Knowl Inf Syst (2017) 53:269–296
DOI 10.1007/s10115-017-1040-6

REGULAR PAPER

L_p-Support vector machines for uplift modeling

Łukasz Zaniewicz[1] · **Szymon Jaroszewicz**[1,2]

Received: 21 October 2015 / Accepted: 29 January 2017 / Published online: 27 March 2017

Abstract 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. In this paper, we present a variant of 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. Further, we extend L_p-SVMs to the case of uplift modeling and demonstrate that they allow for a more stable selection of the size of negative, neutral and positive groups. Finally, we present quadratic and convex optimization methods for efficiently solving the two proposed optimization tasks.

Keywords Uplift modeling · Control group · Support vector machine

1 Introduction

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

---

✉ Szymon Jaroszewicz
s.jaroszewicz@ipipan.waw.pl

[1] Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland

[2] National Institute of Telecommunications, Warsaw, Poland

Table 1 Potential (left) and observed (right) outcomes of a direct marketing campaign

Customer	Potential outcomes		Uplift	Was targeted?	Observed outcomes		Uplift
	Treatment	Control			Treatment	Control	
Adam	1	0	$+1$	Yes	1	–	$\{+1, 0\}$
Betty	1	1	0	No	–	1	$\{0, -1\}$
Cyril	0	0	0	No	–	0	$\{+1, 0\}$
Deborah	0	1	-1	Yes	0	–	$\{0, -1\}$

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.

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 [8], 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 Sect. 6.2, 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 [10], 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 [10] 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., [10]) 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* [10]. 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 shows potential outcomes for an example marketing campaign (1 is considered success, 0 a failure). For example, Adam would not have bought the product had he not been targeted, but he would buy a 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, 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 would not have bought the product regardless of being targeted or not. From the point of view of the 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 [8,21]. 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 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. This fact poses challenges for learning and evaluating uplift models.

In this paper, 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. 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.

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 [1,5] to the uplift modeling problem is then described. Those models are not susceptible to such discontinuities.

1.1 Previous work

Surprisingly, uplift modeling has received relatively little attention in the literature. The most obvious approach 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’. See [21] for an illustrative example.

A few papers addressed decision tree construction for uplift modeling. See, e.g., [4,8,20,21]. Those approaches build a single tree by simultaneously splitting the two training datasets based on modified test selection criteria. For example, Radcliffe and Surry [21] use a criterion based on a statistical test on the interaction term between the treatment indicator and the split variable. In [25], uplift decision trees have been presented which are more in line with modern machine learning algorithms. Splitting criteria are based on information theoretical measures, and a dedicated pruning strategy is presented. The approach has been extended to the case of multiple treatments in [27].

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 Rzepakowski and Jaroszewicz [25,27] with splitting criteria modified to include extra randomization have been described by Guelman et al. [7]. A thorough analysis of various types of ensembles in the uplift setting can be found in [28]. The comparison includes bagging and random forests. A theoretical justification for good performance of uplift ensembles is also provided.

Some regression techniques for uplift modeling are available. Most researchers follow the two-model approach either explicitly or implicitly [16,17], but some dedicated approaches are also available [23,24,30]. In [14], 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 class variable transformation. Recently, Pechyony et al. [18] 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 is beyond the scope of this paper.

Fairly recent and thorough literature overviews on uplift modeling can be found in [25] and [21].

Another type of uplift support vector machines was proposed in [15]. 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 [29] to obtain an algorithm which maximizes the desired quantity. Experimental comparison with our approach is given in Sect. 6.2.

In [13], an SVM-based approach has been presented for uplift modeling in 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 beyond the scope of this paper.

Support vector machines with parallel hyperplanes, similar to our approach, have been analyzed in the context of ordinal classification [27]; here the situation is different as two training datasets are involved.

A preliminary version of this paper appeared in [31]. The current paper significantly extends that first version. The most important addition is the practical and theoretical demonstration of discontinuity problems with L_1-USVMs and the introduction of L_p uplift support vector machines which do not suffer from such problems. The second 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. Thanks to better convergence and efficiency of the new optimizers, the experimental section has been reworked, the presented results now being more stable and repeatable. Finally, we added a definition of a true uplift loss and proved that the proposed model minimizes an upper bound on it.

2 Uplift support vector machines

We now introduce the 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 $\langle \mathbf{x}_1, \mathbf{x}_2 \rangle$.

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.

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 ith data record, m is the number of attributes 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.

In the rest of the paper, 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}. An *uplift model* is defined as a function

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

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 subject 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 no action was 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 $b_1, b_2 \in \mathbb{R}$ are the intercepts. The model predictions are specified by the following equation

$$M(\mathbf{x}) = \begin{cases} +1 & \text{if } \langle \mathbf{w}, \mathbf{x} \rangle > b_1 \text{ and } \langle \mathbf{w}, \mathbf{x} \rangle > b_2, \\ 0 & \text{if } \langle \mathbf{w}, \mathbf{x} \rangle \le b_1 \text{ and } \langle \mathbf{w}, \mathbf{x} \rangle > b_2, \\ -1 & \text{if } \langle \mathbf{w}, \mathbf{x} \rangle \le b_1 \text{ and } \langle \mathbf{w}, \mathbf{x} \rangle \le b_2. \end{cases} \tag{2}$$

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 and neutral points, and H_2 neutral and negative points. Notice that the model is valid iff $b_1 \ge b_2$; in Lemmas 1 and 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*.

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

$$+ C_2 \sum_{\mathbf{D}_+^T \cup \mathbf{D}_-^C} \xi_{i,2} + C_1 \sum_{\mathbf{D}_-^T \cup \mathbf{D}_+^C} \xi_{i,2}, \tag{3}$$

subject to the following constraints

$$\langle \mathbf{w}, \mathbf{x}_i\rangle - b_1 \geq +1 - \xi_{i,1}, \text{ for all } (\mathbf{x}_i, y_i) \in \mathbf{D}_+^T \cup \mathbf{D}_-^C, \tag{4}$$

$$\langle \mathbf{w}, \mathbf{x}_i\rangle - b_1 \leq -1 + \xi_{i,1}, \text{ for all } (\mathbf{x}_i, y_i) \in \mathbf{D}_-^T \cup \mathbf{D}_+^C, \tag{5}$$

$$\langle \mathbf{w}, \mathbf{x}_i\rangle - b_2 \geq +1 - \xi_{i,2}, \text{ for all } (\mathbf{x}_i, y_i) \in \mathbf{D}_+^T \cup \mathbf{D}_-^C, \tag{6}$$

$$\langle \mathbf{w}, \mathbf{x}_i\rangle - b_2 \leq -1 + \xi_{i,2}, \text{ for all } (\mathbf{x}_i, y_i) \in \mathbf{D}_-^T \cup \mathbf{D}_+^C, \tag{7}$$

$$\xi_{i,j} \geq 0, \text{ for all } i = 1, \ldots, n, \;\; j \in \{1, 2\}, \tag{8}$$

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 to the hyperplane H_1 and H_2, respectively. 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 plane and point lying on a positive or negative side of a plane, we implicitly assume that the width of the margin is also taken into account.

The situation is graphically depicted in Fig. 1. Example 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 handpicked 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. 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 $C_1\xi_{i,1}$ 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 2 and 3 in the next section). Analogous penalty is introduced for points in $\mathbf{D}_-^T$ and $\mathbf{D}_+^C$ in the fifth term of (3). In Fig. 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). Those data points are thus additionally penalized for being on the wrong side of the hyperplane H_2 with

---

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

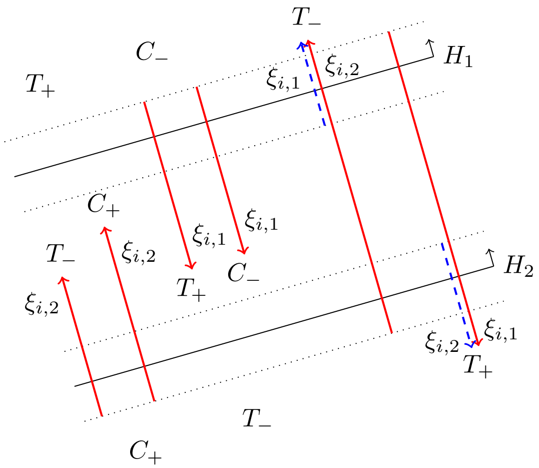

Fig. 1 The uplift SVM optimization problem. Example points belonging to the positive class in the treatment and control groups are marked, respectively, with T_+ and C_+. Analogous notation is used for points in the negative class. The figure shows penalties incurred by points with respect to the two hyperplanes of the USVM. Positive sides of hyperplanes are indicated by *small arrows* at the *right* ends of *lines* in the image. *Red solid arrows* denote the penalties incurred by points which lie on the wrong side of a single hyperplane, and *blue dashed arrows* denote additional penalties for being misclassified also by the second hyperplane

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 Fig. 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 Fig. 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 analog 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

$$l(y^T, y^C, a) = \begin{cases} -u & \text{if } a = +1, \\ u & \text{if } a = -1, \\ 0 & \text{if } a = 0 \text{ and } u = 0, \\ \rho & \text{otherwise}, \end{cases} \tag{9}$$

where $0 \le \rho \le 1$ 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.

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

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

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 Sect. 1) it is not possible in practice. We can, however, optimize an upper bound on it as shown in the following theorem.

Theorem 1 *The quantity optimized in the USVM optimization task given in Eq. 3 is an upper bound on the sum of the true uplift loss l over all training records in $\mathbf{D}^C$ and $\mathbf{D}^T$.*

The proof is found in "Appendix."

3 Properties of the uplift support vector machines (USVMs)

In this section, we analyze some mathematical properties of uplift support vector machines (USVMs), especially in those related to the influence of the parameters C_1 and C_2 on model's behavior. One of the more important results is how the ratio of the penalty parameters $\frac{C_2}{C_1}$ directly influences the number of records which are classified as neutral, or, in other words, how it influences the distance between the two separating hyperplanes. This also sheds light on the interpretation of the model.

Lemma 1 *Let $\mathbf{w}^*, b_1^*, b_2^*$ be a solution to the uplift SVM optimization problem given by Eqs. 3–8. If $C_2 > C_1$ then $b_1^* \geq b_2^*$.*

The proof of this and the remaining lemmas is found in "Appendix." The lemma guarantees that the problem possesses a well- defined solution in the sense of Eq. 2. Moreover, it naturally constrains (together with Lemma 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 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.*

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

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 the relative cost of misclassified points with respect to the margin maximization

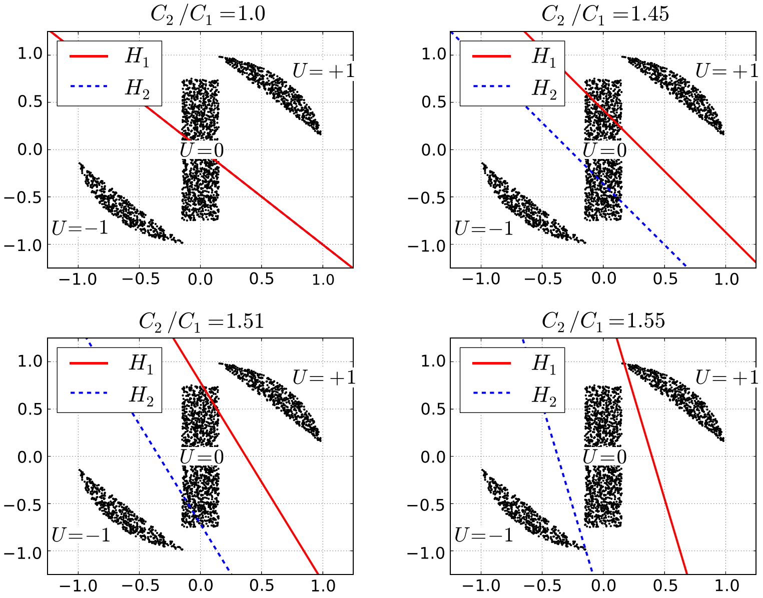

Fig. 2 The effect of the C_2/C_1 ratio on the separating hyperplanes for an artificial example

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 positive and negative 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 2 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.

4 L_p-Uplift support vector machines

Unfortunately, L_1-USVMs suffer from a problem which, in certain cases, makes Lemmas 2 and 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.

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. The lemma is stated for b_1, and the result for b_2 is analogous.

Lemma 4 *Assume that* $\mathbf{w}$ *and* b_2 *are fixed and*

$$\frac{2}{\|\mathbf{w}\|} \geq \max_i\{\langle \mathbf{w}, \mathbf{x}_i \rangle\} - \min_i\{\langle \mathbf{w}, \mathbf{x}_i \rangle\}, \tag{10}$$

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

The proof is found in "Appendix." 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 Sect. 6.1.

Lemma 5 *W.l.o.g. 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.

The proof is found in "Appendix". 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 [22]. 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 [1,5] to uplift modeling, and which, since they depend continuously on the parameter C_2, do not suffer from the aforementioned problem.

4.2 L_p-Uplift support vector machines. definition

To avoid confusion, USVMs from the previous sections will be referred to as L_1-USVMs.

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 [1,5]. In the uplift case, the quantity being optimized (analog of Eq. 3) now becomes

$$\min_{\mathbf{w},b_1,b_2\in\mathbb{R}^{m+2}} \frac{1}{2}\langle\mathbf{w},\mathbf{w}\rangle + C_1 \sum_{\mathbf{D}_+^T\cup\mathbf{D}_-^C} |\xi_{i,1}|^p + C_2 \sum_{\mathbf{D}_-^T\cup\mathbf{D}_+^C} |\xi_{i,1}|^p + C_2 \sum_{\mathbf{D}_+^T\cup\mathbf{D}_-^C} |\xi_{i,2}|^p + C_1 \sum_{\mathbf{D}_-^T\cup\mathbf{D}_+^C} |\xi_{i,2}|^p, \tag{11}$$

and the optimization is performed subject to

$$\langle\mathbf{w},\mathbf{x}_i\rangle - b_1 \geq +1 - \xi_{i,1},\ \text{for all } (\mathbf{x}_i, y_i) \in \mathbf{D}_+^T \cup \mathbf{D}_-^C, \tag{12}$$

$$\langle\mathbf{w},\mathbf{x}_i\rangle - b_1 \leq -1 + \xi_{i,1},\ \text{for all } (\mathbf{x}_i, y_i) \in \mathbf{D}_-^T \cup \mathbf{D}_+^C, \tag{13}$$

$$\langle\mathbf{w},\mathbf{x}_i\rangle - b_2 \geq +1 - \xi_{i,2},\ \text{for all } (\mathbf{x}_i, y_i) \in \mathbf{D}_+^T \cup \mathbf{D}_-^C, \tag{14}$$

$$\langle\mathbf{w},\mathbf{x}_i\rangle - b_2 \leq -1 + \xi_{i,2},\ \text{for all } (\mathbf{x}_i, y_i) \in \mathbf{D}_-^T \cup \mathbf{D}_+^C. \tag{15}$$

Note that these are the first four constraints (4)–(7) used 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 1 and Lemmas 1–3 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.

5 The uplift support vector machine optimization task

Let us now present the dual of the uplift support vector machine optimization task. Later in this section we will introduce the dual for the L_p-USVMs and discuss in detail methods of solving both problems.

We first introduce a class variable transformation

$$z_i = \begin{cases} y_i, & \text{if } (\mathbf{x}_i, y_i) \in \mathbf{D}^T, \\ -y_i, & \text{if } (\mathbf{x}_i, y_i) \in \mathbf{D}^C. \end{cases} \tag{16}$$

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 [14] in the context of uplift modeling and logistic regression.

This variable transformation allows us to simplify the optimization problem given in Eqs. 3–8 by merging (4) with (5) and (6) with (7). The simplified optimization problem is

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

subject to constraints

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

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

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

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 conditions. We begin by deriving w.r.t. $\mathbf{w}$

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

from which we obtain

$$\mathbf{w} = \sum_{i=1}^{n} (\alpha_i + \beta_i) z_i \mathbf{x}_i. \tag{17}$$

We obtain the remaining derivatives in a similar fashion

$$\frac{\partial L}{\partial b_1} = \sum_{i=1}^{n} \alpha_i z_i = 0, \qquad \frac{\partial L}{\partial b_2} = \sum_{i=1}^{n} \beta_i z_i = 0, \tag{18}$$

$$\frac{\partial L}{\partial \xi_{i,1}} = C_1 \mathbb{1}_{[z_i=+1]} + C_2 \mathbb{1}_{[z_i=-1]} - \alpha_i - r_i = 0, \tag{19}$$

$$\frac{\partial L}{\partial \xi_{i,2}} = C_1 \mathbb{1}_{[z_i=-1]} + C_2 \mathbb{1}_{[z_i=+1]} - \beta_i - p_i = 0. \tag{20}$$

Plugging Eqs. 19, 20 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 **w** from Eq. 17 and using Eq. 18, we get

$$\begin{aligned} L &= \frac{1}{2}\sum_{i,j=1}^{n}(\alpha_i+\beta_i)(\alpha_j+\beta_j)z_i z_j\langle \mathbf{x}_i,\mathbf{x}_j\rangle \\ &\quad - \sum_{i,j=1}^{n}(\alpha_i+\beta_i)(\alpha_j+\beta_j)z_i z_j\langle \mathbf{x}_j,\mathbf{x}_i\rangle \\ &\quad + b_1\sum_{i=1}^{n}\alpha_i z_i + \sum_{i=1}^{n}\alpha_i + b_2\sum_{i=1}^{n}\beta_i z_i + \sum_{i=1}^{n}\beta_i \\ &= \sum_{i=1}^{n}(\alpha_i+\beta_i) - \frac{1}{2}\sum_{i,j=1}^{n}(\alpha_i+\beta_i)(\alpha_j+\beta_j)z_i z_j\langle \mathbf{x}_i,\mathbf{x}_j\rangle, \end{aligned} \tag{21}$$

which we maximize over α_i, β_i.

Finally, from the assumption that $r_i, p_i \geq 0$ and (19), (20) combined with the KKT condition on nonnegativity of α_i, β_i and from (18), we obtain the following constraints for the dual optimization problem

$$0 \leq \alpha_i \leq C_1\mathbb{1}_{[z_i=+1]} + C_2\mathbb{1}_{[z_i=-1]}, \tag{22}$$

$$0 \leq \beta_i \leq C_1\mathbb{1}_{[z_i=-1]} + C_2\mathbb{1}_{[z_i=+1]}, \tag{23}$$

$$\sum_{i=1}^{n}\alpha_i z_i = \sum_{i=1}^{n}\beta_i z_i = 0. \tag{24}$$

5.1 Dual optimization task for L_p-USVMs

We use a similar approach to obtain the dual for the L_p-USVM problem. See [1] for an analogous derivation for classification L_p-SVMs.

After applying the variable transformation (16), 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 KKT conditions, 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

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i z_i \mathbf{x}_i + \sum_{i=1}^{n} \beta_i z_i \mathbf{x}_i. \tag{25}$$

Similarly

$$\frac{\partial L}{\partial b_1} = \sum_{i=1}^{n} \alpha_i z_i = 0, \qquad \frac{\partial L}{\partial b_2} = \sum_{i=1}^{n} \beta_i z_i = 0, \tag{26}$$

$$\frac{\partial L}{\partial \xi_{i,1}} = pC_1|\xi_{i,1}|^{p-1} \operatorname{sgn}(\xi_{i,1})\mathbb{1}_{[z_i=+1]} + pC_2|\xi_{i,1}|^{p-1} \operatorname{sgn}(\xi_{i,1})\mathbb{1}_{[z_i=-1]} - \alpha_i = 0, \tag{27}$$

$$\frac{\partial L}{\partial \xi_{i,2}} = pC_1|\xi_{i,2}|^{p-1} \operatorname{sgn}(\xi_{i,2})\mathbb{1}_{[z_i=-1]} + pC_2|\xi_{i,2}|^{p-1} \operatorname{sgn}(\xi_{i,2})\mathbb{1}_{[z_i=+1]} - \beta_i = 0. \tag{28}$$

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 is multiplied by zero. After dropping the signum functions, we obtain

$$|\xi_{i,1}| = \left(\frac{\alpha_i}{pC_1\mathbb{1}_{[z_i=+1]} + pC_2\mathbb{1}_{[z_i=-1]}}\right)^{1/(p-1)}, \tag{29}$$

$$|\xi_{i,2}| = \left(\frac{\beta_i}{pC_1\mathbb{1}_{[z_i=-1]} + pC_2\mathbb{1}_{[z_i=+1]}}\right)^{1/(p-1)}. \tag{30}$$

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 (27) and (28), can be further simplified to

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

Using Eqs. 25, 26, 29, 30, the final form of the Lagrangian is obtained:

$$\begin{aligned}
&-\frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j z_i z_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle - \sum_{i,j=1}^{n} \alpha_i \beta_j z_i z_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\
&-\frac{1}{2}\sum_{i,j=1}^{n} \beta_i \beta_j z_i z_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle + \sum_{i=1}^{n} (\alpha_i + \beta_i) \\
&-\frac{p-1}{p}\sum_{i=1}^{n} \frac{\alpha_i^{p/(p-1)}}{\left(pC_1 \mathbb{1}_{[z_i=+1]} + pC_2 \mathbb{1}_{[z_i=-1]}\right)^{1/(p-1)}} \\
&-\frac{p-1}{p}\sum_{i=1}^{n} \frac{\beta_i^{p/(p-1)}}{\left(pC_1 \mathbb{1}_{[z_i=-1]} + pC_2 \mathbb{1}_{[z_i=+1]}\right)^{1/(p-1)}},
\end{aligned} \tag{31}$$

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

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.

5.2 The optimization algorithm

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 adapted to our problem the dual coordinate descent method [11] used in the LIBLINEAR package which is currently the most popular method of solving SVM-type optimization problems. Unfortunately, the method had poor convergence properties in the case of USVMs. We have thus used the quadratic and convex solvers from the CVXOPT library [2] and developed dedicated solvers for the Karush–Kuhn–Tucker (KKT) systems of equations needed to solve our USVM optimization problems. The solvers exploit the special structure of the systems to offer better performance and numerical accuracy. Details are given in “Appendix”.

6 Experimental evaluation

In this section, we present an experimental evaluation of the proposed uplift support vector machines. We begin with an illustrative example showing the approach applied to two datasets. Later, we present an experimental comparison with other uplift modeling methods on several benchmark datasets.

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 paper, 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 [9]. 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 are available on whether a person visited the Web site and/or made a purchase (conversion). We only focus on visits since very few conversions actually occurred. In this paper, 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 [19]. The first dataset is the bone marrow transplant (`BMT`) data which cover 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 this paper). 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 contain information typical for survival analysis, we used the method from [12] 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 [12], 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 [12] 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 [25], 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. As a postprocessing step, attributes strongly correlated with the split are removed (ideally, the division into treatment and control groups should be independent from all predictive attributes, but this is possible only in a controlled experiment). Multiclass problems are converted to binary problems with the majority class considered to be $+1$ and remaining classes -1. The procedure is described in detail in [25], where a table is given with the exact conditions used to split each dataset.

6.1 An illustrative example

We first show how the method behaves 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 Sect. 6.2.

Figures 3 and 4 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.

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 behavior analyzed theoretically in Sect. 4.1. 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.

The figures validate our interpretation presented earlier in Lemmas 2–3. The analyst can use the parameter $\frac{C_2}{C_1}$ to control the proportion of neutral predictions. Note that, overall, more points are classified as positive than as negative. This is due to the overall beneficial influence of the action.

6.2 Comparison on benchmark datasets

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 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

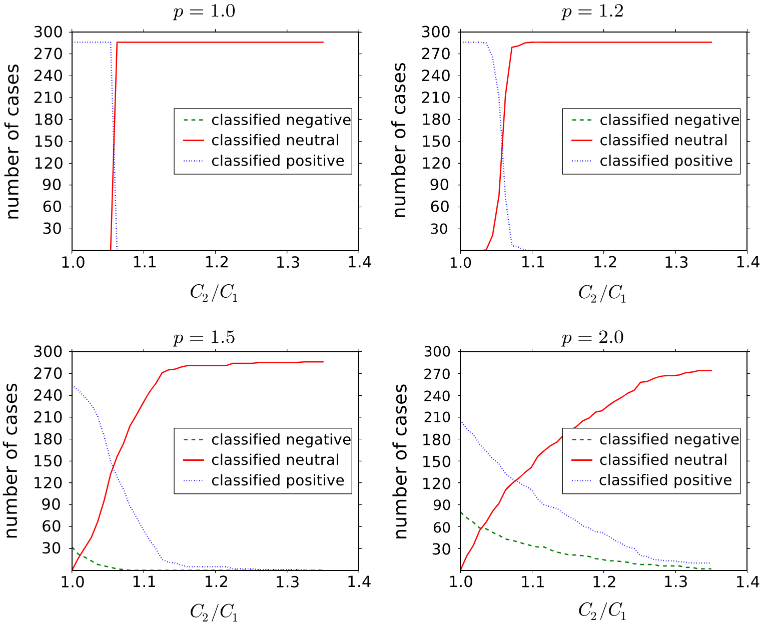

Fig. 3 Number of cases classified as positive, neutral and negative as a function of the quotient $\frac{C_2}{C_1}$ of L_p-USVM penalty coefficients for the `breast-cancer` dataset for different values of p

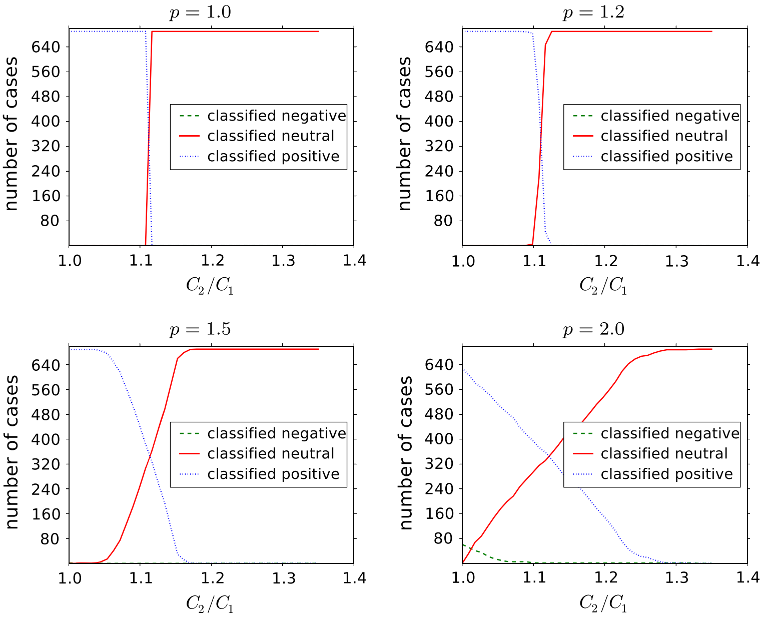

Fig. 4 Number of cases classified as positive, neutral and negative as a function of the quotient $\frac{C_2}{C_1}$ of L_p-USVM penalty coefficients for the `australian` dataset for different values of p

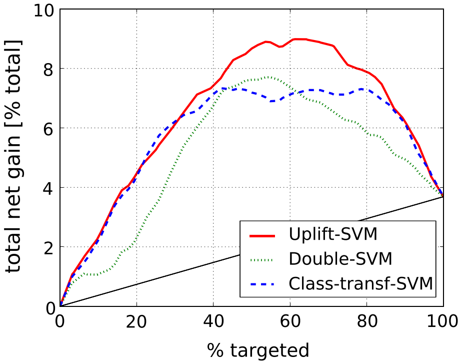

Fig. 5 Uplift curves for the `breast-cancer` dataset for uplift SVM (proposed in this paper), the double SVM approach, and an SVM uplift model based on class variable transformation (`Class-transf-SVM`). The x-axis represents the percentage of the population to which the action has been applied and the y-axis the net gain from performing the action. It can be seen that targeting about 50% 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*

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 [21,25].

Figure 5 shows uplift curves for the `breast-cancer` dataset for three of the uplift models used in the comparison (see below). It can be seen that applying the action only to some proportion of the population leads to significant gains in net success rate. 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.

We now compare 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 classifiers: the method based on building two separate SVM models (`Double-SVM`) on treatment and control groups and subtracting their predicted probabilities as well as a single support vector machine adapted to uplift modeling using the class variable transformation proposed in [14] (`Class-transf-SVM`). Since both those methods require probabilities to be predicted, the SVMs have been calibrated by training logistic regression models on their outputs. The differential prediction SVMs [15] 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 [25,26]. Splitting criterion based on the Euclidean distance was used.

The parameters of all SVM-based models have been chosen using fivefold 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, fivefold cross-validation was used to measure model performance and to pick the best parameter combination.

Table 2 Areas under the uplift curve for six uplift models on real and artificial datasets

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

* Indicates difference larger than one standard deviation
** Larger than two standard deviations

Table 2 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 testing and construction, 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 (resp. two) standard deviations by a ‘*’ (resp. ‘**’).

Table 3 Areas under the uplift curve for L_p uplift support vector machines

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

* Indicates difference larger than one standard deviation

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 [14] on 19 out of 28 datasets. Its 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 [15] 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 due to natural difficulties in predicting uplift where variances are typically much higher than in classification [21].

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 Sect. 6.1) 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 3 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, class variable transformation-based classical L_p-SVMs [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 2 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 1–3).

7 Conclusions

We have presented uplift support vector machines, an adaptation of the SVM methodology to the uplift modeling problem. The uplift SVM minimizes an upper bound on an uplift analog of the 0–1 loss. We have analyzed the proposed method theoretically and demonstrated 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. We have also proposed a modified formulation, which alleviates the problem of large changes in model behavior in response to small changes of parameter values. Finally, we have presented efficient optimization algorithms for both problem formulations.

Acknowledgements This work was supported by Research Grant no. N N516 414938 of the Polish Ministry of Science and Higher Education (Ministerstwo Nauki i Szkolnictwa Wyższego) from research funds for the period 2010–2015. Ł.Z. was co-funded by the European Union from resources of the European Social Fund. Project POKL 'Information technologies: Research and their interdisciplinary applications', Agreement UDA-POKL.04.01.01-00-051/10-00.
The authors would like to thank Finn Kuusisto for providing an implementation of differential prediction SVMs.

Appendix

Proof of Theorem 1

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 auxiliary 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 Eq. 3 (the first is responsible for regularization and is not part of the penalty) is equivalent to minimizing

$$\sum_{\mathbf{D}_+^T \cup \mathbf{D}_-^C} \xi_{i,1} + \frac{C_2}{C_1} \sum_{\mathbf{D}_-^T \cup \mathbf{D}_+^C} \xi_{i,1} + \frac{C_2}{C_1} \sum_{\mathbf{D}_+^T \cup \mathbf{D}_-^C} \xi_{i,2} + \sum_{\mathbf{D}_-^T \cup \mathbf{D}_+^C} \xi_{i,2}, \tag{32}$$

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 (8). Here $a = +1$ and $\tilde{l}(\mathbf{x}_j, g = T, a = +1) = 0 \leq \xi_{j,1} + \frac{C_2}{C_1}\xi_{j,2}$,
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 (4) and $\xi_{j,2} \geq 0$. Here $a = 0$ and $\tilde{l}(\mathbf{x}_j, g = T, a = 0) = \rho \leq \xi_{j,1} + \frac{C_2}{C_1}\xi_{j,2}$,
c) $\langle \mathbf{w}, \mathbf{x}_j \rangle - b_2 < 0$, then $\xi_{j,1} > 1$ and $\xi_{j,2} > 1$ by (4) and (6). Here $a = -1$ and $\tilde{l}(\mathbf{x}_j, g = T, a = -1) = 1 \leq \xi_{j,1} + \frac{C_2}{C_1}\xi_{j,2}$.

Summing over all training records completes the proof. □

Proofs of Lemmas 1–5

Let us begin with an observation which will be used in the proofs. Consider the uplift SVM optimization problem given by Eqs. 3–8. 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 Eq. 4 are $\xi_{i,1}^* = \max\{0, -\langle \mathbf{w}, \mathbf{x}_i \rangle + b_1 + 1\}$, and for those present in Eq. 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 Eqs. 7–8.

Proof of Lemma 1 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) 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 $C_1\xi_{i,1}^*$ 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 $(\mathbf{x}_i, y_i) \in \mathbf{D}_+^T$ contributes to the target function (3) is lower in S' than in S^*.

We now consider points penalized for crossing both hyperplanes. The idea 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^*. □

Proof of Lemma 2 Let us first consider the hyperplane H_1 (argument for H_2 is analogous). Assume that there exists at least one point in $\mathbf{D}^T_- \cup \mathbf{D}^C_+$ 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. □

Proof of Lemma 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) 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 and the second is a function of b_1 and the third of b_2. Moreover, the second term 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. □

Proof of Lemma 4 Pick any b_1 in the range

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

From (10), it follows that for all i, $|\langle \mathbf{w}, \mathbf{x}_i\rangle - b_1| \le 1$, and therefore, (8) follows from (4) to (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. Let us denote by L the goal function given in Eq. 3. Since $\mathbf{w}$ and b_2 are fixed, the first, fourth and fifth terms in (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. □

Proof of Lemma 5 The proof is similar to that for classical SVMs provided by Rifkin et al. [22]. 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 (18)–(20) and, therefore, due to Eq. 17, induces a optimal solution with $\mathbf{w} = 0$. □

Quadratic programming solution to uplift support vector machine optimization problem

In this section, we derive solutions for the KKT equations which exploit special structure of the matrices involved for improved computation speed and numerical accuracy. The KKT system follows the convention used by the CVXOPT library [2].

It is easy to see that the task of maximizing the Lagrangian (21) subject to constraints (22)–(24) can be rewritten in matrix form as minimizing

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

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

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

where $\mathbf{I}$ is the identity matrix, $\mathbf{q} = (1, 1, \ldots, 1)'$, the vector $\mathbf{h}$ is obtained from Eqs. 22 to 23, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)'$ 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, and $\mathbf{D} = \text{diag}(\mathbf{z})[\mathbf{D}^{T'} | \mathbf{D}^{C'}]'$, i.e., 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 KKT system of equations

$$\left[\begin{array}{c|c|c} \mathbf{P} & \mathbf{A}' & \mathbf{G}'\mathbf{W}^{-1} \\ \hline \mathbf{A} & 0 & 0 \\ \hline \mathbf{G} & 0 & -\mathbf{W}' \end{array}\right] \left[\begin{array}{c} \mathbf{u}_x \\ \hline \mathbf{u}_y \\ \hline \mathbf{u}_z \end{array}\right] = \left[\begin{array}{c} \mathbf{b}_x \\ \hline \mathbf{b}_y \\ \hline \mathbf{b}_z \end{array}\right], \tag{33}$$

where the diagonal weight matrix $\mathbf{W}$ and vectors $\mathbf{b}_x$, $\mathbf{b}_y$, $\mathbf{b}_z$ are supplied by the solver. The structure of this system needs to be exploited if an efficient solution is to be obtained. Applying Schur complement [3] reduces (33) to a smaller system

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

$\mathbf{u}_z$ can then be recovered as $\mathbf{u}_z = \mathbf{W}^{-1}(\mathbf{G}\mathbf{u}_x - \mathbf{b}_z)$. Using Schur complement again, we reduce the system further to

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

and solve

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

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 $\mathbf{b}$. 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

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

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 [3]. 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 [3]

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

Applying this formula is sufficient to solve the KKT system efficiently, it is, however, known to have poor numerical stability. In [6], the authors suggested the use of partial Cholesky decomposition for such systems in the case of classical SVMs. This decomposition is, however, not available in standard linear algebra packages. Instead we noticed that the quantity $(\mathbf{I}+\mathbf{X}'\mathbf{Z}^{-1}\mathbf{X})^{-1}\mathbf{X}'\mathbf{Z}^{-1}\mathbf{b}$ can be computed by solving a regularized weighted least squares problem, which can be done using the highly stable singular value decomposition. This is the method we used in our implementation.

Convex programming solution to the Lp-USVM optimization problem

The L_p-USVM is no longer (except for $p = 2$) a quadratic optimization problem so we used CVXOPT's convex optimization routine to solve it. Nevertheless, the solution is similar, with the matrix $\mathbf{P}$ replaced by the Hessian $\mathbf{H}$ of the goal function. We begin by deriving the gradient and the Hessian of the goal function. To simplify notation, define:

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

Using matrix calculus, the gradient and Hessian of (31) can be expressed, respectively, as

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

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

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

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

During each iteration, we need to solve a KKT system very similar to (33) with the matrix $\mathbf{P}$ replaced by the Hessian matrix $\mathbf{H}$. Details have been omitted since the derivation is almost identical.

References

1. Abe S (2002) Analysis of support vector machines. In: Proceedings of the 2002 12th IEEE workshop on neural networks for signal processing, 2002, pp 89–98
2. Andersen MS, Dahl J, Liu Z, Vandenberghe L (2012) Interior-point methods for large-scale cone programming. In: Optimization for machine learning, pp 55–83. MIT Press, Cambridge
3. Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge University Press, Cambridge
4. Chickering DM, Heckerman D (2000) A decision theoretic approach to targeted advertising. In: Proceedings of the 16th conference on Uncertainty in artificial intelligence (UAI), pp 82–88, Stanford, CA
5. Crisp C, Burges D (2000) Uniqueness of the SVM solution. In: Proceedings of the 1999 conference advances in neural information processing systems, vol. 2. MIT Press, Cambridge
6. Fine S, Scheinberg K (2001) Efficient SVM training using low-rank kernel representations. J Mach Learn Res 2:243–264
7. Guelman L, Guillén M, Pérez-Marín AM (2012) Random forests for uplift modeling: an insurance customer retention case. In: Modeling and simulation in engineering, economics and management, Lecture Notes in Business Information Processing (LNBIP), vol. 115, pp 123–133. Springer, Berlin
8. Hansotia B, Rukstales B (2002) Incremental value modeling. J. Interact. Mark. 16(3):35–46
9. Hillstrom K (2008) The MineThatData e-mail analytics and data mining challenge. MineThatData blog. http://blog.minethatdata.com/2008/03/minethatdata-e-mail-analytics-and-data.html. direct link to the data: http://www.minethatdata.com/Kevin_Hillstrom_MineThatData_E-MailAnalytics_DataMiningChallenge_2008.03.20.csv
10. Holland PW (1986) Statistics and causal inference. J Am Stat Assoc 81(396):945–960
11. Hsieh CJ, Chang KW, Lin CJ, Keerthi S, Sundararajan S (2008) A dual coordinate descent method for large-scale linear SVM. In: Proceedings of the 25-th international conference on machine learning (ICML), Helsinki, Finland
12. Jaroszewicz S, Rzepakowski P (2014) Uplift modeling with survival data. In: ACM SIGKDD workshop on health informatics (HI-KDD'14), New York City, August 2014
13. Jaroszewicz S, Zaniewicz Ł (2016) Székely regularization for uplift modeling. In: Matwin S, Mielniczuk J (eds) Challenges in computational statistics and data mining. Springer International Publishing, Cham, pp 135–154
14. Jaśkowski M, Jaroszewicz S (2012) Uplift modeling for clinical trial data. In: ICML 2012 workshop on machine learning for clinical data analysis, Edinburgh, June 2012
15. Kuusisto F, Santos Costa V, Nassif H, Burnside E, Page D, Shavlik J (2014) Support vector machines for differential prediction. In: Proceedings of the ECML-PKDD
16. Larsen K (2011) Net lift models: Optimizing the impact of your marketing. In: Predictive analytics world, 2011. Workshop presentation
17. Lo VSY (2002) The true lift model–a novel data mining approach to response modeling in database marketing. SIGKDD Explor 4(2):78–86
18. Pechyony D, Jones R, Li X (2013) A joint optimization of incrementality and revenue to satisfy both advertiser and publisher. In: WWW 2013 Companion
19. Pintilie M (2006) Competing risks: a practical perspective. Wiley, New York

20. Radcliffe NJ, Surry PD (1999) 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
21. Radcliffe NJ, Surry PD (2011) Real-world uplift modelling with significance-based uplift trees. Portrait Technical Report TR-2011-1, Stochastic Solutions
22. Rifkin R, Pontil M, Verri A (1999) A note on support vector machine degeneracy. In: Algorithmic learning theory, pp 252–263
23. Robins J (1994) Correcting for non-compliance in randomized trials using structural nested mean models. Commun Stat Theory Method 23(8):2379–2412
24. Robins J, Rotnitzky A (2004) Estimation of treatment effects in randomised trials with non-compliance and a dichotomous outcome using structural mean models. Biometrika 91(4):763–783
25. Rzepakowski P, Jaroszewicz S (2010) Decision trees for uplift modeling. In: Proceedings of the 10th IEEE international conference on data mining (ICDM), pp 441–450, Sydney, Australia, December 2010
26. Rzepakowski P, Jaroszewicz S (2012) Decision trees for uplift modeling with single and multiple treatments. Knowl Inf Syst 32:303–327
27. Shashua A, Levin A (2002) Ranking with large margin principle: two approaches. Adv Neural Inf Process Syst 15:937–944
28. Sołtys M, Jaroszewicz S, Rzepakowski P (2015) Ensemble methods for uplift modeling. Data Mining Knowl Discov 29(6):1531–1559. doi:10.1007/s10618-014-0383-9
29. Tsochantaridis I, Hofmann T, Joachims T, Altun Y (2004) Support vector machine learning for interdependent and structured output spaces. In: International conference on machine learning (ICML), pp 104–112
30. Vansteelandt S, Goetghebeur E (2003) Causal inference with generalized structural mean models. J R Stat Soc B 65(4):817–835
31. Zaniewicz Ł, Jaroszewicz S (2013) Support vector machines for uplift modeling. In: The first IEEE ICDM workshop on causal discovery (CD 2013), Dallas, December 2013

Łukasz Zaniewicz is a Ph.D. candidate at the Institute of Computer Science, Polish Academy of Sciences. He received his M.Sc. and B.A. degrees in Mathematics (specialization: Statistics and Data Analysis) from Warsaw University of Technology in 2012 and 2010, respectively. His research is focused on topics that include uplift modeling, especially application of support vector machines to that area.

Szymon Jaroszewicz is a professor at the Institute of Computer Science, Polish Academy of Sciences. He received his Ph.D. degree from University of Massachusetts Boston in 2003 and his Doctor of Science degree from Institute of Computer Science, Polish Academy of Sciences in 2010. His main research interest is uplift modeling, but he is also active in other areas of data mining and statistical data analysis. He is an author of over 50 publications in those fields. He has been a member of Program Committees of several prominent data mining conferences and is a member of the editorial boards of Data Mining and Knowledge Discovery and Fundamenta Informaticae journals.