

$$u(\mathbf{x}) = \Pr(Y_i = 1 \mid \boldsymbol{X}_i = \mathbf{x}, T_i = 1) - \Pr(Y_i = 1 \mid \boldsymbol{X}_i = \mathbf{x}, T_i = 0). \tag{6.2.2}$$

To simplify the notation, we prefer to denote by $m_{yt}(\mathbf{x})$ the corresponding conditional probability $\Pr(Y_i = y \mid \boldsymbol{X}_i = \mathbf{x}, T_i = t)$. Therefore, the uplift is the difference between the two conditional means $m_{11}(\mathbf{x})$ and $m_{10}(\mathbf{x})$. The intuitive approach to model uplift is to build two independent models [Hansotia and Rukstales, 2001]. This consists of fitting two separate conditional probability models: one model for the treated individuals, and another separate model for the untreated individuals. Then, uplift is the difference between these two conditional probability models. These models are called T-learners (T for "two models") in the literature [Künzel et al., 2019]. The asset of T-learners is their simplicity, but they do not perform well in practice, because each model focuses on predicting only one class, so the information about the other treatment is never provided to the learning algorithm [Radcliffe and Surry, 2011]. In addition, differences between the covariates distributions in the two treatment groups can lead to bias in treatment effect estimation. There have been efforts in correcting such drawbacks through a combined classification model known as S-learner, for "single-model" [Künzel et al., 2019]. The idea behind the S-learner is to use the treatment variable as a feature and to add explicit interaction terms between each covariate and the treatment indicator to fit a model, e.g., a logistic regression [Lo, 2002]. The parameters of the interaction terms measure the additional effect of each covariate due to treatment.

Another related method is known as the X-learner [Künzel et al., 2019]. The X-learner estimates the uplift in three stages. First, $m_{11}(\mathbf{x})$ and $m_{10}(\mathbf{x})$ are modeled separately as in the case of T-learners. Then, the fitted values $\hat{m}_{11}(\mathbf{x})$ and $\hat{m}_{10}(\mathbf{x})$ are used to impute the "missing" potential outcomes for each observation, and to create new imputed response variables, $D(1)$ and $D(0)$. These imputed variables are used to fit new models that capture the uplift directly, $\hat{u}^{(1)}(\mathbf{x})$ and $\hat{u}^{(0)}(\mathbf{x})$. The final prediction is given by a weighted average using the propensity score, $e(\mathbf{x})\{\hat{u}^{(1)}(\mathbf{x}) - \hat{u}^{(0)}(\mathbf{x})\} + \hat{u}^{(0)}(\mathbf{x})$. More recently, Nie and Wager [2020] introduced the R-learner. The method uses Robinson's transformation [Robinson, 1988] and assumes "oracle" estimation of the propensity score $e(\mathbf{x})$ and the marginal effect function $m(\mathbf{x}) = \mathbb{E}[Y|\boldsymbol{X} = \mathbf{x}]$ in order to reduce the problem of modeling the uplift to a residual-on-residual ordinary least squares regression. Interestingly, even though the X-learner and R-learner frameworks are valid for continuous and binary Y, both methods aim to find an uplift estimator that minimizes the mean squared error (MSE). The connection with the maximum likelihood framework does not hold in the binary case. Also, in practice, the R-learner is fitted in two stages using cross-fitting to emulate the oracle, thus requiring more observations than S-learners. The methodology introduced in Belbahri et al. [2021] attacks parameter estimation and addresses the loss-metric mismatch in uplift regression. It can be seen as an S-learner which estimates the model's parameters in two stages. The method first applies the pathwise coordinate descent algorithm [Friedman et al., 2007] to compute a sequence of critical regularization values and corresponding (sparse) model parameters. Then, it uses Latin hypercube sampling [McKay et al., 2000] to explore the parameter space in order to find the optimal model.

Several proposed non-parametric methods take advantage of grouped observations in order to model the uplift directly. Some k-nearest neighbours [Cover and Hart, 1967] based methods are adopted for uplift estimation [Crump et al., 2008, Alemi et al., 2009, Su et al., 2012]. The main idea is to estimate the uplift for an observation based on its neighbourhood containing at least one treated and one control observations. However, these methods quickly become computationally expensive for large datasets. State-of-the-art proposed methods view random forests as an adaptive neighborhood metric, and estimate the treatment effect

at the leaf node [Su et al., 2009, Chipman et al., 2010, Wager and Athey, 2018]. Therefore, most active research in uplift modeling is in the direction of classification and regression trees [Breiman et al., 1984] where the majority are modified random forests [Breiman, 2001]. In Radcliffe and Surry [1999, 2011], Hansotia and Rukstales [2001], Rzepakowski and Jaroszewicz [2010], modified split criteria that suited the uplift purpose were studied. The criteria used for choosing each split during the growth of the uplift trees is based on maximization of the difference in uplifts between the two child nodes. Within each leaf, uplift is estimated with the difference between the two conditional means. A good estimate of each mean may lead to a poor estimate of the difference [Radcliffe and Surry, 2011]. However, the existing tree-based uplift optimization problems do not take this common misconception into account. Instead, the focus is on maximizing the heterogeneity in treatment effects. Without careful regularization (e.g., honest estimation [Athey et al., 2019]), splits are likely to be placed next to extreme values (of covariates) because outliers of any group (treatment or control) can influence the choice of a split point. In addition, successive splits tend to group together similar extreme values, introducing more variance in the prediction of uplift [Zhao et al., 2017]. Alternatively some models use the transformed outcome [Athey and Imbens, 2015], an unbiased estimator of the uplift. However, this estimate suffers from higher variance than the difference in conditional means estimator [Powers et al., 2018]. In addition, for both estimators, if random noise is larger than the treatment effect, the model will more likely predict random noise instead of uplift. As a result, based on several experiments on real data, and although the literature suggests that tree-based methods are state-of-the-art for uplift [Sołtys et al., 2015], the published models overfit the training data and predicting uplift still lacks satisfactory solutions.

6.3. An uplift loss function

We formally define the uplift loss function that will be used to fit our models. Our goal is to regularize the conditional means in order to get a better prediction of the quantity of interest, the uplift. Inspired by the work of Athey et al. [2019], Künzel et al. [2019], Nie and Wager [2020], Belbahri et al. [2021] which adapt the optimization problem to the uplift context, we propose to define a composite loss function, which can be separated into two

pieces:

$$\ell(\cdot) = \ell_1(\cdot) + \ell_2(\cdot)$$

and to optimize both simultaneously. Since we generalize the uplift logistic regression, we model the probability of positive response $m_{1t}(\mathbf{x})$. Naturally, the first term can be defined as the negative log-likelihood or the binary cross entropy (BCE) loss, with $\mathbf{y}$ as the response, and $m_{1t}(\mathbf{x})$ as the prediction, that is,

$$\ell_1(\mathbf{y}, m_{1t}(\mathbf{x}) \mid \mathbf{x}) = -\frac{1}{n}\sum_{i=1}^{n}\Big(y_i \log\{m_{1t_i}(\mathbf{x}_i)\} + (1-y_i)\log\{1-m_{1t_i}(\mathbf{x}_i)\}\Big).$$

We define the second term based on a Bayesian interpretation of another measure of treatment effect, the relative risk. First, let us define the relative risk (or risk ratio) as a function of the conditional means.

Definition 6.3.1. *(Relative risk) For any $\boldsymbol{X} = \mathbf{x}$ and $m_{10}(\mathbf{x}) > 0$, the relative risk is defined as follows:*

$$\mathrm{RR}(\mathbf{x}) = \frac{\Pr(Y=1 \mid \boldsymbol{X}=\mathbf{x}, T=1)}{\Pr(Y=1 \mid \boldsymbol{X}=\mathbf{x}, T=0)} = \frac{m_{11}(\mathbf{x})}{m_{10}(\mathbf{x})}.$$

Relative risk is commonly used to present the results of randomized controlled trials. In the medical context, the uplift is known as the absolute risk (or risk difference). In practice, presentation of both absolute and relative measures is recommended [Moher et al., 2012]. If the relative risk is presented without the absolute measure, in cases where the base rate of the outcome $m_{10}(\mathbf{x})$ is low, large or small values of relative risk may not translate to significant effects, and the importance of the effects to the public health can be overestimated. Equivalently, in cases where the base rate of the outcome $m_{10}(\mathbf{x})$ is high, values of the relative risk close to 1 may still result in a significant effect, and their effects can be underestimated. Interestingly, the relative risk can be reformulated as:

$$\mathrm{RR}(\mathbf{x}) = \frac{\Pr(T=1 \mid Y=1, \boldsymbol{X}=\mathbf{x})}{\Pr(T=0 \mid Y=1, \boldsymbol{X}=\mathbf{x})}\left(\frac{1-e(\mathbf{x})}{e(\mathbf{x})}\right),$$

where the propensity score $e(\mathbf{x})$ is given in Definition 6.2.2. For randomized experiments, the propensity score ratio $\{1-e(\mathbf{x})\}/e(\mathbf{x})$ is a constant and, written in that form, the relative risk can be interpreted in Bayesian terms as the normalized posterior propensity score ratio (i.e., after observing the outcome). In the particular case where $e(\mathbf{x}) = 1/2$, it is easy to

show that $\Pr(T = 1 \mid Y = 1, \boldsymbol{X} = \mathbf{x}) = \mathrm{RR}(\mathbf{x})/\{1 + \mathrm{RR}(\mathbf{x})\}$. Moreover, we have the following equalities:

$$\Pr(T = 1 \mid Y = 1, \boldsymbol{X} = \mathbf{x}) = \frac{m_{11}(\mathbf{x})}{m_{11}(\mathbf{x}) + m_{10}(\mathbf{x})}, \tag{6.3.1}$$

$$\Pr(T = 1 \mid Y = 0, \boldsymbol{X} = \mathbf{x}) = \frac{m_{01}(\mathbf{x})}{m_{01}(\mathbf{x}) + m_{00}(\mathbf{x})}. \tag{6.3.2}$$

These two equalities give a lot of information. We call them *posterior propensity scores* and we denote by $p_{yt}(\mathbf{x})$ the corresponding conditional probability $\Pr(T = t \mid Y = y, \boldsymbol{X} = \mathbf{x})$. The posterior propensity scores are functions of the conditional means. The quantity $p_{11}(\mathbf{x})$ can be seen as the proportion of treated observations among those that had positive outcomes and $p_{01}(\mathbf{x})$ can be seen as the proportion of treated observations among those that had negative outcomes. We define the second term of our uplift loss as the BCE loss, but this time, using the observed treatment indicator $\mathbf{t}$ as the "response" variable, and $p_{y1}(\mathbf{x})$ as the "prediction". Formally, it is given by:

$$\ell_2(\mathbf{t}, p_{y1}(\mathbf{x}) \mid \mathbf{x}) = -\frac{1}{n}\sum_{i=1}^{n}\Big(t_i \log\{p_{y_i 1}(\mathbf{x}_i)\} + (1 - t_i)\log\{1 - p_{y_i 1}(\mathbf{x}_i)\}\Big).$$

Taken alone, the second loss models the posterior propensity scores as a function of the conditional means (for positive and negative outcomes). Intuitively, if a treatment has a significant positive (resp. negative) effect on a sub-sample of observations, then within the sample of observations that had a positive (resp. negative) response, we expect a higher proportion of treated. Formally, we define the complete uplift loss function in Definition 6.3.2.

Definition 6.3.2. *Let* $m_{yt} \overset{def}{=} m_{yt}(\mathbf{x}) = \Pr(Y = y|\boldsymbol{X} = \mathbf{x}, T = t)$*, and* $p_{yt} \overset{def}{=} p_{yt}(\mathbf{x}) = m_{yt}/(m_{y1} + m_{y0})$*. We define the uplift loss function as follows:*

$$\ell(\mathbf{y}, \mathbf{t} \mid \mathbf{x}) = -\frac{1}{n}\sum_{i=1}^{n}\Big(\underbrace{\{y_i \log m_{1t_i} + (1 - y_i)\log m_{0t_i}\}}_{\textit{conditional means}} + \underbrace{\{t_i \log p_{y_i 1} + (1 - t_i)\log p_{y_i 0}\}}_{\textit{posterior propensity}}\Big). \tag{6.3.3}$$

Although we considered the case where $e(\mathbf{x}) = 1/2$, the development holds for any constant $e(\mathbf{x})$. An under-sampling or an over-sampling procedure allows to recover the $e(\mathbf{x}) = 1/2$ if the constant is below or above $1/2$ respectively. Interestingly, it is possible to find a connection between the uplift loss function (6.3.3) and the likelihood of the data. Indeed, the described relation between the relative risk and the conditional probabilities

$m_{11}(\mathbf{x})$ and $m_{10}(\mathbf{x})$, as well as the Bayesian interpretation of the posterior propensity scores $p_{yt}(\mathbf{x})$ suggest modeling the joint distribution of Y and T. Formally, the connection can be shown through the following development.

$$\begin{aligned}\Pr(Y = y, T = t \mid \boldsymbol{X} = \mathbf{x}) &= \Pr(T = t \mid Y = y, \boldsymbol{X} = \mathbf{x})\Pr(Y = y \mid \boldsymbol{X} = \mathbf{x}) \\ &= p_{yt}(\mathbf{x})\{m_{y1}(\mathbf{x})e(\mathbf{x}) + m_{y0}(\mathbf{x})[1 - e(\mathbf{x})]\} \\ &= p_{yt}(\mathbf{x})\{m_{y1}(\mathbf{x}) + m_{y0}(\mathbf{x})\}/2,\end{aligned}$$

because $e(\mathbf{x}) = 1/2$. Therefore, the likelihood for n observations is proportional to

$$\prod_{i=1}^{n} p_{y_i 1}^{t_i} p_{y_i 0}^{(1-t_i)} \{m_{11} + m_{10}\}^{y_i} \{m_{01} + m_{00}\}^{(1-y_i)},$$

and the log-likelihood is proportional to

$$\sum_{i=1}^{n} \{y_i \log(m_{11} + m_{10}) + (1 - y_i) \log(m_{01} + m_{00}) + t_i \log p_{y_i 1} + (1 - t_i) \log p_{y_i 0}\}. \qquad (6.3.4)$$

Notice that the functions (6.3.3) and (6.3.4) differ only in that (6.3.3) uses m_{1t} while (6.3.4) specifically uses m_{y1} and m_{y0}, the conditional means under treatment and control. Traditionally, m_{1t} is more common since in practice, each observation can only be treated or not treated. However, we compared the results by fitting uplift models using both functions. The results being very similar, in the rest of the paper, we only present results for models fitted with the augmented loss function (6.3.3). We keep the in-depth analysis of the log-likelihood (6.3.4) for future work.

The loss function (6.3.3) can also be interpreted term by term. The first term is simply the binary cross entropy loss w.r.t the conditional means. The second term can be seen as a regularization term on the conditional means. In the second term, the conditional means are represented through the posterior propensity scores. By minimizing the augmented loss, the first term focuses on estimating the conditional means separately while the second term tries to correct for the posterior propensity scores. Since both terms are minimized simultaneously, this can also be seen as a special case of multi-task learning. As we will show later, this new parameter estimation method greatly improves the predictive performance of the underlying uplift models.

6.4. A twin neural model for uplift

Let's start with the uplift interaction model [Lo, 2002] as a simple preliminary model. This model is based on logistic regression. It is common to add explicit interaction terms between each explanatory variable and the treatment indicator. The parameters of the interaction terms measure the additional effect of each covariate due to treatment. These interactions are important for estimating individual effects since this is what makes it possible to create heterogeneity in the treatment effects.

Logistic regression may be visualised in a model diagram, with a single node to represent the link function, and multiple nodes to represent inputs or outputs. This sort of visualization is very common in neural networks community (see Figure 6.1, left panel).

The uplift interaction model can be represented by a fully-connected neural network with no hidden layer, an intercept, $2p + 1$ input neurons (covariates, treatment variable and interaction terms) and 1 output neuron with sigmoid activation function, where $\sigma(z) = 1/(1 + e^{-z})$, for $z \in \mathbb{R}$ (see Figure 6.1 left panel). Let $\mathbf{x} = (x_1,...,x_p) \in \mathbb{R}^p$ be the covariates vector and $t \in \{0,1\}$, a binary variable. Let us further define θ_j, for $j = 1,\ldots,2p + 1$, the coefficient or weight that connects the jth input neuron to the output and let $\theta_o \in \mathbb{R}$ be the intercept. The uplift interaction model can be written as

$$\mu_{1t}(\mathbf{x}, \boldsymbol{\theta}) = \sigma\Big(\theta_o + \sum_{j=1}^{p} \theta_j x_j + \sum_{j=p+1}^{2p} \theta_j t x_{j-p} + \theta_{2p+1} t\Big),\ t \in \{0,1\}, \tag{6.4.1}$$

where $\sigma(\cdot)$ represents the sigmoid function and $\boldsymbol{\theta}$ denotes the vector of model parameters. The predicted uplift associated with the covariates vector $\mathbf{x}_{n+1}$ of a future individual is

$$\hat{u}(\mathbf{x}_{n+1}) = \mu_{11}(\mathbf{x}_{n+1}, \hat{\boldsymbol{\theta}}) - \mu_{10}(\mathbf{x}_{n+1}, \hat{\boldsymbol{\theta}}),$$

where $\hat{\boldsymbol{\theta}}$ may be estimated by minimizing a loss function such as the one defined in Equation (6.3.3). More generally, let $\mathrm{NN}_{1t}(\mathbf{x}, \boldsymbol{\theta})$ for $t \in \{0,1\}$ be a neural network. We denote by $\mathrm{NN}_{11}(\mathbf{x}, \boldsymbol{\theta})$ and $\mathrm{NN}_{10}(\mathbf{x}, \boldsymbol{\theta})$ the conditional mean model for treated and control observations respectively. In what follows, our goal is to generalize the interaction model (6.4.1) by a more flexible neural network. We focus on a fully-connected network with an input of size $p + 1$ (covariates and treatment variable), and one hidden layer of size $m > 1$ with ReLU activation, where $\mathrm{ReLU}(z) = \max\{0,z\}$, for $z \in \mathbb{R}$. We assume that the intercept (also called bias term) is inherent in the neural model. The hidden layer is then connected to a

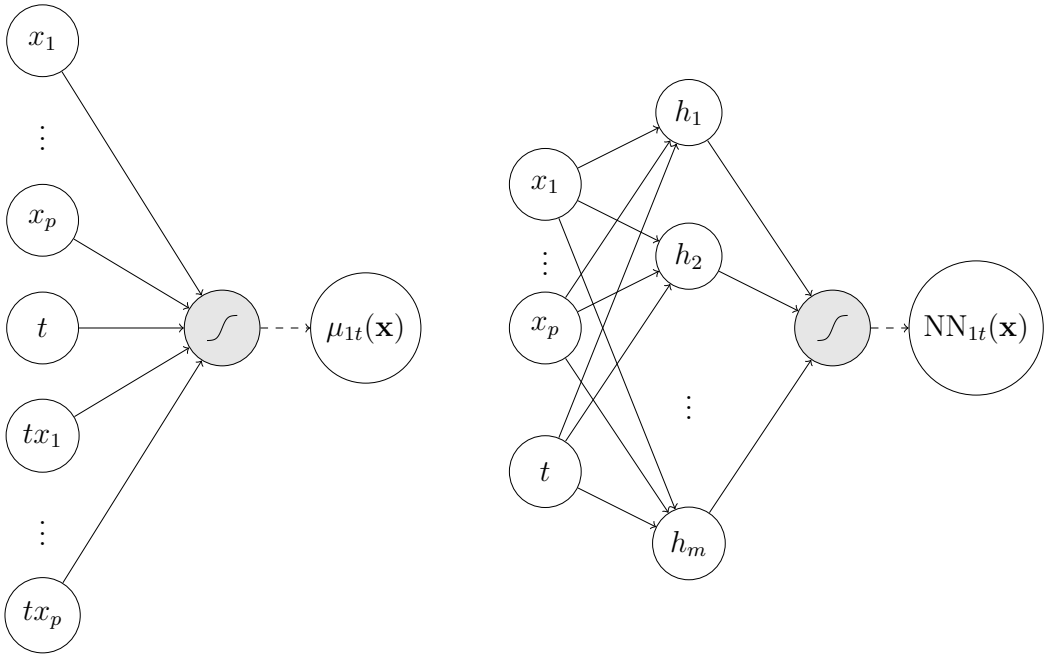

Fig. 6.1. Graphical representation of the uplift interaction model (left panel). Neural network representation of the uplift model (right panel). Note that $h_k = \mathrm{ReLU}(\mathbf{x},t)$ for $k = 1,\ldots,m$. For $m = 2p+1$, Theorem 6.4.1 shows the equivalence between the interaction model (6.4.1) and a particular case of the 1-hidden layer neural network (6.4.2) with ReLU activation. In both diagrams, the gray node represents the sigmoid activation function.

single output neuron with a sigmoid activation (see Figure 6.1 right panel). The output of the neural network $\mathrm{NN}_{1t}(\mathbf{x},\boldsymbol{\theta})$ can be written as

$$\mathrm{NN}_{1t}(\mathbf{x},\boldsymbol{\theta}) = \sigma\left\{\theta_o^{(2)} + \sum_{k=1}^{m}\theta_k^{(2)}\mathrm{ReLU}\left(\theta_{o,k}^{(1)} + \sum_{j=1}^{p}\theta_{j,k}^{(1)}x_j + \theta_{p+1,k}^{(1)}t\right)\right\},\ t\in\{0,1\}, \qquad (6.4.2)$$

where $\theta_{j,k}^{(1)}$ represent the coefficient or weight that connects the jth covariate or input neuron to the kth hidden neuron and $\theta_k^{(2)}$ represents the coefficient that connects the kth hidden neuron to the output. We denote the bias terms for the hidden layer and the output layer by $\theta_{o,k}^{(1)}$, $k = 1,\ldots,m$ and $\theta_o^{(2)}$ respectively. Here, $\boldsymbol{\theta}$ contains all of the neural network's coefficients (or parameters). The predicted uplift associated with the covariates vector $\mathbf{x}_{n+1}$ of a future individual is

$$\hat{u}(\mathbf{x}_{n+1}) = \mathrm{NN}_{11}(\mathbf{x}_{n+1},\hat{\boldsymbol{\theta}}) - \mathrm{NN}_{10}(\mathbf{x}_{n+1},\hat{\boldsymbol{\theta}}),$$

where $\hat{\boldsymbol{\theta}}$ may be estimated by minimizing a loss function such as the one defined in Equation (6.3.3). In the following Theorem, we show that for a judicious choice of the neural network's

coefficients matrix, the two models are equivalent.

Theorem 6.4.1. *Let $\mu_{1t}(\mathbf{x})$ and $\mathrm{NN}_{1t}(\mathbf{x})$ be two uplift models defined as in* (6.4.1) *and* (6.4.2) *respectively. Let $c \in \mathbb{R}^+$ be a positive and finite constant and $m = 2p + 1$. For all $\theta_j \in \mathbb{R}$, $j = 1, \ldots, 2p + 1$ and $\theta_o \in \mathbb{R}$, there exists a matrix of coefficients $\left(\theta^{(1)}_{j,k}\right) \in \mathbb{R}^{(\mathrm{p}+1)\times(2\mathrm{p}+1)}$, such as*

$$\left(\theta^{(1)}_{j,k}\right) = \left(\begin{array}{cccc|cccc|c} 1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 1 & 0 \\ \hline 0 & 0 & \cdots & 0 & c & c & \cdots & c & 1 \end{array}\right),$$

an intercepts vector $\left(\theta^{(1)}_{o,k}\right) \in \mathbb{R}^{2\mathrm{p}+1}$, a vector of coefficients $\left(\theta^{(2)}_{k}\right) \in \mathbb{R}^{2\mathrm{p}+1}$ and an intercept scalar $\theta^{(2)}_o \in \mathbb{R}$ such that for all $\mathbf{x} \in [0,c]^p$ and $t \in \{0,1\}$

$$\mu_{1t}(\mathbf{x}) = \mathrm{NN}_{1t}(\mathbf{x}). \tag{6.4.3}$$

Proof. See Appendix 6.10.1. □

This theorem is interesting since the neural network model is much more flexible and can be seen as a generalization of the interaction model. As we will see in the experimental Section, this flexibility allows a better fit resulting in a higher performance from a prediction point of view.

A twin representation of uplift models. For most existing parametric uplift methods, the uplift prediction is computed in several steps: i) the uplift model is fitted; ii) the conditional probabilities are predicted by fixing the treatment variable T to 1 or 0; iii) the difference is taken to compute the uplift; iv) the uplift is visualized. The fitted model plays a major role in implementing each of these steps. This can be problematic when the fitted model overfit the data at hand. Therefore, most multi-step methods require careful regularization. To simplify this task, we propose to combine the whole process into a single step through a twin model. The twin interaction model diagram is depicted in Figure 6.2. This representation resembles the twin networks [Bromley et al., 1994] in the context of deep learning. Such networks were first introduced in signature verification as an image matching

problem, where the task is to compare two hand-written signatures and infer the identity of the writer.

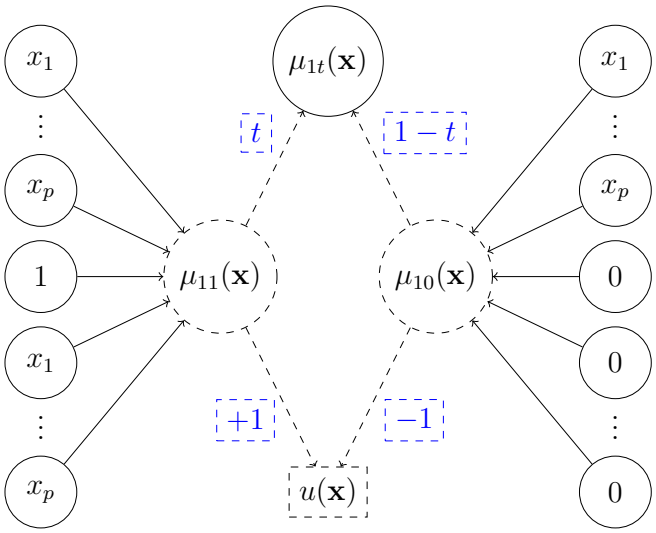

Fig. 6.2. Diagram of the twin logistic interaction model. The original interaction model is separated into two sub-components with the same parameters. For the left sub-component, the treatment variable fixed to 1 and the interaction terms to $\mathbf{x}$. The treatment variable and the interactions terms are fixed to 0 for the right sub-component. The sub-components model the conditional means for treated ($\mu_{11}(\mathbf{x})$) and for control ($\mu_{10}(\mathbf{x})$). The difference gives direct prediction of $u(\mathbf{x})$. At the same time, the predicted conditional mean $\mu_{1t}(\mathbf{x})$ is based on the actual received treatment for each individual $t \in \{0,1\}$, i.e., $\mu_{1t}(\mathbf{x}) = t\mu_{11}(\mathbf{x}) + (1-t)\mu_{10}(\mathbf{x})$.

A twin neural network consists of two models that use the same parameters (or weights) while fitted in parallel on two different input vectors to compute comparable outputs. In our case, the input vectors are almost identical, only the treatment variable is changed. The parameters between the twin networks are shared. Weight sharing guarantees that two individuals with similar characteristics are mapped similarly by their respective networks because each network computes the same function. Such networks are mostly known for their application in face recognition [Chopra et al., 2005, Taigman et al., 2014, Parkhi et al., 2015, Schroff et al., 2015], areal-to-ground image matching [Lin et al., 2015] and large scale video classification [Karpathy et al., 2014], among others.

The twin network representation of the uplift interaction model is easily generalized to neural networks, as show in Figure 6.3. In our case, instead of comparing two distinct images, we duplicate each observation and fix the treatment variable to 1 and 0, while keeping in

memory the true value t of the treatment variable. This makes it possible to fit the twins in parallel and to use the true value t to compute $\mathrm{NN}_{1t}(\mathbf{x})$. The availability of $\mathrm{NN}_{11}(\mathbf{x})$ and $\mathrm{NN}_{10}(\mathbf{x})$ allows to compute the uplift loss function (6.3.3) and to predict the uplift $u(\mathbf{x})$ in a single step, which simplifies the fitting process to a great extent.

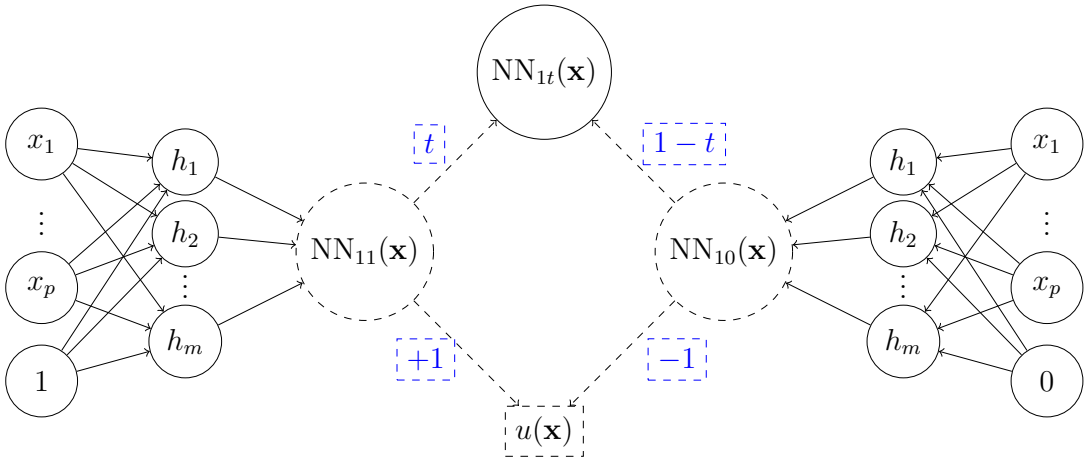

Fig. 6.3. A twin neural model for uplift. The inputs contain the covariates vector $\mathbf{x}$ and, for the left sub-component, the treatment variable fixed to 1. The treatment variable is fixed to 0 for the right sub-component. The sub-components output the predicted conditional means for treated ($\mathrm{NN}_{11}(\mathbf{x})$) and for control ($\mathrm{NN}_{10}(\mathbf{x})$). The difference gives direct prediction of $u(\mathbf{x})$. At the same time, the predicted conditional mean $\mathrm{NN}_{1t}(\mathbf{x})$ is based on the actual received treatment for each individual $t \in \{0,1\}$, i.e., $\mathrm{NN}_{1t}(\mathbf{x}) = t\mathrm{NN}_{11}(\mathbf{x}) + (1-t)\mathrm{NN}_{10}(\mathbf{x})$.

6.5. Parameter estimation

Loss functions are generally convex with respect to $\mathrm{NN}(\cdot)$, but not with respect to the model's parameters $\boldsymbol{\theta}$. Parameter estimation is difficult, and with neural networks, in the best cases we are looking for a good local minimum. Most of the methods are based on some form of gradient descent. Gradient-based optimization is one of the pillars of machine learning. Given the loss function $\ell : \mathbb{R}^{\dim(\boldsymbol{\theta})} \to \mathbb{R}$, classical gradient descent has the goal of finding (local) minima $\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\ \ell(\boldsymbol{\theta})$ via updates of the form $\Delta(\boldsymbol{\theta}) = -\eta\nabla\ell$, where η is a positive step size (or the learning rate).

The prototypical stochastic optimization method is the stochastic gradient method [Robbins and Monro, 1951], which, in the context of minimizing $\ell(\boldsymbol{\theta})$ with a given starting point $\boldsymbol{\theta}^{(0)}$ is

$$\boldsymbol{\theta}^{(q+1)} \leftarrow \boldsymbol{\theta}^{(q)} - \eta \nabla \ell_{i_q} \left(\boldsymbol{\theta}^{(q)} \right), \tag{6.5.1}$$

where $\ell_{i_q}(\cdot)$ is the loss function evaluated using observation $i_q \in \{1,...,n\}$ which is chosen at random. Here we use the term stochastic in the sense that for each parameters update, only a random sample of the data is used. This method is to be distinguished from a gradient descent method using a stochastic learning rate η_q.

There is no particular reason to employ information from only one observation per iteration. Instead, one can employ a *mini-batch* approach in which a small subset of observations $\mathcal{S}_q \subseteq \{1, ..., n\}$, is chosen randomly in each iteration, leading to

$$\boldsymbol{\theta}^{(q+1)} \leftarrow \boldsymbol{\theta}^{(q)} - \frac{\eta}{|\mathcal{S}_q|} \sum_{i \in \mathcal{S}_q} \nabla \ell_i(\boldsymbol{\theta}^{(q)}), \tag{6.5.2}$$

where $\ell_i(\cdot)$ is the loss function evaluated using observation i and $|\mathcal{S}_q|$ is the batch size. Such a mini-batch stochastic gradient method has been widely used in practice. There are some fundamental practical and theoretical reasons why stochastic methods have inherent advantages for large-scale machine learning [Bottou et al., 2018]. Note that optimization algorithms that use only the gradient are called first-order optimization algorithms. Second-order optimization algorithms such as Newton's method use the Hessian matrix. Like all neural network training, we estimate the neural network's parameters using first-order gradient methods.

Neural networks are typically over-parametrized and are prone to overfitting. Hence, regularization is required. Typically the regularization term is a mixture of L_2 (Ridge) and L_1 (lasso) norms, each with its own regularization constant. Similar to classical regression applications, the intercepts are not penalized. The regularization constants are typically small and serve several roles. The L_2 reduces collinearity, and L_1 ignores irrelevant parameters, and both are a remedy to the overfitting, especially in over-parametrized models such as deep neural networks [Efron and Hastie, 2016].

6.5.1. Unstructured sparsity

The parameter estimation process optimizes the uplift loss, so in the interaction model

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \ \ell(\boldsymbol{\theta}) + \lambda \mathcal{R}(\boldsymbol{\theta}), \tag{6.5.3}$$

where $\ell(\boldsymbol{\theta})$ is the loss, $\lambda \in \mathbb{R}^+$ is the regularization constant and $\mathcal{R}(\boldsymbol{\theta})$ is often convex and probably non-differentiable such as $\|\boldsymbol{\theta}\|_1$. The stochastic gradient descent (SGD) for the lasso needs to be modified to make sure the solution remains sparse, so SGD needs to be adapted using the projected gradient update (see Mosci et al. [2010] for more details). Proximal gradient methods are a generalized form of projection used to solve non-differentiable convex optimization problems. For the reader wishing to have more knowledge on proximal operators, we suggest [Beck, 2017, Chapter 6].

Let's focus only on a univariate θ_j. We adjust SGD by splitting θ_j into two positive components, $u_j = \theta_j^+$ and $v_j = \theta_j^-$. Here, θ_j^+ and θ_j^- are the positive and negative parts of $\theta_j \in \mathbb{R}$ so $\theta_j = u_j - v_j$ in which u_j,$v_j \geq 0$ and of course $|\theta_j| = u_j + v_j$. The optimization problem (6.5.3) can be reformulated as

$$\underset{\mathbf{u},\mathbf{v}}{\operatorname{argmin}} \ \ell(\mathbf{u} - \mathbf{v}) + \lambda \mathbf{1}^\top (\mathbf{u} + \mathbf{v}).$$

Let's focus only on a single update of the parameter $\theta_j = u_j - v_j$ and a single batch size. The optimization routine ensures $u_j \geq 0, v_j \geq 0$, by passing the optimizing parameters $\mathbf{u}, \mathbf{v}$ through the proximal projection $\text{ReLU}(\cdot)$ to ensure non-negativity.

Our proximal gradient descent has only two basic steps which are iterated until convergence. For given $u_j^{(0)}, v_j^{(0)}$ the modified SGD is

(1) *gradient step*: define intermediate points $\tilde{u}_j^{(q)}, \tilde{v}_j^{(q)}$ by taking a gradient step such as

$$\begin{aligned} \tilde{u}_j^{(q)} &= u_j^{(q)} - \eta\{\lambda + \nabla \ell_{i_q}(\theta_j^{(q)})\} \\ \tilde{v}_j^{(q)} &= v_j^{(q)} - \eta\{\lambda - \nabla \ell_{i_q}(\theta_j^{(q)})\} \end{aligned} \tag{6.5.4}$$

(2) *projection step*: evaluate the proximal operator at the intermediate points $\tilde{u}_j^{(q)}, \tilde{v}_j^{(q)}$ such as

$$\begin{aligned} u_j^{(q+1)} &\leftarrow \text{ReLU}(\tilde{u}_j^{(q)}) \\ v_j^{(q+1)} &\leftarrow \text{ReLU}(\tilde{v}_j^{(q)}) \\ \theta_j^{(q+1)} &\leftarrow u_j^{(q+1)} - v_j^{(q+1)} \end{aligned} \tag{6.5.5}$$

Exact zero weight updates appear when $u_j < 0$ and $v_j < 0$ to enable unstructured sparsity.

6.5.2. Structured sparsity

Group lasso is a generalization of the lasso method, when features are grouped into disjoint blocks with a total of $G < p$ groups [Yuan and Lin, 2006]. The formulation of group lasso allows us to define what may constitute a suitable group. For pruning the entire hidden nodes, it is enough to define the block to be the weights that define each node. Take the 1-hidden layer uplift neural network $\mathrm{NN}_{1t}(\mathbf{x}, \boldsymbol{\theta})$ defined in (6.4.2). The neural model can be rewritten as

$$\mathrm{NN}_{1t}(\mathbf{x}, \boldsymbol{\theta}, \mathbf{s}) = \sigma\bigg\{\theta_o^{(2)} + \sum_{k=1}^{m} \theta_k^{(2)} \mathrm{ReLU}\Big(s_k\{\theta_{o,k}^{(1)} + \sum_{j=1}^{p} \theta_{j,k}^{(1)} x_j + \theta_{p+1,k}^{(1)} t\}\Big)\bigg\}, \tag{6.5.6}$$

where here, we introduced a scaling factor $s_k \in \mathbb{R}$ for $k = 1, \ldots, m$ (i.e., one scaling factor per hidden node). Introduction of a scaling factor for structured pruning is common in neural networks, see for instance Ramakrishnan et al. [2020]. Let $\mathbf{s} = (s_1, \ldots, s_m)$ be the m-dimensional scaling factors vector. We propose the following optimization problem to enforce structured sparsity in the node level

$$\underset{\mathbf{u},\mathbf{v},\mathbf{a},\mathbf{b}}{\operatorname{argmin}} \; \ell(\mathbf{u}, \mathbf{v}, \mathbf{a}, \mathbf{b}) + \lambda_1 \mathbf{1}^\top (\mathbf{a} + \mathbf{b}) + \lambda_2 \mathcal{R}(\mathbf{u} - \mathbf{v}),$$

where $\lambda_1, \lambda_2 \in \mathbb{R}^+$ are the regularization constants and λ_1 controls the amount of structured sparsity. This formulation allows to use a similar lasso proximal SGD development using the introduced scaling factor s_k. So we define $s_k = a_k - b_k$, for $k = 1, \ldots, m$ in which $a_k, b_k \geq 0$ so that $|s_k| = a_k + b_k$ and $\lambda_1 \|\mathbf{s}\|_1 = \lambda_1 \mathbf{1}^\top (\mathbf{a} + \mathbf{b})$. This results in the following modified SGD updates for given $a_k^{(0)}, b_k^{(0)}$ in addition to the lasso proximal updates (6.5.4) and (6.5.5)

(1) *gradient step*: define intermediate points $\tilde{a}_k^{(q)}, \tilde{b}_k^{(q)}$ by taking the gradient step

$$\tilde{a}_k^{(q)} = a_k^{(q)} - \eta\{\lambda_1 + \nabla \ell_{i_q}(s_k^{(q)})\}$$
$$\tilde{b}_k^{(q)} = b_k^{(q)} - \eta\{\lambda_1 - \nabla \ell_{i_q}(s_k^{(q)})\}$$

(2) *projection step*: evaluate the proximal operator at the intermediate points $\tilde{a}_k^{(q)}, \tilde{b}_k^{(q)}$ such as

$$a_k^{(q+1)} \leftarrow \mathrm{ReLU}(\tilde{a}_k^{(q)})$$
$$b_k^{(q+1)} \leftarrow \mathrm{ReLU}(\tilde{b}_k^{(q)})$$
$$s_k^{(q+1)} \leftarrow a_k^{(q+1)} - b_k^{(q+1)}$$

Pruning the kth node happens when $a_k < 0$ and $b_k < 0$ which enables structured sparsity. This structured sparsity yields an automatic selection of the number of hidden nodes m.

6.6. Model evaluation

In the context of model selection, in practice, given L models and/or hyperparameter settings, we build L estimators $\{\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_L\}$. We intend to maximize expected prediction performance, using some goodness-of-fit statistic. There are several ways to evaluate prediction performance. However, data-splitting to *training data* and *validation data* is the most widely used in practice [Arlot et al., 2010]. Before fitting the models, the observations are randomly split into training samples $\mathcal{T}$and validation samples $\mathcal{V}$. All models are fit on $\mathcal{T}$, but evaluated on $\mathcal{V}$.

Classic evaluation approaches are ineffective for treatment effect estimation, because both treatment and control are not observed in any observation so the true treatment effect is never observed. *Qini coefficient*, which is based on the *Qini curve* [Radcliffe, 2007], is commonly used in the uplift literature as an alternative to the goodness-of-fit statistic (see Definition 6.6.2). The Qini curve separates observations into heterogeneous segments in terms of reactions to the treatment and identify sub-groups with most varying predicted uplifts (see Definition 6.6.1).

Definition 6.6.1. *Given a model, let $\hat{u}_{(1)} \geq \hat{u}_{(2)} \geq ... \geq \hat{u}_{(|\mathcal{V}|)}$ be the sorted predicted uplifts on the validation set $\mathcal{V}$. Let $\phi \in [0,1]$ be a given proportion. Define $N_\phi = \{i : \hat{u}_i \geq \hat{u}_{(\lceil \phi |\mathcal{V}| \rceil)}\} \subset \{1, \ldots, |\mathcal{V}|\}$ as the subset of observations with the $\phi|\mathcal{V}| \times 100\%$ highest predicted uplifts. The Qini curve is the function g of the fraction of population treated ϕ, where*

$$g(\phi) = \left(\sum_{i \in N_\phi} y_i t_i - \sum_{i \in N_\phi} y_i (1 - t_i) \left\{ \sum_{i \in N_\phi} t_i / \sum_{i \in N_\phi} (1 - t_i) \right\} \right) / \sum_{i=1}^{|\mathcal{V}|} t_i. \tag{6.6.1}$$

In practice, the domain of $\phi \in [0,1]$ is partitioned into J bins, or $J + 1$ grid points $0 = \phi_1 < \phi_2 < ... < \phi_{J+1} = 1$. The Qini coefficient is an approximation of the area under the Qini curve.

Definition 6.6.2. *The Qini coefficient is given by:*

$$\hat{q} = \frac{1}{2}\sum_{k=1}^{J}(\phi_{k+1} - \phi_k)\{Q(\phi_{k+1}) + Q(\phi_k)\} \times 100\%. \tag{6.6.2}$$

where $Q(\phi) = g(\phi) - \phi\ g(1)$ *and* $g(1)$ *is the average treatment effect in the validation set.*

Unlike the area under the ROC curve, $\hat{q}$ may take negative values. A negative $\hat{q}$ means the uplift is worse than random targeting. A good uplift model groups the individuals in decreasing uplift bins. This can be measured by the similarity between the theoretical uplift percentiles of predictions compared with empirical percentiles observed in the data. Maximizing the *adjusted Qini coefficient*, given in Definition 6.6.3, maximizes the Qini coefficient and simultaneously promotes grouping the individuals in decreasing uplift bins, which in turn result in concave Qini curves [Belbahri et al., 2021].

Definition 6.6.3. *Let* B_k *denote the* k*th bin* $(\phi_k, \phi_{k+1}] \subseteq (0,1]$, $k = 1, \ldots, J$. *The adjusted Qini coefficient is defined as:*

$$\hat{q}_{\text{adj}} = \rho\ \max\{0, \hat{q}\}, \tag{6.6.3}$$

where ρ *is the Kendall's uplift rank correlation:*

$$\hat{\rho} = \frac{2}{K(K-1)}\sum_{i<j}\text{sign}(\bar{\hat{u}}_i - \bar{\hat{u}}_j)\ \text{sign}(\bar{u}_i - \bar{u}_j), \tag{6.6.4}$$

where $\bar{\hat{u}}_k$ *is the average predicted uplift in bin* B_k, $k \in 1,...,J$, *and* $\bar{u}_k$ *is the observed uplift in the same bin.*

For the remainder of the paper, we use $\hat{q}_{\text{adj}}$ as the models comparison measure.

6.7. Experiments

We demonstrate the utility of our proposed methods in a simulation study. Each simulation is defined by a data-generating process with a known effect function. Each run of each simulation generates a dataset, and split into training, validation, and test subsets. We use the training data to estimate L different uplift functions $\{\hat{u}_l\}_{l=1}^{L}$ using L different methods. The models are fine-tuned using the training and validation observations and results are presented for the test set.

6.7.1. Data generating process

We generate synthetic data similar to Powers et al. [2018]. For each experiment, we generate n observations and p covariates. We draw odd-numbered covariates independently from a standard Gaussian distribution. Then, we draw even-numbered covariates independently from a Bernoulli distribution with probability 1/2. Across all experiments, we define the mean effect function $\mu(\cdot)$ and the treatment effect function $\tau(\cdot)$ for a given noise level σ^2. Given the elements above, our data generation model is, for $i = 1,...,n$,

$$Y_i^* \mid \mathbf{x}_i, t_i \sim \mathcal{N}(\mu(\mathbf{x}_i) + t_i\tau(\mathbf{x}_i), \sigma^2),$$
$$Y_i = \mathbb{1}(Y_i^* > 0 \mid \mathbf{x}_i, t_i)$$

where t_i is the realisation of the random variable $T_i \sim$ Bernoulli(1/2) and Y_i is the binary outcome random variable. Following Powers et al. [2018], within each set of simulations, we make different choices of mean effect function and treatment effect function. In this section, we describe the results associated with the most complex scenario since the conclusions are similar from one scenario to another. For the reader wishing to analyze the results for other scenarios, we present them in Appendix 6.10.2. We fix $n = 20000$, $p = 100$ and $\sigma = 4$ and define $\mu(\mathbf{x})$ and $\tau(\mathbf{x})$ as follows

$$\mu(\mathbf{x}) = 4\mathbb{1}(x_1 > 1)\mathbb{1}(x_3 > 0) + 4\mathbb{1}(x_5 > 1)\mathbb{1}(x_7 > 0) + 2x_8x_9,$$
$$\tau(\mathbf{x}) = \frac{1}{\sqrt{2}}(f_4(\mathbf{x}) + f_5(\mathbf{x})),$$

where

$$\begin{aligned} f_4(\mathbf{x}) = {} & x_2x_4x_6 + 2x_2x_4(1-x_6) + 3x_2(1-x_4)x_6 + 4x_2(1-x_4)(1-x_6) + 5(1-x_2)x_4x_6 \\ & + 6(1-x_2)x_4(1-x_6) + 7(1-x_2)(1-x_4)x_6 + 8(1-x_2)(1-x_4)(1-x_6), \\ f_5(\mathbf{x}) = {} & x_1 + x_3 + x_5 + x_7 + x_8 + x_9 - 2. \end{aligned}$$

Next, we repeat each experiment 20 times and each run generates a dataset, which is divided into training (40%), validation (30%), and test samples (30%). The models are fitted using the training observations and results are presented for the test set.

6.7.2. Regularization

In Table 6.1, we compare the performance of the twin model $\text{NN}_{1t}(\mathbf{x}, \boldsymbol{\theta}, \mathbf{s})$ (defined in (6.5.6)) using different regularization functions, with and without the scaling factors for pruning (i.e., structured sparsity). Next, we will refer to this model as $Twin_{\text{NN}}$. We fix the initial number of hidden neurons (or nodes) to $m = 512$. When structured sparsity regularization is used, we denote by $\hat{m}$ the number of remaining hidden neurons. We run the experiments 20 times. For fitting the models in each experiment, we vary the hyper-parameters from the following grid: *learning rate* $\eta \in \{0.005, 0.01, 0.05, 0.1, 0.2, 0.3\}$, *structured sparsity constant* $\lambda_1 \in \{0, 0.0001, 0.0005, 0.001, 0.005, 0.01\}$, *regularization constant* $\lambda_2 \in \{0, 0.0001, 0.0005, 0.001, 0.005, 0.01\}$ and cross-validate on the $\hat{q}_{\text{adj}}$ for each combination.

Structured Sparsity	$\mathcal{R}(\cdot)$	$\hat{m}/m$	$\hat{q}_{\text{adj}}$
No	L_2	512/512	3.15
No	L_1	512/512	3.35
Yes	0	218/512	3.09
Yes	L_2	417/512	3.32
Yes	L_1	340/512	**3.58**

Tab. 6.1. Average adjusted Qini (20 runs) for the twin model (6.5.6) with structured pruning of nodes and different regularization functions $\mathcal{R}(\cdot)$. The L_1 regularization of weights provides the highest performance. Note that the maximum standard-error is 0.1; we do not report them to simplify the Table.

With the L_1 regularization, the test-set adjusted Qini coefficient is higher than when the L_2 regularization is used. When structured sparsity is used alone, the model removes too many neurons (around half) but $\hat{q}_{\text{adj}}$ decreases to 3.09. On the other hand, when structured sparsity is used in addition to the L_2 regularization, we see a clear improvement over L_2 or structured sparsity alone, and in this case, the number of pruned neurons is smaller. However, it does not outperform the L_1 regularization used alone. Finally, the best combination is the L_1 regularization paired with the structured sparsity achieving an average adjusted Qini coefficient of 3.58.

Now, when structured sparsity is used to prune the hidden nodes, it is possible to refit a new model with $\hat{m}$ hidden nodes. In our experiments, this did not necessarily have a positive impact on prediction performance. In the case where only structured sparsity is used, $\hat{m} = 218$. For a model with 218 hidden neurons fitted with the L_2 regularization, $\hat{q}_{\text{adj}}$ drops to 2.80. With L_1, it increases slightly to reach an average of 3.16. The same results are observed when we first use L_2 and structured sparsity and then refit models with 417 hidden neurons. Finally, in the case where we fit the models directly with L_1 and structured sparsity, the prediction performance is at its maximum. Indeed, when we refit the models with $\hat{m} = 340$ hidden nodes, whether with L_2 or L_1, the adjusted Qini decreases from 3.58 to 3.01 and 3.31 respectively. Other scenarios not presented here yielded similar conclusions. Therefore, it seems better to fit the models in one step, using the L_1 regularization and structured sparsity. This makes it possible to prune a few nodes from the hidden layer, and to get sparse estimation of the remaining parameters. This seems to improve the prediction performance of the underlying model. For the rest of the paper, we fit our models with L_1 and structured sparsity.

6.7.3. Comparison with benchmark models

In this section, we compare our models to different benchmarks. First, let us focus on the simple twin model $\mu_{1t}(\mathbf{x}, \boldsymbol{\theta})$, defined in (6.4.1). The goal is to compare three optimization procedures. The first method optimizes the penalized Binomial likelihood, as in the case of a *lasso* logistic regression. This is the baseline model, and we will refer to it as *Logistic* as a reference to Lo's interaction model [Lo, 2002]. For fair comparison, we use the twin neural architecture. The second method is a two-stage method which is based on a derivative-free optimization of the $\hat{q}_{\text{adj}}$ and imposes sparsity [Belbahri et al., 2021]. We denote this model by *Qini-based* and our proposed method by *Twin*$_\mu$. We also compare to lasso versions of the R-learner [Nie and Wager, 2020] and X-learner [Künzel et al., 2019]. Table 6.2 shows the averaged results on the test dataset. This experience shows the superiority of using our uplift loss function. Results for other scenarios are presented in Appendix 6.10.2.

Next we compare our twin model $\text{NN}_{1t}(\mathbf{x}, \boldsymbol{\theta}, \mathbf{s})$, defined in (6.5.6), with commonly used benchmark models. A method that has proven to be very effective in estimating

Logistic	*Qini-based*	$Twin_{\mu}$	*R-Learner* (`lasso`)	*X-Learner* (`lasso`)
2.59	2.94	**3.12**	2.83	2.91

Tab. 6.2. "Simple" model prediction performance comparison in terms of $\hat{q}_{\text{adj}}$. Note that the maximum standard-error is 0.05; we do not report them to simplify the Table.

treatment effects is one based on generalized random forests [Athey et al., 2019]. This method uses honest estimation, i.e., it does not use the same information for the partition of the covariates space and for the estimation of the uplift. This has the effect of reducing overfitting. Another candidate that we consider in our experiments is also based on random forests and designed for uplift [Guelman et al., 2012]. For this method, we consider two different split criteria, one based on the Kullback-Leibler (KL) divergence and one based on the Euclidean distance (ED). We also compare our method to XGboost [Chen et al., 2015] versions of the R- and X-learners. In Table 6.3, we compare these benchmark models to different versions of our $Twin_{\text{NN}}$ with different values of m (hidden neurons), with and without structured sparsity. Appendix 6.10.2 provides details about model fine-tuning and other scenarios results.

Neural networks have two main hyper-parameters that control the architecture or topology of the network: the number of hidden layers and the number of nodes in each hidden layer. The number of nodes can be seen as a hyper-parameter. If we don't use our pruning technique for structured sparsity, we can search for the right number of nodes over a grid. For comparison purposes, we varied the number of nodes using either $m = 64, 128, 256$ or $m = 512$ nodes. As we can see in Table 6.3, the best model is reached for $m = 512$ (with $\hat{q}_{\text{adj}} = 3.35$). Choosing the number of hidden neurons by cross-validation is very common in practice. This is part of architecture search. However, choosing among a few values does not necessarily mean that all the selected model's neurons are useful. As discussed earlier, the use of the scaling factor penalization on the weight matrix allows to automatically fine-tune the number of hidden neurons. This has the effect of increasing the performance of the underlying models from a predictive point of view, as shown in Table 6.1. Thus, we suggest to start with a fairly large number of hidden neurons (e.g., $m \geq 2p + 1$), and to let the optimization determine the number of active neurons needed for a specific dataset.

Method	Structured Sparsity	Hidden-Layer Size ($\hat{m}/m$)	$\hat{q}_{\text{adj}}$
Twin$_{\text{NN}}$	No	64/64	3.10
Twin$_{\text{NN}}$	No	128/128	3.16
Twin$_{\text{NN}}$	No	201/201	3.14
Twin$_{\text{NN}}$	No	256/256	3.26
Twin$_{\text{NN}}$	No	512/512	3.35
Twin$_{\text{NN}}$	Yes	340/512	**3.58**
Causal Forest			2.79
Causal Forest (Honest)			**3.07**
Uplift Random Forest (KL)			2.19
Uplift Random Forest (ED)			2.33
R-Learner (`XGboost`)			2.12
X-Learner (`XGboost`)			2.37

Tab. 6.3. Models prediction performance comparison in terms of $\hat{q}_{\text{adj}}$. Note that the maximum standard-error is 0.1; we do not report them to simplify the Table.

Based on these experiments, it appears that the twin neural model fitted with L_1 regularization and structured sparsity outperforms all other methods significantly in terms of $\hat{q}_{\text{adj}}$ (e.g., based on Wilcoxon signed-rank test [Wilcoxon, 1992]). We also observe that the random forest with honesty criterion performs best in comparison to the other uplift benchmark models.

Finally, the number of hidden layers can also be seen as a hyper-parameter for the neural network architecture. Note that in our experiments, increasing the number of hidden layers did not improve the prediction performance significantly. For instance, we fixed the initial number of hidden neurons to $p + 1$ and p for a 2-hidden layers twin model and observed an average $\hat{q}_{\text{adj}}$ of 3.39. Results for other scenarios are given in Appendix 6.10.2.

6.8. Application

We have the rare opportunity to have access to real data from a large scale randomized experiment. Indeed, in the uplift domain, it is not easy to find public benchmarks, apart from the recent CRITEO-UPLIFT1 dataset [Diemert Eustache, Betlei Artem et al., 2018].

The CRITEO-UPLIFT1 dataset is constructed by collecting data from a particular randomized trial procedure where a random part of the population is prevented from being targeted by advertising. The dataset consists of $\approx 14M$ rows, each one representing a user with $p = 12$ covariates, a treatment indicator and a binary response variables (visits). Positive responses mean the user visited the advertiser website during the test period (2 weeks). The proportion of treated individuals is 85%. We use an undersampling method in order to bring the treatment/control ratio back to 1. Thus, we have about $4M$ observations. We also create two other balanced datasets by randomly sampling $400K$ and $1M$ observations.

We fit the twin-network $Twin_{\text{NN}}$ and optimize the regularized loss function with lasso and structured sparsity. For comparison purposes, we also consider the honest causal forest, which gave the best benchmark in our simulation study. We split the data into training (40%), validation (30%), and test samples (30%). We use the training and the validation sets to fit and fine-tune the models. Then, we select the best model based on the validation set. Using the best model, we score the samples from the test set and compute the adjusted Qini coefficient $\hat{q}_{\text{adj}}$. We repeat the experiment 20 times and report averaged results as well as their standard-errors in Table 6.4.

Dataset	Sample Size	p	$Twin_{\text{NN}}$	$\hat{m}/m$	*Causal Forest (Honest)*
Insurance data	50K	40	0.19 (0.003)	212/512	0.06 (0.007)
CRITEO-UPLIFT1	100K	12	0.19 (0.008)	187/512	0.14 (0.011)
CRITEO-UPLIFT1	1M	12	0.24 (0.007)	127/512	0.18 (0.019)
CRITEO-UPLIFT1	4M	12	0.28 (0.006)	63/512	0.23 (0.015)

Tab. 6.4. Application on real-data. Prediction performance in terms of $\hat{q}_{\text{adj}}$ averaged over 20 realizations (standard-errors are given in parenthesis).

The same type of analysis is repeated on a data set that was made available to us by a car insurance company with $50K$ customers and $p = 40$ covariates. This company was interested in designing strategies to maximize its conversion rate. An experimental acquisition campaign was implemented for 6 months, for which half of the potential clients were randomly allocated into a treatment group and the other half into a control group. Potential clients under the treatment group were contacted. The goal of the analysis is to

propose a predictive model that maximizes the return on investment of future initiatives (i.e., a maximum $\hat{q}_{\text{adj}}$ on a test set). The observed difference in sales rates between the treated and the control groups shows a slightly positive impact of the marketing initiative, that is, 0.55%. Results are reported in Table 6.4.

For the insurer's data, we also considered a model without hidden layers (i.e., $Twin_{\mu}$). The objective being model interpretability. In this case, the model is not as efficient as the neural network one (see Table 6.5). However, it performs better than the causal forest. Therefore, if the practitioner needs an interpretable model, he can use the simple twin model and still get satisfactory results. For comparison purposes, we also fit the *Logistic* model. In this case, the model's parameters are estimated by maximizing the penalized Binomial log-likelihood.

Method	$\hat{m}/m$	Training	Validation	Test
$Twin_{\mu}$	-	0.112 (0.005)	0.098 (0.004)	0.085 (0.005)
$Twin_{\text{NN}}$	212/512	0.223 (0.006)	0.202 (0.003)	0.187 (0.003)
Logistic	-	0.197 (0.029)	0.062 (0.019)	0.021 (0.007)
Causal Forest (Honest)	-	0.573 (0.028)	0.157 (0.022)	0.059 (0.007)

Tab. 6.5. Adjusted Qini coefficients on the insurance data (standard errors are given in parenthesis). The results are averaged over 20 runs.

For these types of marketing initiatives, it makes sense that the overall impact is small. A potential reason is that customers are already interested in the product since it is mandatory to get car insurance. The effect of the call is slightly positive on average, that is to say that during the call, the advisor allows the customer to understand all the details of the coverage and thus the customer is most likely reassured. Therefore, it is rather unlikely that the call would have a negative effect. We believe that for this data set, the situation is similar to the first scenario from the simulations presented in Appendix 6.10.2. This may explain why it is more difficult to prevent overfitting with the causal forest model.

6.9. Conclusion

We present a meaningful and intuitive twin neural networks architecture for the problem of uplift modeling. We proposed to estimate the model's parameters by optimizing a new

loss function. This loss function is built by leveraging a connection with the Bayesian interpretation of the relative risk. The twin neural network performs well on predictive tasks and overfitting is minimized by using a proper regularization term. We applied our method to synthetic and real-world data and we compared it with the state-of-the-art methods for uplift. We modify the learning algorithm to allow for structured sparse solutions, which significantly helped training uplift models. Our results show that the twin models significantly outperform the common approaches to uplift such as random forests in all scenarios.

Our methodological development has been driven by real data from a large scale random experiment where the treatment is the marketing initiative. The methodology is also readily applicable to other types of datasets, such as randomized clinical trial data, for predicting variability in medical treatment response. In addition to the quality of prediction obtained in optimal settings, we observed that even the one-layer version of the proposed architecture performs well compared to random forest models. In this case, the model can be interpreted in the same way as a logistic regression. This interpretation is of great importance in practice, since the deployment of an interpretable model minimizes risks of applying an inadequate model and facilitates convincing business owners. This is also true in the medical world, where understanding the variability in response to treatment predictions is of great importance for advancing precision medicine, but not at the cost of model explanation.

6.10. Appendix

6.10.1. Proof of Theorem 6.4.1

PROOF. We prove the theorem by finding explicitly the coefficients of $\mathrm{NN}_{1t}(\mathbf{x})$ that verify the equality (6.4.3). First, we fix the vector of coefficients that connects the hidden layer to the output such as

$$\theta_k^{(2)} = \theta_j,$$

for $k = j$, for k,$j = 1, \ldots, 2p + 1$. Moreover, we fix the intercept $\theta_o^{(2)}$ to be equal to the intercept from the first model, that is, $\theta_o^{(2)} = \theta_o$. Therefore, $\mathrm{NN}_{1t}(\mathbf{x})$ can be written as

$$\mathrm{NN}_{1t}(\mathbf{x}) = \sigma\Bigg\{\theta_o + \sum_{k=1}^{2p+1} \theta_k \mathrm{ReLU}\Big(\theta_{o,k}^{(1)} + \sum_{j=1}^{p} \theta_{j,k}^{(1)} x_j + \theta_{p+1,k}^{(1)} t\Big)\Bigg\}. \tag{6.10.1}$$

Next, we need to define the right structure for the matrix of coefficients $\left(\theta_{j,k}^{(1)}\right) \in \mathbb{R}^{(p+1)\times(2p+1)}$ to be able to recover the $\mu_{1t}(\mathbf{x})$ model. First, let us fix the intercepts vector $\left(\theta_{o,k}^{(1)}\right) \in \mathbb{R}^{2p+1}$ such as

$$\theta_{o,k}^{(1)} = \begin{cases} 0 & \text{if } k \in \{1,\ldots,p\} \\ -c & \text{if } k \in \{p+1,\ldots,2p\} \\ 0 & \text{if } k = 2p+1 \end{cases}$$

Finally, we give the following representation to the matrix of coefficients $\left(\theta_{j,k}^{(1)}\right) \in \mathbb{R}^{(p+1)\times(2p+1)}$, that is,

$$\left(\theta_{j,k}^{(1)}\right) = \left(\begin{array}{cccc|cccc|c} 1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 1 & 0 \\ \hline 0 & 0 & \cdots & 0 & c & c & \cdots & c & 1 \end{array}\right).$$

Therefore, based on these attributions, we have the following equalities. For $k \in \{1,\ldots,p\}$,

$$\text{ReLU}\left(\theta_{o,k}^{(1)} + \sum_{j=1}^{p} \theta_{j,k}^{(1)} x_j + \theta_{p+1,k}^{(1)} t\right) = \text{ReLU}(x_k) = x_k. \tag{6.10.2}$$

Then, for $k \in \{p+1,\ldots,2p\}$,

$$\text{ReLU}\left(\theta_{o,k}^{(1)} + \sum_{j=1}^{p} \theta_{j,k}^{(1)} x_j + \theta_{p+1,k}^{(1)} t\right) = \text{ReLU}(-c + x_{k-p} + ct) = t x_{k-p}. \tag{6.10.3}$$

In (6.10.3), the outcome depends on the value of t and the second equality comes from the fact that we have

$$\text{ReLU}(-c + x_{k-p} + ct) = \begin{cases} \text{ReLU}(x_{k-p} - c) = 0 & \text{if } t = 0, \\ \text{ReLU}(x_{k-p}) = x_{k-p} & \text{if } t = 1, \end{cases}$$

therefore, $\text{ReLU}(-c + x_{k-p} + ct) = t x_{k-p}$. Finally, for $k = 2p+1$, we have

$$\text{ReLU}\left(\theta_{o,k}^{(1)} + \sum_{j=1}^{p} \theta_{j,k}^{(1)} x_j + \theta_{p+1,k}^{(1)} t\right) = \text{ReLU}(t) = t. \tag{6.10.4}$$

Hence, the desired result is obtained by simply replacing (6.10.2), (6.10.3) and (6.10.4) in (6.10.1). □

6.10.2. Additional experiments

In this section, we present in more detail the experimental protocol as well as the results associated with different scenarios. Each simulation is defined by a data-generating process with a known effect function. Each run of each simulation generates a dataset, and split into training, validation, and test subsets. We use the training data to estimate L different uplift functions $\{\hat{u}_l\}_{l=1}^{L}$ using L different methods. The models are fine-tuned using the training and validation observations and results are presented for the test set.

We generate synthetic data similar to Powers et al. [2018]. For each experiment, we generate n observations and p covariates. We draw odd-numbered covariates independently from a standard Gaussian distribution. Then, we draw even-numbered covariates independently from a Bernoulli distribution with probability 1/2. Across all experiments, we define the mean effect function $\mu(\cdot)$ and the treatment effect function $\tau(\cdot)$ for a given noise level σ^2. Given the elements above, our data generation model is, for $i = 1,...,n$,

$$
\begin{aligned}
Y_i^* \mid \mathbf{x}_i, t_i &\sim \mathcal{N}(\mu(\mathbf{x}_i) + t_i\tau(\mathbf{x}_i), \sigma^2), \\
Y_i &= \mathbb{1}(Y_i^* > 0 \mid \mathbf{x}_i, t_i)
\end{aligned}
$$

where t_i is the realisation of the random variable $T_i \sim$ Bernoulli(1/2) and Y_i is the binary outcome random variable.

The random variable Y_i follows a Bernoulli distribution with parameter $\Pr(Y_i^* > 0 \mid \mathbf{x}_i, t_i)$ and it is easy to recover the "true" uplift u_i^*, that is,

$$
\begin{aligned}
u_i^* &= \Pr(Y_i^* > 0 \mid \mathbf{x}_i, T_i = 1) - \Pr(Y_i^* > 0 \mid \mathbf{x}_i, T_i = 0) \\
u_i^* &= \Phi\Big(\frac{\mu(\mathbf{x}_i) + \tau(\mathbf{x}_i)}{\sigma}\Big) - \Phi\Big(\frac{\mu(\mathbf{x}_i)}{\sigma}\Big) \qquad (6.10.5)
\end{aligned}
$$

for $i = 1, \ldots, n$, where $\Phi(.)$ is the standard Gaussian cumulative distribution function.

Following Powers et al. [2018], within each set of simulations, we make different choices of mean effect function and treatment effect function, each represents a wide variety of functional forms, univariate and multivariate, additive and interactive, linear and piecewise constant. Table 6.6 gives the mean and treatment effect functions for the different randomized simulations.

	Scenarios			
Parameters	1	2	3	4
n	10000	20000	20000	20000
p	200	100	100	100
$\mu(\mathbf{x})$	$f_7(\mathbf{x})$	$f_3(\mathbf{x})$	$f_2(\mathbf{x})$	$f_6(\mathbf{x})$
$\tau(\mathbf{x})$	$f_4(\mathbf{x})$	$f_5(\mathbf{x})$	$f_7(\mathbf{x})$	$f_8(\mathbf{x})$
σ	1/2	1	1	4

Tab. 6.6. Specifications for the simulation scenarios. The rows of the table correspond, respectively, to the sample size, dimensionality, mean effect function, treatment effect function and noise level.

The functions are

$$f_2(\mathbf{x}) = 5\mathbb{1}(x_1 > 1) - 5$$

$$f_3(\mathbf{x}) = 2x_1 - 4$$

$$f_4(\mathbf{x}) = x_2x_4x_6 + 2x_2x_4(1-x_6) + 3x_2(1-x_4)x_6 + 4x_2(1-x_4)(1-x_6) + 5(1-x_2)x_4x_6$$
$$+ 6(1-x_2)x_4(1-x_6) + 7(1-x_2)(1-x_4)x_6 + 8(1-x_2)(1-x_4)(1-x_6)$$

$$f_5(\mathbf{x}) = x_1 + x_3 + x_5 + x_7 + x_8 + x_9 - 2$$

$$f_6(\mathbf{x}) = 4\mathbb{1}(x_1 > 1)\mathbb{1}(x_3 > 0) + 4\mathbb{1}(x_5 > 1)\mathbb{1}(x_7 > 0) + 2x_8x_9$$

$$f_7(\mathbf{x}) = \frac{1}{2}(x_1^2 + x_2 + x_3^2 + x_4 + x_5^2 + x_6 + x_7^2 + x_8 + x_9^2 - 11)$$

$$f_8(\mathbf{x}) = \frac{1}{\sqrt{2}}(f_4(\mathbf{x}) + f_5(\mathbf{x})).$$

In Figure 6.4, we show the Qini curves associated with the "true" uplift u^*. The curve represents the incremental number of positive responses for a fraction of treated observations relative to the size of the treatment group in the sample. The dashed straight line is the performance of a random strategy. The scenarios are interesting because they represent different situations that we can face with real data. Indeed, scenario 1, although it seems simpler than the others (with a small noise level), is in fact a case that happens often, especially for marketing campaigns. In this scenario, by construction, there are no do-not-disturb clients. This results in an increasing monotonic Qini curve. In scenario 2,

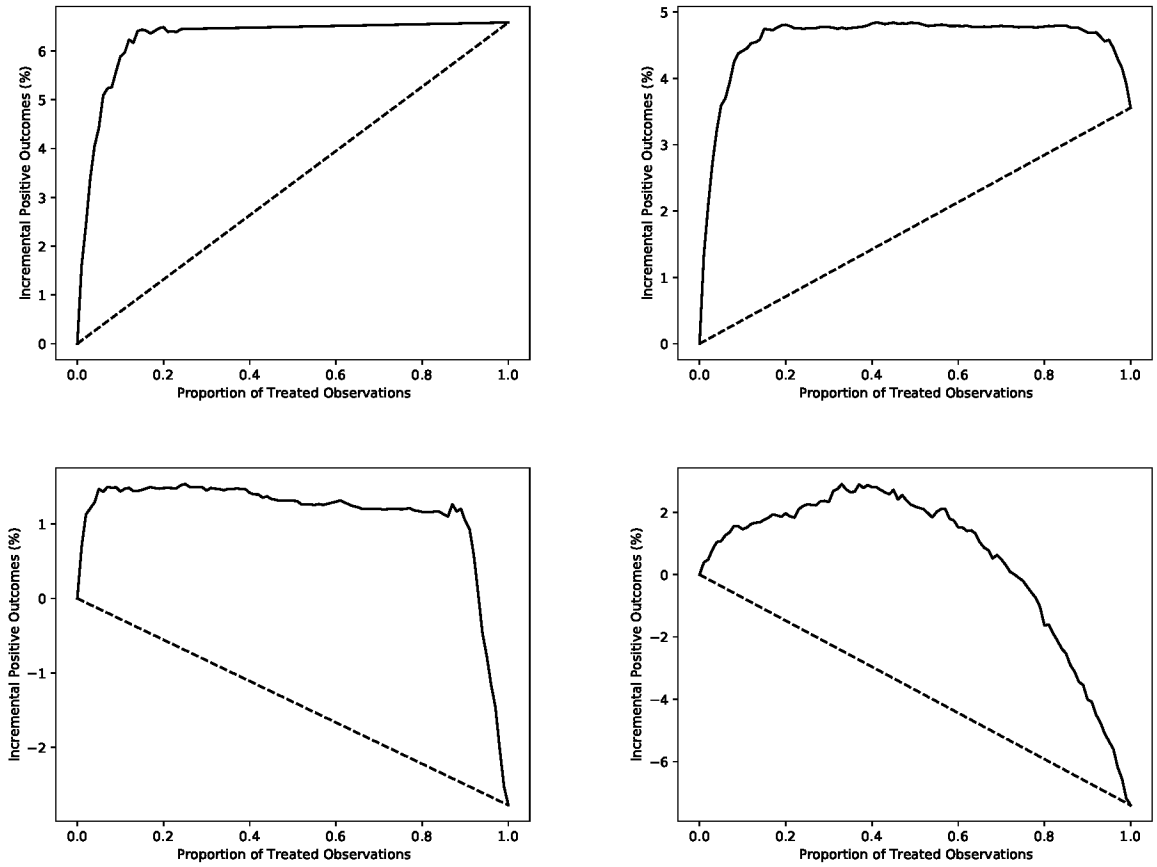

Fig. 6.4. Qini curves based on the "true" uplift with respect to scenarios 1-4 of Table 6.6. Top left: scenario 1; top right: scenario 2; bottom left: scenario 3; bottom right: scenario 4.

we introduce a group of do-not-disturb clients. In addition, we increase the noise level as well as the number of observations, but we reduce the number of variables. The average treatment effect is negative but a small group of persuadables is generated. In scenario 3, we introduce a quadratic treatment effect function and scenario 4 is the most complex one.

Each run of each simulation generates a dataset, which we split into training (40%), validation (30%), and test samples (30%). We use the training and the validation sets to fit and fine-tune the models. Note that fine-tuning is specific to the type of the model, e.g., for a neural network, we fine-tune the learning rate and the regularization constants but for random forests, we fine-tune the number of trees as well as the depth of each tree. Then, we score the test samples with the fine-tuned models and compute the performance metric. We

repeat each experiment 20 times. For each simulated dataset, we implement the following models:

(a) a multivariate logistic regression, i.e., a twin model with no hidden layer that optimizes the Binomial likelihood. This is the baseline model, and we will refer to it as *Logistic*. For fair comparison, we use the twin neural architecture and L_1 regularization.

(b) our twin models, i.e., models (with and without hidden layers) that optimize the uplift loss function (6.3.3) to estimate the regression parameters. We denote these models by $Twin_{\mu}$ and $Twin_{\text{NN}}$.

(c) a Qini-based uplift regression model that uses several LHS structures to search for the optimal parameters (see Belbahri et al. [2021] for more details). We denote this model by *Qini-based*.

(d) two types of random forests designed for causal inference (see Athey et al. [2019] for more details). This method uses honest estimation, i.e., it does not use the same information for the partition of the covariates space and for the estimation of the uplift. We denote these models by *Causal Forest* and *Causal Forest (Honest)* without and with honest estimation respectively.

(e) two types of random forests designed for uplift (see Rzepakowski and Jaroszewicz [2010], Guelman et al. [2012] for more details). This method uses different split criteria. We denote these models by *Uplift Random Forest (KL)* and *Uplift Random Forest (ED)* for Kullback-Leibler and Euclidean split criterion respectively.

(f) two types of R-learners (see Nie and Wager [2020] for more details). This method implements a residual-on-residual regression designed for CATE modeling. We consider two base models that we denote by *R-Learner* (`XGboost`) and *R-Learner* (`lasso`).

(g) two types of X-learners (see Künzel et al. [2019] for more details). This method implements a three-stage regression designed for conditional treatment effect modeling. We consider two base models that we denote by *X-Learner* (`XGboost`) and *X-Learner* (`lasso`).

All results are reported in Table 6.7. If we pick "winners" in each of the simulation scenarios based on which method has the highest $\hat{q}_{\text{adj}}$, $Twin_{\text{NN}}$ would win scenarios 1 and 4. In scenario 2, the *Qini-based*, *R-Learner* (`lasso`) and *X-Learner* (`lasso`) models yield a

higher $\hat{q}_{\text{adj}}$. In general the *Logistic* model is the worst. We observe that the random forests tend to overfit for several scenarios, with the exception of the causal forest with honest estimation which seems to mitigate this problem. In scenario 3, the *R-Learner* (`XGboost`) model has a slightly higher but not significant $\hat{q}_{\text{adj}}$ than our 2-hidden layers $Twin_{\text{NN}}$ model.

Our implementation	Scenarios			
Model (Size: total number of parameters)	1	2	3	4
Logistic $(2p+1)$	0.75	1.80	0.98	2.59
$Twin_{\mu}$ $(2p+1)$	1.04	2.32	1.20	3.12
$Twin_{\text{NN}}$ $((p+1)\times 64+64)$	1.54	2.61	1.22	3.10
$Twin_{\text{NN}}$ $((p+1)\times 128+128)$	1.56	2.54	1.28	3.16
$Twin_{\text{NN}}$ $((p+1)\times(2p+1)+(2p+1))$	1.63	2.65	1.30	3.14
$Twin_{\text{NN}}$ $((p+1)\times 256+256)$	1.57	2.65	1.32	3.26
$Twin_{\text{NN}}$ $((p+1)\times 512+512)$	1.60	2.49	1.18	3.35
$Twin_{\text{NN}}$ with structured sparsity $((p+1)\times 512+512)$	**1.67**	**2.67**	1.35	**3.58**
2-hidden layers $Twin_{\text{NN}}$ $((p+1)\times(p+1)+(p+1)\times(p)+p)$	1.59	2.36	**1.38**	3.39
Open-source implementation				
Qini-based [Belbahri et al., 2021]	**1.02**	2.68	1.09	2.94
Causal Forest [Athey et al., 2019]	0.75	2.22	0.94	2.79
Causal Forest (Honest) [Athey et al., 2019]	0.75	2.51	1.14	**3.07**
Uplift Random Forest (KL) [Guelman et al., 2012]	0.74	2.52	1.01	2.19
Uplift Random Forest (ED) [Guelman et al., 2012]	0.68	2.42	0.99	2.33
R-Learner (`XGboost`) [Nie and Wager, 2020]	0.76	2.63	**1.40**	2.12
R-Learner (`lasso`) [Nie and Wager, 2020]	0.66	2.75	0.87	2.83
X-Learner (`XGboost`) [Künzel et al., 2019]	0.72	2.57	1.31	2.37
X-Learner (`lasso`) [Künzel et al., 2019]	0.77	**2.78**	0.77	2.91

Tab. 6.7. Summary: models comparison in terms of $\hat{q}_{\text{adj}}$ averaged on the test set over 20 runs. Note that the maximum standard-error is 0.15; we do not report them to simplify the Table. The model size indicates the number of parameters to estimate (excluding the bias terms).

Now, looking at our 1-hidden layer $Twin_{\text{NN}}$ with fixed number of hidden neurons, for scenario 1, in which $p = 200$, we see that the best model is reached for $m = 512$; for scenario 2, the model performs best with $m = 201$ or $m = 256$ and with $m = 256$ and $m = 512$ for scenarios 3 and 4. As discussed earlier, the use of the scaling factor penalization on the weight matrix allows to automatically fine-tune the number of hidden neurons (i.e., pruning). This has the effect of increasing the performance of the underlying models from a predictive point of view, as shown in Table 6.7. Finally, in our experiments, increasing the number of hidden layers did not improve the performance significantly. When we fix the initial number of hidden neurons to $p + 1$ and p for the 2 hidden layer, in scenarios 1 and 4, the average $\hat{q}_{\text{adj}}$ are 1.59 and 3.39 respectively.

Computational details

In all of our experiments, we used available implementations of the benchmarks. We used the R4.0.3 libraries **grf** [Tibshirani et al., 2019] and **uplift** [Guelman, 2014] for the random forests implementations. We also used the R4.0.3 library **tools4uplift** [Belbahri et al., 2020] to fit the Qini-based uplift regression. Finally, we used the R4.0.3 library **rlearner** [Nie et al., 2020] to fit the *R*- and *X-learners*. Note that the *Logistic* and *Twin* methods were implemented with **Pytorch1.3.1** in Python3.7.5. Our codes will be made available on GitHub.

Acknowledgements

Mouloud Belbahri and Alejandro Murua were partially funded by The Natural Sciences and Engineering Research Council of Canada grant 2019-05444. Vahid Partovi Nia was supported by the Natural Sciences and Engineering Research Council of Canada grant 418034-2012. Mouloud Belbahri wants to acknowledge Python implementation discussions with Ghaith Kazma and Eyyüb Sari. Mouloud Belbahri and Olivier Gandouet thank Julie Hussin for proof reading an early version of the manuscript and for interesting discussions about potential applications in Bioinformatics.

Bibliography

Farrokh Alemi, Harold Erdman, Igor Griva, and Charles H Evans. Improved statistical methods are needed to advance personalized medicine. *The Open Translational Medicine*

Journal, 1:16, 2009.

Sylvain Arlot, Alain Celisse, et al. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79, 2010.

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

Susan Athey, Julie Tibshirani, Stefan Wager, et al. Generalized random forests. *The Annals of Statistics*, 47(2):1148–1178, 2019.

Amir Beck. *First-order methods in optimization*. SIAM, 2017.

Mouloud Belbahri, Olivier Gandouet, Alejandro Murua, and Vahid Partovi Nia. *tools4uplift: Tools for Uplift Modeling*, 2020. URL `https://CRAN.R-project.org/package=tools4uplift`. R package version 1.0.0.

Mouloud Belbahri, Alejandro Murua, Olivier Gandouet, and Vahid Partovi Nia. Qini-based uplift regression. *The Annals of Applied Statistics*, 2021. to appear.

Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and Regression Trees*. CRC press, 1984.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In *Advances in Neural Information Processing Systems*, pages 737–744, 1994.

Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, et al. Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4), 2015.

Hugh A Chipman, Edward I George, Robert E McCulloch, et al. Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, 2010.

Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.

Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

Richard K Crump, V Joseph Hotz, Guido W Imbens, and Oscar A Mitnik. Nonparametric tests for treatment effect heterogeneity. *The Review of Economics and Statistics*, 90(3): 389–405, 2008.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.

Diemert Eustache, Betlei Artem, Christophe Renaudin, and Amini Massih-Reza. A large scale benchmark for uplift modeling. In *Proceedings of the AdKDD and TargetAd Workshop, KDD, London,United Kingdom, August, 20, 2018*. ACM, 2018.

Bradley Efron and Trevor Hastie. *Computer age statistical inference*, volume 5. Cambridge University Press, 2016.

Jerome Friedman, Trevor Hastie, Holger Höfling, Robert Tibshirani, et al. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

Leo Guelman. *uplift: Uplift Modeling*, 2014. URL `https://CRAN.R-project.org/package=uplift`. R package version 0.3.5.

Leo Guelman, Montserrat Guillén, and Ana M Pérez-Marín. Random forests for uplift modeling: an insurance customer retention case. In *Modeling and Simulation in Engineering, Economics and Management*, pages 123–133. Springer, 2012.

Behram Hansotia and Bradley Rukstales. Direct marketing for multichannel retailers: Issues, challenges and solutions. *Journal of Database Marketing and Customer Strategy Management*, 9(3):259–266, 2001.

Paul W Holland. Statistics and causal inference. *Journal of the American Statistical Association*, 81(396):945–960, 1986.

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.

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

Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

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, 2019.

Andrea Lamont, Michael D Lyons, Thomas Jaki, Elizabeth Stuart, Daniel J Feaster, Kukatharmini Tharmaratnam, Daniel Oberski, Hemant Ishwaran, Dawn K Wilson, and M Lee Van Horn. Identification of predicted individual treatment effects in randomized clinical trials. *Statistical Methods in Medical Research*, 27(1):142–157, 2018.

Tsung-Yi Lin, Yin Cui, Serge Belongie, and James Hays. Learning deep representations for ground-to-aerial geolocalization. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5007–5015, 2015.

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.

Michael D McKay, Richard J Beckman, and William J Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61, 2000.

David Moher, Sally Hopewell, Kenneth F Schulz, Victor Montori, Peter C Gøtzsche, PJ Devereaux, Diana Elbourne, Matthias Egger, and Douglas G Altman. Consort 2010 explanation and elaboration: updated guidelines for reporting parallel group randomised trials. *International Journal of Surgery*, 10(1):28–55, 2012.

Sofia Mosci, Lorenzo Rosasco, Matteo Santoro, Alessandro Verri, and Silvia Villa. Solving structured sparsity regularization with proximal methods. In *Joint European conference on Machine Learning and Knowledge Discovery in Databases*, pages 418–433. Springer, 2010.

Jerzy S Neyman. On the application of probability theory to agricultural experiments. *Annals of Agricultural Sciences*, 10:1–51, 1923.

Xinkun Nie and Stefan Wager. Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika*, 2020.

Xinkun Nie, Alejandro Schuler, and Stefan Wager. *rlearner: R-learner for Heterogeneous Treatment Effect Estimation*, 2020. R package version 1.1.0.

Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. *British Machine Vision Association*, 2015.

Allan Pinkus. Approximation theory of the MLP model in neural networks. *Acta Numerica*, 8(1):143–195, 1999.

Scott Powers, Junyang Qian, Kenneth Jung, Alejandro Schuler, Nigam H Shah, Trevor Hastie, and Robert Tibshirani. Some methods for heterogeneous treatment effect estimation in high dimensions. *Statistics in Medicine*, 37(11):1767–1787, 2018.

Nicholas J Radcliffe and Patrick D Surry. Real-world uplift modelling with significance-based uplift trees. *White Paper TR-2011-1, Stochastic Solutions*, 2011.

NJ Radcliffe. Using control groups to target on predicted lift: Building and assessing uplift models. *Direct Market J Direct Market Assoc Anal Council*, 1:14–21, 2007.

NJ Radcliffe and PD Surry. Differential response analysis: Modeling true response by isolating the effect of a single action. *Credit Scoring and Credit Control VI. Edinburgh, Scotland*, 1999.

Ramchalam Kinattinkara Ramakrishnan, Eyyub Sari, and Vahid Partovi Nia. Differentiable mask for pruning convolutional and recurrent networks. In *2020 17th Conference on Computer and Robot Vision (CRV)*, pages 222–229. IEEE, 2020.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

Peter M Robinson. Root-n-consistent semiparametric regression. *Econometrica: Journal of the Econometric Society*, pages 931–954, 1988.

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, 1983.

Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5):688, 1974.

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

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

Michał Sołtys, Szymon Jaroszewicz, and Piotr Rzepakowski. Ensemble methods for uplift modeling. *Data Mining and Knowledge Discovery*, 29(6):1531–1559, 2015.

Xiaogang Su, Chih-Ling Tsai, Hansheng Wang, David M Nickerson, and Bogong Li. Subgroup analysis via recursive partitioning. *Journal of Machine Learning Research*, 10(Feb): 141–158, 2009.

Xiaogang Su, Joseph Kang, Juanjuan Fan, Richard A Levine, and Xin Yan. Facilitating score and causal inference trees for large observational studies. *Journal of Machine Learning Research*, 13(Oct):2955–2994, 2012.

Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014.

Julie Tibshirani, Susan Athey, and Stefan Wager. *grf: Generalized Random Forests*, 2019. URL `https://CRAN.R-project.org/package=grf`. R package version 0.10.4.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

Michael Tsang, Hanpeng Liu, Sanjay Purushotham, Pavankumar Murali, and Yan Liu. Neural interaction transparency (nit): Disentangling learned interactions for improved interpretability. In *Advances in Neural Information Processing Systems*, pages 5804–5813, 2018.

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.

Frank Wilcoxon. Individual comparisons by ranking methods. In *Breakthroughs in Statistics*, pages 196–202. Springer, 1992.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

Yan Zhao, Xiao Fang, and David Simchi-Levi. A practically competitive and provably consistent algorithm for uplift modeling. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 1171–1176. IEEE, 2017.

Discussion générale

Cette thèse a documenté l'état actuel ainsi que nos contributions à la méthodologie d'apprentissage statistique permettant la modélisation *uplift* - la modélisation du changement de comportement qui résulte spécifiquement d'un traitement ou d'une action telle qu'une intervention de marketing. Nous avons inclus des détails sur les méthodes spécifiques à la modélisation *uplift* actuellement disponibles. Nos contributions sont présentées à travers 3 articles soumis/publiés dans des revues statistiques. Certains des résultats ont été fournis en utilisant la modélisation *uplift* dans la pratique, avec des exemples tirés d'applications et de données réelles de fidélisation de la clientèle. Nous avons eu la rare opportunité de collaborer avec l'industrie afin d'avoir accès à des données provenant de campagnes marketing à grande échelle favorables à l'application de nos méthodes. Ceci nous a permis de guider le développement méthodologique en lien avec les difficultés rencontrées dans l'analyse des données réelles.

Nous avons également étudié et discuté les approches de chacune des étapes majeures impliquées dans la modélisation *uplift* : la sélection de variables, la construction du modèle, l'estimation des paramètres, les mesures de la qualité d'ajustement et l'évaluation de la performance prédictive. Ces étapes nécessitent toutes des approches différentes de la modélisation statistique traditionnelle. En plus de la méthodologie développée, nous avons publié une librairie R incluant des fonctions liées à la plupart de nos méthodes [Belbahri *et al.*, 2020]. Ceci permet aux praticiens de les utiliser avec aise et de se référer aux articles méthodologiques pour les détails.

Dans la plupart des paradigmes de modèles prédictifs, une fonction de perte fixe (souvent construite manuellement) est supposée être un bon proxy pour une mesure d'évaluation sous-jacente du modèle. Nous avons montré dans ce travail que, dans les modèles *uplift*, maximiser la vraisemblance ne permet pas forcément de maximiser le coefficient Qini. Notre

méthodologie, présentée dans le chapitre 3, est basée sur une exploration efficace de l'espace des coefficients de régression. Cette méthode adaptative permet d'optimiser directement la métrique d'évaluation des modèles *uplift*. Nous avons montré empiriquement comment cette formulation améliore l'ajustement des modèles sous-jacents. De plus, notre méthode est applicable à un large éventail de fonctions de perte et de mesures d'évaluation non paramétriques. Nous croyons que la méthode est facilement généralisable pour d'autres tâches prédictives. Par ailleurs, des approches similaires commencent à intéresser les chercheurs. Par exemple, Huang *et al.* [2019] proposent une méthode d'apprentissage par renforcement [Sutton et Barto, 2018] pour adapter la perte de manière dynamique pendant l'entraînement des modèles de classification.

L'inférence causale essaie de répondre à des questions telle que « Quel serait le résultat si nous donnions à ce patient un traitement A au lieu du traitement B? ». La réponse à cette question est ensuite utilisée comme prédiction pour un nouveau patient. Dans la deuxième partie de la thèse, nous avons davantage insisté sur la prédiction. Nous avons présenté un réseau de neurones jumeaux pour l'*uplift* et une nouvelle méthode d'entraînement pour l'inférence causale, facile à mettre en oeuvre; et compatible avec n'importe quelle architecture. Nous avons montré que ce modèle est une généralisation du modèle d'interaction logistique de l'*uplift*. L'estimation des paramètres est basée sur une nouvelle fonction de perte permettant de corriger les moyennes conditionnelles prédites par le score de propension a posteriori. Nous avons également modifié l'algorithme de descente de gradient stochastique pour obtenir des solutions parcimonieuses structurées. Nos expériences démontrent que nos modèles surpassent un certain nombre de méthodes de pointe pour la prédiction de l'effet causal du traitement. Nous partageons ouvertement nos codes Python pour les praticiens désireux de tester nos méthodes du chapitre 6.

Il est possible de généraliser nos méthodes aux données d'observation. En effet, l'une des extensions les plus simples est l'estimation de l'effet causal du traitement en cas d'absence de confusion (ou *unconfoundedness* en anglais). Qualitativement, l'absence de confusion est pertinente lorsqu'on veut estimer l'effet d'un traitement non randomisé, mais aussi bon qu'un traitement aléatoire une fois un ensemble de covariables contrôlé. La démarche peut sembler difficile à mettre en oeuvre en pratique, car elle implique le conditionnement par rapport à des variables aléatoires continues. Cependant, comme le montrent Rosenbaum et

Rubin [1983], elle peut être rendue considérablement plus traitable en considérant le score de propension. Statistiquement, une propriété du score de propension est qu'il s'agit d'un score d'équilibrage : s'il y a absence de confusion, alors il suffit de contrôler $e(\mathbf{x})$ plutôt que les covariables pour éliminer les biais associés à une affectation de traitement non aléatoire.

Le rôle central du score de propension dans l'estimation des effets causaux a été mis en évidence pour la première fois par Rosenbaum et Rubin [1983], tandis que les méthodes d'estimation associées telles que la stratification par score de propension sont discutées dans Rosenbaum et Rubin [1984]. Une autre façon populaire d'exploiter le score de propension dans la pratique est l'appariement de propension (ou *propensity score matching* en anglais), c'est-à-dire l'estimation des effets causaux en comparant des paires d'unités avec des valeurs similaires de $\hat{e}(\mathbf{x})$, l'estimation de $e(\mathbf{x})$. Pour quelques discussions récentes sur l'appariement dans l'inférence causale, voir Abadie et Imbens [2006, 2016] et leurs références. La pondération inversement proportionnelle au score de propension (IPW) est une autre approche simple et transparente de l'estimation de l'effet moyen du traitement en cas d'absence de confusion. Hirano *et al.* [2003] présentent une discussion détaillée sur les propriétés asymptotiques des estimateurs de type IPW. Imbens [2004] donne un aperçu général des méthodes d'estimation de l'effet du traitement en cas d'absence de confusion, y compris une discussion sur les estimations alternatives de l'effet moyen du traitement, comme l'effet moyen du traitement sur les personnes traitées.

Une autre direction que nous aimerions explorer est celle des systèmes de recommandation. En effet, la tâche de ces systèmes de recommandation est classiquement conçue comme une prédiction des préférences des utilisateurs et des évaluations des utilisateurs. Cependant, son esprit est de répondre à une question contrefactuelle : « Quelle serait la note si nous *obligions* l'utilisateur à regarder le film ? » C'est une question sur une intervention, donc d'inférence causale [Wang *et al.*, 2020]. Le principal défi de cette inférence causale réside dans les facteurs de confusion non observés, des variables qui affectent à la fois les éléments avec lesquels les utilisateurs décident d'interagir et la façon dont ils les évaluent. À cette fin, plusieurs chercheurs tentent de développer des algorithmes qui exploitent les modèles de recommandation classiques pour la recommandation causale; voir par exemple Schnabel *et al.* [2016], Bonner et Vasile [2018], Wang et Blei [2019] et leurs références. Le point commun entre ces méthodes est l'utilisation des réseaux de neurones. Nous sommes

d'avis que la méthodologie et l'architecture proposées au chapitre 6 pourraient être adaptées aux systèmes de recommandation. Aussi, les systèmes de recommandation sont très souvent déployés sur des plateformes digitales, par exemple une application de téléphone cellulaire. Les réseaux de neurones profonds (DNN) sont largement utilisés dans de nombreuses applications, notamment pour les systèmes de recommandation. Cependant, leur déploiement sur les téléphones cellulaires a été difficile car ils sont gourmands en ressources. Pour pallier à cette contrainte, nous avons travaillé sur la compression de modèles, plus particulièrement sur les réseaux de neurones binaires (BNN, articles non présentés dans la thèse). Les BNN aident à alléger les besoins prohibitifs en ressources du DNN. Pour ce faire, les activations et les poids sont contraints à deux valeurs possibles (1-bit). Nous avons développé une méthode d'apprentissage binaire statistique (BNN+) [Darabi *et al.*, 2019] et introduit une fonction de régularisation qui encourage l'entraînement des poids autour des valeurs binaires [Nia et Belbahri, 2018]. Afin d'améliorer les performances prédictives du modèle, nous avons également introduit une nouvelle approximation de la dérivée de la fonction signe, basée sur une fonction quasi-convexe étudiée dans Belbahri *et al.* [2019]. Ces ajouts s'appuient sur des opérations faciles à implémenter dans l'entraînement des réseaux de neurones. Ces résultats nous permettront de nous concentrer sur le développement d'un système de recommandation causal, facile à déployer sur des technologies nouvelles.

En conclusion, notre présent travail sur l'inférence causale se veut une modeste contribution à l'interdisciplinarité reliant apprentissage statistique (et/ou réseaux de neurones) et statistique traditionnelle. Plusieurs directions prometteuses en inférence causale n'ont pas été abordées dans cette thèse, notamment l'estimation de l'effet causal dans le contexte de variables médiatrices [Imai *et al.*, 2010] ou l'utilisation de variables instrumentales et de la régression locale pour l'estimation de l'effet causal moyen [Angrist et Imbens, 1995] (ces directions sont plus connues en économétrie). De plus, les méthodes actuelles se limitent généralement à l'estimation/prédiction de l'effet causal, considérant seulement deux traitements disponibles et fixes dans le temps. Toutefois, plusieurs applications considèrent des traitements multiples [VanderWeele et Hernan, 2013, Zhao et Harinen, 2019], ou des traitements dynamiques [Murphy, 2003, Moodie *et al.*, 2007, Heckman *et al.*, 2016]. Enfin, bien que cette thèse se soit concentrée sur une variable réponse binaire, nous comptons adapter nos méthodes aux variables dépendantes continues, ordinales et/ou temporelles.

Bibliographie

Alberto Abadie et Guido W Imbens : Large sample properties of matching estimators for average treatment effects. *Econometrica: Journal of the Econometric Society*, 74(1):235–267, 2006.

Alberto Abadie et Guido W Imbens : Matching on the estimated propensity score. *Econometrica: Journal of the Econometric Society*, 84(2):781–807, 2016.

Joshua Angrist et Guido Imbens : Identification and estimation of local average treatment effects, 1995.

Mouloud Belbahri, Olivier Gandouet, Alejandro Murua et Vahid Partovi Nia : *tools4uplift: Tools for Uplift Modeling*, 2020. URL `https://CRAN.R-project.org/package=tools4uplift`. R package version 1.0.0.

Mouloud Belbahri, Eyyüb Sari, Sajad Darabi et Vahid Partovi Nia : Foothill: A quasiconvex regularization for edge computing of deep neural networks. *In International Conference on Image Analysis and Recognition ICIAR 2019. Lecture Notes in Computer Science*, volume 11663, pages 3–14. Springer, Cham, 2019.

Stephen Bonner et Flavian Vasile : Causal embeddings for recommendation. *In Proceedings of the 12th ACM Conference on Recommender Systems*, pages 104–112, 2018.

Sajad Darabi, Mouloud Belbahri, Matthieu Courbariaux et Vahid Partovi Nia : BNN+: Improved binary network training. *EMC2 Workshop Co-located with the 33rd Conference on Neural Information Processing Systems NeurIPS*, 2019.

James J Heckman, John Eric Humphries et Gregory Veramendi : Dynamic treatment effects. *Journal of Econometrics*, 191(2):276–292, 2016.

Keisuke Hirano, Guido W Imbens et Geert Ridder : Efficient estimation of average treatment effects using the estimated propensity score. *Econometrica*, 71(4):1161–1189, 2003.

Chen Huang, Shuangfei Zhai, Walter Talbott, Miguel Bautista Martin, Shih-Yu Sun, Carlos Guestrin et Josh Susskind : Addressing the loss-metric mismatch with adaptive loss alignment. *In International Conference on Machine Learning*, pages 2891–2900. PMLR, 2019.

Kosuke Imai, Luke Keele et Dustin Tingley : A general approach to causal mediation analysis. *Psychological Methods*, 15(4):309, 2010.

Guido W Imbens : Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and Statistics*, 86(1):4–29, 2004.

Erica EM Moodie, Thomas S Richardson et David A Stephens : Demystifying optimal dynamic treatment regimes. *Biometrics*, 63(2):447–455, 2007.

Susan A Murphy : Optimal dynamic treatment regimes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(2):331–355, 2003.

Vahid Partovi Nia et Mouloud Belbahri : Binary quantizer. *Journal of Computational Vision and Imaging Systems*, 4(1):3–3, 2018.

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

Paul R Rosenbaum et Donald B Rubin : Reducing bias in observational studies using subclassification on the propensity score. *Journal of the American Statistical Association*, 79(387):516–524, 1984.

Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak et Thorsten Joachims : Recommendations as treatments: Debiasing learning and evaluation. *In International Conference on Machine Learning*, pages 1670–1679. PMLR, 2016.

Richard S Sutton et Andrew G Barto : *Reinforcement Learning: An introduction*. MIT press, 2018.

Tyler J VanderWeele et Miguel A Hernan : Causal inference under multiple versions of treatment. *Journal of Causal Inference*, 1(1):1–20, 2013.

Yixin Wang et David M Blei : The blessings of multiple causes. *Journal of the American Statistical Association*, 114(528):1574–1596, 2019.

Yixin Wang, Dawen Liang, Laurent Charlin et David M Blei : Causal inference for recommender systems. *In Fourteenth ACM Conference on Recommender Systems*, pages 426–431, 2020.

Zhenyu Zhao et Totte Harinen : Uplift modeling for multiple treatments with cost optimization. *In 2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 422–431. IEEE, 2019.