Łukasz Zaniewicz

Support Vector Machines for Uplift Modeling

PhD dissertation

Supervisor: dr hab. inż. Szymon Jaroszewicz

Institute of Computer Science
Polish Academy of Sciences
Warsaw, January 2018

Research of the author was supported by Project 'Information technologies: research and their interdisciplinary applications' POKL.04.01.01-00-051/10-00.

Serdecznie dziękuję
Panu dr hab. inż. Szymonowi Jaroszewiczowi
za opiekę naukową, cierpliwość, życzliwość
oraz pomoc podczas pisania pracy.

Dziękuję mojej
kochanej żonie Annie oraz
moim rodzicom i teściom
za wsparcie i wyrozumiałość.

Contents

Abstract

Łukasz Zaniewicz, *Support Vector Machines for Uplift Modeling* Doctoral dissertation supervised by dr hab. inż. Szymon Jaroszewicz, Institute of Computer Science, Polish Academy of Sciences, Warsaw 2018.

Uplift modeling is a branch of Machine Learning which aims to predict not the class itself, but the difference between the class variable behavior in two groups: treatment and control. Objects in the treatment group have been subjected to some action, while objects in the control group have not. By including the control group it is possible to build a model which predicts the *causal* effect of the action for a given individual. As a consequence, uplift modeling is directly applicable in fields where presence of the control group is common in market practice. This is always the case for randomized controlled clinical trials and is becoming a standard in marketing campaigns where it has been realized that, at least in some situations, the true effect of an examined action can only be measured against a background of a control group.

This dissertation presents application of the Support Vector Machines designed specifically for uplift modeling. The SVM optimization task has been reformulated to explicitly model the difference in class behavior between two datasets. The model predicts whether a given object will have a positive, neutral or negative response to a given action, and by tuning a parameter of the model the analyst is able to influence the relative proportion of neutral predictions and thus the conservativeness of the model. Moreover, this work extends also the L_p-SVMs to the case of uplift modeling and demonstrates that this allows for a more stable selection of the size of negative, neutral and positive groups. Furthermore, efficient quadratic and convex optimization methods are presented for efficiently solving the two related optimization tasks. Experiments demonstrate that the proposed methods compare favorably with other uplift modeling approaches.

This dissertation discusses also the issue of nonrandom assignment to treatment and control groups. In general, uplift modeling is best applied to training sets obtained from randomized controlled experiments, but such experiments are not always possible, in which case treatment

assignment is biased. To handle such situations we proposed a modification of the Uplift Support Vector Machines which are less sensitive to such a bias. This is achieved by including in the model formulation an additional term which penalizes models which score treatment and control groups differently. We call the technique Székely regularization since it is based on the energy distance proposed by Székely and Rizzo. Optimization algorithm based on stochastic gradient descent techniques has been developed for this problem. Further, this work demonstrates experimentally that the proposed regularization term does indeed produce uplift models which are less sensitive to biased treatment assignment.

Streszczenie

Łukasz Zaniewicz, *Maszyny wektorów wspierających w modelowaniu różnicowym*. Rozprawa doktorska przygotowana pod kierunkiem dr. hab. inż. Szymona Jaroszewicza, Instytut Podstaw Informatyki Polskiej Akademii Nauk, Warszawa 2018.

Modelowanie różnicowe jest dziedziną uczenia maszynowego, której celem nie jest przewidywanie zmiennej celu, lecz różnic w zachowaniu zmiennej celu między dwoma grupami: eksperymentalną i kontrolną. Obiekty w grupie eksperymentalnej zostały poddane pewnemu działaniu, podczas gdy obiekty w grupie kontrolnej nie. Włączając grupę kontrolną, możliwe staje się zbudowanie modelu przewidującego *przyczynowy* efekt danego działania na poziomie konkretnego obiektu. W konsekwencji modelowanie różnicowe znajduje bezpośrednie zastosowanie w dziedzinach, w których stosowanie grup kontrolnych jest powszechne. Jest tak w przypadku randomizowanych prób klinicznych, staje się też standardem w marketingu bezpośrednim, gdzie zauważono, że, przynajmniej w niektórych sytuacjach, prawdziwą korzyść z badanego działania można zmierzyć tylko biorąc pod uwagę grupę kontrolną.

Niniejsza praca przedstawia maszyny wektorów wspierających zaadaptowane do problemu modelowania różnicowego. Zadanie optymalizacyjne maszyn klasyfikacyjnych zostało przeformułowane, tak aby możliwe było modelowanie różnic w zachowaniu zmiennej celu między dwoma zbiorami danych. Model przewiduje, czy wpływ badanego działania na dany obiekt będzie pozytywny, neutralny czy negatywny. Poprzez odpowiedni dobór parametrów modelu, analityk ma bezpośredni wpływ na względny udział prognoz neutralnych. Dalej, w pracy dostosowano do problemu modelowania różnicowego także model L_p-SVMs, wykorzystujący zmodyfikowaną funkcję straty. Pokazano, że to pozwala na bardziej stabilny wybór proporcji przewidywań negatywnego, neutralnego i pozytywnego wpływu działania. Przedstawiono również wydajne metody optymalizacji kwadratowej i wypukłej służące do rozwiązania problemów optymalizacyjnych związanych z proponowanymi modelami. Eksperymentalnie wykazano, że proponowane algorytmy są konkurencyjne wobec innych metod modelowania różnicowego.

W rozprawie omówiony został również problem nielosowego przypisania obiektów do grupy eksperymentalnej i kontrolnej. Zasadniczo, modelowanie różnicowe najlepiej jest stosować w przypadku przypisania całkowicie losowego, ale takie eksperymenty nie zawsze są możliwe. W praktyce, przydział do grupy kontrolnej często następuje na podstawie decyzji dokonywanych przez człowieka, jest zatem obciążony. Aby możliwe było zastosowanie modelowania różnicowego w takim scenariuszu, w pracy przedstawiono modyfikację proponowanego modelu, która jest mniej wrażliwa na obciążony podział na grupy. Modyfikacja polega na dodaniu do sformułowaniu modelu dodatkowego wyrazu, który karze modele, których predykcje w grupie kontrolnej i eksperymentalnej różnią się. Technikę tę nazywano regularyzacją Székely'ego, ponieważ opiera się ona na odległości energetycznej zaproponowanej przez Székely'ego i Rizzo. Przestawiono również algorytm optymalizacji dla tego typu modeli oparty o techniki stochastycznego spadku gradientu. Ponadto, w pracy wykazano eksperymentalnie, że proponowany dodatkowy wyraz regularyzacyjny istotnie tworzy modele różnicowe, które są mniej wrażliwe na nielosowe przypisanie obiektów do grup eksperymentalnej i kontrolnej.

Chapter 1

Introduction

1.1 Motivation and purpose of the dissertation

The aim of traditional classification methods is to predict the class membership probabilities of new objects based on a given training dataset. In practice, however, a more important question often is how those probabilities change as a result of some action. As an example consider a company which is sending promotional e-mails to customers. Sending an e-mail is beneficial only for customers who buy the product after receiving the offer but would *not* have bought otherwise; otherwise the campaign at best introduces unnecessary costs. However finding such customers requires the knowledge of how receiving the e-mail *changes* their behavior.

Modeling this particular change (or difference) is the scope of *uplift modeling*. It is a subfield of Machine Learning which aims to predict not the outcome of an action itself, but the difference between the outcomes in two groups: *treatment* and *control*. Objects in the treatment group have been subjected to some action, while objects in the control group have not. By including the control group it becomes possible to build a model which predicts the *causal* effect of the action for a given individual and to decide whether it is beneficial, neutral or detrimental. Uplift modeling is directly applicable in case of medical treatments where randomized controlled trails, and the presence of control groups is common. However the technique originated from the field of marketing where it has been realized that the true effect of a direct campaign can only be measured against a background of a control group. Randomized controlled experiments are now frequently used in marketing.

The main difficulty of uplift modeling is that, for a given unit, we only observe one of the outcomes, either after the action has been performed or when the action has not been performed, never both. In statistical literature this problem has been known as the *Fundamental Problem of*

Causal Inference. As a result we never know whether the action was beneficial for a given individual which makes uplift modeling less intuitive and more difficult than traditional classification.

The goal of this dissertation is to establish whether the popular Support Vector Machine framework can be adapted to work in the uplift modeling context. The reason why we have choosen SVMs is that they are very popular and successful models used frequently in machine learning. They are well understood and known to have good predictive accuracy.

To address this goal, in this dissertation we present a few variants of Uplift Support Vector Machines (USVMs) which are an application of the Support Vector Machine methodology to the problem of uplift modeling. The first variant is the most direct adaptation, which allows for predicting whether the action will be detrimental, neutral or beneficial for a given object. Several properties of this model have been rigorously proven, including the upper bounding of an uplift analogue of zero-one loss. A modification has also been introduced which responds more smoothly to changes in model parameters. Finally a variant involving an additional regularization term has been proposed to be applied in situations where treatment assignment is not fully randomized.

1.2 The problem of Uplift Modeling

Uplift modeling is a predictive modeling technique which directly models the change in behaviour resulting from a specified action. Examples of such actions are a controlled medical trial or a direct marketing campaign. The main feature of uplift modeling is the need to have two training datasets. The first one called the *treatment group*, contains data on objects on which the action has been taken. The second one, the *control group*, contains data on objects on which the action has *not* been taken. The main motivation behind uplift modeling is to estimate the effect of such an action, that is to assess whether it will be beneficial or detrimental. The estimate is made against the background of refraining from taking the action.

First, let us briefly discuss the deficiencies of traditional (non uplift) approach, i.e. response modeling. Traditional classification methods predict the conditional class probability distribution based on a model built on a training dataset. In practical applications this dataset often describes individuals on whom some action, such as a marketing campaign or a medical treatment, has been performed. The model is then used to select cases from the general population to which the action should be applied. This approach is, however, usually incorrect. Standard classification methods are only able to model what happens *after* the action has been taken, not what happens *because* of the action. The reason is that such models do not take into account what would have happened had the action not been taken.

Customer	Potential outcomes		uplift
	treatment	control	
Adam	1	0	$+1$
Betty	1	1	0
Cyril	0	0	0
Deborah	0	1	-1

Was targeted?	Observed outcomes		uplift
	treatment	control	
Yes	1	—	$\{+1, 0\}$
No	—	1	$\{0, -1\}$
No	—	0	$\{+1, 0\}$
Yes	0	—	$\{0, -1\}$

Table 1.2.1: Potential (left) and observed (right) outcomes of a direct marketing campaign

This is easiest to see in the context of direct marketing campaigns. Some of the customers who bought after receiving a campaign would have bought anyway, the action incurred unnecessary cost. Worse, some customers who were going to buy got annoyed by the action, refrained from purchase and may even churn. The existence of such 'negative' groups is a well-known phenomenon in the marketing literature [13] and detecting them is often crucial for the success of a campaign.

Uplift modeling, in contrast, allows for the use of an additional control dataset and aims at explicitly modeling the difference in outcome probabilities between the two groups, thus being able to identify cases for which the outcome of the action will be truly positive, neutral or negative. In Section 6.4 we will experimentally compare uplift modeling with traditional classification confirming its superior performance. Moreover, when the assignment to treatment and control groups is random, the model assumes a probabilistic causal interpretation [15], that is, it allows for predicting how class probabilities will change if the action is applied to a given individual. The reason is that, due to randomization, characteristics of both groups are expected to be identical in terms of both observed and latent features, see [15] for a detailed discussion.

The main problem of uplift modeling is that for each data point we know only one of the outcomes, either after the action has been performed or when the action has not been performed, never both. The problem has been known in statistical literature (see e.g. [15]) as the *Fundamental Problem of Causal Inference*. This makes the task less intuitive than standard classification, and formulating optimization tasks becomes significantly more difficult.

To further clarify the differences between classical and uplift modeling, we will consider a simple example stated in terms of the so called *potential outcomes framework* [15]. The framework assumes that for each possible target (customer) there are two *potential* outcomes: one for the case when the customer is targeted (treatment) and the other for the case when the customer is not targeted (control). The outcomes are called *potential* because, due to the Fundamental Problem of Causal Inference they may not be directly observable. The left part of Table 1.2.1 shows potential outcomes for an example marketing campaign (the outcome 1 is considered a

success, 0 a failure). For example Adam, would not have bought the product had he not been targeted, but he would buy the product if he had been a target of the campaign. The fourth column ('uplift') in the left part of the table is the difference between the potential treatment and control outcomes and shows the true gain from performing the action on a given individual. Targeting Adam is truly beneficial, so the value is $+1$.

The second customer in Table 1.2.1, Betty, would have bought the product after the campaign, but was going to buy the product anyway, so the campaign would have had no effect and only incurred unnecessary cost. The third customer, Cyril, would not have bought the product regardless of being targeted or not. From the point of view of a marketer both cases are analogous since there is zero gain from targeting such individuals, as indicated in the fourth column. The fourth customer, Deborah, is quite interesting. She was going to buy the product but the campaign put her off (this is indicated by a -1 in the 'uplift' column). The existence of such cases is well known to marketers [13, 31]. Note that classical modeling, which does not use the control group, cannot tell the difference between Adam and Betty or between Cyril and Deborah.

If both potential outcomes were known to us, we could build a three-valued classifier with the uplift column used as the target variable. Unfortunately, due to the Fundamental Problem of Causal Inference, for each customer only the treatment or the control outcome is known, never both: once a customer has been targeted she cannot be made to forget about the offer received. The situation we encounter in practice is shown in the right part of Table 1.2.1 which shows the data based on which we are supposed to build an uplift model. Notice that for each customer one of the outcomes is unknown; therefore, unlike in case of traditional classification, we do not know the true outcome (i.e. whether the campaign was beneficial, neutral or harmful) for any of the training cases. We are only able to give a *set* of two class values to which a case may belong (depending on the missing outcome) as indicated in the last column of Table 1.2.1. This fact poses challenges for learning and evaluating uplift models.

1.3 Literature overview

Surprisingly, uplift modeling has received relatively little attention in the literature. In this section we give a brief overview of current developments. A more detailed, but somewhat dated overview can be found in [31, 38]. A newer reference is [17].

The most obvious approach to uplift modeling uses two separate probabilistic models, one built on the treatment and the other on the control dataset, and subtracts their predicted probabilities. The advantage of the two-model approach is that it can be applied with any classification

model. Moreover, if uplift is strongly correlated with the class attribute itself, or if the amount of training data is sufficient for the models to predict the class probabilities accurately, the two-model approach will perform very well. The disadvantage is, that when uplift follows a different pattern than the class distributions, both models will focus on predicting the class, instead of focusing on the weaker 'uplift signal'. An illustrative example has been presented in [31]. In the example, the class variable in both groups was strongly influenced by one variable, while the effect of the action was weakly influenced by a second one. A double decision tree model failed to take the second variable into account. Most research on uplift modeling has thus concentrated on building dedicated models which predict uplift directly.

A few papers addressed decision tree construction for uplift modeling. See e.g. [6, 13, 30, 31, 38, 39]. Those approaches build a single uplift tree by simultaneously splitting the two training datasets based on modified test selection criteria. The criteria typically aim at maximizing differences between treatment and control probabilities after the split. For example, in [31] the authors use a criterion based on a statistical test on the interaction term between the treatment indicator and the split variable in a linear model predicting the response. In [38] uplift decision trees have been presented which are more in line with modern machine learning algorithms. Splitting criteria are based on information theoretical measures such the Kullback-Leibler divergence. A dedicated pruning strategy is also presented. The approach has been extended to the case of multiple treatments in [39].

As is the case in classical machine learning, uplift decision trees can be combined into ensembles. Uplift random forests which use ensembles of trees from [38, 39] with splitting criteria modified to include extra randomization have been described in [12]. A thorough analysis of various types of ensembles in the uplift setting can be found in [43]. The comparison includes bagging and random forests. It is noted that bagging performs very well in the uplift setting, often giving very significant improvements in performance. Some theoretical justification for good performance of uplift ensembles is also provided.

Some regression techniques for uplift modeling have been proposed. Most researchers follow the two model approach either explicitly or implicitly [25, 26] by including interactions between the treatment indicator and the predictor variables. Some dedicated approaches are also available, most notably g-estimation [33, 34, 48]. This approach is, however, not statistically efficient. Its efficiency can be improved by, in fact, changing it into the double model approach.

In [20] a method has been presented which makes it possible to convert a classical logistic regression model (or in fact any other probabilistic classifier) into an uplift model. The approach is based on a simple class variable transformation. The transformation reverts the class variable

in the control group and reweighs data records such that the weight of records in treatment and control groups becomes equal. The two groups are then concatenated and a single probabilistic classifier is built on the combined dataset. Any such classifier can thus easily be converted into an uplift model. Support Vector Machines used with this transformation are included in our experiments.

Recently, [27] extended the approach to work in the context of online advertising, where it is necessary to not only maximize uplift (the difference between success rate in the treatment and control datasets) but also to increase advertiser's gains through maximizing response. This type of problems are beyond the scope of this dissertation.

Uplift Support Vector Machines proposed in [49] were to the best of our knowledge the first adaptation of the framework to uplift modeling. Later, another type of uplift Support Vector Machines was proposed in [24]. The approach is based on direct maximization of the area under the uplift curve. The authors proceed by noticing a direct relationship between area under the ROC curve and the area under the cumulative gains curve. The connection is then used together with the SVM struct algorithm [47] to obtain an algorithm which maximizes the desired quantity. Experimental comparison with our approaches is given in Section 6.4.

Support Vector Machines with parallel hyperplanes, similar to our approach, have been analyzed in the context of ordinal classification [40]. The situation analyzed in this dissertation is, however, different since two training datasets are involved.

We now list the publications in which Uplift Support Vector Machines described in this dissertation have first appeared. Our first proposed variant of the USVMs has appeared in [49]. A following paper [50] significantly extends that first version. The most important addition is the practical and theoretical demonstration of discontinuity problems with USVMs and the introduction of L_p Uplift Support Vector Machines which do not suffer from such problems. The second novel contribution is the development of improved optimization algorithms based on convex and quadratic programming techniques and efficient solutions to structured Karush-Kuhn-Tucker (KKT) systems. Finally it has been proven that USVMs minimize an upper bound of an uplift analogue of zero-one loss.

In [19] the Uplift SVM approach has been extended to the case of nonrandom treatment-control group assignment. An additional regularization term has been added which enforces similar model behavior in both groups. The problem of nonrandom treatment assignment is presented in Chapter 5.

1.4 Contributions

We now briefly summarize the contributions of this dissertation.

In this dissertation we present Uplift Support Vector Machines (USVMs) which are an application of the SVM methodology to the problem of uplift modeling. The SVM optimization problem has been reformulated such that the machine accepts two training datasets: treatment and control, and models the differences in class behavior between those sets. While other uplift modeling methods return the score of an instance; USVMs are the first such method we are aware of, which aims to explicitly predict whether an outcome of an action for a given case will be positive, negative or neutral. What is especially important is that the model identifies the negative group allowing for minimizing the adverse impact of the action. Moreover, by proper choice of parameters, the analyst is able to decide on the relative proportion of neutral predictions, adjusting model's confidence in predicting positive and negative cases. Moreover, we have proved that Uplift Support Vector Machines minimize an upper bound on an uplift analogue of the 0-1 loss and have shown and proven several other interesting Uplift SVM properties. We have also developed optimization algorithms based on convex and quadratic programming techniques and efficient solutions to solve the related quadratic optimization problems.

Further, we demonstrate theoretically and experimentally, that USVMs may, in some cases, suffer from a problem of very abrupt changes in predictions in response to tiny changes in parameter values. In the most extreme case, predictions for *all* data points may simultaneously change from neutral to positive or negative. An adaptation of L_p-Support Vector Machines [1, 8] to the uplift modeling problem is then described. Those models are not susceptible to such discontinuities.

Finally, we propose a novel approach to remedy a situation when the assignment to treatment and control groups is not random. This is achieved by adding an additional regularizer term based on the so called energy distance between probability distributions.

1.5 Outline of the dissertation

The dissertation is organized as follows. In Chapter 1 we present the problem of Uplift Modeling and include an overview of literature related to this topic. Next, in Chapter 2 we give a short description of classical Support Vector Machines. In Chapter 3 we introduce and formally define the L_1 and L_p Uplift Support Vector Machines (USVMs), describe their respective optimization tasks and analyze their properties. Later, in Chapter 4 we provide concrete optimization algorithms

for the proposed models. Chapter 5 addresses the problem of biased treatment selection and introduces a specially designed regularizer for this case. Chapter 6 contains results of experimental evaluation of the proposed methods and Chapter 7 concludes the dissertation.

Chapter 2

Support Vector Machines for classification

The origins of the famous Support Vector Machine algorithm date back to the late 50s [37] but the current and nowadays widely used formulation was introduced by Vapnik and Cortes in the 90s [7]. It is one of the most popular supervised learning algorithms, with applications in multiple areas. The main idea behind the SVMs has been extended to e.g. regression analysis (Support Vector Regression) [42] and clustering (Support Vector Clustering) [4]. In this dissertation we do not cover those topics and focus entirely on SVM used as a classifier and in further chapters we present the adaptation of the SVM methodology to the uplift modeling problem. A more detailed exposition on Support Vector Machines can be found in [41].

In this chapter we consider only the binary (two-class) classification problem. Let us consider the training dataset of the form $\mathbf{D} = \{(\mathbf{x}_i, y_i) : i = 1, \ldots, n\}$, where $\mathbf{x}_i \in \mathbb{R}^m$ are the values of the predictor variables, and $y_i \in \{-1, 1\}$ is the class label of the $i-$th data record. By $\langle \mathbf{x}_1, \mathbf{x}_2 \rangle$ we denote the scalar product of vectors $\mathbf{x}_1$ and $\mathbf{x}_2$.

Binary classification methods are often presented in terms of a real-valued function f. The classifier takes an input vector $\mathbf{x}$ and assigns it to the positive class $(+1)$ if $f(\mathbf{x}) \geq 0$ and otherwise to the negative class (-1). In a linear model, f can be written in the form

$$f(x) = \sum_{i=1}^{m} w_i x_i - b = \langle \mathbf{w}, \mathbf{x} \rangle - b, \tag{2.0.1}$$

where $\mathbf{w} \in \mathbb{R}^m$ is the weight vector and $b \in \mathbb{R}$ the intercept. The decision is then made depending on $\operatorname{sgn}(f(\mathbf{x}))$. We assume that $\operatorname{sgn}(0) = +1$.

The geometric intuition behind a linear classifier is that the hyperplane given by the equation $\langle \mathbf{w}, \mathbf{x} \rangle - b = 0$ splits the input space into two parts, each corresponding to one of the classes. The weight vector $\mathbf{w}$ is perpendicular to the hyperplane and defines its direction. The intercept b decides how far the hyperplane is from the origin.

2.1 The linearly separable case

Let now assume that our training dataset is *linearly separable*, that is, there exists a hyperplane with all of the points from class $+1$ on one side of the hyperplane and all the class -1 points on the other side. If no such hyperplane exists, we say that the dataset is *linearly non-separable*. We begin by describing Support Vector Machines for the linearly separable case, the non-separable case will be presented in the next section.

The Support Vector Machine framework is based on the concept of a *margin*:

Definition 1. The *functional margin* of an observation $(\mathbf{x}_i, y_i)$ with respect to a hyperplane $H : \langle \mathbf{w}, \mathbf{x} \rangle - b = 0$ is the quantity

$$\gamma_i^H = y_i \left(\langle \mathbf{w}, \mathbf{x}_i \rangle - b \right). \tag{2.1.1}$$

Taking the minimum of γ_i^H over all training examples we define the (functional) *margin of a hyperplane* H:

$$\gamma^H = \min_{i=1,\ldots,n} \{\gamma_i^H\}.$$

When the weight vector $\mathbf{w}$ is normalized, i.e. $\|\mathbf{w}\| = 1$, γ_i^H is the Euclidean distance of the point $\mathbf{x}_i$ from H. In this case we call it a *geometric margin* of an observation $(\mathbf{x}_i, y_i)$ with respect to a hyperplane H and denote it $\gamma_i^{H,G}$. By taking the minimum over the whole training sample $\mathbf{D}$ we get $\gamma^{H,G}$ the geometric margin of the hyperplane H.

Finally, we call the maximum geometric margin over all hyperplanes, the *geometric margin of a dataset* $\mathbf{D}$ and define the *maximum margin hyperplane* as a hyperplane realising this maximum.

Learning a Support Vector Machine means finding a maximum margin hyperplane which separates the positive examples from the negative examples. We expect that such a model should be more resistant to noise and reduce the risk of misclassification compared to models with smaller margins. We will now describe the corresponding optimization problem.

Note that the decision function f does not change if we multiply it by a constant $\lambda > 0$. Such scaling will affect the functional margin, but the geometric margin remains unchanged. Hence, while optimizing the geometric margin, we can fix the functional margin to be equal to 1 without

any loss of generality. The condition that there exists hyperplane H with a functional margin at least 1 can be expressed using the following constraints

$$\begin{cases} \langle \mathbf{w}, \mathbf{x}_i \rangle - b \geq 1 & \text{if } y_i = +1, \quad i = 1, \ldots, n, \\ \langle \mathbf{w}, \mathbf{x}_i \rangle - b \leq -1 & \text{if } y_i = -1, \quad i = 1, \ldots, n, \end{cases} \tag{2.1.2}$$

which can be simplified to

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle - b) \geq 1, \qquad i = 1, \ldots, n. \tag{2.1.3}$$

The data points for which the condition becomes an equality are called the *support vectors*, their margin is equal to that of the dataset $\mathbf{D}$.

Dividing both sides of Equation 2.1.1 by $\|\mathbf{w}\|$ we get

$$\frac{\gamma_i^H}{\|\mathbf{w}\|} = y_i \left(\langle \frac{\mathbf{w}}{\|\mathbf{w}\|}, \mathbf{x}_i \rangle - \frac{b}{\|\mathbf{w}\|} \right) = \gamma_i^{H,G}$$

and using the fact $\gamma^H = 1$ the geometric margin of a hyperplane H is simply $\frac{1}{\|\mathbf{w}\|}$.

Finding the maximum margin separating hyperplane is thus equivalent to solving the following constrained optimization task (the factor $\frac{1}{2}$ is used for convenience)

$$\min \frac{1}{2} \|\mathbf{w}\|^2 \tag{2.1.4}$$

subject to

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle - b) \geq 1, \qquad i = 1, \ldots, n. \tag{2.1.5}$$

This is a quadratic optimization problem with linear constraints. We will not discuss this problem in detail and will, instead, focus on the more practically important problem of classifying linearly non-separable data.

2.2 The linearly non-separable case

Linear separability is a nice theoretical concept but, in practice, this assumption is usually not satisfied. In order to handle linearly non-separable datasets, a modification of the SVM formulation is needed. The inventors of Support Vector Machines modified the linear constraints given in Equation 2.1.5 by introducing *slack variables* ξ_i which allow for misclassification of the training

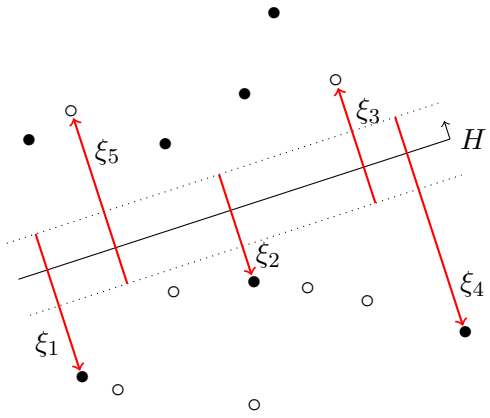

Figure 2.2.1: Illustration of SVM problem on an artificial linearly non-separable example. Points belonging to the classes $+1$ and -1 are marked respectively by the filled and empty circles. Red arrows denote the penalties ξ_i for the missclassified points.

examples [7]. The new constraints are

$$\langle \mathbf{w}, \mathbf{x} \rangle - b \geq 1 - \xi_i, \text{ when } y_i = +1,$$
$$\langle \mathbf{w}, \mathbf{x} \rangle - b \leq -1 + \xi_i, \text{ when } y_i = -1,$$

where $\xi_i \in \mathbb{R}, \xi_i \geq 0, i = 1, \ldots, n$. The modified optimization problem is

$$\min \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{n} \xi_i \tag{2.2.1}$$

subject to

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle - b) \geq 1 - \xi_i, \qquad i = 1, \ldots, n, \tag{2.2.2}$$
$$\xi_i \geq 0, \qquad i = 1, \ldots, n. \tag{2.2.3}$$

The constant C decides on the relative importance of margin width and correct classification of as many training points as possible.

Figure 2.2.1 illustrates the slack variables $\xi_i, i = 1, \ldots, n$ graphically. The solid line is the hyperplane H and the dashed lines represent the functional margin of 1.

The values of the slack variables can be interpreted as the amount by which the constraints need to be modified in order not to be violated. Moreover, since in the formulation we use

a separating hyperplane with a functional margin equal to 1, $\xi_i - 1$ can be interpreted as a scaled distance of $\mathbf{x}_i$ from the decision boundary, given that $\mathbf{x}_i$ is misclassified. More precisely, a misclassification occurs when $\xi_i > 1$. Therefore, the sum of the slack variables ξ_i is an upper bound on the total number of misclassified points (from the training dataset).

2.3 The SVM optimization problem

Before we discuss methods of solving the optimization problem given above in more detail, we will introduce some basic notions of optimization theory, namely the so called KKT conditions.

2.3.1 Karush-Kuhn-Tucker conditions

We now present the Karush-Kuhn-Tucker (KKT) conditions for optimality of a solution to a constrained optimization problem which will be used throughout the dissertation. We only focus on aspects of the KKT theory which are relevant to the optimization problems encountered in this work. A more general description can be found in [5].

Consider a minimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^m} f(\mathbf{x}) \tag{2.3.1}$$

subject to

$$h_i(\mathbf{x}) \leq 0, \qquad i = 1, \ldots, k, \tag{2.3.2}$$

where f is a convex objective or goal function and h_i are linear inequality constraint functions. The following theorem states the necessary and sufficient conditions for a point $\mathbf{x}^*$ to be a solution of this optimization task.

Theorem 2.3.1. Suppose that the minimized function $f : \mathbb{R}^m \to \mathbb{R}$ is convex and continuously differentiable at $\mathbf{x}^*$, and the inequality constraint functions $h_i : \mathbb{R}^m \to \mathbb{R}$ are linear. A point $\mathbf{x}^*$ is a solution to the optimization problem given in Equations 2.3.1 and 2.3.2 if and only if there exist constants λ_i, $i = 1, \ldots, k$ called Lagrange (or KKT) multipliers satisfying the following conditions

- stationarity $\left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^*} - \sum_{i=1}^{k} \lambda_i \left.\frac{\partial h_i}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^*} = 0,$
- primal feasibility $h_i(\mathbf{x}^*) \leq 0, \quad \text{for } i = 1, \ldots, k,$
- dual feasibility $\lambda_i \geq 0, \quad \text{for } i = 1, \ldots, k,$
- complementary slackness $\lambda_i h_i(\mathbf{x}^*) = 0, \quad \text{for } i = 1, \ldots, k.$

The proof of the theorem can be found in [5].

2.3.2 Support Vector Machines in the non-separable case: the optimization problem

To solve the optimization problem given in Equations 2.2.1–2.2.3, one can apply the method of Lagrange multipliers and the KKT conditions. The Lagrangian is

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i \left\{ y_i \left(\langle \mathbf{w}, \mathbf{x}_i \rangle - b \right) - 1 + \xi_i \right\} - \sum_{i=1}^{n} r_i \xi_i, \tag{2.3.3}$$

where $\alpha_i \geq 0$ and $r_i \geq 0$ are Lagrange multipliers and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$. We first apply the Karush-Kuhn-Tucker stationarity condition to L and obtain

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0, \tag{2.3.4}$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{n} \alpha_i y_i = 0, \tag{2.3.5}$$

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - r_i = 0, \qquad i = 1, \ldots, n. \tag{2.3.6}$$

From the first equation above we get

$$\mathbf{w} = \sum_{i=1}^{n} y_i \alpha_i \mathbf{x}_i. \tag{2.3.7}$$

Using the above equations, the Lagrangian can be rewritten as

$$L(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle. \tag{2.3.8}$$

This is the dual form of the SVM optimization problem. To solve it, we need to maximize the above quantity subject to the following constraints

$$\sum_{i=1}^{n} y_i \alpha_i = 0 \qquad \text{and} \qquad \alpha_i \geq 0, \quad i = 1, \ldots, n. \tag{2.3.9}$$

From the complementary slackness KKT condition of the original problem we get

$$\alpha_i \left\{ y_i \left(\langle \mathbf{w}, \mathbf{x}_i \rangle - b \right) - 1 + \xi_i \right\} = 0 \qquad i = 1, \ldots, n. \tag{2.3.10}$$

From this equation and the fact that $\xi_i \geq 0$, we see that non-zero multipliers α_i correspond to support vectors, since they must satisfy $y_i\left(\langle \mathbf{w}, \mathbf{x}_i\rangle - b\right) = 1 - \xi_i$. Moreover, from the condition $C - \alpha_i - r_i = 0$ together with $r_i \geq 0$, we get $\alpha_i \leq C$. Notice also that $\xi_i > 0$ only if $r_i = 0$ in which case $\alpha_i = C$.

The solution of the dual optimization problem is a vector

$$\boldsymbol{\alpha}^* = (\alpha_1^*, \ldots, \alpha_n^*), \tag{2.3.11}$$

which gives us the desired optimal separating hyperplane

$$\sum_{i=1}^{n} y_i \alpha_i^* \langle \mathbf{x}_i, \mathbf{x}\rangle - b^* = 0. \tag{2.3.12}$$

The above sum in fact involves only indices of examples which are support vectors. The intercept b^* is, however, not obtained directly by maximization of L. We can calculate it, for example, by solving Equation 2.3.10 for any of the support vectors.

2.4 Nonlinear SVM using the kernel trick

In the previous section we presented a short description of the linear Support Vector Machines. However, one of the reasons the SVM method gained its popularity is the possibility to perform nonlinear classification via the use of the so-called *kernel trick*. Here we present only a short, general overview of nonlinear SVMs, for details see [7, 41].

Notice that in the dual SVM formulation the separating hyperplane depends on training data points $\mathbf{x}_i, \mathbf{x}_j$ only via scalar products $\langle \mathbf{x}_i, \mathbf{x}_j\rangle$. One way of obtaining a nonlinear classifier is by mapping the input vectors $\mathbf{x}_i$, $\mathbf{x}_j$ to a new, higher dimensional space using a nonlinear transformation Φ. The scalar products are then computed in the new space. It turns out that the function $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)\rangle$ can frequently be computed without performing the actual mapping. As a result, the target space may even have an infinite number of dimensions. This concept is known as the *kernel trick*.

The Uplift Support Vector Machines described in this dissertation can also use the kernel trick. However, the main focus of this work are linear models so we do not address this issue in later chapters. Moreover, experiments we have performed did not show significant improvements from using nonlinear kernels on benchmark datasets available to us.

Chapter 3

Uplift Support Vector Machines

In this chapter we will present Uplift Support Vector Machines (USVMs) which are the main contribution of this dissertation.

3.1 Uplift Support Vector Machines

Let us first introduce the necessary notation and formally define Uplift Support Vector Machines (USVMs). The class +1 will be considered the *positive*, or desired outcome. The scalar product of vectors $\mathbf{x}_1$, $\mathbf{x}_2$ will be denoted with $\langle \mathbf{x}_1, \mathbf{x}_2 \rangle$. Here, and in the remaining part of the dissertation we will continue to follow the convention that all quantities related to the treatment group will be denoted with superscript ^{T} and those related to the control group with superscript ^{C}.

Unlike standard classification, in uplift modeling we have two training samples: the *treatment group*, $\mathbf{D}^T = \{(\mathbf{x}_i, y_i) : i = 1, \ldots, n^T\}$ and the *control group* $\mathbf{D}^C = \{(\mathbf{x}_i, y_i) : i = 1, \ldots, n^C\}$, where $\mathbf{x}_i \in \mathbb{R}^m$ are the values of the predictor variables, and $y_i \in \{-1, 1\}$ is the class of the i-th data record, m is the number of variables in the data, and n^T and n^C are the numbers of records in the treatment and control groups respectively. Objects in the treatment group have been subjected to some *action* or *treatment*, while objects in the control group have not.

SVMs are designed primarily for classification, not probability modeling, so in order to adapt SVMs to the analyzed setting we first recast the uplift modeling problem as a three-class classification problem. This differs from the typical formulation which aims at predicting the difference in class probabilities between treatment and control groups. An *uplift model* is defined as a function

$$M(\mathbf{x}) : \mathbb{R}^m \to \{-1, 0, +1\}, \tag{3.1.1}$$

which assigns to each point in the input space one of the values $+1$, 0 and -1, interpreted, respectively, as positive, neutral and negative impact of the action. In other words, the positive prediction $+1$ means that we expect the object's class to be $+1$ if it is subjected to treatment and -1 if it is not, the negative prediction means that we expect the class to be -1 after treatment and $+1$ if it was not performed, and neutral if the object's class is identical (either $+1$ or -1) regardless of whether the action was taken or not.

The proposed Uplift Support Vector Machine (USVM), which performs uplift prediction, uses two parallel hyperplanes

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

where $\mathbf{w} \in \mathbb{R}^m$ is the weight vector and $b_1, b_2 \in \mathbb{R}$ are the intercepts. The model predictions are specified by the following equation

$$M(\mathbf{x}) = \begin{cases} +1 & \text{if } \langle \mathbf{w}, \mathbf{x} \rangle > b_1 \text{ and } \langle \mathbf{w}, \mathbf{x} \rangle > b_2, \\ 0 & \text{if } \langle \mathbf{w}, \mathbf{x} \rangle \leq b_1 \text{ and } \langle \mathbf{w}, \mathbf{x} \rangle > b_2, \\ -1 & \text{if } \langle \mathbf{w}, \mathbf{x} \rangle \leq b_1 \text{ and } \langle \mathbf{w}, \mathbf{x} \rangle \leq b_2. \end{cases} \tag{3.1.2}$$

Intuitively, the point is classified as positive if it lies on the positive side of both hyperplanes, neutral if it lies on the positive side of hyperplane H_2 only, and classified as negative if it lies on the negative side of both hyperplanes. In other words, H_1 separates positive points from neutral points, and H_2 neutral points from negative points. Notice that the model is valid if and only if $b_1 \geq b_2$; in Lemmas 3.3.1 and 3.3.3 we will give sufficient conditions for this inequality to hold.

Let us now formulate the optimization task which allows for finding the model's parameters $\mathbf{w}, b_1, b_2$. We use $\mathbf{D}_+^T = \{(\mathbf{x}_i, y_i) \in \mathbf{D}^T : y_i = +1\}$ to denote treatment data points belonging to the positive class and $\mathbf{D}_-^T = \{(\mathbf{x}_i, y_i) \in \mathbf{D}^T : y_i = -1\}$ to denote treatment data points belonging to the negative class. Analogous notation is used for points in the control group. Denote $n = |\mathbf{D}^T| + |\mathbf{D}^C|$.

The parameters of an USVM can be found by solving the following optimization problem,

which we call the *USVM optimization problem.*

$$\min_{\mathbf{w},b_1,b_2\in\mathbb{R}^{m+2}} \frac{1}{2}\langle \mathbf{w},\mathbf{w}\rangle + C_1 \sum_{\mathbf{D}_+^T\cup\mathbf{D}_-^C} \xi_{i,1} + C_2 \sum_{\mathbf{D}_-^T\cup\mathbf{D}_+^C} \xi_{i,1} + C_2 \sum_{\mathbf{D}_+^T\cup\mathbf{D}_-^C} \xi_{i,2} + C_1 \sum_{\mathbf{D}_-^T\cup\mathbf{D}_+^C} \xi_{i,2} \tag{3.1.3}$$

subject to the following constraints

$$\langle \mathbf{w},\mathbf{x}_i\rangle - b_1 \geq +1 - \xi_{i,1}, \text{ forall } (\mathbf{x}_i,y_i)\in \mathbf{D}_+^T\cup\mathbf{D}_-^C, \tag{3.1.4}$$

$$\langle \mathbf{w},\mathbf{x}_i\rangle - b_1 \leq -1 + \xi_{i,1}, \text{ forall } (\mathbf{x}_i,y_i)\in \mathbf{D}_-^T\cup\mathbf{D}_+^C, \tag{3.1.5}$$

$$\langle \mathbf{w},\mathbf{x}_i\rangle - b_2 \geq +1 - \xi_{i,2}, \text{ forall } (\mathbf{x}_i,y_i)\in \mathbf{D}_+^T\cup\mathbf{D}_-^C, \tag{3.1.6}$$

$$\langle \mathbf{w},\mathbf{x}_i\rangle - b_2 \leq -1 + \xi_{i,2}, \text{ forall } (\mathbf{x}_i,y_i)\in \mathbf{D}_-^T\cup\mathbf{D}_+^C, \tag{3.1.7}$$

$$\xi_{i,j} \geq 0, \text{ forall } i=1,\ldots,n,\ j\in\{1,2\}, \tag{3.1.8}$$

where C_1, C_2 are penalty parameters and $\xi_{i,j}$ slack variables allowing for misclassified training cases. Note that $\xi_{i,1}$ and $\xi_{i,2}$ are slack variables related respectively to the hyperplane H_1 and H_2. We will now give an intuitive justification for this formulation of the optimization problem, later we formally prove that the USVM minimizes an upper bound on an uplift specific loss function.

Below, when we talk about 'distance of a point from a hyperplane' and 'a point lying on a positive or negative side of a hyperplane' we implicitly assume that the width of the margin is also taken into account.

The situation is graphically depicted in Figure 3.1.1. Sample points belonging to $\mathbf{D}_+^T$ are marked with T_+, points belonging to $\mathbf{D}_-^T$, respectively with T_-. Analogous notation is used for example points in the control group which are marked with C_+ and C_-. The points and hyperplane locations are hand picked to illustrate the USVM penalties.

In an ideal situation, points for which a positive (+1) prediction is made include only cases in $\mathbf{D}_+^T$ and $\mathbf{D}_-^C$, that is points which do not contradict the positive effect of the action (see the first row of Table 1.2.1). Note that for the remaining points, which are in $\mathbf{D}_-^T$ or in $\mathbf{D}_+^C$, the effect of an action can at best be neutral[1]. Therefore points in $\mathbf{D}_+^T$ and $\mathbf{D}_-^C$ (marked T_+ and C_- respectively in the figure) are not penalized when on the positive side of hyperplane H_1. Analogously points in $\mathbf{D}_-^T$ and $\mathbf{D}_+^C$ (marked T_- and C_+) which are on the negative side of H_2 are not penalized.

Points in $\mathbf{D}_+^T$ and $\mathbf{D}_-^C$ which lie on the negative side of H_1 are penalized with penalty $C_1\xi_{i,1}$,

[1]Recall from Section 1.2 that the true gain from performing an action on a specific case is unknown to us and see the last column of Table 1.2.1.

where $\xi_{i,1}$ is the distance of the point from the plane and C_1 is a penalty coefficient. Those penalties prevent the model from being overly cautious and classifying all points as neutral (see Lemmas 3.3.2 and 3.3.3 in the next section). An analogous penalty is introduced for points in $\mathbf{D}_-^T$ and $\mathbf{D}_+^C$ in the fifth term of (3.1.3). In Figure 3.1.1, those points are sandwiched between H_1 and H_2, and their penalties are marked with solid red arrows.

Consider now points in $\mathbf{D}_+^T$ and $\mathbf{D}_-^C$ which lie on the negative side of both hyperplanes, i.e. in the region where the model predicts a negative impact (-1). Clearly, model's predictions are wrong in this case, since, if the outcome was positive in the treatment group, the impact of the action can only be positive or neutral (see the last column of Table 1.2.1). Those data points are thus additionally penalized for being on the wrong side of the hyperplane H_2 with penalty $C_2\xi_{i,2}$. Analogous penalty is of course applied to points in $\mathbf{D}_-^T$ and $\mathbf{D}_+^C$ which lie on the positive side of both hyperplanes. Such additional penalties are marked with dashed blue arrows in the figure.

To summarize, the penalty coefficient C_1 is used to punish points being on the wrong side of a single hyperplane (solid red arrows in Figure 3.1.1) and the coefficient C_2 controls additional penalty incurred by a point being on the wrong side of also the second hyperplane (dashed blue arrows in Figure 3.1.1). In the next section we give a more detailed analysis of how the penalties influence the model's behavior.

We now present a more formal analysis of the quantity optimized by an USVM. We begin by defining an analogue of the 0-1 loss function for uplift modeling. Let y^T and y^C denote the respective potential outcomes after a given individual received the treatment and was left as a control; denote by $u = y^T - y^C$ the true gain from performing the action on a given individual. Let $g \in \{T, C\}$ be the group to which the individual is assigned (respectively treatment or control). Further, let $a \in \{-1, 0, +1\}$ be the prediction of the model.

We define the *true uplift loss* as

$$l(y^T, y^C, a) = \begin{cases} -u & \text{if } a = +1, \\ u & \text{if } a = -1, \\ 0 & \text{if } a = 0 \text{ and } u = 0, \\ \rho & \text{otherwise,} \end{cases} \tag{3.1.9}$$

where $0 \leq \rho \leq 1$ is a constant. To make the loss easier to understand the following table summarizes its values depending on the model prediction a and the true gain u for a given individual.

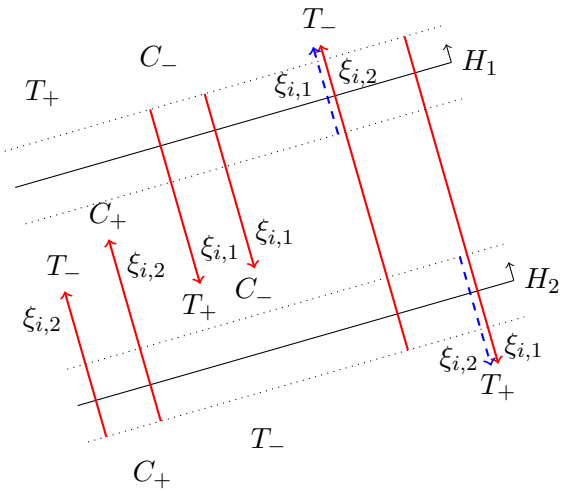

Figure 3.1.1: The Uplift SVM optimization problem. Example points belonging to the positive class in the treatment and control groups are marked respectively with T_+ and C_+. Analogous notation is used for points in the negative class. The figure shows penalties incurred by points with respect to the two hyperplanes of the USVM. Positive sides of hyperplanes are indicated in the image by small arrows at the right ends of lines. Red solid arrows denote the penalties incurred by points which lie on the wrong side of a single hyperplane, blue dashed arrows denote additional penalties for being misclassified also by the second hyperplane.

	$u=-1$	$u=0$	$u=1$
$a=+1$	1	0	-1
$a=0$	ρ	0	ρ
$a=-1$	-1	0	1

For example, when the model suggests treating an individual ($a = +1$) but the true gain is negative, the loss is 1. If, on the other hand, the true gain is $u = +1$ the loss is -1 indicating that we actually gained from performing the treatment. The constant ρ penalizes neutral predictions when the true gain is not zero. Since wrongly classifying a case as neutral is potentially less harmful than wrongly recommending treatment, ρ will typically be less than 1.

Notice that computing $l(y^T, y^C, a)$ requires the knowledge of both potential outcomes, so due to the Fundamental Problem of Causal Inference (see Section 1.2) it is not possible in practice. We can, however, optimize an upper bound on it as shown in the following theorem.

Theorem 3.1.1. The quantity optimized in the USVM optimization task given in Equation 3.1.3 is an upper bound on the sum of the true uplift loss l over all training records in $\mathbf{D}^C$ and $\mathbf{D}^T$.

Proof. Let y be the actual outcome observed, i.e. y^T if the object was treated and y^C otherwise. Define an auxiliary loss function

$$\tilde{l}(y, g, a) = \begin{cases} \max_{y^C} l(y, y^C, a) & \text{if } g = T, \\ \max_{y^T} l(y^T, y, a) & \text{if } g = C. \end{cases}$$

It is clear that the unknown true uplift loss $l(y^T, y^C, a)$ is upper-bounded by the auxilliary loss $\tilde{l}(y, g, a)$ so it is enough to show that USVMs optimize an upper bound on $\tilde{l}$.

Notice that minimizing the last four terms of Equation 3.1.3 (the first is responsible for regularization and is not part of the penalty) is equivalent to minimizing

$$\sum_{\mathbf{D}_+^T \cup \mathbf{D}_-^C} \xi_{i,1} + \frac{C_2}{C_1} \sum_{\mathbf{D}_-^T \cup \mathbf{D}_+^C} \xi_{i,1} + \frac{C_2}{C_1} \sum_{\mathbf{D}_+^T \cup \mathbf{D}_-^C} \xi_{i,2} + \sum_{\mathbf{D}_-^T \cup \mathbf{D}_+^C} \xi_{i,2}, \tag{3.1.10}$$

where $\frac{C_2}{C_1} \geq 0$. Take a point $\mathbf{x}_j \in \mathbf{D}_+^T$ (the reasoning in the three remaining cases is analogous). There are three possibilities

a) $\langle \mathbf{w}, \mathbf{x}_j \rangle - b_1 \geq 0$. We have $\xi_{j,1} \geq 0$ and $\xi_{j,2} \geq 0$ by (3.1.8). Here $a = +1$ and $\tilde{l}(+1, g = T, a = +1) = 0 \leq \xi_{j,1} + \frac{C_2}{C_1} \xi_{j,2}$,

b) $\langle \mathbf{w}, \mathbf{x}_j \rangle - b_1 < 0$ and $\langle \mathbf{w}, \mathbf{x}_j \rangle - b_2 \geq 0$, then $\xi_{j,1} > 1$ by (3.1.4) and $\xi_{j,2} \geq 0$. Here $a = 0$ and $\tilde{l}(+1, g = T, a = 0) = \rho \leq \xi_{j,1} + \frac{C_2}{C_1}\xi_{j,2}$,

c) $\langle \mathbf{w}, \mathbf{x}_j \rangle - b_2 < 0$, then $\xi_{j,1} > 1$ and $\xi_{j,2} > 1$ by (3.1.4) and (3.1.6). Here $a = -1$ and $\tilde{l}(+1, g = T, a = -1) = 1 \leq \xi_{j,1} + \frac{C_2}{C_1}\xi_{j,2}$.

Summing over all training records completes the proof. □

3.2 The Uplift Support Vector Machine optimization task

Let us now present the dual formulation of the Uplift Support Vector Machine optimization task. Methods for solving the optimization problem will be discussed in detail in the next chapter.

We first introduce a class variable transformation

$$z_i = \begin{cases} y_i, & \text{if } (\mathbf{x}_i, y_i) \in \mathbf{D}^T, \\ -y_i, & \text{if } (\mathbf{x}_i, y_i) \in \mathbf{D}^C. \end{cases} \tag{3.2.1}$$

In other words, z_i is obtained by keeping the class variable in the treatment group and reversing it in the control. Note that this is the same transformation which has been introduced in [20] in the context of uplift modeling and logistic regression.

This variable transformation allows us to simplify the optimization problem given in Equations 3.1.3–3.1.8 by merging (3.1.4) with (3.1.5) and (3.1.6) with (3.1.7). The simplified optimization problem is

$$\begin{aligned} \min_{\mathbf{w}, b_1, b_2 \in \mathbb{R}^{m+2}} \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle + C_1 \sum_{\mathbf{D}_+^T \cup \mathbf{D}_-^C} \xi_{i,1} + C_2 \sum_{\mathbf{D}_-^T \cup \mathbf{D}_+^C} \xi_{i,1} \\ + C_2 \sum_{\mathbf{D}_+^T \cup \mathbf{D}_-^C} \xi_{i,2} + C_1 \sum_{\mathbf{D}_-^T \cup \mathbf{D}_+^C} \xi_{i,2} \end{aligned}$$

subject to constraints

$$\begin{aligned} z_i(\langle \mathbf{w}, \mathbf{x}_i \rangle - b_1) - 1 + \xi_{i,1} \geq 0, \text{ forall } i = 1, \ldots, n, \\ z_i(\langle \mathbf{w}, \mathbf{x}_i \rangle - b_2) - 1 + \xi_{i,2} \geq 0, \text{ forall } i = 1, \ldots, n, \\ \xi_{i,j} \geq 0, \text{ forall } i = 1, \ldots, n, \ j \in \{1, 2\}. \end{aligned}$$

We will now obtain the dual form of the optimization problem. We begin by writing the

following Lagrange function

$$\begin{aligned}
&L(\mathbf{w}, b_1, b_2, \alpha_i, \beta_i, \xi_{i,1}, \xi_{i,2}, r_i, p_i) \\
&= \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle + C_1 \sum_{\mathbf{D}_+^T \cup \mathbf{D}_-^C} \xi_{i,1} + C_2 \sum_{\mathbf{D}_-^T \cup \mathbf{D}_+^C} \xi_{i,1} + C_2 \sum_{\mathbf{D}_+^T \cup \mathbf{D}_-^C} \xi_{i,2} + C_1 \sum_{\mathbf{D}_-^T \cup \mathbf{D}_+^C} \xi_{i,2} \\
&\quad - \sum_{i=1}^{n} \alpha_i \big(z_i(\langle \mathbf{w}, \mathbf{x}_i \rangle - b_1) - 1 + \xi_{i,1} \big) - \sum_{i=1}^{n} \beta_i \big(z_i(\langle \mathbf{w}, \mathbf{x}_i \rangle - b_2) - 1 + \xi_{i,2} \big) \\
&\quad - \sum_{i=1}^{n} r_i \xi_{i,1} - \sum_{i=1}^{n} p_i \xi_{i,2},
\end{aligned}$$

where $\alpha_i, \beta_i \in \mathbb{R}$ are Lagrange multipliers and $r_i, p_i \geq 0$.

Now we need to calculate partial derivatives and equate them to 0 in order to satisfy the Karush-Kuhn-Tucker stationarity condition (see Section 2.3.1). We begin by deriving with respect to $\mathbf{w}$

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n} \alpha_i z_i \mathbf{x}_i - \sum_{i=1}^{n} \beta_i z_i \mathbf{x}_i = 0,$$

from which we obtain

$$\mathbf{w} = \sum_{i=1}^{n} (\alpha_i + \beta_i) z_i \mathbf{x}_i. \tag{3.2.2}$$

We compute the remaining derivatives in a similar fashion

$$\frac{\partial L}{\partial b_1} = \sum_{i=1}^{n} \alpha_i z_i = 0, \qquad \frac{\partial L}{\partial b_2} = \sum_{i=1}^{n} \beta_i z_i = 0, \tag{3.2.3}$$

$$\frac{\partial L}{\partial \xi_{i,1}} = C_1 \mathbb{1}_{[z_i=+1]} + C_2 \mathbb{1}_{[z_i=-1]} - \alpha_i - r_i = 0, \tag{3.2.4}$$

$$\frac{\partial L}{\partial \xi_{i,2}} = C_1 \mathbb{1}_{[z_i=-1]} + C_2 \mathbb{1}_{[z_i=+1]} - \beta_i - p_i = 0. \tag{3.2.5}$$

Plugging Equations 3.2.4, 3.2.5 back into the Lagrange function we obtain, after simplifications,

$$L = \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle - \sum_{i=1}^{n} \alpha_i \big(z_i(\langle \mathbf{w}, \mathbf{x}_i \rangle - b_1) - 1 \big) - \sum_{i=1}^{n} \beta_i \big(z_i(\langle \mathbf{w}, \mathbf{x}_i \rangle - b_2) - 1 \big).$$

Substituting $\mathbf{w}$ from Equation 3.2.2 and using Equation 3.2.3 we get

$$\begin{aligned} L =& \frac{1}{2}\sum_{i,j=1}^{n}(\alpha_i+\beta_i)(\alpha_j+\beta_j)z_iz_j\langle \mathbf{x}_i,\mathbf{x}_j\rangle \\ &- \sum_{i,j=1}^{n}(\alpha_i+\beta_i)(\alpha_j+\beta_j)z_iz_j\langle \mathbf{x}_j,\mathbf{x}_i\rangle \\ &+ b_1\sum_{i=1}^{n}\alpha_iz_i + \sum_{i=1}^{n}\alpha_i + b_2\sum_{i=1}^{n}\beta_iz_i + \sum_{i=1}^{n}\beta_i \\ =& \sum_{i=1}^{n}(\alpha_i+\beta_i) - \frac{1}{2}\sum_{i,j=1}^{n}(\alpha_i+\beta_i)(\alpha_j+\beta_j)z_iz_j\langle \mathbf{x}_i,\mathbf{x}_j\rangle, \end{aligned} \tag{3.2.6}$$

which we maximize over α_i, β_i.

Finally, from the assumption that $r_i, p_i \geq 0$ (KKT dual feasibility condition) and (3.2.4), (3.2.5) combined with the KKT condition on nonnegativity of α_i, β_i (dual feasibility) and from (3.2.3) we obtain the following constraints for the dual optimization problem

$$0 \leq \alpha_i \leq C_1\mathbb{1}_{[z_i=+1]} + C_2\mathbb{1}_{[z_i=-1]}, \tag{3.2.7}$$

$$0 \leq \beta_i \leq C_1\mathbb{1}_{[z_i=-1]} + C_2\mathbb{1}_{[z_i=+1]}, \tag{3.2.8}$$

$$\sum_{i=1}^{n}\alpha_iz_i = \sum_{i=1}^{n}\beta_iz_i = 0. \tag{3.2.9}$$

3.3 Properties of the Uplift Support Vector Machines (USVMs)

In this section we analyze some of the mathematical properties of Uplift Support Vector Machines (USVMs), especially those related to the influence of the parameters C_1 and C_2 on model's behavior. One of the more important results is showing that the ratio $\frac{C_2}{C_1}$ of the penalty parameters directly influences the number of records which are classified as neutral, or, in other words, that it determines the distance between the two separating hyperplanes.

Lemma 3.3.1. Let $\mathbf{w}^*, b_1^*, b_2^*$ be a solution to the Uplift SVM optimization problem given by Equations 3.1.3−3.1.8. If $C_2 > C_1$ then $b_1^* \geq b_2^*$.

Proof. Let us begin with an observation which will be used in this and the following proofs. Consider the Uplift SVM optimization problem given by Equations 3.1.3−3.1.8. Notice that when $\mathbf{w}, b_1, b_2$ are fixed, the optimal values of slack variables $\xi_{i,j}$ are uniquely determined. Optimal values for slack variables present in Equation 3.1.4 are $\xi_{i,1}^* = \max\{0, -\langle \mathbf{w}, \mathbf{x}_i\rangle + b_1 + 1\}$, and for

those present in Equation 3.1.5, $\xi^*_{i,1} = \max\{0, \langle \mathbf{w}, \mathbf{x}_i \rangle - b_1 + 1\}$. Analogous formulas can be given for $\xi^*_{i,2}$ and Equations 3.1.6−3.1.7. Now we come back to the proof.

Let $S^* = \langle \mathbf{w}^*, b_1^*, b_2^* \rangle$ be an optimal solution with $b_1^* < b_2^*$. Consider also a set of parameters $S' = \langle \mathbf{w}^*, b_2^*, b_1^* \rangle$ with the values of b_1^*, b_2^* interchanged and look at the target function (3.1.3) for both sets of parameters.

Take a point $(\mathbf{x}_i, y_i) \in \mathbf{D}_+^T \cup \mathbf{D}_-^C$ for which, under the set of parameters S', $\xi'_{i,1} > 0$ and $\xi'_{i,2} = 0$, that is the point is penalized only for crossing the hyperplane H_1. Under the parameters S^* the point will be penalized not with $C_1\xi'_{i,1}$ for crossing H_1 but, instead, with $C_2\xi^*_{i,2}$ for crossing H_2. Since, by switching from S^* to S' the hyperplanes simply exchange intercepts, we have $\xi^*_{i,1} = \xi'_{i,2}$ and, from the assumption, $C_2\xi^*_{i,1} > C_1\xi'_{i,2}$. Thus the amount every point $(\mathbf{x}_i, y_i) \in \mathbf{D}_+^T \cup \mathbf{D}_-^C$ contributes to the target function (3.1.3) is lower in S' than in S^*.

We now consider points penalized for crossing both hyperplanes. The idea of the proof is analogous to the first case. Take a point $(\mathbf{x}_i, y_i) \in \mathbf{D}_+^T \cup \mathbf{D}_-^C$ with $\xi^*_{i,1}, \xi^*_{i,2} > 0$. Denote by $P_i^* = C_1\xi^*_{i,1} + C_2\xi^*_{i,2}$ the penalty incurred by the point under S^* and by $P'_i = C_1\xi'_{i,1} + C_2\xi'_{i,2}$ the penalty of the same point under S'. Notice that $\xi'_{i,1} = \xi^*_{i,2}$ and $\xi'_{i,2} = \xi^*_{i,1}$. Hence

$$\begin{aligned} P_i^* - P'_i &= C_1\xi^*_{i,1} + C_2\xi^*_{i,2} - C_1\xi'_{i,1} - C_2\xi'_{i,2} = C_1\xi^*_{i,1} + C_2\xi^*_{i,2} - C_1\xi^*_{i,2} - C_2\xi^*_{i,1} \\ &= \xi^*_{i,1}(C_1 - C_2) + \xi^*_{i,2}(C_2 - C_1) = \underbrace{(C_1 - C_2)}_{<0}\underbrace{(\xi^*_{i,1} - \xi^*_{i,2})}_{<0} > 0 \end{aligned}$$

giving $P'_i < P_i^*$. Analogous argument holds for points in $\mathbf{D}_-^T \cup \mathbf{D}_+^C$. Therefore penalties incurred by all penalized points are lower in S' than in S^* contradicting the optimality of S^*.

□

The lemma guarantees that the problem possesses a well defined solution in the sense of Equation 3.1.2. Moreover, it naturally constrains (together with Lemma 3.3.3 below) the penalty C_2 to be greater than or equal to C_1. From now on, instead of working with the coefficient C_2, it will be more convenient to talk about the penalty coefficient C_1 and the quotient $\frac{C_2}{C_1} \geq 1$.

Lemma 3.3.2. For sufficiently large value of $\frac{C_2}{C_1}$ none of the observations is penalized with a term involving the C_2 factor in the solution to the USVM optimization problem.

Proof. Let us first consider the hyperplane H_1 (argument for H_2 is analogous). Assume that there exists at least one point in $\mathbf{D}_-^T \cup \mathbf{D}_+^C$ (analogous argument holds for $\mathbf{D}_+^T \cup \mathbf{D}_-^C$) which is punished with a term involving the C_2 penalty coefficient, and therefore lies on the wrong side of H_1. Out of all such points choose the one $(\tilde{\mathbf{x}}_i, \tilde{y}_i)$ which is furthest from H_1 and denote by $\tilde{\xi}_{i,1}$, $\tilde{\xi}_{i,2}$ its slack

variables w.r.t. H_1 and H_2 respectively. The penalty incurred by $(\tilde{\mathbf{x}}_i, \tilde{y}_i)$ equals

$$C_2\tilde{\xi}_{i,1} + C_1\tilde{\xi}_{i,2}.$$

Let us now shift the hyperplane H_1 by exactly $\tilde{\xi}_{i,1}$; as a result, the point is only penalized by $C_1\tilde{\xi}_{i,2}$. The same is true for all other points from $\mathbf{D}_-^T \cup \mathbf{D}_+^C$. On the other hand, after shifting H_1, penalties w.r.t. H_1 of points in $\mathbf{D}_+^T \cup \mathbf{D}_-^C$ could have increased, but the increase is bounded by $C_1\tilde{\xi}_{i,1}$ per point.

Denote $n_1 = |\mathbf{D}_-^T \cup \mathbf{D}_+^C|$, $n_2 = |\mathbf{D}_+^T \cup \mathbf{D}_-^C|$. The change in penalties caused by shifting H_1 is bounded from above by

$$C_1\tilde{\xi}_{i,2} - (C_2\tilde{\xi}_{i,1} + C_1\tilde{\xi}_{i,2}) + n_2C_1\tilde{\xi}_{i,1} = \tilde{\xi}_{i,1}(n_2C_1 - C_2),$$

which is negative for sufficiently large value of C_2, such that the shift of H_1 is guaranteed to decrease the target function. □

Equivalently the lemma states that for a large enough value of $\frac{C_2}{C_1}$, none of the points will be on the wrong side of both hyperplanes. This is possible only when the hyperplanes are maximally separated, resulting in most (often all) points classified as neutral.

Lemma 3.3.3. If $C_1 = C_2 = C$ and the solution is unique then both hyperplanes coincide: $b_1 = b_2$.

Proof of Lemma 3.3.3. Let us fix any $\mathbf{w}$ and optimize with respect to b_1, b_2. Under the assumption of the lemma, the target function (3.1.3) can be rewritten as

$$\frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{\mathbf{D}^T \cup \mathbf{D}^C} \xi_{i,1} + C \sum_{\mathbf{D}^T \cup \mathbf{D}^C} \xi_{i,2}.$$

Note, that the first term is constant, the second is a function of b_1 and the third of b_2. Moreover the second and third term are fully symmetric so the target function can be rewritten as $const. + f(b_1) + f(b_2)$, where f is some function of b_1 or b_2. Notice that optimization over b_1 is done independently of optimization over b_2 and since the optimized functions f are identical, the resulting optima for b_1 and b_2 must be identical if the solution is unique. The result follows since the argument is valid for any $\mathbf{w}$. □

We are now ready to give an interpretation of the C_1 and $\frac{C_2}{C_1}$ parameters of the Uplift SVM. The parameter C_1 plays the role analogous to the penalty coefficient C in classical SVMs controlling

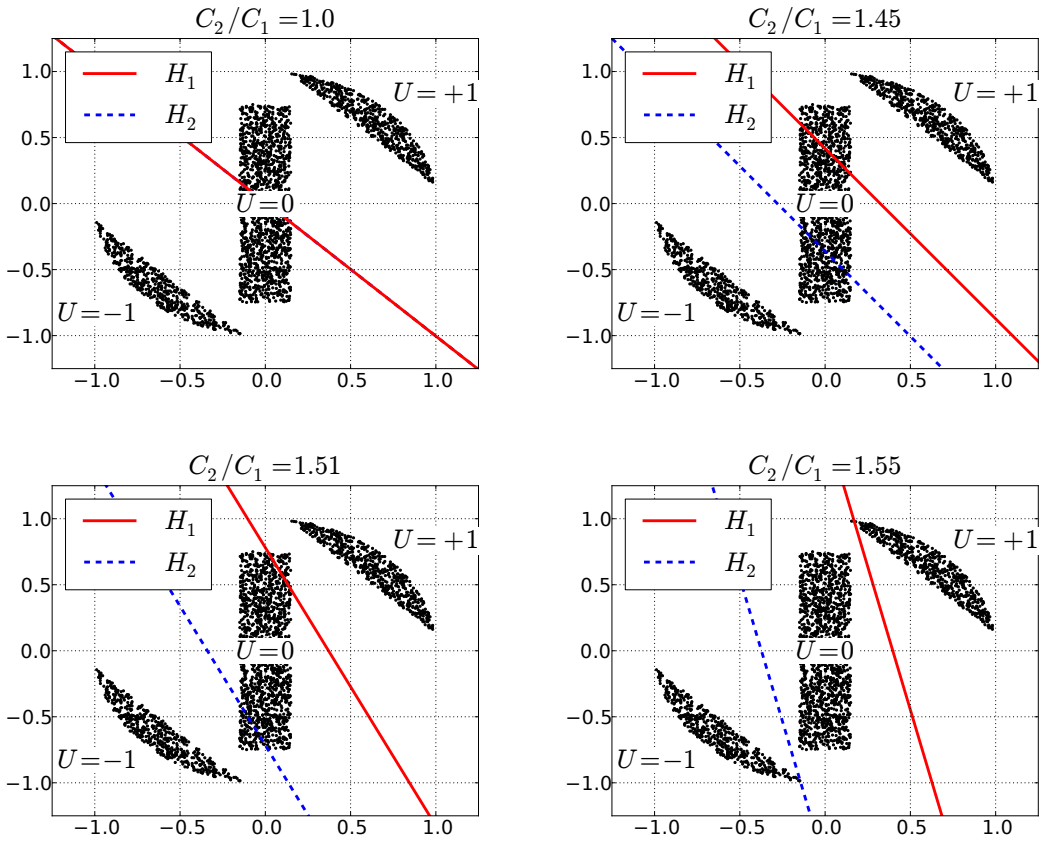

Figure 3.3.1: The effect of the C_2/C_1 ratio on the separating hyperplanes for an artificial example.

the relative cost of misclassified points with respect to the margin maximization term $\frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle$. The quotient $\frac{C_2}{C_1}$ allows the analyst to decide what proportion of points should be classified as positive or negative. In other words, it allows for controlling the size of the neutral prediction.

Note that this is *not* equivalent to selecting thresholds in data scored using a single model. For each value of $\frac{C_2}{C_1}$ a different model is built which is optimized for a specific proportion of neutral predictions. We believe that this property of USVMs is very useful for practical applications, as it allows for tuning the model specifically to the desired size of the campaign.

Figure 3.3.1 shows, on an artificial example, how the weight vector $\mathbf{w}$ adapts to a specific size of the neutral set. The treatment and control datasets consist of three randomly generated point clouds (treatment and control points are both marked with black dots to avoid clutter), each with a different value of the net gain from performing the action, denoted U in the pictures. The two crescents have gains -1 and $+1$ respectively, and in the middle rectangle the effect of the action

is neutral. The value of the parameter C_1 was set to 1. It can be seen that when $C_1 = C_2$ the separating hyperplanes coincide and are aligned with the crescents where the impact is positive or negative. The neutral part of data is ignored. As the ratio C_2/C_1 grows, the hyperplanes become more and more separated and begin changing direction, taking into account not only the crescents but also the neutral group. In the last chart, the neutral rectangle falls between both hyperplanes and the three groups are well separated.

3.4 L_p Uplift Support Vector Machines

Unfortunately, USVMs suffer from a problem which, in certain cases, makes Lemmas 3.3.2 and 3.3.3 lose their practical significance. We begin by analyzing the problem theoretically. Later, in order to alleviate it, we adapt L_p-SVMs [1] to the uplift case. L_p-SVMs are a variant of classical SVMs where, in the optimization problem, the slack variables have been raised to power p resulting in a smoother loss function. Details will be given later in the section. Uplift Support Vector Machines defined in the previous section will be referred to as L_1-USVMs.

3.4.1 A problem with L_1-USVMs. Theoretical analysis

We begin with a lemma on the nonuniqueness of the intercepts b_1 and b_2 in the USVM optimization problem. The lemma is stated for b_1, the result for b_2 is analogous.

Lemma 3.4.1. Assume that $\mathbf{w}$ and b_2 are fixed and

$$\frac{2}{\|\mathbf{w}\|} \geq \max_i\{\langle \mathbf{w}, \mathbf{x}_i\rangle\} - \min_i\{\langle \mathbf{w}, \mathbf{x}_i\rangle\}, \tag{3.4.1}$$

i.e. the margin is wide enough to encompass all data points. Assume further, that $\frac{C_2}{C_1} = \frac{|\mathbf{D}_+^T \cup \mathbf{D}_-^C|}{|\mathbf{D}_-^T \cup \mathbf{D}_+^C|}$. Then the optimal value of b_1 is not unique and can be chosen anywhere in the interval

$$\left[\max_i\{\langle \mathbf{w}, \mathbf{x}_i\rangle - 1\},\ \ \min_i\{\langle \mathbf{w}, \mathbf{x}_i\rangle + 1\}\right].$$

Proof. Pick any b_1 in the range $[\max_i\{\langle \mathbf{w}, \mathbf{x}_i\rangle - 1\},\ \min_i\{\langle \mathbf{w}, \mathbf{x}_i\rangle + 1\}]$. From (3.4.1) it follows that for all i, $|\langle \mathbf{w}, \mathbf{x}_i\rangle - b_1| \leq 1$, and therefore (3.1.8) follows from (3.1.4)−(3.1.7), implying

$$\xi_{i,1} = \begin{cases} 1 - (\langle \mathbf{w}, \mathbf{x}_i\rangle - b_1) & \text{for } \mathbf{x}_i \in \mathbf{D}_+^T \cup \mathbf{D}_-^C, \\ 1 + (\langle \mathbf{w}, \mathbf{x}_i\rangle - b_1) & \text{for } \mathbf{x}_i \in \mathbf{D}_-^T \cup \mathbf{D}_+^C, \end{cases}$$

for all i in an optimal solution. Let us denote by L the goal function given in Equation 3.1.3. Since $\mathbf{w}$ and b_2 are fixed, the first, fourth, and fifth terms in (3.1.3) do not depend on b_1, and L becomes

$$\begin{aligned}
L(b_1) &= \text{const.} + C_1 \sum_{\mathbf{D}_+^T \cup \mathbf{D}_-^C} \xi_{i,1} + C_2 \sum_{\mathbf{D}_-^T \cup \mathbf{D}_+^C} \xi_{i,1} \\
&= \text{const.} + C_1 \sum_{\mathbf{D}_+^T \cup \mathbf{D}_-^C} (1 - (\langle \mathbf{w}, \mathbf{x}_i \rangle - b_1)) + C_2 \sum_{\mathbf{D}_-^T \cup \mathbf{D}_+^C} (1 + (\langle \mathbf{w}, \mathbf{x}_i \rangle - b_1)) \\
&= \text{const.} + \underbrace{C_1 |\mathbf{D}_+^T \cup \mathbf{D}_-^C| - C_1 \sum_{\mathbf{D}_+^T \cup \mathbf{D}_-^C} \langle \mathbf{w}, \mathbf{x}_i \rangle}_{\text{const. indep. of } b_1} + C_1 b_1 |\mathbf{D}_+^T \cup \mathbf{D}_-^C| \\
&\quad \underbrace{+ C_2 |\mathbf{D}_-^T \cup \mathbf{D}_+^C| + C_2 \sum_{\mathbf{D}_-^T \cup \mathbf{D}_+^C} \langle \mathbf{w}, \mathbf{x}_i \rangle}_{\text{const. indep. of } b_1} - C_2 b_1 |\mathbf{D}_-^T \cup \mathbf{D}_+^C| \\
&= \text{const.} - b_1 (C_2 |\mathbf{D}_-^T \cup \mathbf{D}_+^C| - C_1 |\mathbf{D}_+^T \cup \mathbf{D}_-^C|).
\end{aligned}$$

Clearly, if $\frac{C_2}{C_1} = \frac{|\mathbf{D}_+^T \cup \mathbf{D}_-^C|}{|\mathbf{D}_-^T \cup \mathbf{D}_+^C|}$ the value of the goal function does not depend on b_1. □

Note that when $b_1 = \min_i\{-1 - \langle \mathbf{w}, \mathbf{x}_i \rangle\}$ all points are classified as positive, at the other extreme all points are classified as neutral. As a result, for some values of the parameter C_2 *all* points are classified as neutral, then, when the parameter crosses the threshold given in the statement of the above lemma, all data points are classified as positive with no intermediate steps.

It may seem that the condition that the margin be wide enough to encompass all data points is unlikely to occur in practice. The following lemma shows that this is not the case, and the margin can in fact be infinite. Real examples are given in Section 6.3.

Lemma 3.4.2. Without loss of generality assume $|\mathbf{D}_+^T \cup \mathbf{D}_-^C| \geq |\mathbf{D}_-^T \cup \mathbf{D}_+^C|$. Suppose there exist multipliers ω_i such that

$$0 \leq \omega_i \leq 1, \qquad \sum_{\mathbf{D}_-^T \cup \mathbf{D}_+^C} \mathbf{x}_i = \sum_{\mathbf{D}_+^T \cup \mathbf{D}_-^C} \omega_i \mathbf{x}_i, \qquad \sum_{\mathbf{D}_+^T \cup \mathbf{D}_-^C} \omega_i = |\mathbf{D}_-^T \cup \mathbf{D}_+^C|,$$

then the optimal weight vector $\mathbf{w}$ is 0.

Proof. The proof is similar to that for classical SVMs provided in [32]. Given any ω_i satisfying

the assumptions one can easily check (taking into account that $C_2 \geq C_1$) that setting

$$\alpha_i = \begin{cases} C_1 & \text{for } i : z_i = -1, \\ \omega_i C_1 & \text{for } i : z_i = +1, \end{cases} \qquad \beta_i = \begin{cases} C_1 & \text{for } i : z_i = -1, \\ \omega_i C_1 & \text{for } i : z_i = +1. \end{cases}$$

satisfies the KKT conditions (3.2.3)−(3.2.5) and, therefore, due to Equation 3.2.2, induces a optimal solution with $\mathbf{w} = 0$. □

The lemma implies, for example, that if the averages of predictor variables in $\mathbf{D}_-^T \cup \mathbf{D}_+^C$ and $\mathbf{D}_+^T \cup \mathbf{D}_-^C$ are identical, the margin is infinitely wide and encompasses all data points. Note that an analogous condition is true also for classical SVMs [32]. In uplift modeling, the prediction task is often difficult, resulting in large overlap between convex hulls of $\mathbf{D}_+^T \cup \mathbf{D}_-^C$ and $\mathbf{D}_-^T \cup \mathbf{D}_+^C$. As a result, the conditions of the lemma are relatively easy to satisfy.

To solve those problems we now introduce L_p-USVMs, which are an adaptation of L_p-SVMs [1, 8] to uplift modeling, and which, since they depend on the parameter C_2 in a continuous fashion, do not suffer from the aforementioned problem.

3.4.2 L_p Uplift Support Vector Machines. Definition

Let $p > 1$ be a constant. The idea behind L_p-SVMs is to raise the slack variables used in the SVM optimization problem to the power p [1, 8]. In the uplift case, the quantity being optimized (analogue of Equation 3.1.3) now becomes

$$\begin{aligned} \min_{\mathbf{w}, b_1, b_2 \in \mathbb{R}^{m+2}} \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle + C_1 \sum_{\mathbf{D}_+^T \cup \mathbf{D}_-^C} |\xi_{i,1}|^p + C_2 \sum_{\mathbf{D}_-^T \cup \mathbf{D}_+^C} |\xi_{i,1}|^p \\ + C_2 \sum_{\mathbf{D}_+^T \cup \mathbf{D}_-^C} |\xi_{i,2}|^p + C_1 \sum_{\mathbf{D}_-^T \cup \mathbf{D}_+^C} |\xi_{i,2}|^p, \end{aligned} \tag{3.4.2}$$

and the optimization is performed subject to

$$\langle \mathbf{w}, \mathbf{x}_i \rangle - b_1 \geq +1 - \xi_{i,1}, \text{ forall } (\mathbf{x}_i, y_i) \in \mathbf{D}_+^T \cup \mathbf{D}_-^C, \tag{3.4.3}$$

$$\langle \mathbf{w}, \mathbf{x}_i \rangle - b_1 \leq -1 + \xi_{i,1}, \text{ forall } (\mathbf{x}_i, y_i) \in \mathbf{D}_-^T \cup \mathbf{D}_+^C, \tag{3.4.4}$$

$$\langle \mathbf{w}, \mathbf{x}_i \rangle - b_2 \geq +1 - \xi_{i,2}, \text{ forall } (\mathbf{x}_i, y_i) \in \mathbf{D}_+^T \cup \mathbf{D}_-^C, \tag{3.4.5}$$

$$\langle \mathbf{w}, \mathbf{x}_i \rangle - b_2 \leq -1 + \xi_{i,2}, \text{ forall } (\mathbf{x}_i, y_i) \in \mathbf{D}_-^T \cup \mathbf{D}_+^C. \tag{3.4.6}$$

Note that these are the first four constraints (3.1.4)-(3.1.7) used also in the L_1-USVM case. It is easy to see that the fifth constraint is no longer needed. Indeed, a solution with any $\xi_{i,j} < 0$ cannot be optimal because the corresponding constraints $\langle \mathbf{w}, \mathbf{x}_i \rangle + b_1 \geq 1 - \xi_{i,j}$ and $\langle \mathbf{w}, \mathbf{x}_i \rangle - b_1 \leq -1 + \xi_{i,j}$ would be also satisfied for $\xi_{i,j} = 0$ which gives a lower value of objective function. The absolute values are used to ensure that the $\xi_{i,j}$'s can be raised to noninteger powers.

It is easy to see that Theorem 3.1.1 and Lemmas 3.3.1–3.3.3 remain true also in the L_p formulation, so the L_p-USVM minimizes an upper bound on the true uplift loss and the properties regarding the values of parameters C_1 and C_2 directly carry over to this case.

3.4.3 Dual optimization task for L_p-USVMs

We use an approach similar to that in Section 3.2 to obtain the dual for the L_p-USVM optimization problem. See [1] for an analogous derivation for L_p-SVMs in the classification problem.

After applying the variable transformation (3.2.1) the Lagrangian becomes

$$\begin{aligned} L(\mathbf{w}, b_1, b_2, \alpha_i, \beta_i, \xi_{i,1}, \xi_{i,2}) &= \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle \\ &+ C_1 \sum_{\mathbf{D}_+^T \cup \mathbf{D}_-^C} |\xi_{i,1}|^p + C_2 \sum_{\mathbf{D}_-^T \cup \mathbf{D}_+^C} |\xi_{i,1}|^p + C_2 \sum_{\mathbf{D}_+^T \cup \mathbf{D}_-^C} |\xi_{i,2}|^p + C_1 \sum_{\mathbf{D}_-^T \cup \mathbf{D}_+^C} |\xi_{i,2}|^p \\ &- \sum_{i=1}^{n} \alpha_i \big(z_i (\langle \mathbf{w}, \mathbf{x}_i \rangle - b_1) - 1 + \xi_{i,1} \big) - \sum_{i=1}^{n} \beta_i \big(z_i (\langle \mathbf{w}, \mathbf{x}_i \rangle - b_2) - 1 + \xi_{i,2} \big). \end{aligned}$$

From the KKT stationarity condition we get

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n} \alpha_i z_i \mathbf{x}_i - \sum_{i=1}^{n} \beta_i z_i \mathbf{x}_i = 0$$

and consequently

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i z_i \mathbf{x}_i + \sum_{i=1}^{n} \beta_i z_i \mathbf{x}_i. \tag{3.4.7}$$

Similarly

$$\frac{\partial L}{\partial b_1} = \sum_{i=1}^{n} \alpha_i z_i = 0, \qquad \frac{\partial L}{\partial b_2} = \sum_{i=1}^{n} \beta_i z_i = 0, \tag{3.4.8}$$

$$\frac{\partial L}{\partial \xi_{i,1}} = pC_1 |\xi_{i,1}|^{p-1} \operatorname{sgn}(\xi_{i,1}) \mathbb{1}_{[z_i=+1]} + pC_2 |\xi_{i,1}|^{p-1} \operatorname{sgn}(\xi_{i,1}) \mathbb{1}_{[z_i=-1]} - \alpha_i = 0, \tag{3.4.9}$$

$$\frac{\partial L}{\partial \xi_{i,2}} = pC_1 |\xi_{i,2}|^{p-1} \operatorname{sgn}(\xi_{i,2}) \mathbb{1}_{[z_i=-1]} + pC_2 |\xi_{i,2}|^{p-1} \operatorname{sgn}(\xi_{i,2}) \mathbb{1}_{[z_i=+1]} - \beta_i = 0. \tag{3.4.10}$$

Notice that we can omit the factors $\operatorname{sgn}(\xi_{i,j})$ in last two equations since, as noted above, optimal values of $\xi_{i,j}$ have to be nonnegative and when $\xi_{i,j} = 0$ the factor disappears since it's multiplied by zero. After dropping the signum functions we obtain

$$|\xi_{i,1}| = \left(\frac{\alpha_i}{pC_1\mathbb{1}_{[z_i=+1]} + pC_2\mathbb{1}_{[z_i=-1]}}\right)^{1/(p-1)}, \tag{3.4.11}$$

$$|\xi_{i,2}| = \left(\frac{\beta_i}{pC_1\mathbb{1}_{[z_i=-1]} + pC_2\mathbb{1}_{[z_i=+1]}}\right)^{1/(p-1)}. \tag{3.4.12}$$

After reformulating the Lagrangian (using nonnegativity of $\xi_{i,j}$ to replace it with $|\xi_{i,j}|$) we obtain

$$\begin{aligned} L =& \frac{1}{2}\langle \mathbf{w}, \mathbf{w}\rangle - \sum_{i=1}^{n} \alpha_i\big(z_i(\langle \mathbf{w}, \mathbf{x}_i\rangle - b_1) - 1\big) - \sum_{i=1}^{n} \beta_i\big(z_i(\langle \mathbf{w}, \mathbf{x}_i\rangle - b_2) - 1\big) \\ &+ \frac{1}{p}\sum_{i=1}^{n} |\xi_{i,1}|\big(pC_1|\xi_{i,1}|^{p-1}\mathbb{1}_{[z_i=+1]} + pC_2|\xi_{i,1}|^{p-1}\mathbb{1}_{[z_i=-1]} - \alpha_i - (p-1)\alpha_i\big) \\ &+ \frac{1}{p}\sum_{i=1}^{n} |\xi_{i,2}|\big(pC_1|\xi_{i,2}|^{p-1}\mathbb{1}_{[z_i=-1]} + pC_2|\xi_{i,2}|^{p-1}\mathbb{1}_{[z_i=+1]} - \beta_i - (p-1)\beta_i\big), \end{aligned}$$

which, using (3.4.9) and (3.4.10), can be further simplified to

$$\begin{aligned} &\frac{1}{2}\langle \mathbf{w}, \mathbf{w}\rangle - \sum_{i=1}^{n} \alpha_i\big(z_i(\langle \mathbf{w}, \mathbf{x}_i\rangle - b_1) - 1\big) - \sum_{i=1}^{n} \beta_i\big(z_i(\langle \mathbf{w}, \mathbf{x}_i\rangle - b_2) - 1\big) \\ &\qquad - \frac{p-1}{p}\sum_{i=1}^{n} |\xi_{i,1}|\alpha_i - \frac{p-1}{p}\sum_{i=1}^{n} |\xi_{i,2}|\beta_i. \end{aligned}$$

Using Equations 3.4.7, 3.4.8, 3.4.11, 3.4.12 the final form of the Lagrangian is obtained:

$$\begin{aligned} &-\frac{1}{2}\sum_{i,j=1}^{n} \alpha_i\alpha_j z_i z_j\langle \mathbf{x}_i, \mathbf{x}_j\rangle - \sum_{i,j=1}^{n} \alpha_i\beta_j z_i z_j\langle \mathbf{x}_i, \mathbf{x}_j\rangle \\ &\qquad - \frac{1}{2}\sum_{i,j=1}^{n} \beta_i\beta_j z_i z_j\langle \mathbf{x}_i, \mathbf{x}_j\rangle + \sum_{i=1}^{n}(\alpha_i + \beta_i) \\ &\qquad - \frac{p-1}{p}\sum_{i=1}^{n} \frac{\alpha_i^{p/(p-1)}}{\big(pC_1\mathbb{1}_{[z_i=+1]} + pC_2\mathbb{1}_{[z_i=-1]}\big)^{1/(p-1)}} \\ &\qquad - \frac{p-1}{p}\sum_{i=1}^{n} \frac{\beta_i^{p/(p-1)}}{\big(pC_1\mathbb{1}_{[z_i=-1]} + pC_2\mathbb{1}_{[z_i=+1]}\big)^{1/(p-1)}}, \end{aligned} \tag{3.4.13}$$

which needs to be maximized subject to $\alpha_i, \beta_i \geq 0$ and Equation 3.4.8.

Unfortunately most optimization algorithms require the goal function to be twice differentiable in the optimization domain, which limits the choice of p to values for which $\frac{p}{p-1}$ is an integer, e.g. $p = 2, \frac{3}{2}, \frac{4}{3}, \frac{5}{4}, \frac{6}{5}, \ldots$. Note, however, that those values are actually the most interesting from our perspective since they include the smooth $p = 2$ case and allow for arbitrarily close smooth approximations of the L_1-USVM.

Chapter 4

Optimiztion Algorithms

The two optimization problems presented above can be solved using off the shelf constrained optimization software or using methods designed specifically for Support Vector Machines. We have primarily followed the first approach and applied quadratic and convex solvers from the CVXOPT library [2] to the dual formulations of Uplift SVMs. In order to make the solutions efficient, we developed dedicated solvers for the Karush-Kuhn-Tucker (KKT) systems of equations used by CVXOPT. The solvers exploit the structure of Uplift SVMs to offer high computational efficiency and numerical stability. Details are given in Sections 4.2.1 and 4.3.

Additionally we have adapted to our problem the dual coordinate descent method used in the LIBLINEAR package [16] which is currently the most popular method for solving classical SVM-type optimization problems. The method is described in Section 4.2.2. Unfortunately the method had poor convergence properties in the case of USVMs so all our experiments use the method based on quadratic and convex programming using CVXOPT.

4.1 Selected linear algebra concepts

Before we present the optimization algorithms used to find optimal solutions to the problems from the previous chapter, we first give a brief description of a few linear algebra concepts: the Schur complement, the Woodbury matrix identity and the solution of the weighted regularized least squares problem, which will be used later in this section. Further details on those topics can be found in [5] and [11].

4.1.1 The Schur complement

Let $\mathbf{M}$ be an $n \times n$ matrix which is expressed in block-matrix form as

$$\mathbf{M} = \left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{D} \end{array}\right],$$

where $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ are respectively $p \times p$, $p \times q$, $q \times p$ and $q \times q$ matrices, with $n = p + q$. Suppose that we have a system of linear equations involving the matrix $\mathbf{M}$. Using the block matrix form the system can be rewritten as a system of two matrix equations

$$\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} = \mathbf{a},$$
$$\mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{y} = \mathbf{b},$$

where $\mathbf{x}$, $\mathbf{a}$ and $\mathbf{y}$, $\mathbf{b}$ are, respectively, p and q dimensional real vectors. Suppose we want to solve the system in a 'blockwise' fashion, by first finding $\mathbf{x}$ and then $\mathbf{y}$.

Assuming the matrix $\mathbf{D}$ is invertible, we start (similar to Gaussian elimination), by solving the second matrix equation for $\mathbf{y}$:

$$\mathbf{y} = \mathbf{D}^{-1}(\mathbf{b} - \mathbf{C}\mathbf{x}).$$

Then we substitute this result into the first equation obtaining

$$\mathbf{A}\mathbf{x} + \mathbf{B}(\mathbf{D}^{-1}(\mathbf{b} - \mathbf{C}\mathbf{x})) = \mathbf{a},$$

which is equivalent to

$$(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})\mathbf{x} = \mathbf{a} - \mathbf{B}\mathbf{D}^{-1}\mathbf{b}.$$

If the matrix $\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C}$, which is called the *Schur complement of* $\mathbf{D}$ *in* $\mathbf{M}$, is invertible, we can solve the equation for $\mathbf{x}$, and then, by using the second matrix equation $\mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{y} = \mathbf{b}$, solve for $\mathbf{y}$.

Hence, assuming that $\mathbf{D}$ and its Schur complement are invertible, the problem of inverting a $(p + q) \times (p + q)$ matrix reduces to the problem of inverting two $p \times p$ and $q \times q$ matrices, which is especially useful in case of solving systems of linear equations where one of the matrices has a special form and can be easily inverted. In a similar fashion we can define the Schur complement $\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}$ of the matrix $\mathbf{A}$ in $\mathbf{M}$, given that $\mathbf{A}$ is invertible.

4.1.2 Woodbury matrix identity

As shown above, the concept of Schur complement allows us to decompose larger systems of equations such that special structure of their subsystems can be exploited. One such special case is when the system has the form

$$(\mathbf{A} + \mathbf{BC})\mathbf{x} = \mathbf{b}, \tag{4.1.1}$$

where $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are respectively $n \times n$, $n \times p$ and $p \times n$ matrices, $\mathbf{A}$ is nonsingular and $\mathbf{x}$ and $\mathbf{b}$ are n-dimensional real vectors. This special form can be exploited when the inverse of the matrix $\mathbf{A}$ can easily be computed and $p < n$. The derivation below follows the one presented in [5, Section C.4.3].

Introduce a new variable $\mathbf{y} = \mathbf{Cx}$ and rewrite Equation 4.1.1 as

$$\begin{aligned} \mathbf{Ax} + \mathbf{By} &= \mathbf{b} \\ \mathbf{y} &= \mathbf{Cx}, \end{aligned}$$

which can be expressed in matrix form as

$$\left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & -\mathbf{I} \end{array}\right] \left[\begin{array}{c} \mathbf{x} \\ \hline \mathbf{y} \end{array}\right] = \left[\begin{array}{c} \mathbf{b} \\ \hline 0 \end{array}\right]. \tag{4.1.2}$$

Interestingly, the Schur complement of $-\mathbf{I}$ in the above matrix is exactly $\mathbf{A} + \mathbf{BC}$. Now, in the system 4.1.2 we eliminate the original variable $\mathbf{x}$. We get $\mathbf{x} = \mathbf{A}^{-1}(\mathbf{b} - \mathbf{By})$ and by substituting this into the second equation $\mathbf{y} = \mathbf{Cx}$ we get

$$(\mathbf{I} + \mathbf{CA}^{-1}\mathbf{B})\mathbf{y} = \mathbf{CA}^{-1}\mathbf{b},$$

which can be transformed into

$$\mathbf{y} = (\mathbf{I} + \mathbf{CA}^{-1}\mathbf{B})^{-1}\mathbf{CA}^{-1}\mathbf{b}.$$

Now, by using $\mathbf{x} = \mathbf{A}^{-1}(\mathbf{b} - \mathbf{By})$, we obtain

$$\mathbf{x} = (\mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{I} + \mathbf{CA}^{-1}\mathbf{B})^{-1}\mathbf{CA}^{-1})\mathbf{b}.$$

Note that $\mathbf{b}$ can be arbitrary, so we finally get

$$(\mathbf{A}+\mathbf{BC})^{-1} = (\mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{I}+\mathbf{CA}^{-1}\mathbf{B})^{-1}\mathbf{CA}^{-1}). \tag{4.1.3}$$

The above formula is known as the *Woodbury matrix identity* or the *matrix inversion lemma*. Notice that it is assumed that $\mathbf{A}$ is easily invertible so one only needs to explicitly invert $\mathbf{I}+\mathbf{CA}^{-1}\mathbf{B}$ which is of size $p \times p$, typically much smaller than the original $n \times n$.

4.1.3 Weighted regularized least squares

The last topic that we present in this section is the matrix formula for the solution of the weighted regularized least squares problem. We will start with simple linear least squares and modify it by adding weights and a regularization term. Consider a system of linear equations

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta}, \tag{4.1.4}$$

where $\mathbf{X}$ is a $n \times m$ real matrix, $\mathbf{y}$ an n-dimensional real vector and $\boldsymbol{\beta}$ an m-dimensional real vector of regression coefficients. Since the system is typically overdetermined, no solution $\boldsymbol{\beta}$ exists. The least squares problem is to find the 'best' possible $\hat{\boldsymbol{\beta}}$ in the sense of the following quadratic minimization problem

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{m} X_{ij}\beta_j \right)^2 = \arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2. \tag{4.1.5}$$

When $\mathbf{X}$ is of full column rank, this problem has a unique solution given by [5, Chapter 6]

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}.$$

The weighted version differs slightly, as we simply introduce weights into the objective function 4.1.5

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} w_i \left(y_i - \sum_{j=1}^{m} X_{ij}\beta_j \right)^2 = \arg\min_{\boldsymbol{\beta}} \|\mathbf{W}^{1/2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\|^2, \tag{4.1.6}$$

where $w_i > 0$ is the weight of the i-th case, and $\mathbf{W} = \operatorname{diag}(w_1, \ldots, w_n)$ is and $n \times n$ diagonal weight matrix. Then the solution is given by [5, Chapter 6]

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}\mathbf{y}.$$

We now further modify (4.1.6) by adding an L_2 regularization term

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} w_i \left(y_i - \sum_{j=1}^{m} X_{ij}\beta_j \right)^2 + \lambda \|\boldsymbol{\beta}\|^2 = \arg\min_{\boldsymbol{\beta}} \|\mathbf{W}^{1/2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\|^2 + \lambda \|\boldsymbol{\beta}\|^2.$$

The solution to this system has the form

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{W}\mathbf{X} + \lambda \mathbf{I})^{-1}\mathbf{X}'\mathbf{W}\mathbf{y}. \tag{4.1.7}$$

The solutions given above do not have good numerical properties, so the regularized weighted least squares problem is typically solved using the Singular Value Decomposition due to its high numerical stability [11].

4.2 Optimization for L_1 Uplift Support Vector Machines

In this section we describe optimization algorithms for L_1 Uplift Support Vector Machines. The first approach uses the CVXOPT library [2] and the second is an adaptation of the stochastic dual coordinate descent method.

4.2.1 Quadratic programming solution to Uplift Support Vector Machine optimization problem

We now present a solution to the L_1 USVM optimization problem using quadratic programming routines from the CVXOPT library [2]. The library works by solving, during each iteration, a system of equations called the KKT system; it is possible to provide a custom solver for the system for improved computation speed and numerical accuracy. We now derive solutions for the KKT equations which exploit special structure of the L_1 USVM optimization problem. The KKT system described below follows the conventions used by the CVXOPT library [2].

It is easy to see that the task of maximizing the Lagrangian (3.2.6) subject to constraints (3.2.7)−(3.2.9) can be rewritten in matrix form as minimizing

$$\frac{1}{2}\mathbf{u}'\mathbf{P}\mathbf{u} + \mathbf{q}'\mathbf{u} \qquad \text{subject to} \qquad \mathbf{G}\mathbf{u} \leq \mathbf{h}, \;\; \mathbf{A}\mathbf{u} = \mathbf{b},$$

where $\leq$ means elementwise inequality of vectors, $'$ denotes matrix transpose and

$$\mathbf{u} = \left[\begin{array}{c} \boldsymbol{\alpha} \\ \hline \boldsymbol{\beta} \end{array}\right], \quad \mathbf{P} = \left[\begin{array}{c|c} \mathbf{DD}' & \mathbf{DD}' \\ \hline \mathbf{DD}' & \mathbf{DD}' \end{array}\right], \quad \mathbf{A} = \left[\begin{array}{c|c} \mathbf{z}' & 0 \\ \hline 0 & \mathbf{z}' \end{array}\right], \quad \mathbf{G} = \left[\begin{array}{c} -\mathbf{I} \\ \hline \mathbf{I} \end{array}\right],$$

where $\mathbf{I}$ is the $2n \times 2n$ identity matrix, $\mathbf{q} = (1, 1, \ldots, 1)'$, the vector $\mathbf{h}$ is obtained from Equations 3.2.7 and 3.2.8, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)'$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_n)'$ are column vectors of the optimized dual coefficients, $\mathbf{z} = (z_1, \ldots, z_n)'$ is the vector of transformed class variables in treatment and control groups (Equation 3.2.1), and

$$\mathbf{D} = \mathrm{diag}(\mathbf{z}) \left[\begin{array}{c} \mathbf{D}^T \\ \hline \mathbf{D}^C \end{array}\right],$$

i.e. it is the concatenation of the treatment and control datasets with each row multiplied by the transformed class value z_i.

Each CVXOPT iteration requires solving the following KKT system of equations [2]

$$\left[\begin{array}{c|c|c} \mathbf{P} & \mathbf{A}' & \mathbf{G}'\mathbf{W}^{-1} \\ \hline \mathbf{A} & 0 & 0 \\ \hline \mathbf{G} & 0 & -\mathbf{W}' \end{array}\right] \left[\begin{array}{c} \mathbf{u}_x \\ \hline \mathbf{u}_y \\ \hline \mathbf{u}_z \end{array}\right] = \left[\begin{array}{c} \mathbf{b}_x \\ \hline \mathbf{b}_y \\ \hline \mathbf{b}_z \end{array}\right], \tag{4.2.1}$$

where the diagonal weight matrix $\mathbf{W}$ and vectors $\mathbf{b}_x, \mathbf{b}_y, \mathbf{b}_z$ are supplied by the solver. All diagonal elements of $\mathbf{W}$ are guaranteed to be positive, so the matrix is invertible. Note that the dimension of $\mathbf{W}$ is $4n \times 4n$. The structure of this system needs to be exploited if an efficient solution is to be obtained. Applying the Schur complement for $\mathbf{W}'$ in the leftmost matrix in (4.2.1) reduces the system to a smaller one

$$\left[\begin{array}{c|c} \mathbf{P} + \mathbf{G}'\mathbf{W}^{-2}\mathbf{G} & \mathbf{A}' \\ \hline \mathbf{A} & 0 \end{array}\right] \cdot \left[\begin{array}{c} \mathbf{u}_x \\ \hline \mathbf{u}_y \end{array}\right] = \left[\begin{array}{c} \mathbf{c}_x \\ \hline \mathbf{b}_y \end{array}\right], \text{ where } \mathbf{c}_x = \mathbf{b}_x + \mathbf{G}'\mathbf{W}^{-2}\mathbf{b}_z.$$

$\mathbf{u}_z$ can then be recovered as $\mathbf{u}_z = \mathbf{W}^{-1}(\mathbf{G}\mathbf{u}_x - \mathbf{b}_z)$. Using now the Schur complement[1] of $\mathbf{P} + \mathbf{G}'\mathbf{W}^{-2}\mathbf{G}$ we reduce the system further to

$$-\mathbf{A}(\mathbf{P} + \mathbf{G}'\mathbf{W}^{-2}\mathbf{G})^{-1}\mathbf{A}'\mathbf{u}_y = \mathbf{b}_y - \mathbf{A}(\mathbf{P} + \mathbf{G}'\mathbf{W}^{-2}\mathbf{G})^{-1}\mathbf{c}_x \tag{4.2.2}$$

[1] From (4.2.4) it follows that $\mathbf{P} + \mathbf{G}'\mathbf{W}^{-2}\mathbf{G}$ is a sum of a nonnegative definite and two positive definite matrices and is thus positive definite and invertible.

and solve

$$(\mathbf{P} + \mathbf{G}'\mathbf{W}^{-2}\mathbf{G})\mathbf{u}_x = \mathbf{c}_x - \mathbf{A}'\mathbf{u}_y \tag{4.2.3}$$

to recover $\mathbf{u}_x$. The above system of equations requires solving three linear systems of the form $(\mathbf{P} + \mathbf{G}'\mathbf{W}^{-2}\mathbf{G})\mathbf{v} = \mathbf{b}$ for various $\mathbf{b}$. The first two solutions are needed to compute $(\mathbf{P} + \mathbf{G}'\mathbf{W}^{-2}\mathbf{G})^{-1}\mathbf{A}'$ and $(\mathbf{P} + \mathbf{G}'\mathbf{W}^{-2}\mathbf{G})^{-1}\mathbf{c}_x$ in (4.2.2) with $\mathbf{b}$ equal respectively to $\mathbf{A}'$ and $\mathbf{c}_x$. The third occurs directly in (4.2.3).

In order to solve the system efficiently we need to exploit the structure of the matrix $\mathbf{P} + \mathbf{G}'\mathbf{W}^{-2}\mathbf{G}$. Note that it can be expressed as

$$\left[\begin{array}{c}\mathbf{D}\\ \hline \mathbf{D}\end{array}\right][\mathbf{D}'|\mathbf{D}'] + \mathbf{W}_1^{-2} + \mathbf{W}_2^{-2}, \tag{4.2.4}$$

where $\mathbf{W}_i^{-2}$ are the diagonal blocks of $\mathbf{W}^{-2}$ (recall that $\mathbf{W}$ is diagonal). This matrix has a 'diagonal plus low rank' structure which frequently occurs in optimization problems (see e.g. [5, Appendix C.4]). Denote $\mathbf{X} = [\mathbf{D}'|\mathbf{D}']'$, $\mathbf{Z} = \mathbf{W}_1^{-2} + \mathbf{W}_2^{-2}$. Solution to the system $(\mathbf{X}\mathbf{X}' + \mathbf{Z})\mathbf{v} = \mathbf{b}$ can be obtained using the Woodbury matrix identity given in Equation 4.1.3:

$$\mathbf{v} = \mathbf{Z}^{-1}\mathbf{b} - \mathbf{Z}^{-1}\mathbf{X}(\mathbf{I} + \mathbf{X}'\mathbf{Z}^{-1}\mathbf{X})^{-1}\mathbf{X}'\mathbf{Z}^{-1}\mathbf{b}. \tag{4.2.5}$$

Applying this formula directly allows for solving the KKT system efficiently, it is however known to have poor numerical stability. In [10] the authors suggested the use of partial Cholesky decomposition for such systems, but this decomposition is not available in standard linear algebra packages. Instead we noticed that computing the quantity $(\mathbf{I} + \mathbf{X}'\mathbf{Z}^{-1}\mathbf{X})^{-1}\mathbf{X}'\mathbf{Z}^{-1}\mathbf{b}$ in Equation 4.2.5 is equivalent to solving a regularized weighted least squares problem (see Equation 4.1.7), which can be achieved using the highly stable Singular Value Decomposition. This is the method we used in our implementation.

4.2.2 Stochastic dual coordinate descent solver for the L_1-USVM optimization problem

We have also developed an optimization algorithm based on the stochastic dual coordinate descent approach used for classical SVMs in the LIBLINEAR library [16]. However (contrary to classification SVMs) the algorithm worked worse than the quadratic programming algorithm and the convergence was often slow. Its description is included below for completeness.

The method works by solving the dual optimization problem for each of the dual coefficients α_i,

1. $\mathbf{w} \leftarrow \sum_{i=1}^{n}(\alpha_i + \beta_i) z_i \mathbf{x}_i$
2. $\boldsymbol{\gamma} \leftarrow (\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_n)$
3. Repeat until convergence:
4. For $j \in \{1, 2\}$:
5. For $i \in \{1, \ldots, n\}$:
6. $G \leftarrow z_i \langle \mathbf{w}, \mathbf{x}_i \rangle - 1$
7. $U_i \leftarrow C_1 \mathbb{1}_{[z_i = -1]} + C_2 \mathbb{1}_{[z_i = +1]}$
8. $k \leftarrow n(j-1) + i$
9. $$PG = \begin{cases} \min(G, 0) & \text{if } \gamma_k = 0 \\ \max(G, 0) & \text{if } \gamma_k = U_i \\ G & \text{otherwise} \end{cases}$$
10. If $PG \neq 0$:
11. $\bar{\gamma}_k \leftarrow \gamma_k$
12. $\gamma_k \leftarrow \min(\max(\gamma_k - G / \langle \mathbf{x}_i, \mathbf{x}_i \rangle, 0), U_i)$
13. $\mathbf{w} \leftarrow \mathbf{w} + (\gamma_k - \bar{\gamma}_k) z_i \mathbf{x}_i$

Figure 4.2.1: The dual coordinate descent method for the Uplift SVM optimization problem

β_i in turn in random order. The algorithm for our problem is similar to the original formulation in [16]. Necessary adaptations included taking into account the presence of two sets of dual coefficients and using new constraints on those coefficients given by (3.2.7) and (3.2.8). The algorithm is presented in Figure 4.2.1. A detailed derivation for the classical case can be found in [16]. For simplicity, dual coefficients are updated sequentially in the figure, in the actual implementation loops in steps 4 and 5 are executed in random order. The notation differs slightly from that used in the previous section to make it easier to compare with the description in [16].

Note that the method solves an unbiased version of the SVM optimization problem, namely one which assumes the intercept to be zero. Nonzero intercepts are handled by adding an additional constant column to the data, which is set to some large value in order to avoid regularizing the corresponding coefficient [16]. In our case, this resulted in dropping the constraints (3.2.9) and adding two extra variables to the data to emulate the two intercepts b_1 and b_2. One of the new variables is zero in $\mathbf{D}^T$ and equal to c in $\mathbf{D}^C$, the other is zero in $\mathbf{D}^C$ and equal to c in $\mathbf{D}^T$, where

c is a constant, in our case equal to 10.

The following theorem establishes the convergence of the algorithm.

Theorem 4.2.1. The algorithm in Figure 4.2.1 globally converges to an optimal solution. The convergence rate is at least linear.

Proof. Note that the Lagrangian (3.2.6) can be written as

$$-\frac{1}{2}\boldsymbol{\gamma}^T\mathbf{Q}\boldsymbol{\gamma} + \mathbf{e}^T\boldsymbol{\gamma}, \tag{4.2.6}$$

where $\mathbf{e} = (1, 1, \ldots, 1)$ is a vector of $2n$ ones, the vector $\boldsymbol{\gamma}$ is defined as $\boldsymbol{\gamma} = (\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_n)$, and the matrix $\mathbf{Q}$ as

$$\mathbf{Q} = \left[\begin{array}{c|c} \mathbf{R} & \mathbf{R} \\ \hline \mathbf{R} & \mathbf{R} \end{array}\right],$$

where $\mathbf{R}_{ij} = z_i z_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle$. The quantity in Equation 4.2.6 needs to be maximized subject to constraints (3.2.7) and (3.2.8). After rewriting in this form, the argument given in the proof of Theorem 1 in [16] can be directly applied. □

4.3 L_p uplift SVM optimization

We now describe the convex programming solution to dual form of the L_p-USVM optimization problem.

The L_p-USVM optimization problem is no longer (except for $p = 2$) quadratic so we used CVXOPT's convex optimization routine to solve it [2]. Nevertheless, the solution is similar to that presented in Section 4.2.1. The KKT system still has the form given in Equation 4.2.1 with the matrix $\mathbf{P}$ replaced by the Hessian $\mathbf{H}$ of the goal function. Moreover the matrices $\mathbf{P}$ and $\mathbf{H}$ have similar structures. We begin by deriving the gradient and the Hessian of the goal function. To simplify notation define:

$$k_{i,1} = \left(pC_1\mathbb{1}_{[z_i=+1]} + pC_2\mathbb{1}_{[z_i=-1]}\right)^{1/(p-1)},$$
$$k_{i,2} = \left(pC_1\mathbb{1}_{[z_i=-1]} + pC_2\mathbb{1}_{[z_i=+1]}\right)^{1/(p-1)}.$$

Please note that, instead of maximizing the Lagrange function L given in Equation 3.4.13, we minimize $-L$ which, just for notational convenience, we will denote by $\mathcal{L}$. We calculate the

gradient

$$\frac{\partial \mathcal{L}}{\partial \alpha_k} = \sum_{i=1}^{n} \alpha_i z_k z_i \langle \mathbf{x}_k, \mathbf{x}_i \rangle + \sum_{i=1}^{n} \beta_i z_k z_i \langle \mathbf{x}_k, \mathbf{x}_i \rangle - 1 + \frac{\alpha_k^{1/(p-1)}}{k_{i,1}},$$
$$\frac{\partial \mathcal{L}}{\partial \beta_k} = \sum_{i=1}^{n} \beta_i z_k z_i \langle \mathbf{x}_k, \mathbf{x}_i \rangle + \sum_{i=1}^{n} \alpha_i z_k z_i \langle \mathbf{x}_k, \mathbf{x}_i \rangle - 1 + \frac{\beta_k^{1/(p-1)}}{k_{i,2}}$$

and the Hessian

$$\begin{aligned}
\frac{\partial^2 \mathcal{L}}{\partial \alpha_k \partial \alpha_k} &= z_k^2 \langle \mathbf{x}_k, \mathbf{x}_k \rangle + \frac{\alpha_k^{(2-p)/(p-1)}}{(p-1)k_{i,1}}, \\
\frac{\partial^2 \mathcal{L}}{\partial \alpha_k \partial \alpha_l} &= z_k z_l \langle \mathbf{x}_k, \mathbf{x}_l \rangle \qquad \text{for } k \neq l, \\
\frac{\partial^2 \mathcal{L}}{\partial \alpha_k \partial \beta_l} &= \frac{\partial^2 \mathcal{L}}{\partial \beta_k \partial \alpha_l} = z_k z_l \langle \mathbf{x}_k, \mathbf{x}_l \rangle, \\
\frac{\partial^2 \mathcal{L}}{\partial \beta_k \partial \beta_k} &= z_k^2 \langle \mathbf{x}_k, \mathbf{x}_k \rangle + \frac{\beta_k^{(2-p)/(p-1)}}{(p-1)k_{i,2}}, \\
\frac{\partial^2 \mathcal{L}}{\partial \beta_k \partial \beta_l} &= z_k z_l \langle \mathbf{x}_k, \mathbf{x}_l \rangle \qquad \text{for } k \neq l.
\end{aligned}$$

Both the gradient and the Hessian can be expressed in concise matrix form respectively as

$$\mathbf{P}\mathbf{u} + \mathbf{d}_g, \qquad \mathbf{P} + \operatorname{diag}(\mathbf{d}_h),$$

where the matrix $\mathbf{P}$ and the vector of dual coefficients $\mathbf{u}$ are defined as in Section 4.2.1, and the vectors $\mathbf{d}_g$, $\mathbf{d}_h$ are defined as follows

$$\mathbf{d}_g = \left(\frac{\alpha_1^{1/(p-1)}}{k_{1,1}} - 1, \ldots, \frac{\alpha_n^{1/(p-1)}}{k_{n,1}} - 1, \frac{\beta_1^{1/(p-1)}}{k_{1,2}} - 1, \ldots, \frac{\beta_n^{1/(p-1)}}{k_{n,2}} - 1 \right),$$
$$\mathbf{d}_h = \left(\frac{\alpha_1^{(2-p)/(p-1)}}{(p-1)k_{1,1}}, \ldots, \frac{\alpha_n^{(2-p)/(p-1)}}{(p-1)k_{n,1}}, \frac{\beta_1^{(2-p)/(p-1)}}{(p-1)k_{1,2}}, \ldots, \frac{\beta_n^{(2-p)/(p-1)}}{(p-1)k_{n,2}} \right).$$

During each iteration we need to solve a KKT system very similar to (4.2.1) with the matrix $\mathbf{P}$ replaced by the Hessian matrix given above. The system can be solved efficiently using a procedure almost identical to that presented in Section 4.2.1. Details have thus been omitted.

Chapter 5

Székely regularized Support Vector Machines

5.1 Biased treatment assignment problem

A very important aspect of uplift modeling is how the cases are assigned to treatment and control groups. The best scenario is a randomized controlled experiment, where the assignment is random and, therefore, does not depend on neither observed nor unobserved features of the cases. Unfortunately, such an experiment is not always possible (e.g. for ethical or financial reasons) or only historical data may be available where, for example, the treatment was applied to patients which the doctor considered most suitable.

If treatment assignment was not random and biased then the effect of the action cannot, usually, be estimated directly. Consider, for example, a medical treatment with potentially severe side effects. The doctor might then decide not to apply it to patients in serious condition who will thus be placed in the control group. However, such cases are also less likely to recover from the disease making the control group outcomes look worse and the treatment more effective than it is in reality.

In this chapter we present Uplift Support Vector Machines originally proposed in Chapter 3 with an additional penalty term, which we call the *Székely regularizer*. As a result, we obtain uplift models which are additionally forced to make similar predictions in the treatment and control groups, thus helping to reduce the effect of treatment assignment bias.

The additional regularizer is based on so called *energy distance* between probability distributions which was proposed by Székely and Rizzo [22, 44, 45]. The distance has the property that

it is zero (in the population setting) if and only if the distributions are identical, it can thus be used to enforce similar distributions of model scores in the treatment and control groups.

5.2 Székely regularized Support Vector Machines

One way to view an uplift model is as a function which maps feature vectors into the set $\{-1, 0, +1\}$ as we did in Chapter 3. The value is interpreted as a decision on whether the action applied to a given case will be beneficial, neutral, or detrimental. Another approach is for the model to return a *score*: a real number being an increasing function of the predicted probability that the action will be beneficial. In this chapter we are going to define our regularized Uplift Support Vector Machines using the discrete prediction model but for testing and regularization purposes we will use the linear score $\langle \mathbf{w}, \mathbf{x} \rangle$.

5.2.1 Distributions of scores in controlled randomized experiments

Let us now state an important property of score based uplift models used in controlled randomized experiments. Let M_s be an uplift model returning a score and $M_s(\mathbf{x})$ the score returned by the model for a specific instance $\mathbf{x}$. When the feature vector $\mathbf{x}$ is picked at random from the population distribution, then $M_s(\mathbf{x})$ is a random variable. Suppose $\mathbf{x}^T$ is picked at random from the treatment population and $\mathbf{x}^C$ from the control population. In a randomized controlled trial $\mathbf{x}^T$ and $\mathbf{x}^C$ follow the same distribution and therefore *$M_s(\mathbf{x}^T)$ and $M_s(\mathbf{x}^C)$ are random variables following the same distribution.* If the treatment assignment is not random, the distributions of $\mathbf{x}^T$ and $\mathbf{x}^C$ differ and so may those of $M_s(\mathbf{x}^T)$ and $M_s(\mathbf{x}^C)$.

In this chapter we will use this property to obtain models which are less sensitive to treatment assignment bias. This will be achieved by adding a regularization term penalizing models which yield different score distributions in the treatment and control training sets.

5.2.2 The energy distance

In this chapter we make use the concept of *energy distance* (also called *E-statistics*) $e^{(\alpha)}$ proposed in 2005 by Székely and Rizzo [22, 44, 45, 46]. Initially this concept was introduced as a measure of distance between clusters, but it is in fact a general statistical distance between two or more probability distributions or samples. The name comes from the fact that it was first derived for applications in physics; later Székely applied this concept to statistics.

Let $A = \{\mathbf{a}_1, \mathbf{a}_2, \dots \mathbf{a}_{n_1}\}$, $B = \{\mathbf{b}_1, \mathbf{b}_2, \dots \mathbf{b}_{n_2}\}$ be two nonempty sets of points in $\mathbb{R}^d$. Formally, $e^{(\alpha)}(A, B)$ is defined as

$$\begin{aligned} e^{(\alpha)}(A,B) =& \frac{n_1 n_2}{n_1 + n_2}\Bigg[\frac{2}{n_1 n_2}\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\|\mathbf{a}_i - \mathbf{b}_j\|^\alpha \\ & -\frac{1}{n_1^2}\sum_{i=1}^{n_1}\sum_{j=1}^{n_1}\|\mathbf{a}_i - \mathbf{a}_j\|^\alpha - \frac{1}{n_2^2}\sum_{i=1}^{n_2}\sum_{j=1}^{n_2}\|\mathbf{b}_i - \mathbf{b}_j\|^\alpha\Bigg], \end{aligned} \tag{5.2.1}$$

where $\|\cdot\|$ is the Euclidean norm and α is parameter that influences the behavior of the distance. If $\alpha = 2$ then the distance is equal to zero if and only if the means of A and B are equal. The case $\alpha \in (0, 2)$ is more interesting, since the distance is then equal to zero if and only if the sets A and B are equal. Moreover, if A and B are random samples and $\alpha \in (0, 2)$, then, as the size of A and B grows to infinity, the distance between them tends to zero if and only if A and B are drawn from the same distribution (for $\alpha = 2$ the distributions from which they are drawn only need to have equal means). This property is important for the task the distance will be used for in this dissertation. Notice also that for $d = 1$ the Euclidean norms reduce to absolute values.

5.2.3 Model formulation

We modify the risk function of Uplift Support Vector Machines by adding an extra term responsible for penalizing the difference in score distributions in the treatment and control groups. We call this term the *Székely regularization term*. The version presented here has another, minor, difference compared to that given in Section 3.1: the soft margin penalties are averaged separately over the treatment and control groups. As a result both groups have the same impact on the optimized risk. The optimization problem is to find weights $\mathbf{w}$ maximizing the function $R(\mathbf{w})$ defined as

$$\begin{aligned} R(\mathbf{w}) = \frac{1}{2}\langle \mathbf{w}, \mathbf{w}\rangle &+ \frac{C_1}{n^T}\sum_{\mathbf{D}_+^T}\xi_{i,1} + \frac{C_2}{n^T}\sum_{\mathbf{D}_-^T}\xi_{i,1} + \frac{C_2}{n^T}\sum_{\mathbf{D}_+^T}\xi_{i,2} + \frac{C_1}{n^T}\sum_{\mathbf{D}_-^T}\xi_{i,2} \\ &+ \frac{C_1}{n^C}\sum_{\mathbf{D}_-^C}\xi_{i,1} + \frac{C_2}{n^C}\sum_{\mathbf{D}_+^C}\xi_{i,1} + \frac{C_2}{n^C}\sum_{\mathbf{D}_-^C}\xi_{i,2} + \frac{C_1}{n^C}\sum_{\mathbf{D}_+^C}\xi_{i,2} \\ &+ C_3 S(\mathbf{D}^T, \mathbf{D}^C, \mathbf{w}). \end{aligned} \tag{5.2.2}$$

The risk is optimized subject to the same constraints as in the USVM optimization problem, which are given in Equations 3.1.4–3.1.7. Above, $S(\mathbf{D}^T, \mathbf{D}^C, \mathbf{w})$ is the *Székely regularizer* given by

$$\begin{aligned} S(\mathbf{D}^T, \mathbf{D}^C, \mathbf{w}) =& \frac{2}{n^T n^C} \sum_{i=1}^{n^T} \sum_{j=1}^{n^C} |\langle \mathbf{w}, \mathbf{x}_i^T \rangle - \langle \mathbf{w}, \mathbf{x}_j^C \rangle|^\alpha \\ &- \frac{1}{(n^T)^2} \sum_{i=1}^{n^T} \sum_{j=1}^{n^T} |\langle \mathbf{w}, \mathbf{x}_i^T \rangle - \langle \mathbf{w}, \mathbf{x}_j^T \rangle|^\alpha \\ &- \frac{1}{(n^C)^2} \sum_{i=1}^{n^C} \sum_{j=1}^{n^C} |\langle \mathbf{w}, \mathbf{x}_i^C \rangle - \langle \mathbf{w}, \mathbf{x}_j^C \rangle|^\alpha. \end{aligned} \tag{5.2.3}$$

Note that $\langle \mathbf{w}, \mathbf{x}_i \rangle$ is the score assigned by the model to a data record $\mathbf{x}_i$, and (5.2.3) is thus the energy distance (5.2.1) applied to the sets of scores assigned by the model to records in the treatment and control groups. Due to the properties of the energy distance the term will penalize models for which distributions of scores in both groups differ. The factor C_3 determines the strength of the penalty. The fraction $\frac{n^T n^C}{n^T + n^C}$ from (5.2.1) is absorbed into C_3 for the ease of exposition.

Let us now discuss the choice of the exponent α. Since we want to guarantee equal score distributions we need $\alpha \in (0, 2)$ [45]. However for $\alpha < 1$ the function S exhibits strong non-convexity and is thus more difficult to optimize. We should, therefore, choose α from the interval $[1, 2)$. We found values close to 1 to work better in practice but for $\alpha = 1$ the function S is not differentiable. We thus settled for $\alpha = 1.1$ which gives good properties and a smoother function to optimize. Note, however that S may not be convex even for $\alpha \in [1, 2)$.

5.2.4 Properties of Székely regularized Uplift Support Vector Machines

All three lemmas presented in Section 3.3 are still valid in case of Székely regularized USVMs, only small modifications to the proofs were needed. The key observation is that intercepts b_1, b_2 are independent from the Székely regularizer term $S(\mathbf{D}^T, \mathbf{D}^C, \mathbf{w})$.

Lemma 5.2.1. Let $\mathbf{w}^*, b_1^*, b_2^*$ be a solution to the Uplift SVM optimization problem given by Equations 5.2.2 and constraints 3.1.4–3.1.8. If $C_2 > C_1$ then $b_1^* \geq b_2^*$.

Proof. Let $S^* = \langle \mathbf{w}^*, b_1^*, b_2^* \rangle$ be an optimal solution with $b_1^* < b_2^*$. Consider also a set of parameters $S' = \langle \mathbf{w}^*, b_2^*, b_1^* \rangle$ with the values of b_1^*, b_2^* interchanged and look at the target function (5.2.2) for both sets of parameters.

Take a point $(\mathbf{x}_i, y_i) \in \mathbf{D}_+^T$ for which, under the set of parameters S', $\xi'_{i,1} > 0$ and $\xi'_{i,2} = 0$, that is the point is penalized only for crossing the hyperplane H_1. Under the parameters S^* the point will be penalized not with $\frac{C_1}{n^T}\xi'_{i,1}$ for crossing H_1 but, instead, with $\frac{C_2}{n^T}\xi^*_{i,2}$ for crossing H_2. Since, by switching from S^* to S' the hyperplanes simply exchange intercepts, we have $\xi^*_{i,1} = \xi'_{i,2}$ and, from the assumption, $\frac{C_2}{n^T}\xi^*_{i,1} > \frac{C_1}{n^T}\xi'_{i,2}$. Thus the amount every point $(\mathbf{x}_i, y_i) \in \mathbf{D}_+^T$ contributes to the target function (5.2.2) is lower in S' than in S^*.

By a similar argument (see proof of Lemma 3.3.1) one can see that for a point $(\mathbf{x}_i, y_i) \in \mathbf{D}_+^T$ for which, under S', $\xi'_{i,1}, \xi'_{i,2} > 0$ (i.e. it is penalized for crossing both hyperplanes) a switch to parameter set S^* increases the target function by $(\xi^*_{i,1} - \xi^*_{i,2})(\frac{C_1 - C_2}{n^T}) > 0$.

Analogous arguments hold for points in $\mathbf{D}_-^T$, $\mathbf{D}_+^C$, and $\mathbf{D}_-^C$ contradicting the optimality of S^*. □

Lemma 5.2.2. For sufficiently large value of $\frac{C_2}{C_1}$ none of the observations is penalized with a term involving the C_2 factor in the solution to the Székely regularized USVM optimization problem.

Proof. Let us first consider only the hyperplane H_1. Assume that there exists at least one point in $\mathbf{D}_-^T$ which is punished with a term involving the C_2 penalty coefficient, and therefore lies on the wrong side of H_1. Out of all such points choose the one $(\tilde{\mathbf{x}}_i, \tilde{y}_i)$ which is furthest from H_1 and denote by $\tilde{\xi}_{i,1}$, $\tilde{\xi}_{i,2}$ its slack variables w.r.t. H_1 and H_2 respectively. The penalty incurred by $(\tilde{\mathbf{x}}_i, \tilde{y}_i)$ equals

$$\frac{C_2}{n^T}\tilde{\xi}_{i,1} + \frac{C_1}{n^T}\tilde{\xi}_{i,2}.$$

Let us now shift the hyperplane H_1 by exactly $\tilde{\xi}_{i,1}$; as a result, the point is only penalized by $\frac{C_1}{n^T}\tilde{\xi}_{i,2}$. The same is true for all other points from $\mathbf{D}_-^T$. On the other hand, after shifting H_1, penalties w.r.t. H_1 of points in $\mathbf{D}_+^T \cup \mathbf{D}_-^C$ could have increased, but the increase is bounded by $\frac{C_1}{\min\{n^T, n^C\}}\tilde{\xi}_{i,1}$ per point.

Denote $n_1 = |\mathbf{D}_-^T \cup \mathbf{D}_+^C|$, $n_2 = |\mathbf{D}_+^T \cup \mathbf{D}_-^C|$. The change in penalties caused by shifting H_1 is bounded from above by

$$\frac{C_1}{n^T}\tilde{\xi}_{i,2} - \left(\frac{C_2}{n^T}\tilde{\xi}_{i,1} + \frac{C_1}{n^T}\tilde{\xi}_{i,2}\right) + \frac{n_2 C_1}{\min\{n^T, n^C\}}\tilde{\xi}_{i,1} = \tilde{\xi}_{i,1}\left(\frac{n_2 C_1}{\min\{n^T, n^C\}} - \frac{C_2}{n^T}\right),$$

which is negative for sufficiently large value of C_2, such that shifting H_1 is guaranteed to decrease the target function. Analogous results hold for points in $\mathbf{D}_+^C$, $\mathbf{D}_-^C$, and $\mathbf{D}_+^T$ completing the proof. □

Lemma 5.2.3. If $C_1 = C_2 = C$ and the solution is unique then both hyperplanes coincide: $b_1 = b_2$.

The proof is the same as that of Lemma 3.3.3 since the Székely regularizer does not depend on b_1 and b_2.

5.3 Optimization

We now describe the method used to optimize (5.2.2) subject to the constraints given in Equations 3.1.4–3.1.7. As a first step we rewrite the problem as an unconstrained optimization problem using the hinge loss:

$$\begin{aligned} R(\mathbf{w}) =& \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle \\ &+ \frac{C_1}{n^T} \sum_{\mathbf{D}_+^T} h(y_i^T(\langle \mathbf{w}, \mathbf{x}_i^T \rangle - b_1)) + \frac{C_2}{n^T} \sum_{\mathbf{D}_+^T} h(y_i^T(\langle \mathbf{w}, \mathbf{x}_i^T \rangle - b_2)) \\ &+ \frac{C_2}{n^T} \sum_{\mathbf{D}_-^T} h(y_i^T(\langle \mathbf{w}, \mathbf{x}_i^T \rangle - b_1)) + \frac{C_1}{n^T} \sum_{\mathbf{D}_-^T} h(y_i^T(\langle \mathbf{w}, \mathbf{x}_i^T \rangle - b_2)) \\ &+ \frac{C_2}{n^C} \sum_{\mathbf{D}_+^C} h(-y_i^C(\langle \mathbf{w}, \mathbf{x}_i^C \rangle - b_1)) + \frac{C_1}{n^C} \sum_{\mathbf{D}_+^C} h(-y_i^C(\langle \mathbf{w}, \mathbf{x}_i^C \rangle - b_2)) \\ &+ \frac{C_1}{n^C} \sum_{\mathbf{D}_-^C} h(-y_i^C(\langle \mathbf{w}, \mathbf{x}_i^C \rangle - b_1)) + \frac{C_2}{n^C} \sum_{\mathbf{D}_-^C} h(-y_i^C(\langle \mathbf{w}, \mathbf{x}_i^C \rangle - b_2)) \\ &+ C_3 S(\mathbf{D}^T, \mathbf{D}^C, \mathbf{w}), \end{aligned} \tag{5.3.1}$$

where h is the *hinge loss* function given by

$$h(q) = \max\{0, 1 - q\}.$$

To see why such a rewrite is possible fix the vector $\mathbf{w}$. The target function then depends only on $\xi_{i,j}$ and, due to constraints, attains a minimum value for $\xi_{i,j} = h(y_i^T(\langle \mathbf{w}, \mathbf{x}_i^T \rangle - b_j))$ for points in $\mathbf{D}^T$ and $\xi_{i,j} = h(-y_i^C(\langle \mathbf{w}, \mathbf{x}_i^C \rangle - b_j))$ for points in $\mathbf{D}^C$. A similar argument for classical SVMs can be found in [9].

We are going to use the Averaged Stochastic Gradient Descent algorithm [29] in order to optimize (5.2.2). The reason is that the algorithm is fast, stable and works well with non-smooth functions. Note that in our optimization problem the derivatives of the target function are not guaranteed to exist so methods such as conjugate gradient descent are not applicable.

In order to optimize the expression given in (5.3.1) we first need to compute its subgradient (since $R(\mathbf{w})$ is not everywhere differentiable there are values of $\mathbf{w}$ for which the gradient does not exist). Note that the subgradient of the hinge loss $h(q)$ is

$$\frac{\partial h(q)}{\partial q} = \begin{cases} -1 & \text{if } q < 1, \\ \text{any value in } [-1, 0] & \text{if } q = 1, \\ 0 & \text{if } q > 1. \end{cases}$$

Since for $q = 1$ any value in $[-1, 0]$ can be picked we will simply set

$$\frac{\partial h(q)}{\partial q} = -\mathbb{1}_{[1-q>0]}. \tag{5.3.2}$$

We can now give the expression for the subgradient of the minimized risk given in (5.3.1)

$$\begin{aligned} \frac{\partial R(\mathbf{w})}{\partial \mathbf{w}} = \mathbf{w} &- \frac{C_1}{n^T} \sum_{\mathbf{D}_+^T} \mathbb{1}_{\left[1-y_i^T(\langle \mathbf{w}, \mathbf{x}_i^T \rangle - b_1)>0\right]} \mathbf{x}_i^T y_i^T - \frac{C_2}{n^T} \sum_{\mathbf{D}_+^T} \mathbb{1}_{\left[1-y_i^T(\langle \mathbf{w}, \mathbf{x}_i^T \rangle - b_2)>0\right]} \mathbf{x}_i^T y_i^T \\ &- \frac{C_2}{n^T} \sum_{\mathbf{D}_-^T} \mathbb{1}_{\left[1-y_i^T(\langle \mathbf{w}, \mathbf{x}_i^T \rangle - b_1)>0\right]} \mathbf{x}_i^T y_i^T - \frac{C_1}{n^T} \sum_{\mathbf{D}_-^T} \mathbb{1}_{\left[1-y_i^T(\langle \mathbf{w}, \mathbf{x}_i^T \rangle - b_2)>0\right]} \mathbf{x}_i^T y_i^T \\ &+ \frac{C_2}{n^C} \sum_{\mathbf{D}_+^C} \mathbb{1}_{\left[1+y_i^C(\langle \mathbf{w}, \mathbf{x}_i^C \rangle - b_1)>0\right]} \mathbf{x}_i^C y_i^C + \frac{C_1}{n^C} \sum_{\mathbf{D}_+^C} \mathbb{1}_{\left[1+y_i^C(\langle \mathbf{w}, \mathbf{x}_i^C \rangle - b_2)>0\right]} \mathbf{x}_i^C y_i^C \\ &+ \frac{C_1}{n^C} \sum_{\mathbf{D}_-^C} \mathbb{1}_{\left[1+y_i^C(\langle \mathbf{w}, \mathbf{x}_i^C \rangle - b_1)>0\right]} \mathbf{x}_i^C y_i^C + \frac{C_2}{n^C} \sum_{\mathbf{D}_-^C} \mathbb{1}_{\left[1+y_i^C(\langle \mathbf{w}, \mathbf{x}_i^C \rangle - b_2)>0\right]} \mathbf{x}_i^C y_i^C \\ &+ C_3 \frac{\partial S(\mathbf{D}^T, \mathbf{D}^C, \mathbf{w})}{\partial \mathbf{w}}. \end{aligned} \tag{5.3.3}$$

Since for $\alpha > 1$, $|x|^\alpha$ is differentiable for all x including 0 we have

$$\begin{aligned}
&\frac{\partial S(\mathbf{D}^T, \mathbf{D}^C, \mathbf{w})}{\partial \mathbf{w}} \\
&= \frac{2}{n^T n^C} \sum_{i=1}^{n^T} \sum_{j=1}^{n^C} \alpha |\langle \mathbf{w}, \mathbf{x}_i^T - \mathbf{x}_j^C \rangle|^{\alpha-1} \mathrm{sgn}(\langle \mathbf{w}, \mathbf{x}_i^T - \mathbf{x}_j^C \rangle)(\mathbf{x}_i^T - \mathbf{x}_j^C) \\
&\quad - \frac{1}{(n^T)^2} \sum_{i=1}^{n^T} \sum_{j=1}^{n^T} \alpha |\langle \mathbf{w}, \mathbf{x}_i^T - \mathbf{x}_j^T \rangle|^{\alpha-1} \mathrm{sgn}(\langle \mathbf{w}, \mathbf{x}_i^T - \mathbf{x}_j^T \rangle)(\mathbf{x}_i^T - \mathbf{x}_j^T) \\
&\quad - \frac{1}{(n^C)^2} \sum_{i=1}^{n^C} \sum_{j=1}^{n^C} \alpha |\langle \mathbf{w}, \mathbf{x}_i^C - \mathbf{x}_j^C \rangle|^{\alpha-1} \mathrm{sgn}(\langle \mathbf{w}, \mathbf{x}_i^C - \mathbf{x}_j^C \rangle)(\mathbf{x}_i^C - \mathbf{x}_j^C). \qquad (5.3.4)
\end{aligned}$$

5.3.1 Averaged Stochastic Gradient Descent algorithm for Székely regularized USMVs

The Stochastic Gradient Descent algorithm typically works by picking random data points, computing the contribution of those points to the gradient and updating current weights with a decreasing update coefficient.

Notice however, that each term in the Székely regularizer given in (5.2.3) operates on a pair of treatment data points and a pair of control data points. In order to apply a stochastic optimization algorithm to the problem we thus take, at each iteration, four randomly selected records, two from the treatment training set and two from control. The weight update is then computed based on four training points instead of one.

The algorithm is given in Figure 5.3.1. The expressions for $\partial S(\mathbf{x}_i^T, \mathbf{x}_j^T, \mathbf{x}_k^C, \mathbf{x}_l^C, y_i^T, y_j^T, y_k^C, y_l^C, \mathbf{w})/\partial \mathbf{w}$ and $\partial l(\mathbf{w}, \mathbf{x}_i^T, y_i^T)/\partial \mathbf{w}$ used in the algorithm will be given below. Notice that in step 10 we take the average of the weight vectors $\mathbf{w}_t$ obtained during all steps of the algorithm. This is the so called Polyak-Ruppert averaging [3, 23, 29] which improves the convergence properties of the algorithm.

In order to provide the expressions for $\partial l/\partial \mathbf{w}$ and $\partial S/\partial \mathbf{w}$ used in the algorithm, as well as to prove its convergence, we first need to compute the contribution of each random sample to the subgradient of the target risk function (5.3.4). Since we are dealing with pairs of treatment and control points, each sample will involve four data records: $\mathbf{x}_i^T, \mathbf{x}_j^T, \mathbf{x}_k^C, \mathbf{x}_l^C$ and their corresponding class values $y_i^T, y_j^T, y_k^C, y_l^C$. The subgradient of the risk for the given sample is given by the

1. $\mathbf{w}_0 = 0$
2. For $t \leftarrow 1, 2, \ldots$
3. Draw two samples $(\mathbf{x}_i^T, y_i^T)$, $(\mathbf{x}_j^T, y_j^T)$ uniformly at random from $\mathbf{D}^T$
4. Draw two samples $(\mathbf{x}_k^C, y_k^C)$, $(\mathbf{x}_l^C, y_l^C)$ uniformly at random from $\mathbf{D}^C$
5. $\mathbf{g} \leftarrow \mathbf{w}_{t-1} + \frac{1}{2}\frac{\partial l(\mathbf{w}_{t-1}, \mathbf{x}_i^T, y_i^T)}{\partial \mathbf{w}_{t-1}} + \frac{1}{2}\frac{\partial l(\mathbf{w}_{t-1}, \mathbf{x}_j^T, y_j^T)}{\partial \mathbf{w}_{t-1}}$
6. $\mathbf{g} \leftarrow \mathbf{g} + \frac{1}{2}\frac{\partial l(\mathbf{w}_{t-1}, \mathbf{x}_k^C, y_k^C)}{\partial \mathbf{w}_{t-1}} + \frac{1}{2}\frac{\partial l(\mathbf{w}_{t-1}, \mathbf{x}_l^C, y_l^C)}{\partial \mathbf{w}_{t-1}}$
7. $\mathbf{g} \leftarrow \mathbf{g} + C_3 \frac{\partial S(\mathbf{x}_i^T, \mathbf{x}_j^T, \mathbf{x}_k^C, \mathbf{x}_l^C, y_i^T, y_j^T, y_k^C, y_l^C, \mathbf{w}_{t-1})}{\partial \mathbf{w}_{t-1}}$
8. $\gamma_t \leftarrow \frac{1}{\sqrt{t}}$
9. $\mathbf{w}_t \leftarrow \mathbf{w}_{t-1} - \gamma_t \mathbf{g}$
10. $\mathbf{w}^a \leftarrow \frac{1}{t}\sum_{t'=1}^{t} \mathbf{w}_{t'}$
11. If converged:
12. Return $\mathbf{w}^a$

Figure 5.3.1: The Averaged Stochastic Gradient Descent algorithm for Székely regularized Uplift Support Vector Machines.

following equation

$$\begin{aligned}\frac{\partial R(\mathbf{x}_i^T, \mathbf{x}_j^T, \mathbf{x}_k^C, \mathbf{x}_l^C, y_i^T, y_j^T, y_k^C, y_l^C, \mathbf{w})}{\partial \mathbf{w}} = \\ \mathbf{w} + \frac{1}{2}\frac{\partial l(\mathbf{w}, \mathbf{x}_i^T, y_i^T)}{\partial \mathbf{w}} + \frac{1}{2}\frac{\partial l(\mathbf{w}, \mathbf{x}_j^T, y_j^T)}{\partial \mathbf{w}} \\ + \frac{1}{2}\frac{\partial l(\mathbf{w}, \mathbf{x}_k^C, y_k^C)}{\partial \mathbf{w}} + \frac{1}{2}\frac{\partial l(\mathbf{w}, \mathbf{x}_l^C, y_l^C)}{\partial \mathbf{w}} \\ + C_3 \frac{\partial S(\mathbf{x}_i^T, \mathbf{x}_j^T, \mathbf{x}_k^C, \mathbf{x}_l^C, y_i^T, y_j^T, y_k^C, y_l^C, \mathbf{w})}{\partial \mathbf{w}},\end{aligned} \tag{5.3.5}$$

where the parts resulting from differentiating the hinge loss are

$$\frac{\partial l(\mathbf{w},\mathbf{x},y)}{\partial \mathbf{w}} = \mathbf{x}y \cdot \begin{cases} -C_1 \mathbb{1}_{[1-y(\langle \mathbf{w},\mathbf{x}\rangle - b_1)>0]} - C_2 \mathbb{1}_{[1-y(\langle \mathbf{w},\mathbf{x}\rangle - b_2)>0]} & \text{if } (\mathbf{x},y) \in \mathbf{D}_+^T \\ -C_2 \mathbb{1}_{[1-y(\langle \mathbf{w},\mathbf{x}\rangle - b_1)>0]} - C_1 \mathbb{1}_{[1-y(\langle \mathbf{w},\mathbf{x}\rangle - b_2)>0]} & \text{if } (\mathbf{x},y) \in \mathbf{D}_-^T \\ C_1 \mathbb{1}_{[1+y(\langle \mathbf{w},\mathbf{x}\rangle - b_1)>0]} + C_2 \mathbb{1}_{[1+y(\langle \mathbf{w},\mathbf{x}\rangle - b_2)>0]} & \text{if } (\mathbf{x},y) \in \mathbf{D}_-^C \\ C_2 \mathbb{1}_{[1+y(\langle \mathbf{w},\mathbf{x}\rangle - b_1)>0]} + C_1 \mathbb{1}_{[1+y(\langle \mathbf{w},\mathbf{x}\rangle - b_2)>0]} & \text{if } (\mathbf{x},y) \in \mathbf{D}_+^C \end{cases} \tag{5.3.6}$$

and the part for the subgradient of the Székely regularizer is

$$\begin{aligned} &\frac{\partial S(\mathbf{x}_i^T, \mathbf{x}_j^T, \mathbf{x}_k^C, \mathbf{x}_l^C, y_i^T, y_j^T, y_k^C, y_l^C, \mathbf{w})}{\partial \mathbf{w}} \\ &\quad = \frac{\alpha}{2}\big[|\langle \mathbf{w}, \mathbf{x}_i^T - \mathbf{x}_k^C\rangle|^{\alpha-1}\mathrm{sgn}(\langle \mathbf{w}, \mathbf{x}_i^T - \mathbf{x}_k^C\rangle)(\mathbf{x}_i^T - \mathbf{x}_k^C) \\ &\qquad + |\langle \mathbf{w}, \mathbf{x}_i^T - \mathbf{x}_l^C\rangle|^{\alpha-1}\mathrm{sgn}(\langle \mathbf{w}, \mathbf{x}_i^T - \mathbf{x}_l^C\rangle)(\mathbf{x}_i^T - \mathbf{x}_l^C) \\ &\qquad + |\langle \mathbf{w}, \mathbf{x}_j^T - \mathbf{x}_k^C\rangle|^{\alpha-1}\mathrm{sgn}(\langle \mathbf{w}, \mathbf{x}_j^T - \mathbf{x}_k^C\rangle)(\mathbf{x}_j^T - \mathbf{x}_k^C) \\ &\qquad + |\langle \mathbf{w}, \mathbf{x}_j^T - \mathbf{x}_l^C\rangle|^{\alpha-1}\mathrm{sgn}(\langle \mathbf{w}, \mathbf{x}_j^T - \mathbf{x}_l^C\rangle)(\mathbf{x}_j^T - \mathbf{x}_l^C)\big] \\ &\quad - \alpha|\langle \mathbf{w}, \mathbf{x}_i^T - \mathbf{x}_j^T\rangle|^{\alpha-1}\mathrm{sgn}(\langle \mathbf{w}, \mathbf{x}_i^T - \mathbf{x}_j^T\rangle)(\mathbf{x}_i^T - \mathbf{x}_j^T) \\ &\quad - \alpha|\langle \mathbf{w}, \mathbf{x}_k^C - \mathbf{x}_l^C\rangle|^{\alpha-1}\mathrm{sgn}(\langle \mathbf{w}, \mathbf{x}_k^C - \mathbf{x}_l^C\rangle)(\mathbf{x}_k^C - \mathbf{x}_l^C). \end{aligned}$$

A necessary condition for the Stochastic Gradient Descent algorithm to converge is that the expectation (taken over the randomly sampled vectors) of the subgradient (5.3.5) be equal to the subgradient computed on the full dataset given in (5.3.3). Since in the algorithm given in Figure 5.3.1 we are using four randomly sampled data points we need to take the expectation over all of them.

We will denote the expectation over $(\mathbf{x}_i^T, y_i^T)$ by $\mathbf{E}_i^T[\cdot]$, analogous notation will be used for expectations over records in the control group. Notice that, since the records in the stochastic optimization algorithm are chosen uniformly at random, we have

$$\mathbf{E}_i^T[f(\mathbf{x}_i^T, y_i^T)] = \frac{1}{n^T}\sum_{i=1}^{n^T} f(\mathbf{x}_i^T, y_i^T), \quad \mathbf{E}_k^C[f(\mathbf{x}_i^C, y_i^C)] = \frac{1}{n^C}\sum_{i=1}^{n^C} f(\mathbf{x}_i^C, y_i^C). \tag{5.3.7}$$

Further, denote by $\mathbf{E}[\cdot]$ the expectation over all four randomly chosen samples $(\mathbf{x}_i^T, y_i^T)$, $(\mathbf{x}_j^T, y_j^T)$, $(\mathbf{x}_k^C, y_k^C)$, $(\mathbf{x}_l^C, y_l^C)$, i.e.

$$\mathbf{E}[\cdot] = \mathbf{E}_i^T\mathbf{E}_j^T\mathbf{E}_k^C\mathbf{E}_l^C[\cdot].$$

Let us now compute, term by term, the expectation of the subgradient given in (5.3.5). Clearly

$\mathbf{E}\mathbf{w} = \mathbf{w}$. Also, using (5.3.6) and (5.3.7) we get

$$\begin{aligned}
\mathbf{E}\frac{\partial l(\mathbf{w}, \mathbf{x}_i^T, y_i^T)}{\partial \mathbf{w}} &= \mathbf{E}_i^T \frac{\partial l(\mathbf{w}, \mathbf{x}_i^T, y_i^T)}{\partial \mathbf{w}} \\
&= -\frac{1}{n^T} \sum_{i=1}^{n^T} \mathbf{x}_i^T y_i^T \begin{cases} C_1 \mathbb{1}_{\left[1 - y_i^T(\langle \mathbf{w}, \mathbf{x}_i^T \rangle - b_1) > 0\right]} + C_2 \mathbb{1}_{\left[1 - y_i^T(\langle \mathbf{w}, \mathbf{x}_i^T \rangle - b_2) > 0\right]} & \text{if } (\mathbf{x}_i^T, y_i^T) \in \mathbf{D}_+^T \\ C_2 \mathbb{1}_{\left[1 - y_i^T(\langle \mathbf{w}, \mathbf{x}_i^T \rangle - b_1) > 0\right]} + C_1 \mathbb{1}_{\left[1 - y_i^T(\langle \mathbf{w}, \mathbf{x}_i^T \rangle - b_2) > 0\right]} & \text{if } (\mathbf{x}_i^T, y_i^T) \in \mathbf{D}_-^T \end{cases} \\
&= -\frac{C_1}{n^T} \sum_{\mathbf{D}_+^T} \mathbb{1}_{\left[1 - y_i^T(\langle \mathbf{w}, \mathbf{x}_i \rangle - b_1) > 0\right]} \mathbf{x}_i^T y_i^T - \frac{C_2}{n^T} \sum_{\mathbf{D}_+^T} \mathbb{1}_{\left[1 - y_i^T(\langle \mathbf{w}, \mathbf{x}_i \rangle - b_2) > 0\right]} \mathbf{x}_i^T y_i^T \\
&\quad - \frac{C_2}{n^T} \sum_{\mathbf{D}_-^T} \mathbb{1}_{\left[1 - y_i^T(\langle \mathbf{w}, \mathbf{x}_i^T \rangle - b_1) > 0\right]} \mathbf{x}_i^T y_i^T - \frac{C_1}{n^T} \sum_{\mathbf{D}_-^T} \mathbb{1}_{\left[1 - y_i^T(\langle \mathbf{w}, \mathbf{x}_i^T \rangle - b_2) > 0\right]} \mathbf{x}_i^T y_i^T.
\end{aligned}$$

We now move to computing the expectation of the subgradient of the Székely regularizer. Note that

$$\begin{aligned}
&\mathbf{E}|\langle \mathbf{w}, \mathbf{x}_i^T - \mathbf{x}_k^C \rangle|^{\alpha-1} \text{sgn}(\langle \mathbf{w}, \mathbf{x}_i^T - \mathbf{x}_k^C \rangle)(\mathbf{x}_i^T - \mathbf{x}_k^C) \\
&\qquad = \mathbf{E}_i^T \mathbf{E}_k^C |\langle \mathbf{w}, \mathbf{x}_i^T - \mathbf{x}_k^C \rangle|^{\alpha-1} \text{sgn}(\langle \mathbf{w}, \mathbf{x}_i^T - \mathbf{x}_k^C \rangle)(\mathbf{x}_i^T - \mathbf{x}_k^C) \\
&\qquad = \frac{1}{n^T n^C} \sum_{i=1}^{n^T} \sum_{k=1}^{n^C} |\langle \mathbf{w}, \mathbf{x}_i^T - \mathbf{x}_k^C \rangle|^{\alpha-1} \text{sgn}(\langle \mathbf{w}, \mathbf{x}_i^T - \mathbf{x}_k^C \rangle)(\mathbf{x}_i^T - \mathbf{x}_k^C).
\end{aligned}$$

By symmetry, the three other pairs of treatment and control points lead to the same expected value. Similarly

$$\begin{aligned}
&\mathbf{E}|\langle \mathbf{w}, \mathbf{x}_i^T - \mathbf{x}_j^T \rangle|^{\alpha-1} \text{sgn}(\langle \mathbf{w}, \mathbf{x}_i^T - \mathbf{x}_j^T \rangle)(\mathbf{x}_i^T - \mathbf{x}_j^T) \\
&\qquad = \frac{1}{(n^T)^2} \sum_{i=1}^{n^T} \sum_{j=1}^{n^T} |\langle \mathbf{w}, \mathbf{x}_i^T - \mathbf{x}_j^T \rangle|^{\alpha-1} \text{sgn}(\langle \mathbf{w}, \mathbf{x}_i^T - \mathbf{x}_j^T \rangle)(\mathbf{x}_i^T - \mathbf{x}_j^T).
\end{aligned}$$

The expression for the pair of control points is analogous. Finally we get

$$\begin{aligned}
&\mathbf{E}\frac{\partial S(\mathbf{x}_i^T, \mathbf{x}_j^T, \mathbf{x}_k^C, \mathbf{x}_l^C, y_i^T, y_j^T, y_k^C, y_l^C, \mathbf{w})}{\partial \mathbf{w}} \\
&\qquad = 2\alpha \frac{1}{n^T n^C} \sum_{i=1}^{n^T} \sum_{k=1}^{n^C} |\langle \mathbf{w}, \mathbf{x}_i^T - \mathbf{x}_k^C \rangle|^{\alpha-1} \mathrm{sgn}(\langle \mathbf{w}, \mathbf{x}_i^T - \mathbf{x}_k^C \rangle)(\mathbf{x}_i^T - \mathbf{x}_k^C) \\
&\qquad - \alpha \frac{1}{(n^T)^2} \sum_{i=1}^{n^T} \sum_{j=1}^{n^T} |\langle \mathbf{w}, \mathbf{x}_i^T - \mathbf{x}_j^T \rangle|^{\alpha-1} \mathrm{sgn}(\langle \mathbf{w}, \mathbf{x}_i^T - \mathbf{x}_j^T \rangle)(\mathbf{x}_i^T - \mathbf{x}_j^T) \\
&\qquad - \alpha \frac{1}{(n^C)^2} \sum_{i=1}^{n^C} \sum_{j=1}^{n^C} |\langle \mathbf{w}, \mathbf{x}_i^C - \mathbf{x}_j^C \rangle|^{\alpha-1} \mathrm{sgn}(\langle \mathbf{w}, \mathbf{x}_i^C - \mathbf{x}_j^C \rangle)(\mathbf{x}_i^C - \mathbf{x}_j^C).
\end{aligned}$$

Combining the above results we get exactly the subgradient of the risk which is minimized by Székely regularized Uplift Support Vector Machines given in (5.3.3) and (5.3.4).

Therefore the necessary condition for convergence is satisfied. For sufficiency, let us first examine the properties of the optimization problem (5.3.1). Notice first, that although the term $\frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle$ is strongly convex and the remaining terms are convex, the Székely penalty term is not. Therefore the optimized function need not be convex and we cannot guarantee global convergence. Suppose that there exists a bound D such that$\|\mathbf{w}_t\| \leq D$ for all iteration steps t and $\|\mathbf{w}^*\| \leq D$, where $\mathbf{w}^*$ is a (possibly local) optimum to which the algorithm should converge. Note that the risk function is Lipschitz continuous on any closed region of the parameter space. This is obviously true for the closed ball of radius D centered at origin. By Weierstrass theorem this implies that a local minimum does exist, possibly on the boundary of the ball. It also follows that the subgradient of R is bounded throughout the algorithm and the convergence is guaranteed based on results given in [23, Section 11.0] for $\gamma_t = Ct^{-\frac{1}{2}}$. The constant C was chosen to be 1 in our implementation.

Let us now briefly comment on the existence of the bound D. Without additional assumptions we cannot formally guarantee that at every iteration we have $\|\mathbf{w}_t\| \leq D$. To obtain such guarantees, an extra step can be added to Algorithm 5.3.1, which, after each iteration, projects $\mathbf{w}_t$ onto a ball of some radius D [23]. In practice we saw no convergence problems and the extra step was not necessary.

If we make an additional assumption that the Székely penalty S is locally convex around the minimum we can guarantee fast convergence rates. Since $\frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle$ is strongly convex and a sum a convex and strongly convex function is strongly convex, the risk $R(\mathbf{w})$ given in (5.3.1) becomes strongly convex. The convergence rate is then $O(t^{-1})$ for $\gamma_t = Ct^{-\frac{1}{2}}$ following the results in [3, Theorem 3]. The constant C was chosen to be 1 in our implementation.

Note that the step size used guarantees convergence for non-strongly convex functions and fast convergence for strongly convex ones.

Chapter 6

Experimental evaluation

In this section we present an experimental evaluation of all the proposed variants of Uplift Support Vector Machines. We begin by a short general discussion of evaluation of uplift models and the description of benchmark datasets used in experiments. We follow with an illustrative example showing the difference between L_1 and L_p Uplift Support Vector Machines. Later, we present an experimental comparison with other uplift modeling methods on several benchmark datasets. Finally we discuss model evaluation for the biased treatment assignment problem and evaluate Szekely regularized USVMs on such a dataset.

6.1 Evaluation of Uplift models

Evaluation of uplift models is more difficult than in case of classification. Due to the Fundamental Problem of Causal Inference (see Section 1.2) the true gain is never known for specific data records. Evaluating correctness of predictions is possible only on groups of records. Luckily some of the curves used to assess classifiers can be adapted also to uplift models.

Let us now discuss evaluation of uplift models using so called uplift curves. One of the tools used for assessing the performance of standard classification models are lift curves (also known as cumulative gains curves or cumulative accuracy profiles). For lift curves, the x axis corresponds to the number of cases targeted and the y axis to the number of successes captured by the model. In our case both numbers are expressed as percentages of the total population.

The *uplift curve* is computed by subtracting the lift curve obtained on the control test set from the lift curve obtained on the treatment test set. Both curves are computed using the same uplift model. Recall that the number of successes on the y axis is expressed as a percentage of the total population which guarantees that the curves can be meaningfully subtracted. An uplift curve

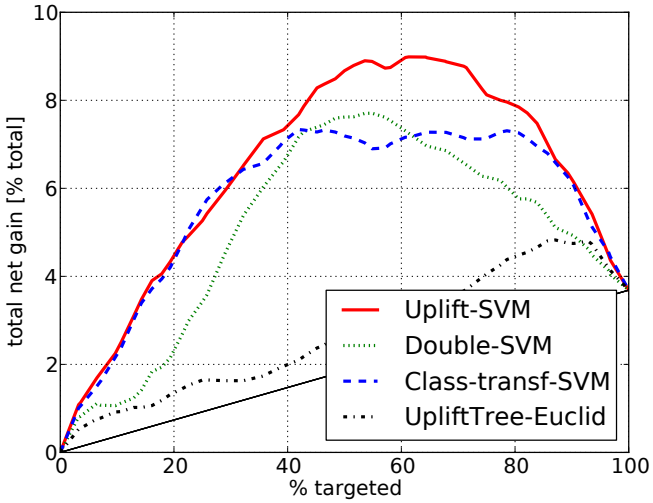

Figure 6.1.1: Uplift curves for the `breast-cancer` dataset for Uplift SVM, the double SVM approach, and an SVM uplift model based on class variable transformation (`Class-transf-SVM`). The x-axis represents the percentage of the population to which the action has been applied and the y-axis the net gain from performing the action based on model selection. It can be seen that targeting about 60% of the population according to models' predictions gives significant gains over targeting nobody or the whole population. The proposed Uplift SVM model achieves the best performance over the whole range of the plot.

can be interpreted as follows: on the x axis we select the percentage of the population on which an action is to be performed and on the y axis we read the difference between the success rates in the treatment and control groups. A point at $x = 100\%$ gives the gain in success probability we would obtain if the action was performed on the whole population. The diagonal corresponds random selection. The Area Under the Uplift Curve (AUUC) can be used as a single number summarizing model performance. We subtract the area under the diagonal line from this value in order to obtain more meaningful numbers. More details on evaluating uplift models and on uplift curves can be found in [31, 38].

Let us now present an example providing the intuition behind uplift curves. Figure 6.1.1 shows uplift curves for the `breast-cancer` dataset (see below for a description of how it has been used for uplift modeling) for three of the uplift models used in our experiments. The curves in the figure have been generated by averaging over 128 random train test splits; the same method has been used for other experiments in this section and is described in detail below. It can be seen that applying the action only to some proportion of the population leads to significant gains in net

success rate. For example, applying the action to the whole population results in four percentage points gain in success rate. If however the action is applied to about 60% of the population based on the USVM model, a total gain of 9 percentage points is possible. It can also be seen that, on this dataset, Uplift Support Vector Machines outperformed all remaining models. An uplift decision tree performed especially poorly compared to all SVM based uplift models.

6.2 Description of the datasets used in experiments

While testing uplift modeling algorithms one encounters the problem of the lack of publicly available datasets. Even though control groups are ubiquitous in medicine and become common in marketing, there are very few publicly available datasets which include a control group as well as a reasonable number of predictive attributes. In this dissertation we will use the few publicly available datasets we are aware of, as well as some artificially generated examples based on datasets from the UCI repository. We describe the two approaches in turn.

The first publicly available dataset, provided on Kevin Hillstrom's MineThatData blog, contains results of an e-mail campaign for an Internet based retailer [14]. The dataset contains information about 64 000 customers with basic marketing information such as the amount of money spent in the previous year or when the last purchase was made. The customers have been randomly split into three groups: the first received an e-mail campaign advertising men's merchandise, the second, a campaign advertising women's merchandise, and the third was kept as control. Data is available on whether a person visited the website and/or made a purchase (conversion). We only focus on visits since very few conversions actually occurred. In this chapter we use the dataset in two ways: combining both e-mailed groups into a single treatment group (`Hillstrom-visit`) and using only the group who received advertisement for women's merchandise and the control group (`Hillstrom-visit-w`). Women's merchandise group was selected since the campaign selling the men's merchandise was ineffective, with very few visits.

Additionally, we found two suitable publicly available clinical trial datasets which accompany a book on survival analysis [28]. The first dataset is the Bone Marrow Transplant (`BMT`) data which covers patients who received two types of bone marrow transplant: taken from the pelvic bone (which we used as the control group since this is the procedure commonly used at the time the data was created) or from the peripheral blood (a novel approach, used as the treatment group in our experiments). The peripheral blood transplant is generally the preferred treatment, so minimizing its side effects is highly desirable. There are only three randomization time variables available: the type and extent of the disease, as well as patients age. There are two target variables

representing the occurrence of the chronic (`cgvh`) and acute (`agvh`) graft versus host disease.

Note that even though the `BMT` dataset does not, strictly speaking, include a control group, uplift modeling can still be applied. The role of the control group is played by one of the treatments, and the method allows for selection of patients to whom an alternative treatment should be applied.

The second clinical trial dataset we analyze (`tamoxifen`) comes from the study of treatment of breast cancer with a drug tamoxifen. The control group received tamoxifen alone and the treatment group tamoxifen combined with radio therapy. We attempt to model the variable `stat` describing whether the patient was alive at the time of the last follow-up. The dataset contains six variables. Since the data contains information typical for survival analysis we used the method from [18] to convert it to a standard uplift problem. The method simply ignores censoring and treats all observed survival times greater than some threshold (median in our case) as successes. In [18] it is shown that such a method preserves correctness of decisions made by the model.

We have also used clinical trial datasets available in the `survival` and `kmsurv` packages of the `R` statistical system. Since all those datasets involve survival data, the method from [18] was used in all cases with median observed survival time used as the threshold. The `kmsurv` package includes two datasets: `burn` and `hodg`. Their description is available in the package documentation and is omitted here. The `survival` package contains four suitable datasets: `pbc`, `bladder`, `colon` and `veteran`. The datasets are described in the package documentation. The `colon` dataset involves two possible treatments (levamisole and levamisole combined with 5FU: Fluorouracil), and a control group, as well as two possible targets: patient death and tumor recurrence. Since the analyzed setting assumes a single treatment and a single target variable we formed six different datasets, three for each target variable (indicated by the suffix 'death' and 'recur'). The `colon-death` and `colon-recur` datasets combine the two treatments into a single treatment group. The datasets `colon-lev-death` and `colon-lev-recur` use only the group treated with levamisole alone and the control cases. Finally `colon-lev5fu-death` and `colon-lev5fu-recur` compare the combined therapy (levamisole with 5FU) with control cases.

As can be seen, there are very few real uplift datasets available, moreover, they all have a limited number of attributes (up to 10) and/or data. In [38] an approach has been proposed to split standard UCI datasets into treatment and control groups suitable for uplift modeling. The conversion is performed by first picking one of the data attributes which either has a causal meaning or splits the data evenly into two groups. Table 6.2.1 shows the UCI datasets used as well as the condition used to select the treatment group.

As a postprocessing step, attributes strongly correlated with the split are removed (ideally, the

dataset	treatment selection condition	#removed/total attributes
acute inflam.	a3 = 'YES'	2/6
australian	a1 = '1'	2/14
breast-cancer	menopause = 'PREMENO'	2/9
credit-a	a7 $\neq$ 'V'	3/15
dermatology	exocytosis ≤ 1	16/34
diabetes	insu > 79.8	2/8
heart-c	sex = 'MALE'	2/13
hepatitis	steroid = 'YES'	1/19
hypothyroid	on_thyroxine = 'T'	2/29
labor	education-allowance = 'YES'	4/16
liver-disorders	drinks < 2	2/6
nursery	children $\in$ {'3','MORE'}	1/8
primary-tumor	sex ='MALE'	2/17
splice	attribute1 $\in$ {'A','G'}	2/61
winequal-red	sulfur dioxide < 46.47	2/11
winequal-white	sulfur dioxide < 138.36	3/11

Table 6.2.1: Artificial datasets used in the experiments. Source: [38]

division into treatment and control groups should be independent from all predictive attributes, but this is possible only in a controlled experiment). The removal was based on a simple measure of dependence:

1. Numerical attributes were removed if their means in treatment and control groups differed by more than 25%.

2. A categorical attribute was removed if the probabilities of one of its possible values differed between treatment and control groups by more than 0.25.

The number of removed attributes is also given in Table 6.2.1. Multiclass problems are converted to binary problems with the majority class considered to be $+1$ and remaining classes -1.

6.3 An illustration of the difference between L_1 and L_p Uplift Support Vector Machines

We now show the difference between L_1-USVMs and L_p-USVMs on two example datasets from the UCI repository: `breast-cancer` and `australian`. More specifically, we show how the choice

of the parameter $\frac{C_2}{C_1}$ affects model behavior. Since this section has a mainly illustrative purpose, all curves are drawn based on the full dataset; more rigorous experiments involving test sets are given in Section 6.4.

Figures 6.3.1 and 6.3.2 show the number of cases classified as positive, neutral and negative depending on the quotient $\frac{C_2}{C_1}$ for the two datasets. The numbers shown were obtained on the full dataset and are averages of respective numbers of cases in treatment and control groups. The parameter C_1 was set to 5, but for other values we obtained very similar results. Each figure comprises four charts, one for L_1-USVMs and three for L_p-USVMs for three possible values of the parameter p.

It can clearly be seen that for low values of the quotient, the neutral class is empty, but as the quotient increases, more and more cases are classified as neutral. Finally, almost no cases are classified as positive or negative. Notice that for $p = 1$ we have an abrupt jump between all cases being classified as neutral and all cases being classified as negative. This is an example of the undesirable behavior analyzed theoretically in Section 3.4.1. The model is practically useless for prediction since it always predicts the same class for all points (note however that such a model may still be useful for ranking as will be seen in the next section). As the values of p become larger the transition becomes smoother. For $p = 1.2$ the behavior is close to that of L_1-USVMs, and for $p = 2$ the transition is very smooth and points are assigned to all three classes.

The figures validate our interpretation presented earlier in Lemmas 3.3.2−3.3.3. The analyst can use the parameter $\frac{C_2}{C_1}$ to control the proportion of neutral predictions, especially for L_p-USVMs. Note that, overall, more points are classified as positive than as negative. This is due to the overall beneficial influence of the action.

6.4 Comparison of model performance on benchmark datasets

In this section we compare the performance of the proposed uplift models with other uplift modeling approaches. The performance will be measured in term of Areas Under the Uplift Curves (AUUCs).

We begin by comparing the performance of L_1 Uplift Support Vector Machines (`Uplift-SVM`) and five other uplift modeling methods on several benchmark datasets. Four of the models are also based on Support Vector Machine classifiers. The first is the method based on building two separate SVM models (`Double-SVM`) on treatment and control groups and subtracting their predicted probabilities and the second, a single Support Vector Machine adapted to uplift modeling using the class variable transformation proposed in [20] (`Class-transf-SVM`). Since both those

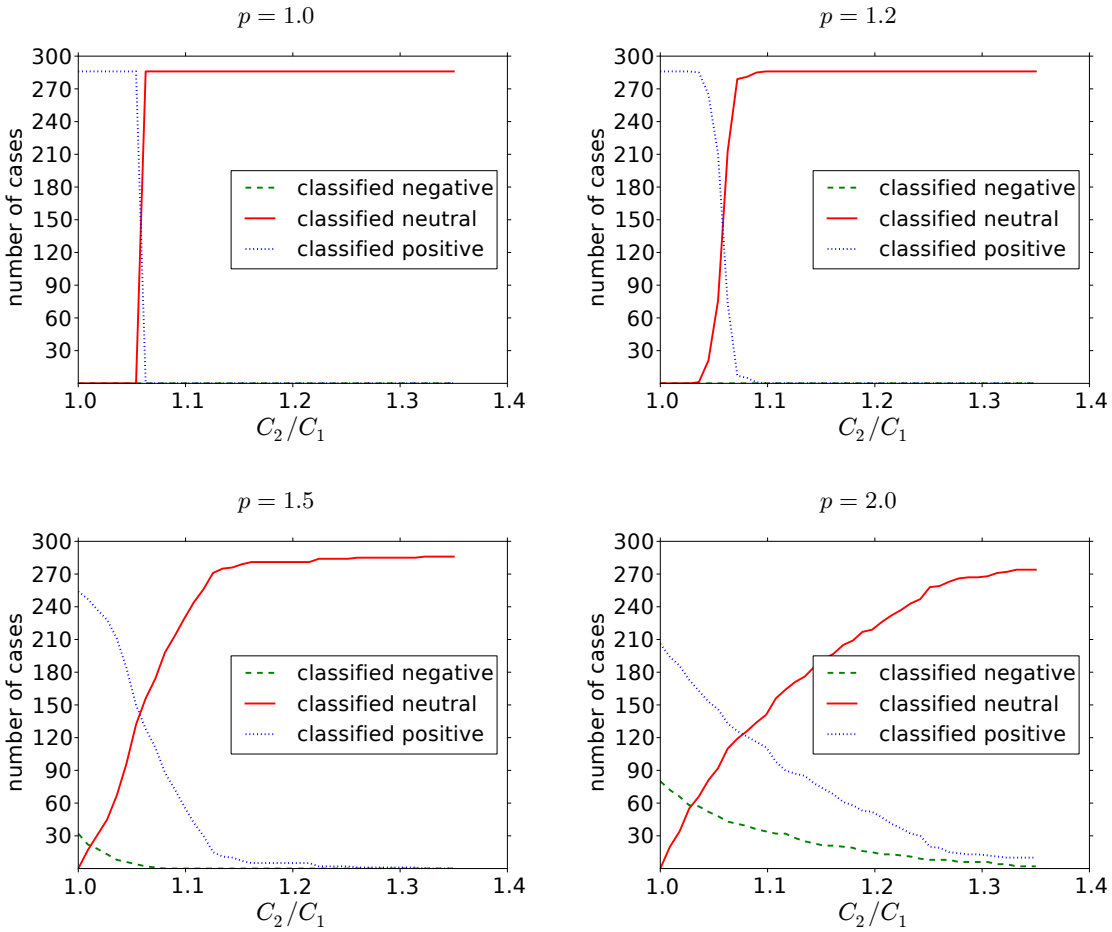

Figure 6.3.1: Number of cases classified as positive, neutral and negative as a function of the quotient $\frac{C_2}{C_1}$ of L_p-USVM penalty coefficients for the `breast-cancer` dataset for different values of p.

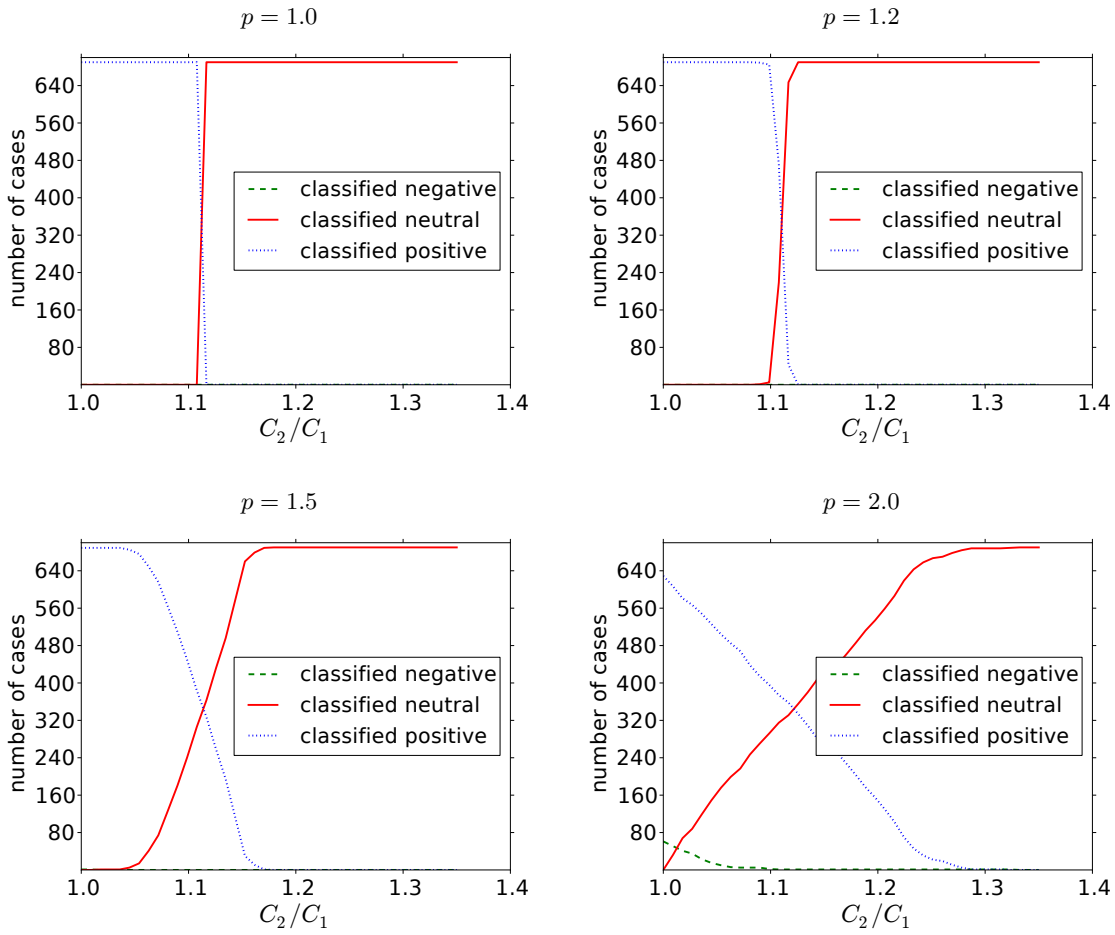

Figure 6.3.2: Number of cases classified as positive, neutral and negative as a function of the quotient $\frac{C_2}{C_1}$ of L_p-USVM penalty coefficients for the `australian` dataset for different values of p.

dataset	Uplift SVM	Double SVM	Class-transf SVM	Diff-pred SVM	Treatment SVM	Uplift Tree
BMT-agvh	−0.024	−0.019	−0.038	−0.019	0.001	−0.016
BMT-cgvh	0.040	0.046	0.021	0.049	0.017	0.023
Hillstrom-visit	0.004	0.005	0.003	0.004	0.003	0.004
Hillstrom-visit-w	0.008	0.008	0.008	0.006	0.004**	0.006
australian	−0.002	0.023*	−0.005	−0.008	0.013*	0.004
bladder	−0.048	−0.030	−0.042	—	0.005	0.004*
breast cancer	0.043	0.035	0.041	0.038	0.002*	0.008*
burn	0.038	0.097*	0.042	0.034	0.007**	0.069
colon-death	−0.014	−0.008	−0.017	−0.015	−0.009	0.003
colon-recur	0.003	0.015	0.001	0.008	−0.009	0.003
colon-lev5fu-death	0.008	0.008	0.010	0.005	0.006	0.012
colon-lev5fu-recur	0.006	0.001	−0.015	−0.015	−0.007	0.000
colon-lev-death	0.002	−0.012	−0.022*	−0.024*	−0.013	−0.001
colon-lev-recur	−0.004	−0.009	−0.012	−0.015	−0.010	0.003
credit-a	0.062	0.011**	0.059	0.049	0.004**	0.022*
dermatology	0.080	0.056	0.079	0.076	−0.045**	0.068
diabetes	−0.002	0.005	−0.003	−0.010	0.010	0.016
diagnosis	0.151	−0.003**	0.142	0.148	0.018**	0.139
heart-c	0.023	−0.001	0.028	0.016	0.016	0.017
hepatitis	0.015	0.009	0.003	0.025	−0.002	−0.001
hodg	0.050	0.043	0.053	0.074	0.056	0.019
labor	−0.016	−0.005	−0.024	−0.013	−0.005	−0.019
liver-disorders	0.001	0.029	0.012	0.021	0.028	0.020
pbc	0.000	−0.006	−0.012	−0.009	−0.016	−0.010
primary-tumor	0.041	0.011	0.037	0.039	0.022	0.010*
veteran	0.057	0.034	0.060	0.061	−0.007*	0.038
winequality-red	0.019	0.014	0.020	0.021	0.013	0.034*
winequality-white	0.020	0.021	0.019	0.023	0.004**	0.040**
USVM Win/total		14/28	19/28	16/28	20/28	15/28

Table 6.4.1: Areas under the Uplift Curve for six uplift models on real and artificial datasets. '*' indicates difference larger than one standard deviation; '**' larger than two standard deviations

methods require probabilities to be predicted, the SVMs have been calibrated by training logistic regression models on the scores (i.e. $\langle \mathbf{w}, \mathbf{x} \rangle$) they return. The differential prediction SVMs [24] are included under the name (`Diff-pred-SVM`). The next method included in the comparison, `Treatment-SVM`, is the standard classification approach, i.e. a Support Vector Machine built only on the treatment group, ignoring the control cases. Finally, to compare with a different type of model we include results for uplift decision trees described in [38, 39]. Splitting criterion based on the Euclidean distance was used.

The parameters of all SVM based models have been chosen using 5-fold cross-validation by maximizing the Area Under the Uplift Curve (AUUC). The parameter C for classical SVMs was chosen from the set $\{10^{-2}, 10^{-1}, \ldots, 10^{5}\}$. For L_1 Uplift Support Vector Machines the parameter C_1 was selected from the set $\{10^{-2}, 10^{-1}, \ldots, 10^{3}\}$ and the parameter ratio $\frac{C_2}{C_1}$ from ten points evenly spaced on the interval $[1, 2.5]$. For each grid point 5-fold cross-validation was used to measure model performance and to pick the best parameter combination.

Table 6.4.1 compares Areas Under the Uplift Curve for Uplift SVMs against the five other modeling approaches. The areas are given in terms of percentages of the total population (used also on the y-axis). Testing was performed by repeating 128 times a random train/test split with 80% of data used for training (and cross-validation based parameter tuning). The remaining 20% were used for testing. Large number of repetitions reduces the influence of randomness in model learning and testing, making the experiments repeatable. The last row of the table lists the number of times Uplift-SVM was better than each respective method. We were not able to run the differential prediction SVM on the `bladder` dataset which is indicated with a dash in the table.

We have used the 128 samples to estimate the standard deviation of the AUUCs and indicated differences larger than one (respectively two) standard deviations by a '*' (respectively '**').

Let us first compare the performance of our method with traditional classification which ignores the control group. It can be seen that the method wins in 20 out of 28 cases, sometimes by a wide margin (e.g. the `diagnosis` dataset). The results are often statistically significant. One can thus conclude that the use of a control group in the modeling process has the potential to bring significant gains when working with data from randomized experiments.

We now compare with other uplift modeling methods. Uplift SVM outperforms the method based on class variable transformation proposed in [20] on 19 out of 28 datasets. It's performance is on par with the method based on double SVMs, which it outperforms on half of the datasets. Notice also, that the class variable transformation based method performs similarly (although usually worse) to USVMs, but the double SVM method tends to perform poorly when USVMs

give good results and vice-versa. The methods thus appear complementary to each other. The differential prediction SVM [24] also performs comparably with USVMs.

Unlike in the case of comparison with traditional classification the differences in AUUCs are usually not statistically significant. This is most probably due to natural difficulties in predicting uplift where variances are typically much higher than in classification [31].

We believe that the experimental results clearly demonstrate that USVMs are a useful addition to the uplift modeling toolbox. Overall, our method performs comparably to or better than current state-of-the-art uplift modeling methods. We also believe, that other advantages of the proposed Uplift SVMs are equally important. For example, it allows for natural prediction of cases with positive, negative and neutral outcomes (as shown in Section 6.3) which is very useful in practice. The negative group is especially important from the point of view of practical applications. Being able to detect this group and refraining from targeting it was crucial for many successful marketing campaigns. Additionally, through the choice of the parameter $\frac{C_2}{C_1}$ the analyst is able to decide how conservative should the model be when selecting those groups.

We now move to experimental analysis of L_p-USVMs. Table 6.4.2 shows AUUCs for L_p-USVMs with $p = 1.2, 1.5, 2.0$. The experimental procedure has been identical to L_1-USVMs, except that the parameter ratio $\frac{C_2}{C_1}$ was selected from the range $[1, 5]$. For comparison, the uplift models based on class variable transformations and L_p-SVM classifiers [1] are also included.

It can be seen that L_p-USVMs generally perform comparably to the class variable transformation based methods. Moreover, comparing with Table 6.4.1 we can see that L_p-USVMs performance is generally similar to L_1-USVMs, especially for values of p closer to 1. At the same time, they guarantee that the analyst is able to reliably control the percentage of neutral predictions (according to Lemmas 3.3.1–3.3.3).

6.5 Experimental evaluation of Székely regularized USMVs

In this section we present experimental evaluation of Székely regularized Uplift Support Vector Machines. We begin by describing the dataset used for testing, then discuss the methodology we used for testing uplift models in the presence of biased treatment assignment and finally present actual experimental results.

6.5.1 The Right Heart Catheterization dataset

The right heart catheterization dataset [21] contains data about 5735 patients admitted to hospitals in serious condition. 2184 of them were subjected to the right heart catheterization procedure

dataset	$p = 2$		$p = 1.5$		$p = 1.2$		$p = 1.0$	
	Uplift SVM	Class-tr. SVM	Uplift SVM	Class-tr. SVM	Uplift SVM	Class-tr. SVM	Uplift SVM	Class-tr. SVM
BMT-agvh	−0.026	−0.025	−0.026	−0.022	−0.027	−0.021	−0.024	−0.038
BMT-cgvh	0.037	0.040	0.036	0.042	0.037	0.040	0.040	0.021
Hillstrom-visit	0.003	0.003	0.003	0.003	0.003	0.003	0.004	0.003
Hillstrom-visit-w	0.007	0.007	0.007	0.007	0.007	0.007	0.008	0.008
australian	−0.007	−0.007	−0.007	−0.008	−0.008	−0.008	−0.002	−0.005
bladder	−0.047	−0.046	−0.048	−0.047	−0.047	−0.046	−0.048	−0.042
breast cancer	0.039	0.038	0.038	0.037	0.039	0.038	0.043	0.041
burn	0.028	0.026	0.030	0.022	0.028	0.025	0.038	0.042
colon-death	−0.015	−0.015	−0.015	−0.016	−0.016	−0.017	−0.014	−0.017
colon-recur	0.007	0.007	0.007	0.007	0.007	0.006	0.003	0.001
colon-lev5fu-death	0.006	0.006	0.006	0.010	0.008	0.012	0.008	0.010
colon-lev5fu-recur	−0.015	−0.015	−0.015	−0.013	−0.014	−0.012	0.006	−0.015
colon-lev-death	−0.024*	−0.023*	−0.024*	−0.021*	−0.024*	−0.017	0.002	−0.022*
colon-lev-recur	−0.015	−0.015	−0.015	−0.015	−0.015	−0.012	−0.004	−0.012
credit-a	0.055	0.054	0.054	0.060	0.059	0.067	0.062	0.059
dermatology	0.079	0.078	0.078	0.078	0.079	0.079	0.080	0.079
diabetes	−0.005	−0.005	−0.005	−0.005	−0.005	−0.005	−0.002	−0.003
diagnosis	0.146	0.146	0.146	0.145	0.146	0.146	0.151	0.142
heart-c	0.019	0.020	0.019	0.021	0.019	0.018	0.023	0.028
hepatitis	0.003	0.008	0.001	0.016	0.009	0.018	0.015	0.003
hodg	0.071	0.067	0.072	0.062	0.068	0.064	0.050	0.053
labor	−0.006	−0.006	−0.005	−0.007	−0.006	−0.009	−0.016	−0.024
liver-disorders	0.015	0.014	0.015	0.012	0.015	0.009	0.001	0.012
pbc	−0.009	−0.008	−0.009	−0.004	−0.007	−0.002	0.000	−0.012
primary-tumor	0.039	0.040	0.039	0.041	0.039	0.042	0.041	0.037
veteran	0.055	0.055	0.054	0.054	0.054	0.051	0.057	0.060
winequality-red	0.020	0.020	0.020	0.020	0.020	0.020	0.019	0.020
winequality-white	0.014	0.014	0.014	0.014	0.014	0.015	0.020	0.019

Table 6.4.2: Areas under the Uplift Curve for L_p Uplift Support Vector Machines

(RHC) and constitute the treatment group; the remaining 3551 did not receive the procedure and are the controls. The data does not come from a randomized study, the application of RHC was decided based on patients' condition, so the group selection is biased, in fact, it was done retrospectively based on historical data. Because of this characteristics, as well as its relatively large size, the dataset is ideal to test our algorithm.

The class variable was the attribute `Death` denoting patient death during the first 180 days after hospital admission. Patient survival was considered the positive outcome. To avoid information leaks we removed other outcome related variables such as date of death or date of last contact.

The predictive attributes describe various characteristics of the patient such as age, sex, education, income, medical insurance, the disease the patient suffers from. Also present are results of diagnostics performed at admission such as blood pressure, temperature, results of blood tests, and various scores describing the severity of patient's condition. A full list of predictive variables is given in Table 6.5.1.

Some of the variables contained missing values. All missing values have been replaced by the mean of available values of the respective variable. Categorical variables have been converted to 0-1 real valued variables before imputation.

6.5.2 Testing methodology for biased group selection

Testing the performance of the models was, however, more challenging than in case of randomized controlled trials. As discussed in Section 6.1 testing uplift models is based on an assumption that groups of treatment and control records with similar scores are indeed similar. Unfortunately, this is usually not the case for biased treatment selection.

In order to test the model's predictions we thus had to correct the bias in the test sets. In practice, such corrections are typically achieved using so called propensity scores [36]. A propensity score is the probability that a given patient, described by a feature vector $\mathbf{x}$, will be assigned to the treatment group. There are several ways propensity scores can be used to correct for nonrandom group assignment. In this chapter we are going to use inverse probability of treatment weighting (IPTW) [35].

The IPTW method assigns to each treatment group record a weight inversely proportional to the probability that a record with those characteristics is selected for treatment. This way, cases to which treatment is applied disproportionately often are given lower weights and underrepresented cases higher weights. The control group records are, analogously, given weights proportional to the inverse of the probability that a record with a given feature vector is not given the treatment. Note that for a randomized controlled trial all records within a group receive equal weights.

Variable name	Variable Definition
Age	Age
Sex	Sex
Race	Race
Edu	Years of education
Income	Income
Ninsclas	Medical insurance
Cat1	Primary disease category
Cat2	Secondary disease category
Resp	Respiratory Diagnosis
Card	Cardiovascular Diagnosis
Neuro	Neurological Diagnosis
Gastr	Gastrointestinal Diagnosis
Renal	Renal Diagnosis
Meta	Metabolic Diagnosis
Hema	Hematologic Diagnosis
Seps	Sepsis Diagnosis
Trauma	Trauma Diagnosis
Ortho	Orthopedic Diagnosis
Adld3p	ADL
Das2d3pc	DASI (Duke Activity Status Index)
Dnr1	DNR status on day1
Ca	Cancer
Surv2md1	Support model estimate of the prob. of surviving 2 months
Aps1	APACHE score
Scoma1	Glasgow Coma Score
Wtkilo1	Weight
Temp1	Temperature
Meanbp1	Mean blood pressure
Resp1	Respiratory rate
Hrt1	Heart rate
Pafi1	PaO2/FIO2 ratio
Paco21	PaCo2
Ph1	PH
Wblc1	WBC
Hema1	Hematocrit
Sod1	Sodium
Pot1	Potassium
Crea1	Creatinine
Bili1	Bilirubin
Alb1	Albumin
Urin1	Urine output
Cardiohx	Severe and Very Severe Cardiovascular Symptoms
Chfhx	Congestive Heart Failure
Dementhx	Dementia, Stroke or Cerebral Infarct, Parkinson's Disease
Psychhx	Psychiatric History, Active Psychosis or Severe Depression
Chrpulhx	Pulmonary Disease
Renalhx	Chronic Renal Disease, Chronic Hemodialysis or Peritoneal Dialysis
Liverhx	Cirrhosis, Hepatic Failure
Gibledhx	Upper GI Bleeding
Malighx	Cancer related condition
Immunhx	Immunological health issues
Transhx	Transfer ($>$ 24 Hours) from Another Hospital
Amihx	Definite Myocardial Infarction
Death	Death at any time up to 180 Days

Table 6.5.1: Variables in the Right Heart Catheterization dataset. Reproduced based on [21]

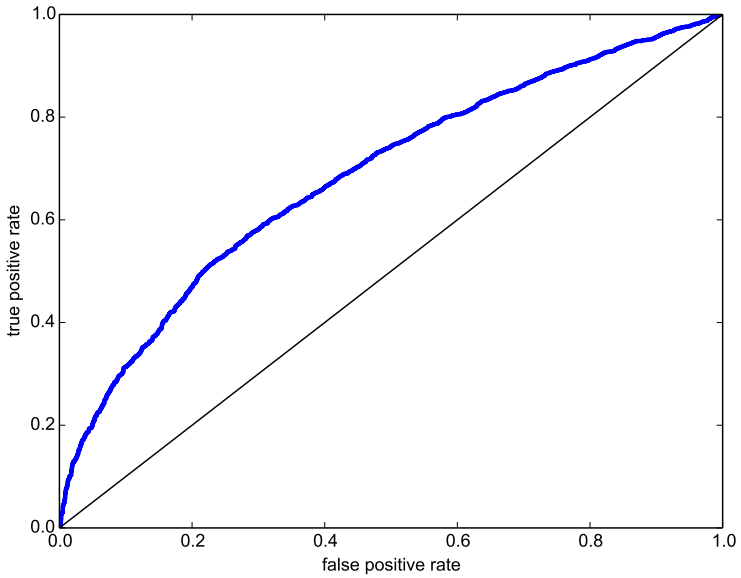

Figure 6.5.1: ROC curve for the propensity score model.

To summarize, our testing procedure works as follows: we build a model with nonrandom treatment assignment using the Székely penalty term to correct for the bias, then we test the model on treatment and control test sets on which the bias has been corrected using the inverse probability of treatment weighting. Since different bias correction procedures are used for model construction and testing, we believe that it is less likely that the estimated model performance is a result of an uncorrected treatment assignment bias.

In order to use the IPTW procedure one needs to know the probability of treatment assignment conditional on patients' characteristics. Unfortunately, this probability is usually unknown and needs to be estimated. Here, we use a logistic regression model trained on full data before crossvalidation splits. The ROC curve for the model is shown in Figure 6.5.1 (area under the ROC curve is 0.686). It can be seen that the model is able to predict reasonably well whether a given patient will receive the RHC procedure. One can conclude that treatment assignment is indeed seriously biased.

6.5.3 Experimental results

We will now present the experimental results. Figure 6.5.2 shows uplift curves drawn for several values of the Székely penalty coefficient C_3. All experiments were performed for $C_1 = C_2 = 1$,

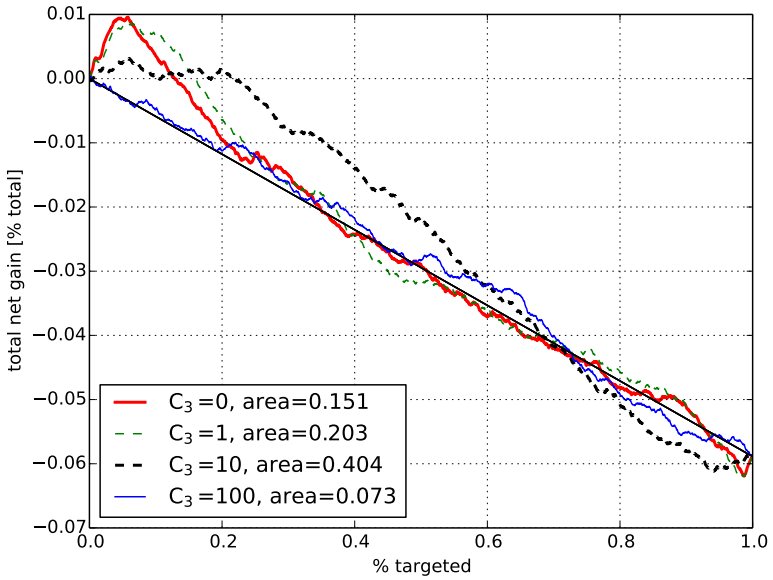

Figure 6.5.2: Uplift curves for Uplift SVM models with different Székely penalty coefficients.

Table 6.5.2: The influence of the Székely penalty coefficient C_3 on the area under the uplift curve and the differences between scores in the treatment and control groups.

C_3 penalty coefficient	0	0.01	0.1	1	10	100
AUUC	0.1505	0.1511	0.1559	0.2030	0.4035	0.0727
difference between score means	0.1051	0.1050	0.1036	0.0839	0.0202	0.004
Kolmogorov-Smirnov statistic	0.3436	0.3433	0.3411	0.3026	0.1103	0.0320

only the value of C_3 was changed. Ten times ten-fold crossvalidation was used to obtain the curves. The curves are drawn based on data weighted using inverse probability of treatment weighting (IPTW) to correct treatment assignment bias. More detailed data on areas under the uplift curves are given in the second row of Table 6.5.2.

Overall the treatment is not effective and the application of the right heart catheterization procedure seems to decrease patients chances of survival. This is in line with the findings presented in [21].

It can be seen that without the Székely correction ($C_3 = 0$) the curve follows the diagonal line corresponding to a model assigning treatment at random, except for the 20% of highest scored cases for which RHC does indeed bring improvement in survival rate over random selection.

With increasing values of the Székely penalty coefficient C_3, the area under the uplift curve is steadily increasing, up to $C_3 = 100$ where the performance rapidly drops. This shows that the application of the Székely penalty does indeed improve model performance under treatment assignment bias.

The best performance is achieved for $C_3 = 10$, and Figure 6.5.2 shows that this particular model achieves good performance over a wide range of scores, bringing improvement over random selection for about 75% of the population. The area under the uplift curve is more than two and a half times better than for the unregularized model.

The drop in performance for very high value of the regularization parameter is typical for regularized models in general: too high a penalty leads to the model ignoring the data and focusing only on the regularization term.

To further examine the effect of the Székely penalty on model behavior we examine the distributions of model scores in treatment and control groups. Figure 6.5.3 shows the empirical cumulative distribution functions of model score distributions in the treatment and control groups for various strength of the Székely regularization term. The charts were obtained on a single repetition of the ten fold crossvalidation. Additionally, we computed two types of statistics summarizing the discrepancies: the difference between score means in the two groups and the Kolmogorov-Smirnov statistic, i.e. the maximum difference between the empirical cumulative distribution functions of the two groups. The summary statistics are given in the third and fourth rows of Table 6.5.2 and are shown graphically in Figure 6.5.4.

It can be seen that for the unregularized model, the distributions of scores in both groups differ significantly. The score means differ by about 0.1, which is a fairly large value since the scores range roughly from -0.5 to 0.5. The value of the Kolmogorov-Smirnov statistic is almost 0.35.

When the Székely penalty increases, the distributions become closer to each other. For $C_3 = 0.01$ and $C_3 = 0.1$ the decrease is tiny but noticeable and is accompanied by a tiny but noticeable improvement in the area under the uplift curve. When $C_3 = 1$ the score distributions already come much closer to each other with the difference between means decreasing to about 0.084; at the same time AUUC increased by about 35% with respect to the unregularized model. A further tenfold increase of the penalty coefficient makes the distributions very similar; the difference in means is just 0.02 and the Kolmogorov-Smirnov statistic just 0.11. The AUUC is 2.68 times higher than for the unregularized model.

A further tenfold increase in the value of C_3 makes the score distributions in the treatment and control groups practically identical, however the regularization is too strong and the model

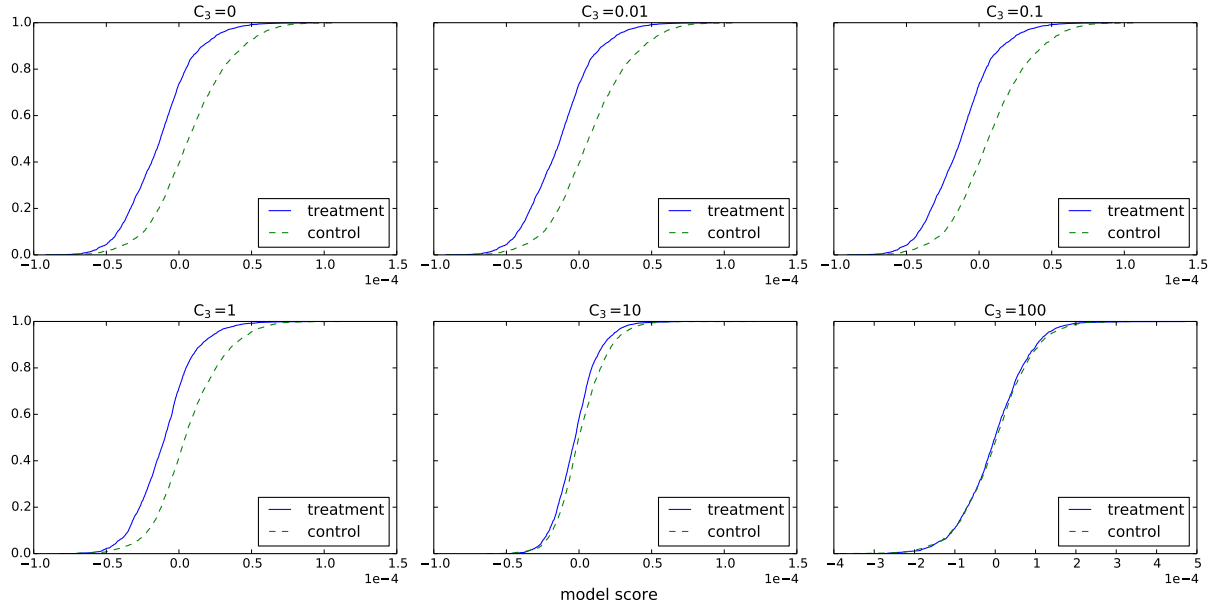

Figure 6.5.3: Empirical cumulative distribution functions of scores in the treatment and control groups for various Székely regularization coefficient values.

no longer correctly predicts for whom the RHC procedure is beneficial. In fact its predictions are not better than random assignments.

Let us now summarize our experimental findings. First, it was shown that the unregularized model behaves poorly after treatment assignment bias correction is applied. Moreover, it produces significantly different scores in the treatment and control groups likely modeling not the real causal impact of the action but the differences in group assignment. As the Székely penalty term increased, the differences between scores the model assigns to treatment and control records became much smaller, accompanied by large improvements in model performance. One can thus conclude that using the Székely penalty term does indeed reduce model's susceptibility to treatment assignment bias, proving the main claim of this chapter.

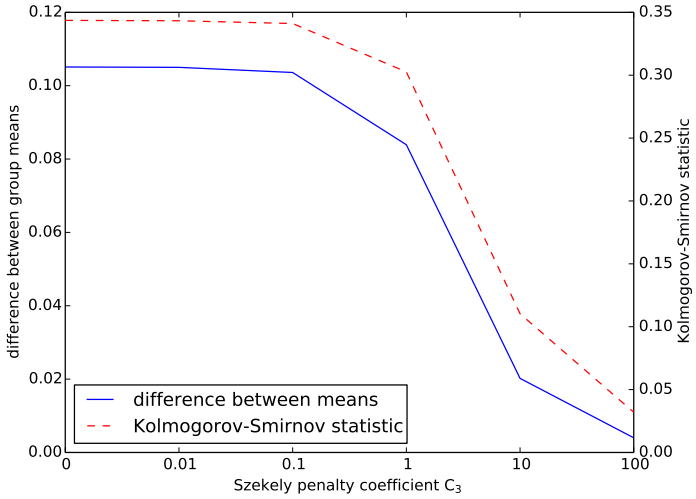

Figure 6.5.4: Statistics summarizing the differences between score distributions in the treatment and control groups for various Székely regularization coefficient values.

Chapter 7

Conclusions

The goal of this dissertation was to establish whether the popular Support Vector Machine framework can be adapted to work in the uplift modeling context. The answer to this research question turned out to be positive.

The main contribution of this dissertation is the adaptation of the Support Vector Machines framework to the uplift modeling problem, including complete reformulation of the optimization task. The proposed methods have been analyzed theoretically, and have been shown to possess several interesting properties including the fact that by an appropriate choice of model parameters, one is able to tune how conservative the model is in declaring a positive or negative impact of an action.

Experimental verification demonstrated that the proposed Uplift Support Vector Machines offer competitive performance to other uplift modeling methods, while possessing some interesting advantages such as being able to classify points into three distinct classes. A theoretical analysis has shown that the Uplift SVMs minimize an upper bound on an uplift analog of the zero-one loss, thus showing that some of the theoretical properties of classical SVMs can be carried over to the uplift case.

During experiments, it turned out that the original proposed uplift SVMs suffered from a problem of abrupt model changes in response to small modifications of model parameters. An adaptation of L_p-SVMs to uplift modeling has thus been proposed which is more stable and reacts smoothly to changes in penalty parameters. Experiments have demonstrated that ranking performance of L_p-USVMs is comparable to other models, but they reliably split the data into three classes.

Moreover, we have also presented a regularization method which corrects the behavior of uplift models under non-randomized treatment assignment. The approach is based on an energy

distance proposed by Székely and Rizzo which offers a practical way of ensuring the similarity of model scores in the treatment and control datasets.

Finally, we have presented efficient optimization algorithms for both L_1 and L_p problem formulations and for the version with Székely regularization.

Bibliography

[1] S. Abe. Analysis of support vector machines. In *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on*, pages 89–98, 2002.

[2] M.S. Andersen, J. Dahl, Z. Liu, and L. Vandenberghe. Interior-point methods for large-scale cone programming. In *Optimization for Machine Learning*, pages 55–83. MIT Press, 2012.

[3] F. Bach and E. Moulines. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Proceedings of Advances in Neural Information Processing Systems 24 (NIPS 2011)*, 2011.

[4] A. Ben-Hur, D. Horn, H.T. Siegelmann, and V. Vapnik. Support vector clustering. *Journal of Machine Learning Research*, 2:125–137, 2001.

[5] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[6] D.M. Chickering and D. Heckerman. A decision theoretic approach to targeted advertising. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence, UAI'00*, pages 82–88, Stanford, CA, 2000.

[7] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

[8] C. Crisp and D. Burges. Uniqueness of the svm solution. In *Advances in Neural Information Processing Systems 12: Proceedings of the 1999 Conference*, volume 12. MIT Press, 2000.

[9] T. Evgeniou, M. Pontil, and T. Poggio. A unified framework for regularization networks and support vector machines. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA, 1999.

[10] S. Fine and K. Scheinberg. Efficient SVM training using low-rank kernel representations. *Journal of Machine Learning Research*, (2):243–264, 2001.

[11] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 2013.

[12] L. Guelman, M. Guillén, and A.M. Pérez-Marín. Random forests for uplift modeling: An insurance customer retention case. In *Modeling and Simulation in Engineering, Economics and Management*, volume 115 of *Lecture Notes in Business Information Processing (LNBIP)*, pages 123–133. Springer, 2012.

[13] B. Hansotia and B. Rukstales. Incremental value modeling. *Journal of Interactive Marketing*, 16(3):35–46, 2002.

[14] K. Hillstrom. The MineThatData e-mail analytics and data mining challenge. MineThatData blog, `http://blog.minethatdata.com/2008/03/minethatdata-e-mail-analytics-and-data.html`, 2008. Retrieved on 26.01.2018.

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

[16] C.J. Hsieh, K.W. Chang, C.J. Lin, S. Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear SVM. In *Proc. of the 25-th International Conference on Machine Learning (ICML)*, Helsinki, Finland, 2008.

[17] S. Jaroszewicz. Uplift modeling. In Claude Sammut and Geoffrey I. Webb, editors, *Encyclopedia of Machine Learning and Data Mining*, pages 1304–1309. Springer US, Boston, MA, 2017.

[18] S. Jaroszewicz and P. Rzepakowski. Uplift modeling with survival data. In *ACM SIGKDD Workshop on Health Informatics (HI-KDD'14)*, New York City, USA, August 2014.

[19] S. Jaroszewicz and Ł. Zaniewicz. Székely regularization for uplift modeling. In Stan Matwin and Jan Mielniczuk, editors, *Challenges in Computational Statistics and Data Mining*, pages 135–154. Springer International Publishing, Cham, 2016.

[20] M. Jaśkowski and S. Jaroszewicz. Uplift modeling for clinical trial data. In *ICML 2012 Workshop on Machine Learning for Clinical Data Analysis*, Edinburgh, Scotland, June 2012.

[21] A.F. Connors Jr., T. Speroff, N.V. Dawson NV, et al. The effectiveness of right heart catheterization in the initial care of critically ill patients. *JAMA*, 276(11):889–897, 1996.

[22] J. Koronacki and J. Ćwik. *Statystyczne systemy uczące się*. Exit, Warsaw, 2008. In Polish.

[23] H.J. Kushner and G.G. Yin. *Stochastic approximation and recursive algorithms and applications*. Springer-Verlag, 2003.

[24] F. Kuusisto, V. Santos Costa, H. Nassif, E. Burnside, D. Page, and J. Shavlik. Support vector machines for differential prediction. In *Proceedings of the European Conference on Machine Learning (ECML)*, pages 50–65, 2014.

[25] K. Larsen. Net lift models: Optimizing the impact of your marketing. In *Predictive Analytics World*, 2011. Workshop presentation.

[26] V.S.Y. Lo. The true lift model – a novel data mining approach to response modeling in database marketing. *SIGKDD Explorations*, 4(2):78–86, 2002.

[27] D. Pechyony, R. Jones, and X. Li. A joint optimization of incrementality and revenue to satisfy both advertiser and publisher. In *Proceedings of the 22Nd International Conference on World Wide Web*, pages 123–124, New York, NY, USA, 2013. ACM.

[28] M. Pintilie. *Competing risks : a practical perspective*. John Wiley & Sons Inc., 2006.

[29] B.T. Polyak and A.B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM J. of Control and Optimization*, 30(4):838–855, 1992.

[30] N.J. Radcliffe and P.D. Surry. Differential response analysis: Modeling true response by isolating the effect of a single action. In *Proceedings of Credit Scoring and Credit Control VI*. Credit Research Centre, University of Edinburgh Management School, 1999.

[31] N.J. Radcliffe and P.D. Surry. Real-world uplift modelling with significance-based uplift trees. Portrait Technical Report TR-2011-1, Stochastic Solutions, 2011.

[32] R. Rifkin, M. Pontil, and A. Verri. A note on support vector machine degeneracy. In *Algorithmic Learning Theory*, pages 252–263, 1999.

[33] J. Robins. Correcting for non-compliance in randomized trials using structural nested mean models. *Communications in Statistics - Theory and Methods*, 23(8):2379–2412, 1994.

[34] J. Robins and A. Rotnitzky. Estimation of treatment effects in randomised trials with non-compliance and a dichotomous outcome using structural mean models. *Biometrika*, 91(4): 763–783, 2004.

[35] P.R. Rosenbaum. Model-based direct adjustment. *Journal of the American Statistical Association*, 82(398):387–394, 1987.

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

[37] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[38] P. Rzepakowski and S. Jaroszewicz. Decision trees for uplift modeling. In *Proc. of the 10th IEEE International Conference on Data Mining (ICDM)*, pages 441–450, Sydney, Australia, December 2010.

[39] P. Rzepakowski and S. Jaroszewicz. Decision trees for uplift modeling with single and multiple treatments. *Knowledge and Information Systems*, 32:303–327, August 2012.

[40] A. Shashua and A. Levin. Ranking with large margin principle: Two approaches. *Advances in neural information processing systems*, 15:937–944, 2002.

[41] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, UK, 2004.

[42] A.J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

[43] M. Sołtys, S. Jaroszewicz, and P. Rzepakowski. Ensemble methods for uplift modeling. *Data Mining and Knowledge Discovery*, 29(6):1531–1559, Nov 2015.

[44] G.J. Székely and M.L. Rizzo. Testing for equal distributions in high dimension. *Interstat*, November 2004.

[45] G.J. Székely and M.L. Rizzo. Hierarchical clustering via joint between-within distances: Extending ward's minimum variance method. *Journal of Classification*, 22(2):151–183, 2005.

[46] G.J. Székely, M.L. Rizzo, and N.K. Bakirov. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6):2769–2794, 12 2007.

[47] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *International Conference on Machine Learning (ICML)*, pages 104–112, 2004.

[48] S. Vansteelandt and E. Goetghebeur. Causal inference with generalized structural mean models. *Journal of the Royal Statistical Society B*, 65(4):817––835, 2003.

[49] Ł. Zaniewicz and S. Jaroszewicz. Support vector machines for uplift modeling. In *The First IEEE ICDM Workshop on Causal Discovery (CD 2013)*, Dallas, Texas, December 2013.

[50] Ł. Zaniewicz and S. Jaroszewicz. L_p-support vector machines for uplift modeling. *Knowledge and Information Systems*, 53(1):269–296, October 2017.