Knowl Inf Syst (2012) 32:303–327
DOI 10.1007/s10115-011-0434-0

REGULAR PAPER

Decision trees for uplift modeling with single and multiple treatments

Piotr Rzepakowski · Szymon Jaroszewicz

Received: 27 January 2011 / Revised: 16 May 2011 / Accepted: 12 July 2011 /
Published online: 29 July 2011

Abstract Most classification approaches aim at achieving high prediction accuracy on a given dataset. However, in most practical cases, some action such as mailing an offer or treating a patient is to be taken on the classified objects, and we should model not the class probabilities themselves, but instead, the *change* in class probabilities caused by the action. The action should then be performed on those objects for which it will be most profitable. This problem is known as uplift modeling, differential response analysis, or true lift modeling, but has received very little attention in machine learning literature. An important modification of the problem involves several possible actions, when for each object, the model must also decide which action should be used in order to maximize profit. In this paper, we present tree-based classifiers designed for uplift modeling in both single and multiple treatment cases. To this end, we design new splitting criteria and pruning methods. The experiments confirm the usefulness of the proposed approaches and show significant improvement over previous uplift modeling techniques.

Keywords Uplift modeling · Decision trees · Randomized controlled trial · Information theory

1 Introduction and notation

In most practical problems involving classification, the aim of building models is to later use them to select subsets of cases to which some action is to be applied. A typical example is

P. Rzepakowski · S. Jaroszewicz (✉)
National Institute of Telecommunications, Warsaw, Poland
e-mail: s.jaroszewicz@itl.waw.pl

P. Rzepakowski
Faculty of Electronics and Information Technology, Warsaw University of Technology, Warsaw, Poland
e-mail: p.rzepakowski@gmail.com

S. Jaroszewicz
Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland

training a classifier, after a pilot campaign, to predict which customers are most likely to buy *after* a marketing action. The offer is then targeted to the customers which, according to the model's predictions, are the most likely buyers. Unfortunately, this is not what the marketers want. They want to target people who will buy *because* they received the offer.

These two aims are clearly not equivalent, and certain customers may buy the product even if they have not been targeted by a campaign. Targeting them at best incurs additional cost. At worst, excessive marketing may annoy them and prevent any future purchases. It is in fact well known in the advertising community that campaigns do put off some percentage of customers and there are however no easy means of identifying them. See [12,28,25,27,29] for more information.

Similar problems arise very frequently in medicine. In a typical clinical trial, a random subgroup of patients is assigned treatment A and the other, treatment B or placebo. A statistical test is then performed to assess the *overall* difference between the two groups. If, however, treatment A only works for a subgroup of people (e.g., people with some genetic traits) and not for others, such a fact might go undetected. In some cases, the analysis is carried out separately in several subgroups, but there is no systematic methodology for automatic detection of such subgroups or modeling differences in response directly.

Despite its ubiquity and importance, the problem has received scarce attention in literature [12,28,26,29], where it is known as uplift modeling, differential response analysis, incremental value modeling, or true lift modeling. Typically, a random sample of the population is selected and subjected to the action being analyzed (a medical treatment or a marketing campaign). This sample is called the *treatment* dataset. Another, disjoint, random sample is also selected, to which the action is not applied. This is the *control* dataset, which serves as the background against which the results of the action will be evaluated. The task now is to build a model predicting not the probability of objects belonging to a given class, but the *difference* between such probabilities on the two sets of data: treatment and control.

An important modification to the problem is the case when several treatments are available to us, and we need to decide not only whether an action should be applied or not, but also which action is most profitable in a given case. The multitreatment version of the problem requires, apart from the control training data, a separate training set for each possible action. The differences from the control class probabilities now need to be modeled for each treatment, such that for each new case, the best action can be decided.

If the assignment of cases to the control and (possibly several) treatment groups is completely random, uplift modeling has another advantage: it allows for modeling the effect *caused* by the action. Objects are often subject to other actions (such as competitor's marketing campaigns) which are beyond our control and the influence of which cannot be taken into account directly. By selecting random treatment and control groups, we automatically factor out all such effects, as they apply equally to those groups. A more thorough motivation for uplift modeling can be found in [28].

While decision trees are no longer an active research area, they are still widely used in the industry (included in practically all commercial analytical products) and, as a historically first machine learning approach, are a natural first candidate to adapt to the uplift methodology. Adapting other models will be a topic of future research.

We now describe the contribution of our paper. While approaches to uplift decision tree learning are already present in the literature [6,12,28], they are typically quite basic and use simple splitting criteria which maximize class differences directly. Also, no specific pruning methodology is described ([29] being an exception). The uplift decision trees we propose are more in the style of modern algorithms [4,22,24], which use splitting criteria based on information theory. Unlike [12], which only allows two class problems and binary splits, our

algorithm can handle arbitrary number of classes and multiway splits. In contrast to other approaches, we also consider the case of multiple treatments.

Moreover, all steps of the proposed methods are carefully designed such that they are direct generalizations of standard decision trees used in classification, by which we specifically mean the CART [4] and C4.5 [24] approaches. That is, when the control group and all but one treatment groups are empty, they behave identically to decision trees known in the literature. The advantages of this approach are twofold: first, when no control data are present (this can frequently happen at lower levels of the tree), it is natural to just try to predict the class, even though we are no longer able to perform uplift modeling; second, the fact that, as a special case, the methods reduce to well known, well justified, and well researched approaches, corroborates the intuitions behind them and the design principles used.

The rest of the paper is organized as follows: the following three sections deal with the case of single treatment, while Sect. 5 describes the case of multiple treatments. The remaining part of this section introduces the notation, Sect. 2 gives an overview of the related work, Sect. 3 describes uplift decision tree construction for the single treatment case, Sect. 4 presents the experimental evaluation, and finally, Sect. 6 concludes. Proofs of the theorems are given in the Appendix.

1.1 Notation

Let us now introduce the notation used when describing uplift modeling for the single treatment case. Section 5 extends the notation to the case of multiple treatments.

Recall that nonleaf nodes of decision trees are labeled with *tests* [24]. A test may have a finite number of outcomes. We create a single test for each categorical attribute, and the outcomes of this test are all attribute's values, as is done for example in C4.5. For each numerical attribute X, we create several tests of the form $X < v$, where v is a real number. A test is created for each v being a midpoint between two consecutive different values of the attribute X present in data (treatment and control datasets are concatenated for this purpose). We omit further details as they can be found in any book on decision trees [4,24].

Tests will be denoted with uppercase letter A. The distinguished class attribute will be denoted with the letter Y. The class attribute is assumed to have a finite domain, and all tests are assumed to have finite numbers of outcomes, so all probability distributions involved are discrete. Values from the domains of attributes and test outcomes will be denoted by corresponding lowercase letters, e.g., a will denote an outcome of a test A, and y one of the classes. Similarly, $\sum_a$ is the sum over all outcomes of a test A, and $\sum_y$ is the sum over all classes.

The situation considered here is different from the standard machine learning setting in that we now have *two* datasets (samples): treatment and control. This presence of double datasets necessitates a special notation. The probabilities estimated based on the treatment dataset will be denoted by P^T and those based on the control dataset by P^C. $P^T(Y)$ will denote the probability distribution of the attribute Y estimated on the treatment sample, and $P^T(y)$ the corresponding estimate of the probability of the event $Y = y$; notation for tests and the control sample is analogous. Conditional probabilities are denoted in the usual manner, for example, $P^C(Y|a)$ is the class probability distribution conditional on the test outcome $A = a$ estimated from the control sample.

We will always use Laplace correction while estimating the probabilities P^T and P^C.

Additionally, let N^T and N^C denote the number of records in the treatment and control samples, respectively, and $N^T(a)$ and $N^C(a)$, the number of records in which the outcome of a test A is a. Finally, let $N = N^T + N^C$ and $N(a) = N^T(a) + N^C(a)$.

2 Related work

Despite its practical importance, the problem of uplift modeling has received surprisingly little attention in literature. Below, we discuss the handful of available research papers.

There are two overall approaches to uplift modeling. The obvious approach is to build two separate classifiers, one on the treatment and the other on the control dataset. For each classified object, we then subtract the class probabilities predicted by the control group classifier from those predicted by the treatment group model. This approach suffers from a major drawback: the pattern of differences between probabilities can be quite different than the pattern of the probabilities themselves, so predicting treatment and control probabilities separately can result in poor model performance [6,12,19,29]. In case of decision trees, it does not necessarily favor splits which lead to *different* responses in treatment and control groups, just splits which lead to predictable outcomes in each of the groups separately. This brings us to the second class of methods, which attempt to directly model the difference between treatment and control probabilities.

The first paper explicitly discussing uplift modeling was [28]. It presents a thorough motivation including several use cases. A modified decision tree learning algorithm is also proposed, albeit with very few details given. Recently, a thorough description of the approach has been published [29]: the decision trees have been specially adapted to the uplift case using a splitting criterion based on statistical tests of the differences between treatment and control probabilities introduced by the split. There is also a variance based pruning technique. See [29] for more details.

Hansotia and Rukstales [12] offer a detailed description of their uplift approach. They describe two ideas, one based on logistic regression and the other on decision trees. The decision tree part of [12] again describes two approaches. The first is based on building two separate trees for treatment and control groups with cross-validation used to improve the accuracy of probability estimates. The second approach, most relevant to this work, builds a single tree which explicitly models the difference between responses in treatment and control groups.

The algorithm uses a splitting criterion called $\Delta\Delta P$, which selects tests maximizing the difference between the differences between treatment and control probabilities in the left and right subtrees. Suppose we have a test A with outcomes a_0 and a_1. The splitting criterion used in [12] is defined as

$$\Delta\Delta P(A) = \left|\left(P^T(y_0|a_0) - P^C(y_0|a_0)\right) - \left(P^T(y_0|a_1) - P^C(y_0|a_1)\right)\right|,$$

where y_0 is a selected class. The criterion is based on maximizing the desired difference directly, while our approach follows the more modern criteria based on information theory. Our experiments demonstrate that this results in significant performance improvements. Moreover, $\Delta\Delta P$ works only for binary trees and two-class problems, while our approach works for multiway splits and with an arbitrary number of classes (in Sect. 4 we generalize the $\Delta\Delta P$ measure to multiway splits).

In [6], the authors propose a decision tree building method for uplift modeling. The tree is modified such that every path ends with a split on whether a given person has been treated (mailed an offer) or not. Otherwise, the algorithm is a standard decision tree construction procedure from [5], so all remaining splits are selected such that the class is well predicted, while our approach selects splits that lead to large differences between treatment and control distributions. In [18], logistic regression has been applied, along with a simple approach based on building two separate Naive Bayes classifiers. Recently, Larsen [16] proposed a

method based on so called *net weight of evidence* to train naive Bayes and so called *bifurcated regression* models.

The problem has been more popular in medical literature where the use of treatment and control groups is common. Several approaches have been proposed for modeling the difference between treatment and control responses based on regression analysis. One example are nested mean models [10,31,32] similar to regression models proposed in [18]. An overview with a list of related literature can be found in [3]. The purpose of those methods is different from the problem discussed here, as the main goal of those approaches is to demonstrate that the treatment works after taking into account confounding factors, while our goal is to *find* subgroups in which the treatment works best. Also, only linear models are used, and typically, the problem of regression, not classification is addressed.

In [1], the authors set themselves an ambitions goal of modeling long-term influence of various advertising channels on the customer. Our approach can be seen as a small part of such a process that only deals with a single campaign. Otherwise, the approach is completely different from ours.

Action rules discovery [2,9,13,30] is concerned with finding actions which should be taken to achieve a specific goal. This is different from our approach as we are trying to identify groups on which a predetermined action will have the desired effect.

Methods for measuring the performance of uplift models are discussed in [6,12,26]; these include analogs of ROC and lift curves. See Sect. 4 for more details.

A preliminary version of this paper [33] was presented at the ICDM conference in 2010. This paper contains several important extensions. Primarily, the problem of uplift modeling in the presence of multiple treatments has been analyzed. Extensions to uplift decision tree construction allowing for multiple treatments are presented in Sect. 5. Additionally, splitting criterion based on chi-squared divergence is introduced and analyzed. Several parts of the text have also been clarified and extended.

3 Decision trees for uplift modeling: the single treatment case

In this section, we describe each step of uplift decision tree construction and application in the case of a single treatment.

3.1 Splitting criterion

A key part of a decision tree learning algorithm is the criterion used to select tests in nonleaf nodes of the tree. In this section, we present two splitting criteria designed especially for the uplift modeling problem.

While previous approaches [12] used directly the difference between response probabilities, i.e., the predicted quantity, we follow an approach more typical to decision trees, which is modeling the amount of *information* that a test gives about this difference.

We will now describe several postulates that a splitting criterion should satisfy, and later, we will prove that our criteria do indeed satisfy those postulates.

1. The value of the splitting criterion should be minimum if and only if the class distributions in the treatment and control groups are the same in all branches. More formally this happens when for all outcomes of a test A we have

$$P^T(Y|a) = P^C(Y|a).$$

2. If A is statistically independent of Y in both treatment and control groups then the value of the splitting criterion should be zero.
3. If the control group is empty, the criterion should reduce to one of the classical splitting criteria used for decision tree learning.

Postulate 1 is motivated by the fact that we want to achieve as high a difference between class distributions in the treatment and control groups as possible. Postulate 2 says that tests statistically independent of the class should not be used for splitting, just as in standard decision trees. Note, however, that the analogy in this case is not perfect. It is in fact possible for the treatment and control class distributions after the split to be more similar than before, so the splitting criterion can take negative values. This means that an independent split is not necessarily the worst. Theorem 3.2 and the discussion below further clarify the situation.

3.2 Splitting criteria based on distribution divergences

As we want to maximize the differences between class distributions in treatment and control sets, it is natural that the splitting criteria we propose are based on distribution divergences [7, 11,15,17]. A distribution divergence is a measure of how much two probability distributions differ. We will only require that the divergence of two discrete distributions be nonnegative and equal to zero if and only if the two distributions are identical.

We will use three distribution divergence measures, the Kullback-Leibler divergence [7,11], the squared Euclidean distance [17], and the chi-squared divergence [7,15]. Those divergences, from a distribution $Q = (q_1, \ldots, q_n)$ to a distribution $P = (p_1, \ldots, p_n)$, are defined, respectively, as

$$KL(P : Q) = \sum_i p_i \log \frac{p_i}{q_i},$$

$$E(P : Q) = \sum_i (p_i - q_i)^2,$$

$$\chi^2(P : Q) = \sum_i \frac{(p_i - q_i)^2}{q_i}.$$

The Kullback-Leibler divergence is a well-known and widely used information theoretic measure. The squared Euclidean distance is less frequently applied to compare distributions, but has been used in literature [17,34], and applied for example to Schema Matching [14]. Chi-squared divergence has been used, e.g., to measure interestingness of rules [15].

We will argue that the squared Euclidean distance has some important advantages that make it an attractive alternative to the Kullback-Leibler and chi-squared measures. First, it is symmetric, which will have consequences for tree learning when only control data are present. We note, however, that the asymmetry of Kullback-Leibler and chi-squared divergences is not necessarily a problem in our application, as the control dataset is a natural background from which the treatment set is supposed to differ.

A second, more subtle advantage of squared Euclidean distance is its higher stability. The KL and chi-squared divergences tend to infinity if one of the q_i probabilities tends to zero, while the corresponding p_i remains nonzero. This makes estimates of its value extremely uncertain in such cases. Moreover, it is enough for just one of control group probabilities in one of the tree branches to have a small value for the divergence to be extremely large, which may result in selection of a wrong attribute.

The proposed splitting criterion for a test A is defined for any divergence measure D as

$$D_{gain}(A) = D\left(P^T(Y) : P^C(Y)|A\right) - D\left(P^T(Y) : P^C(Y)\right),$$

where $D\left(P^T(Y) : P^C(Y)|A\right)$ is the conditional divergence defined below. Substituting for D the KL-divergence, squared Euclidean distance, and the chi-squared divergence, we obtain our three proposed splitting criteria KL_{gain}, E_{gain}, and χ^2_{gain}.

The intuition behind the definition is as follows: we want to build the tree such that the distributions in the treatment and control groups differ as much as possible. The first part of the expression picks a test which leads to most divergent class distributions in each branch. We subtract the divergence between class distributions on the whole dataset in order to obtain the increase or *gain* of the divergence resulting from splitting with test A. This is completely analogous to how entropy gain [24] and Gini gain [4] are defined for standard decision trees. In fact, we will show that the analogy goes deeper, and when the control set is missing, KL_{gain} reduces to entropy gain, and E_{gain} and χ^2_{gain} reduce to Gini gain. Additionally, E_{gain} reduces to Gini gain also when the treatment set is missing. Recall that we use Laplace correction while estimating P^C and P^T, so that absent datasets lead to uniform class probability distributions.

The key problem is the definition of conditional divergence. Conditional KL-divergences have been used in literature [11] but the definition is not directly applicable to our case. The difficulty stems from the fact that the probability distributions of the test A may differ in the treatment and control groups. We have thus chosen the following definition (recall that $N = N^T + N^C$ and $N(a) = N^T(a) + N^C(a)$):

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

where the relative influence of each test value is proportional to the total number of training examples falling into its branch in both treatment and control groups. Notice that when treatment and control distributions of A are identical, the definition reduces to conditional divergence as defined in [11].

The theorem below shows that the proposed splitting criteria do indeed satisfy our postulates.

Theorem 3.1 *The KL_{gain}, E_{gain} and χ^2_{gain} test selection criteria satisfy postulates 1–3. Moreover, if the control group is empty, KL_{gain} reduces to entropy gain [22] and E_{gain} and χ^2_{gain} reduce to Gini gain [4]. Additionally when the treatment set is empty E_{gain} also reduces to Gini gain.*

The proof can be found in the Appendix. More properties of divergences can be found in [7,11]. The $\Delta\Delta P$ splitting criterion used in [12] only satisfies the first two postulates.

Notice that the value of KL_{gain}, χ^2_{gain}, and E_{gain} can be negative. Splitting a dataset can indeed lead to more similar treatment and control distributions in all leaves as can be seen in the following example.

Example 3.1 Suppose the class Y and the test A, both take values in $\{0, 1\}$. Assume now that $P^T(A = 0) = 0.3$ and $P^C(A = 0) = 0.9$ and that after splitting the data according to A we have in the left branch of the tree $P^T(Y = 1|A = 0) = 0.7$, $P^C(Y = 1|A = 0) = 0.9$ and in the right branch $P^T(Y = 1|A = 1) = 0.1$, $P^C(Y = 1|A = 1) = 0.4$. It is easy to calculate $P^T(Y = 1) = P^T(Y = 1|A = 0)P^T(A = 0) + P^T(Y = 1|A = 1)P^T(A = 1) = 0.7 \cdot 0.3 + 0.1 \cdot 0.7 = 0.28$ and similarly $P^C(Y = 1) = 0.85$. It is immediately clear that

control and treatment probabilities in both leaves are more similar than in the root. Indeed, $KL(P^T(Y) : P^C(Y)) = 1.1808$ and, assuming $N^T = N^C$, $KL(P^T(Y) : P^C(Y)|A) = 0.2636$ giving $KL_{gain} = -0.9172$.

Notice, however, the strong dependence of the test A on the split between the treatment and control datasets. The negative value of the gain is in fact a variant of the well-known Simpson's paradox [21]. In practice, it is usually desirable that the assignment of cases to treatment and control groups be independent from all attributes in the data. For example in clinical trials, great care is taken to ensure that this assumption does indeed hold. We then have the following theorem, which ensures that in such a case, all three gains stay nonnegative, just as is the case with entropy and Gini gains for classification trees.

Theorem 3.2 *If outcomes of a test A are independent of the assignment to treatment and control groups, i.e., $P^C(A) = P^T(A)$ then $KL_{gain}(A)$, $E_{gain}(A)$ and $\chi^2_{gain}(A)$ are nonnegative.*

The proof can be found in the Appendix. Recall that in this case, KL_{gain} becomes the conditional divergence known in the literature [11].

3.3 Normalization: correcting for tests with large number of splits and imbalanced treatment and control splits

In order to prevent bias toward tests with high number of outcomes, standard decision tree learning algorithms normalize the information gain dividing it by the information value (usually measured by entropy) of the test itself [24]. In our case, the normalization factor is more complicated, as the information value can be different in the control and treatment groups.

Moreover, we would like to punish tests that split the control and treatment groups in different proportions since such splits indicate that the test is not independent from the assignment of cases between the treatment and control groups. Apart from violating the assumptions of randomized trials, such splits lead to problems with probability estimation. As an extreme example, consider a test that puts all treatment group records in one subtree and all control records in another; the tree construction will proceed based on only one dataset, as in classification tree learning (except for KL_{gain} and χ^2_{gain} in case of empty treatment dataset, when they do not reduce to standard splitting criteria), but the ability to detect uplift will be completely lost.

Additionally, the normalizing factor makes our splitting criterion sensitive to relative population sizes after the split, contrary to what is claimed in [29].

The proposed normalization value for a test A is given by (recall again that $N = N^T + N^C$ is the total number of records in both treatment and control datasets)

$$I(A) = H\left(\frac{N^T}{N}, \frac{N^C}{N}\right) KL(P^T(A) : P^C(A)) + \frac{N^T}{N} H(P^T(A)) + \frac{N^C}{N} H(P^C(A)) + \frac{1}{2} \tag{2}$$

for the KL_{gain} criterion, and

$$J(A) = Gini\left(\frac{N^T}{N}, \frac{N^C}{N}\right) D(P^T(A) : P^C(A)) + \frac{N^T}{N} Gini(P^T(A)) + \frac{N^C}{N} Gini(P^C(A)) + \frac{1}{2}$$

for the E_{gain} and χ^2_{gain} criteria (D denotes, respectively, the E or χ^2 divergence). For the sake of symmetry, we use entropy related measures for KL_{gain} and Gini index-related measures for E_{gain} and χ^2_{gain}, although one can also imagine combining different types of gain and normalization factors.

The first term is responsible for penalizing uneven splits. The unevenness of splitting proportions is measured using the divergence between the distributions of the test outcomes in treatment and control datasets. Tests that are strongly dependent on group assignment will thus be strongly penalized (note that the value of $I(A)$ can be arbitrarily close to infinity). However, penalizing uneven splits only makes sense if there is enough data in *both* treatment and control groups. The $KL(P^T(A) : P^C(A))$ term is thus multiplied by $H\left(\frac{N^T}{N}, \frac{N^C}{N}\right)$ which is close to zero when there is a large imbalance between the number of data in treatment and control groups (analogous Gini-based measures are used for E_{gain} and χ^2_{gain}). The result is that when only treatment or only control data are available, the first term in the expression is zero, as penalizing uneven splits no longer makes sense.

The following two terms penalize tests with large number of outcomes [24]. Additionally, those terms allow the criterion to take into account relative sample sizes after the split. We use the sum of entropies (Gini indices) of the test outcomes in treatment and control groups weighted by the number of records in those groups.

One problem we encountered was that small values of the normalizing factor can give high preference to some tests despite their low information gain. Solutions described in literature [38] involve selecting a test only if its information gain is greater or equal to the average gain of all remaining attributes and other heuristics. We found however that just adding $\frac{1}{2}$ to the value of I or J gives much better results. Since the value is always at least $\frac{1}{2}$, it cannot inflate too much the information value of a test.

Notice that when $N^C = 0$, the criterion reduces to $H(P^T(A)) + \frac{1}{2}$ which is identical to normalization used in standard decision tree learning (except for the extra $\frac{1}{2}$). After taking the normalizing factors into account, the final splitting criteria become

$$\frac{KL_{gain}(A)}{I(A)}, \qquad \frac{E_{gain}(A)}{J(A)} \qquad \text{and} \qquad \frac{\chi^2_{gain}(A)}{J(A)}.$$

The key step of tree pruning will be discussed after the next section which describes assigning scores and actions to leaves of the tree.

3.4 Application of the tree

Once the tree has been built, its leaves will contain subgroups of objects for which the treatment class distribution differs from control class distribution. The question now is how to apply the tree to score new data and make decisions on whether the action (treatment) should be applied to objects falling into a given leaf. In general, the action should be applied only if it is expected to be profitable. We thus annotate each leaf with an expected profit, which will also be used for scoring new data.

We assign profits to leaves using an approach similar to [6,12] generalized to more than two classes. Each class y is assigned a profit v_y, that is, the expected gain if a given object (whether treated or not) falls into this class. There is also a fixed cost c of performing a given action (treatment) on a single object. Let $P^T(Y|l)$ and $P^C(Y|l)$ denote treatment and control class distributions in a leaf l. If each object in a leaf is treated, the expected profit (per object) is equal to $-c + \sum_y P^T(y|l)v_y$. If no object in the leaf is treated, the expected profit

is $\sum_y P^C(y|l)v_y$. So the expected gain from treating each object falling into that leaf is

$$-c + \sum_y v_y \Big(P^T(y|l) - P^C(y|l) \Big). \tag{3}$$

Objects falling into l should be treated only if this value is greater than zero. The value itself is used for scoring new data.

It might be beneficial to include costs directly in the uplift tree construction process. An example of such an approach can be found in [37,40], where several types of costs such as misclassification costs and costs associated with tests in the tree nodes are directly taken into account.

3.5 Pruning

Decision tree pruning is a step which has decisive influence on the generalization performance of the model. There are several pruning methods, based on statistical tests, minimum description length principle, and so on. Full discussion is beyond the scope of this paper, see [20, 23,24,38] for details.

We chose the simplest, but nevertheless effective pruning method based on using a separate validation set [20,23]. For the classification problem, after the full tree has been built on the training set, the method works by traversing the tree bottom up and testing for each node, whether replacing the subtree rooted at that node with a single leaf would improve accuracy on the validation set. If this is the case, the subtree is replaced and the process continues.

In the case of uplift modeling, obtaining an analog of accuracy is not easy. One option is assigning costs/profits to each class (see the previous section) and pruning subtrees based on the total increase in profits obtained by replacing a subtree with a leaf. Unfortunately, this method is ineffective. The total expected profit obtained in the leaves is identical to that obtained in the root of a subtree. To see this sum (3) over all leaves weighting by the probability of ending up in each leaf.

We have thus devised another measure of improvement, the *maximum class probability difference* which can be viewed as a generalization of classification accuracy to the uplift case. The idea is to look at the differences between treatment and control probabilities in the root of the subtree and in its leaves, and prune if, overall, the differences in leaves are not greater than the difference in the root. In each node, we only look at the class for which the difference was largest on the training set, and in addition, remember the sign of that difference such that only differences that have the same sign in the training and validation sets contribute to the increase in our analog of accuracy. This procedure is consistent with the goal of maximizing the difference between treatment and control probabilities.

More precisely, while building the tree on the *training* set, for each node t, we remember the class $y^*(t)$ for which the difference $\left| P^T(y^*|t) - P^C(y^*|t) \right|$ is maximal 0and also remember the sign of this difference $s^*(t) = \text{sgn}(P^T(y^*|t) - P^C(y^*|t))$. During the pruning step, suppose we are examining a subtree with root r and leaves $l_1, \ldots, l_k$. We calculate the following quantities with the stored values of y^* and s^* and all probabilities computed on the *validation* set:

$$d_1(r) = \sum_{i=1}^{k} \frac{N(l_i)}{N(r)} s^*(l_i) \Big(P^T(y^*(l_i)|l_i) - P^C(y^*(l_i)|l_i) \Big),$$

$$d_2(r) = s^*(r) \Big(P^T(y^*(r)|r) - P^C(y^*(r)|r) \Big),$$

where $N(l_i)$ is the number of validation examples (both treatment and control) falling into leaf l_i. The first quantity is the maximum class probability difference of the unpruned subtree, and the second is the maximum class probability difference we would obtain on the validation set if the subtree was pruned and replaced with a single leaf. The subtree is pruned if $d_1(r) \leq d_2(r)$.

The class y^* is an analog of the predicted class in standard classification trees. In case either the treatment or the control dataset is absent, the missing probabilities are set to zero (we do not use Laplace correction in this step). It is then easy to see that, as long as the same sets are missing in the training and validation data, d_1 and d_2 reduce to standard classification accuracies of the unpruned and pruned subtree (note, that when the treatment set is missing, the value of s^* will be negative guaranteeing that both d_1 and d_2 are nonnegative).

4 Experimental evaluation

In this section, we present the results of experimental evaluation of the proposed models. We compare four models: uplift decision trees based on E_{gain} and KL_{gain}, the method of [12] based on the $\Delta\Delta P$ criterion, and an approach which builds separate decision trees for the treatment and control groups. Throughout this section, we will refer to those models, respectively, as "Euclid", "KL", "DeltaDeltaP", and "DoubleTree". In order to improve clarity, we do not include the χ^2_{gain} in the main experiments. Instead, at the end of the section, we compare it with E_{gain} and KL_{gain}.

In order to be able to compare against the DeltaDeltaP method [12], we had to modify the $\Delta\Delta P$ criterion to work for tests with more than two outcomes. The modification is

$$\Delta\Delta P(A) = \max_{a,a'} \left[\left(P^T(y_0|a) - P^C(y_0|a) \right) - \left(P^T(y_0|a') - P^C(y_0|a') \right) \right],$$

where a and a' vary over all outcomes of the test A, and y_0 is a selected class (say the first). In other words, we take the maximum difference between any two branches, which reduces to the standard $\Delta\Delta P$ criterion for binary tests.

For the DoubleTree classifier, we used our own implementation of decision trees, identical in all possible respects to the uplift based models. This decision was made in order to avoid biasing the comparison by different procedures used during tree construction, such as different details of the pruning strategy or the use of Laplace corrections.

4.1 Methods of evaluating uplift classifiers

Discussions on assessing the quality of uplift models can be found in [12,26]. In most classifier testing schemes, some amount of data is set aside before training and is later used to assess performance. Using this approach with an uplift classifier is more difficult. We now have two test sets, one containing treated and the other control objects. The test set for the treatment group is scored using the model, and the scores can be used to calculate profits and draw lift curves. However, in order to assess the *gain* in profit, we need to take into account the behavior on the control group. This is not easy, as records in the treatment group do not have natural counterparts in the control group.

To select appropriate background data, the control dataset is scored using the same model. The gain in profits resulting from performing the action on p percent of the highest scored objects is estimated by subtracting the profit on the p percent highest scored objects from the control set from the profit on the highest scored p percent of objects from the treatment

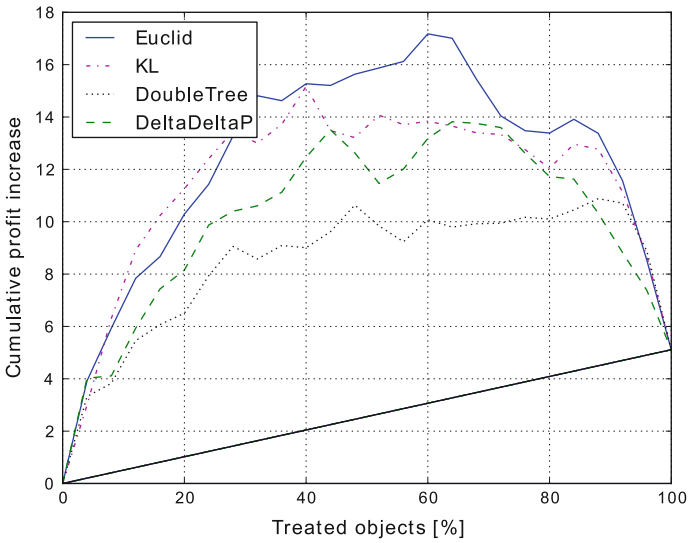

Fig. 1 The uplift curve for the `splice` dataset

dataset. This solution is not ideal as there is no guarantee that the highest scoring examples in the treatment and control groups are similar, but it works well in practice. All approaches in literature use this method [12,26].

Note that when the sizes of treatment and control datasets differ, profits calculated on the control group should be weighted to compensate for the difference.

From an equivalent point of view, this approach consists of drawing two separate lift curves for treatment and control groups using the same model and then subtracting the curves. The result of such a subtraction will be called an *uplift curve*. In this work, we will use those curves to assess model performance. To obtain comparable numerical values, we computed Areas Under the Uplift Curves (AUUC) and the heights of the curve at the 40th percentile.

Notice that, contrary to lift curves, uplift curves can achieve negative values (the results of an action can be worse than doing nothing), and the area under an uplift curve can also be negative. Figure 1 shows the uplift curves for the four analyzed classifiers on the `splice` dataset.

4.2 Dataset preparation

The biggest problem we faced was the lack of suitable data to test uplift models. While the problem itself has wide applicability, for example, in clinical trials or marketing, there seems to be very little publicly available data involving treatment and control groups. This has been noted in other papers, such as [1], where simulated data were used in experiments.

We resorted to another approach: using publicly available datasets from the UCI repository and splitting them artificially into treatment and control groups. Table 1 shows the datasets used in our study, as well as the condition used for splitting each dataset. For example, the `hepatitis` dataset was split into a treatment dataset containing records for which the condition *steroid* = *"YES"* holds and a control dataset containing the remaining records.

Table 1 Datasets used in the experiments

Dataset	Treatment/control split condition	# Removed attrs/total
acute-inflammation	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

Overall, the group assignment condition was chosen using the following rules:

1. If there is an attribute related to an action being taken, pick it (for example the steroid attribute in the `hepatitis` data).
2. Otherwise, pick the first attribute which gives a reasonably balanced split between the treatment and control groups.

We note that the selection of the splitting conditions was done **before** any experiments were carried out in order to avoid biasing the results.

A further preprocessing step was necessary in order to remove attributes that are too correlated with the splitting condition. The presence of such attributes would bias the results, since the KL and Euclid methods use the normalization factors I and J which penalize the use of such attributes, while other methods do not. A simple heuristic was used:

1. A numerical attribute was removed if its averages in the treatment and control datasets differed by more than 25%.
2. A categorical attribute was removed if the probabilities of one of its values differed between treatment and control datasets by more than 0.25.

Again, we note that the decision to remove such attributes has been made and the thresholds selected, **before** any experiments have been performed. The number of removed attributes (vs. the total number of attributes) is shown in Table 1.

Class profits were set to 1 for the most frequent class and 0 for the remaining classes. The cost of applying an action was set to 0. This way, the profits reflect the difference between the probabilities of the most frequent class.

4.3 Experimental results

To test the significance of differences between classifiers, we use the statistical testing methodology described in [8]. First, all classifiers are compared using Friedman's test, a

Table 2 Area under the uplift curve (AUUC) for various models and datasets

Dataset	DeltaDeltaP	DoubleTree	Euclid	KL
acute-inflammation	−46.86	−53.34	**−46.36**	−47.76
australian	0.22	6.02	11.20	**12.96**
breast-cancer	26.28	28.00	**36.49**	25.90
credit-a	32.47	39.32	**43.41**	42.36
dermatology	270.90	280.20	**305.33**	275.10
diabetes	88.69	82.27	**113.65**	103.71
heart-c	149.39	145.37	156.91	**162.92**
hepatitis	10.45	20.10	**22.80**	12.90
hypothyroid	−43.66	−26.85	−17.78	**−11.21**
labor	2.00	**4.52**	4.22	4.14
liver-disorders	40.69	27.96	**51.05**	43.56
nursery	−5.00	−6.00	−3.90	**−2.70**
primary-tumor	75.89	**87.65**	64.04	62.38
splice	253.12	211.65	**309.35**	289.06
winequal-red	**713.19**	626.10	708.70	658.34
winequal-white	1747.58	1351.32	1647.79	**1765.41**

Best performing method has been marked in bold

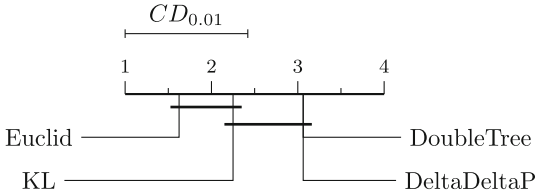

Fig. 2 Comparison of all classifiers using the Nemenyi test at $p = 0.01$. Results for area under uplift curve

nonparametric analog of ANOVA. If the test shows significant differences, a post hoc Nemenyi test is used to assess which of the models are significantly different.

All algorithm parameters have been tuned on artificial data, **not** on the datasets shown in Table 1.

Table 2 shows the results of applying the classifiers to the datasets in Table 1. Each cell contains the AUUC (Area Under the Uplift Curve) obtained by 2×5 cross-validation. The best classifier for each dataset is marked in bold. It can be seen that the model based on the squared Euclidean distance had a clear advantage. We now proceed to quantify these results using statistical tests.

We first applied the Friedman's test to check whether there are significant differences between the classifiers. The test result was that the models are significantly different with the p value of 0.0029. We thus proceeded with the post hoc Nemenyi test in order to assess the differences between specific classifiers. Figure 2 displays the results graphically. The scale marks the average rank of each model over all datasets; lower rank means a better model. For example, the model based on the squared Euclidean distance criterion had an average rank of 1.625, while the DoubleTree-based approach, an average rank of 3.06. The horizontal line in the upper part of the chart shows the *critical difference* at the significance

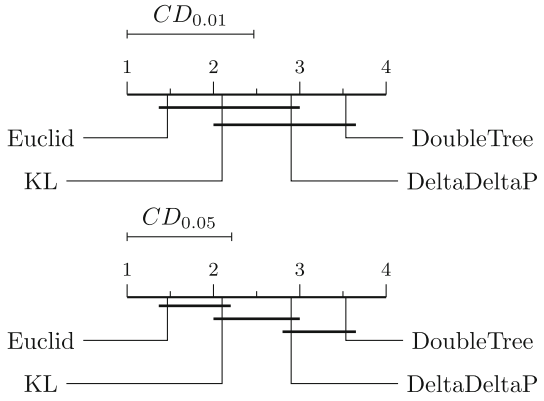

Fig. 3 Comparison of all classifiers using the Nemenyi test at $p = 0.01$ and $p = 0.05$. Results for the height of the uplift curve at the 40th percentile

level of 0.01, i.e., the minimum difference between the average ranks of classifiers, which is deemed significant. The thick lines connect models which are not statistically distinguishable.

It can be seen that Euclid is a clear winner. It is significantly better than both the DoubleTree and DeltaDeltaP approaches. The two methods we propose in this paper, KL and Euclid are not significantly different, but the Euclidean distance-based version did perform better. Also, the KL algorithm is not significantly better than other approaches.

We conclude that methods designed specifically for uplift modeling (Euclid) are indeed better than building two separate classifiers. Moreover, this approach significantly outperforms the DeltaDeltaP criterion [12,28]. In fact, there was no significant difference between DeltaDeltaP and DoubleTree. We suspect that the KL method also outperforms the DeltaDeltaP and DoubleTree approaches, but more experiments are needed to demonstrate this rigorously.

We also compared the results for the height of the uplift curve at the 40th percentile. Friedman's test showed significant differences (with the p value of 5.4×10^{-5}), so we proceeded with the Nemenyi test to further investigate the differences. We only show the results graphically in Fig. 3. The conclusions are confirmed also in this case, although sometimes only at the significance level of 0.05.

We have also ran tests to compare the χ^2_{gain} against the two other proposed uplift attribute selection measures. The χ^2_{gain} came out better than the KL_{gain} but worse than E_{gain}. The results were not statistically significant: the p value of the Friedman's test was equal to 0.06 which was above the 0.05 threshold we assumed; moreover, Nemenyi test showed that the most significant difference was between the KL_{gain} and E_{gain}. We are however convinced that the same ordering would become significant with more datasets.

5 The multiple treatments case

We now move to the case when more than one treatment is possible. The problem now is to chose, for each object, not only whether it should be treated or not, but also to pick, out of several treatments, the one which is likely to give the best result. A typical example here is a marketing campaign where more than one communication channel is possible or an e-mail campaign with several possible messages advertising the same product. Another example is

choosing an appropriate treatment for a given patient or, indeed, choosing not to treat her at all (for example because side effects of all possible treatments are too severe).

In this section, it is assumed that there are k different possible treatments $T_1, \ldots, T_k$ as well as the option of not treating an object at all. We want to built a model predicting which action is likely to give the highest gain for a given object described by an attribute vector x, i.e., pick i^*, which maximizes the quantity

$$-c_i + \sum_y v_y \left(P^{T_i}(y|x) - P^C(y|x) \right), \tag{4}$$

where c_i is the cost of applying the ith treatment and $P^{T_i}(Y|x)$ is the population class distribution resulting from the application of the ith treatment to an object with characteristics x. If, for $i = i^*$, the value is positive, treatment T_{i^*} should be used, otherwise no treatment should be applied.

Let us now introduce notation needed to describe tree construction with multiple treatments, which is a direct extension of the notation introduced in Sect. 1.1. We now have $k+1$ training datasets, one for each treatment and one for the control group. Of course, we also need $k+1$ validation and test datasets. Let N^{T_i} be the number of samples in the training set containing objects subjected to the treatment T_i. Define the sample sizes $N^T = \sum_{i=1}^k N^{T_i}$, $N = N^C + N^T$ and, for an outcome a of a test A, $N^T(a) = \sum_{i=1}^k N^{T_i}(a)$ and $N(a) = N^C(a) + N^T(a)$ analogously to the definition in Sect. 1.1. Additionally, let $P^T(A)$ denote the probability distribution of the test A in the data obtained by combining datasets for all treatments $P^T(A = a) = \frac{N^T(a)}{N^T}$.

We are now ready to describe all the stages of uplift decision tree construction in the case of multiple treatments.

Splitting criteria. Splitting criteria in the presence of multiple treatments require simultaneous comparison of several probability distributions. A few information theoretical divergence measures that involve more than two distributions have been proposed in the literature. In [36], a generalization of the so called J-divergence to multiple distributions has been proposed. Let $P_1, \ldots, P_k$ be probability distributions. Their J-divergence is defined as

$$J(P_1, \ldots, P_k) = \frac{1}{k^2} \sum_{i=1}^k \sum_{j=1}^k KL(P_i : P_j).$$

Notice that the KL-divergence in the above formula can be replaced with any other divergence measure. A few other measures, together with parametric generalizations can be found in [35].

Unfortunately, there is no single divergence measure for multiple distributions which is satisfactory in all cases. We will now introduce the proposed measure of divergence of multiple distributions which can be tuned using several parameters. This measure takes into account the special role of the control distribution and is thus more suited to the problem at hand. Define

$$\begin{aligned} &D(P^{T_1}(Y), \ldots, P^{T_k}(Y) : P^C(Y)) \\ &= \alpha \sum_{i=1}^k \lambda_i D(P^{T_i}(Y) : P^C(Y)) + (1-\alpha) \sum_{i=1}^k \sum_{j=1}^k \gamma_{ij} D(P^{T_i}(Y) : P^{T_j}(Y)), \end{aligned} \tag{5}$$

where D is an arbitrary divergence between two distributions; $\alpha \in [0, 1]$, $\lambda_i \geq 0$, and $\gamma_{ij} \geq 0$ are constants such that $\sum_{i=1}^{k} \lambda_i = 1$ and $\sum_{i=1}^{k} \sum_{j=1}^{k} \gamma_{ij} = 1$. We will now discuss the influence of the constants on the divergence and some guidelines regarding their choice.

The first term in the definition measures the divergence between all treatments and the control group. It does not measure whether the treatments differ between themselves, all that matters is that treatments should be different from the control. The second term measures the difference between treatments themselves. By changing the parameter α, the relative importance of those two terms can be changed. For $\alpha = 1$, only the difference between the treatments and the control is important and the differences between treatments themselves are ignored. This is justified if the costs of all treatments are similar and we have no preference as to which one to use. If, on the other hand, the value of α is significantly smaller than one, differences between the treatments will play an important role in choosing the splits. This is useful when the costs of treatments differ, and we would prefer to treat at least some of the objects with cheaper treatments.

For example, when we need to choose between several text messages that should be sent to a customer, it is reasonable to set $\alpha = 1$. The costs of sending each message are identical, and we have no a-priori preference toward any of them. If, on the other hand, the choice is between doing nothing, sending a text message and mailing a brochure, then the cost of the second treatment is significantly higher, and we would want to separate out the customers for whom it works best. Setting $\alpha = \frac{1}{2}$ is more appropriate.

The weights λ_i and γ_{ij} decide on the relative importance of specific treatments. Setting $\lambda_i = \frac{1}{k}$ and $\gamma_{ij} = \frac{1}{k^2}$ gives all treatments equal weights. Another choice is $\lambda_i = \frac{N^{T_i}}{N^T}$, $\gamma_{ij} = \frac{N^{T_i} N^{T_j}}{(N^T)^2}$, which has the advantage that when data on one of the treatments are missing (e.g., at lower levels during the tree construction), the divergence is identical to the divergence for the remaining treatments. Also, when data for only one treatment are present, the criterion reduces to that for uplift decision trees discussed in the preceding parts of this paper. This is not necessarily the case for other choices of λ_i and γ_{ij}. There are of course other possibilities, like setting the weights proportionally to the cost of each treatment.

We now proceed to define conditional divergence between multiple probability distributions and the corresponding gain. Conditional multiple divergence is defined as

$$\begin{aligned} &D(P^{T_1}(Y), \ldots, P^{T_k}(Y) : P^C(Y)|A) \\ &= \sum_a \frac{N(a)}{N} D\left(P^{T_1}(Y|a), \ldots, P^{T_k}(Y|a) : P^C(Y|a)\right), \end{aligned} \tag{6}$$

where D is the multiple divergence defined in Eq. 5. Of course, the conditional divergence depends on all the parameters present in (5), which are omitted from the formula. The gain is defined as

$$\begin{aligned} D_{gain}(A) &= D(P^{T_1}(Y), \ldots, P^{T_k}(Y) : P^C(Y)|A) \\ &\quad - D(P^{T_1}(Y), \ldots, P^{T_k}(Y) : P^C(Y)). \end{aligned} \tag{7}$$

We now give a theorem which shows that the desired properties of the gain continue to hold in the case of multiple treatments.

Theorem 5.1 *The following statements hold*

1. *The value of $D_{gain}(A)$ is minimum if and only if the class distributions in all treatment datasets and the control dataset are the same for all outcomes of the test A.*

2. *If A is statistically independent of Y in all treatment datasets and in the control dataset, then $D_{gain}(A) = 0$.*
3. *If $\lambda_i = \frac{N^{T_i}}{N^T}$ and $\gamma_{ij} = \frac{N^{T_i} N^{T_j}}{(N^T)^2}$ then, when only one treatment dataset is nonempty, the criterion reduces to $D_{gain}(A)$ discussed in the single treatment case.*
4. *If the outcomes of the test A are independent of the assignment to treatment and control groups, i.e., for all $1 \leq i \leq k$, $P^{T_i}(A) = P^C(A)$ then KL_{gain}, E_{gain} and χ^2_{gain} are nonnegative in the case of multiple treatments.*

Normalization. We also need to derive a normalization factor correcting the bias toward tests with large number of values and discouraging splits which result in imbalanced sizes of treatment and control training data. The equation below gives the normalizing factor based on entropy and Kullback-Leibler divergence

$$\begin{aligned} I(A) = {} & \alpha H\left(\frac{N^T}{N}, \frac{N^C}{N}\right) KL(P^T(A) : P^C(A)) \\ & + (1-\alpha) \sum_{i=1}^{k} H\left(\frac{N^{T_i}}{N^{T_i}+N^C}, \frac{N^C}{N^{T_i}+N^C}\right) KL(P^{T_i}(A) : P^C(A)) \\ & + \sum_{i=1}^{k} \frac{N^{T_i}}{N} H(P^{T_i}(A)) + \frac{N^C}{N} H(P^C(A)) + \frac{1}{2}, \end{aligned}$$

where $P^T(A)$ is the distribution of A in all treatment datasets combined. The formula is a direct extension of (2). Analogous equation can be given for the Gini index and the squared Euclidean distance or chi-squared divergence.

The first term measures the imbalance of the split between all the treatments combined and the control set. The second term measures the imbalance of the split for each treatment separately. The parameter α allows for setting the relative importance of those terms. This is in complete analogy to the α parameter in the definition of multidistribution divergence (5); $\alpha = 1$ means that we are only interested in the data being evenly distributed between treatment and control without differentiating between the treatments. When $\alpha = 0$, we require the test to split the data in a similar fashion for all the treatments. The following two terms penalize attributes with large numbers of values by summing the test entropy over all the treatment and control datasets.

Scoring. For each leaf, we compute the expected profit (4) for each treatment and pick the treatment for which the profit is maximized. Those values are stored in the leaf. The expected profit is used for scoring, and the stored treatment will be suggested for all new cases falling into that leaf.

Pruning. In order to prune uplift decision trees with multiple treatments, we will update the *maximum class probability difference* approach introduced for the single treatment case in Sect. 3.5.

For each node t, we examine the difference $\left|P^{T_i}(y|t) - P^C(y|t)\right|$ and remember the class $y^*(t)$ and treatment $i^*(t)$ for which it was maximized

$$i^*(t), y^*(t) = \arg\max_{i,y} \left|P^{T_i}(y|t) - P^C(y|t)\right|.$$

Additionally, we remember the sign of this difference $s^*(t) = \mathrm{sgn}(P^{T_{i^*}}(y^*|t) - P^C(y^*|t))$. In the pruning step, suppose we are examining a subtree with root r and leaves $l_1, \ldots, l_m$. We calculate the following quantities with the stored values of y^* and s^*, with all probabilities computed on the *validation* set:

$$d_1(r) = \sum_{i=1}^{m} \frac{N^{T_{i^*}}(l_i) + N^C(l_i)}{N^{T_{i^*}}(r) + N^C(r)} s^*(l_i) \left(P^{T_{i^*}}(y^*(l_i)|l_i) - P^C(y^*(l_i)|l_i) \right),$$

$$d_2(r) = s^*(r) \left(P^{T_{i^*}}(y^*(r)|r) - P^C(y^*(r)|r) \right),$$

where $N^{T_{i^*}}(l_i)$ is the number of validation examples in treatment T_{i^*} falling into the leaf l_i. The first quantity is the maximum class probability difference of the unpruned subtree, and the second is the maximum class probability difference we would obtain on the validation set if the subtree was pruned and replaced with a single leaf. The subtree is pruned if $d_1(r) \leq d_2(r)$.

Testing. After a model has been built, its performance should be assessed on test data. Testing in the presence of multiple treatments proves more difficult than for uplift decision trees discussed above. For each case in the test set, the model gives not only the score but also the treatment which should be used. Unfortunately, each test case has had a specific treatment applied, not necessarily the one predicted by the model.

We have decided to retain only those records in the test set for which the treatment suggested by the model matches the one actually applied; all other cases are discarded. While this approach may lead to significant data loss, it ensures correct interpretation of the results. The control group is treated in exactly the same manner as for the single treatment case described in Sect. 4.

An example. We now present an application of the multiple treatment uplift decision trees to the `splice` dataset. The dataset has been artificially split into the control and two treatment sets based on `attribute1`. Records with `attribute1` = "A" were assumed to have received treatment T_1 and records for which `attribute1` = 'G' treatment T_2. The remaining cases were assigned to the control group. As in the previous section, the choice was made before the experiment was performed.

While building a two-treatment uplift decision tree, we used the squared Euclidean distance based divergence with $\alpha = \frac{1}{2}$, λ_i set to $\frac{N^{T_i}}{N^T}$, and γ_{ij} to $\frac{N^{T_i} N^{T_j}}{(N^T)^2}$. All other aspects of the methodology, such as the use of 2×5 cross-validation, were identical to Sect. 4.

Figure 4 shows the resulting uplift curve. For comparison, we also include the curve from Fig. 1, where only a single treatment was used. This single treatment can be viewed as an "average" between T_1 and T_2. In order to make the results comparable, the curves have been renormalized by the number of examples used to compute them (recall that parts of the test set are discarded in the case of multiple treatments, as some records do not match the prescribed treatment). The costs of both treatments were identical.

It can be seen that using two different treatments and applying them alternatively according to model suggestions gave better results than using a single "average" treatment.

6 Conclusions

The paper presents a method for decision tree construction for uplift modeling. The case when several possible treatments are available has also been considered and analyzed. Splitting

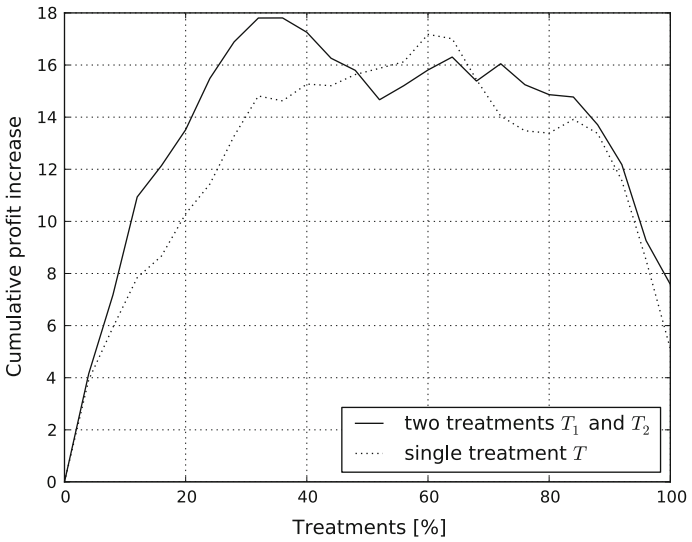

Fig. 4 Uplift curve for the `splice` dataset with two treatments

criteria and a tree pruning method have been designed specifically for the uplift modeling case and demonstrated experimentally to significantly outperform previous approaches. The methods are more in style of modern decision tree learning and in fact reduce to standard decision trees if the control dataset is missing. We are also planing on applying our information measures to recommender systems [39].

Acknowledgments 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 2010–2012.

Appendix

Proof (*of Theorem* 3.1). Recall that $N = N^T + N^C$ and $N(a) = N^T(a) + N^C(a)$. It is a well-known property of Kullback-Leibler, χ^2, and E divergences that they are zero if and only if their arguments are identical distributions and are greater than zero otherwise. Combined with the fact that the unconditional terms in the definitions of KL_{gain}, χ^2_{gain}, and E_{gain} do not depend on the test this proves postulate 1.

To prove postulate 2 notice that when the test A is independent from Y then $P^T(Y|a) = P^T(Y)$ and $P^C(Y|a) = P^C(Y)$ for all a. Thus, for any divergence D,

$$\begin{aligned} D(P^T(Y) : P^C(Y)|A) &= \sum_a \frac{N(a)}{N} D\left(P^T(Y|a) : P^C(Y|a)\right) \\ &= \sum_a \frac{N(a)}{N} D\left(P^T(Y) : P^C(Y)\right) = D\left(P^T(Y) : P^C(Y)\right), \end{aligned}$$

giving

$$\begin{aligned} D_{gain}(A) &= D\left(P^T(Y) : P^C(Y)|A\right) - D\left(P^T(Y) : P^C(Y)\right) \\ &= D\left(P^T(Y) : P^C(Y)\right) - D\left(P^T(Y) : P^C(Y)\right) = 0. \end{aligned} \tag{8}$$

To prove 3, let n be the number of classes and U the uniform distribution over all classes. It is easy to check that

$$\begin{aligned} KL\left(P^T(Y) : U\right) &= \log n - H\left(P^T(Y)\right), \\ E\left(P^T(Y) : U\right) &= \frac{n-1}{n} - Gini\left(P^T\right), \\ \chi^2\left(P^T(Y) : U\right) &= nE\left(P^T(Y) : U\right). \end{aligned}$$

Now, if $P^C(Y) = U$ and, for all a, $P^C(Y|a) = U$ (recall the use of Laplace correction while estimating the probabilities), and since $N^C = 0$, we have $N = N^T$, and $N(a) = N^T(a)$. It follows that

$$\begin{aligned} KL_{gain}(A) &= KL\left(P^T(Y) : U|A\right) - KL\left(P^T(Y) : U\right) \\ &= -\log n + H\left(P^T(Y)\right) + \sum_a \frac{N(a)}{N}\left(\log n - H\left(P^T(Y|a)\right)\right) \\ &= H\left(P^T(Y)\right) - \sum_a \frac{N^T(a)}{N^T} H\left(P^T(Y|a)\right). \end{aligned}$$

Similarly

$$\begin{aligned} E_{gain}(A) &= E\left(P^T(Y) : U|A\right) - E\left(P^T(Y) : U\right) \\ &= Gini\left(P^T(Y)\right) - \frac{n-1}{n} + \sum_a \frac{N(a)}{N}\left(\frac{n-1}{n} - Gini\left(P^T(Y|a)\right)\right) \\ &= Gini\left(P^T(Y)\right) - \sum_a \frac{N^T(a)}{N^T} Gini\left(P^T(Y|a)\right). \end{aligned}$$

Due to (8)

$$\chi^2_{gain}(A) = nE_{gain}(A) = n\left(Gini\left(P^T(Y)\right) - \sum_a \frac{N^T(a)}{N^T} Gini\left(P^T(Y|a)\right)\right).$$

Since multiplying by the constant n will not affect the choice of tests, the criterion is again equivalent to the Gini gain. The symmetry of E implies that when the treatment dataset is empty, $E_{gain}(A)$ is equal to the Gini gain of A on the control sample. □

Proof (*of Theorem* 3.2). From the independence assumption it follows that $P^T(a) = P^C(a) = P(a)$, which will be used several times in the proof. Notice that $KL(P^T(Y) : P^C(Y))$ can be written as $\sum_y P^C(y) f\left(\frac{P^T(y)}{P^C(y)}\right)$ with $f(z)$ equal to $z \log z$; f is strictly

convex. For every class y we have

$$f\left(\frac{P^T(y)}{P^C(y)}\right) = f\left(\sum_a \frac{P^T(y,a)}{P^C(y)}\right) = f\left(\sum_a \frac{P^C(y,a)}{P^C(y)} \cdot \frac{P^T(y,a)}{P^C(y,a)}\right)$$
$$\leq \sum_a \frac{P^C(y,a)}{P^C(y)} f\left(\frac{P^T(y,a)}{P^C(y,a)}\right) = \sum_a \frac{P^C(y,a)}{P^C(y)} f\left(\frac{P^T(y|a)P(a)}{P^C(y|a)P(a)}\right)$$
$$= \sum_a \frac{P^C(y,a)}{P^C(y)} f\left(\frac{P^T(y|a)}{P^C(y|a)}\right),$$

where the inequality follows from Jensen's inequality and the convexity of f. The desired result follows:

$$KL\left(P^T(Y) : P^C(Y)\right) = \sum_y P^C(y) f\left(\frac{P^T(y)}{P^C(y)}\right)$$
$$\leq \sum_a P(a) \sum_y P^C(y|a) f\left(\frac{P^T(y|a)}{P^C(y|a)}\right) = KL\left(P^T(Y) : P^C(Y)|A\right).$$

A similar proof can be found in [7]. The proof for the chi-squared based criterion is identical, except that $f(z) = (z-1)^2$ needs to be used. For the squared Euclidean distance, notice that for every class y

$$\left(P^T(y) - P^C(y)\right)^2 = \left(\sum_a P(a)\left(P^T(y|a) - P^C(y|a)\right)\right)^2$$
$$\leq \sum_a P(a)\left(P^T(y|a) - P^C(y|a)\right)^2,$$

where the inequality follows from Jensen's inequality and the convexity of z^2. We now have

$$E(P^T(Y) : P^C(Y)) = \sum_y \left(P^T(y) - P^C(y)\right)^2$$
$$\leq \sum_a P(a) \sum_y \left(P^T(y|a) - P^C(y|a)\right)^2 = E\left(P^T(Y) : P^C(Y)|A\right).$$

□

Proof (*of Theorem* 5.1). Notice that if for all a and all $1 \leq i \leq k$, $P^{T_i}(Y|a) = P^C(Y|a)$, then also $P^{T_i}(Y|a) = P^{T_j}(Y|a)$ for all $1 \leq i, j \leq k$, and, consequently, for all $1 \leq i \leq k$, $D(P^{T_i}(Y|a) : P^C(Y|a)) = 0$ and for all $1 \leq i, j \leq k$, $D(P^{T_i}(Y|a) : P^{T_j}(Y|a)) = 0$. Therefore, $D_{gain} = 0$ and parts 1 and 2 follow. For part 3, notice that λ_i and γ_{ij} have been chosen such that if data for one of the treatments is absent, all terms involving this treatment become zero.

Suppose now that for all $1 \leq i \leq k$, $P^{T_i}(A) = P^C(A)$. This obviously means that for all $1 \leq i, j \leq k$, $P^{T_i}(A) = P^{T_j}(A)$. Notice that D_{gain} can be written as (denoting $N(a)/N = P(a)$)

$$
\begin{aligned}
D_{gain}(A) &= -D(P^{T_1}(Y), \ldots, P^{T_k}(Y) : P^C(Y)) \\
&\quad + \sum_a \frac{N(a)}{N} D\left(P^{T_1}(Y|a), \ldots, P^{T_k}(Y|a) : P^C(Y|a)\right) \\
&= \alpha \sum_{i=1}^{k} \lambda_i \left[-D(P^{T_i}(Y) : P^C(Y)) + \sum_a P(a) D(P^{T_i}(Y|a) : P^C(Y|a)) \right] \\
&\quad + (1-\alpha) \sum_{i=1}^{k} \sum_{j=1}^{k} \gamma_{ij} \left[- D(P^{T_i}(Y) : P^{T_j}(Y)) \right. \\
&\quad \left. + \sum_a P(a) D(P^{T_i}(Y|a) : P^{T_j}(Y|a)) \right].
\end{aligned}
$$

Each term of the above two sums is nonnegative by nonnegativity of gain for two distributions (see proof of Theorem 3.2). □

References

1. Abe N, Verma N, Apte C, Schroko R (2004) Cross channel optimized marketing by reinforcement learning. In: Proceedings of the 10th ACM SIGKDD international conference on knowledge discovery and data mining (KDD-2004), pp 767–772
2. Adomavicius G, Tuzhilin A (1997) Discovery of actionable patterns in databases: The action hierarchy approach. In: Proceedings of the 3rd ACM SIGKDD international conference on knowledge discovery and data mining (KDD-1997), pp 111–114
3. Bellamy S, Lin J, Ten Have T (2007) An introduction to causal modeling in clinical trials. Clin Trials 4(1):58–73
4. Brieman L, Friedman JH, Olshen RA, Stone CJ (1984) Classification and regression trees. Wadsworth, Belmont
5. Buntine W (1992) Learning classification trees. Stat Comput 2(2):63–73
6. Chickering DM, Heckerman D (2000) A decision theoretic approach to targeted advertising. In: Proceedings of the 16th conference on uncertainty in artificial intelligence (UAI-2000), Stanford, CA, pp 82–88
7. Csiszár I, Shields P (2004) Information theory and statistics: a tutorial. Found Trends Commun Inf Theory 1(4):417–528
8. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
9. Drabent W, Małuszyński J (2010) Hybrid rules with well-founded semantics. Knowl Inf Syst 25(1): 137–168
10. Goetghebeur E, Lapp K (1997) The effect of treatment compliance in a placebo-controlled trial: regression with unpaired data. Appl Stat 46(3):351–364
11. Han TS, Kobayashi K (2001) Mathematics of information and coding. American Mathematical Society, USA
12. Hansotia B, Rukstales B (2002) Incremental value modeling. J Interact Market 16(3):35–46
13. Im S, Raś Z, Wasyluk H (2010) Action rule discovery from incomplete data. Knowl Inf Syst 25(1):21–33
14. Jaroszewicz S, Ivantysynova L, Scheffer T (2008) Schema matching on streams with accuracy guarantees. Intell Data Anal 12(3):253–270
15. Jaroszewicz S, Simovici DA (2001) A general measure of rule interestingness. In: Proceedings of the 5th European conference on principles of data mining and knowledge discovery (PKDD-2001), Freiburg, Germany, pp 253–265
16. Larsen K (2011) Net lift models: optimizing the impact of your marketing. In: Predictive Analytics World. Workshop presentation
17. Lee L (1999) Measures of distributional similarity. In: Proceedings of the 37th annual meeting of the association for computational linguistics (ACL-1999), pp 25–32
18. Lo VSY (2002) The true lift model—a novel data mining approach to response modeling in database marketing. SIGKDD Explor 4(2):78–86

19. Manahan C (2005) A proportional hazards approach to campaign list selection. In: Proceedings of the thirtieth annual SAS users group international conference (SUGI), Philadelphia, PA
20. Mitchell T (1997) Machine learning. McGraw Hill, New York
21. Pearl J (2000) Causality: models, reasoning, and inference. Cambridge University Press, Cambridge
22. Quinlan JR (1986) Induction of decision trees. Mach Learn 1:81–106
23. Quinlan JR (1987) Simplifying decision trees. Int J Man-Mach Stud 27(3):221–234
24. Quinlan JR (1992) C4.5: programs for machine learning. Morgan Kauffman, Los Altos
25. Radcliffe NJ (2007) Generating incremental sales. White paper, Stochastic Solutions Limited
26. Radcliffe NJ (2007) Using control groups to target on predicted lift: building and assessing uplift models. Direct Market J Direct Market Assoc Anal Council 1:14–21
27. Radcliffe NJ, Simpson R (2007) Identifying who can be saved and who will be driven away by retention activity. White paper, Stochastic Solutions Limited
28. 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
29. Radcliffe NJ, Surry PD (2011) Real-world uplift modelling with significance-based uplift trees. Portrait Technical Report TR-2011-1, Stochastic Solutions
30. Raś Z, Wyrzykowska E, Tsay L-S (2009) Action rules mining. In: Encyclopedia of Data Warehousing and Mining, vol 1, pp 1–5. IGI Global
31. Robins J (1994) Correcting for non-compliance in randomized trials using structural nested mean models. Commun Stat Theory Methods 23(8):2379–2412
32. 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
33. Rzepakowski P, Jaroszewicz S (2010) Decision trees for uplift modeling. In: Proceedings of the 10th IEEE international conference on data mining (ICDM-2010), Sydney, Australia, pp 441–450
34. Salicrú M (1992) Divergence measures: invariance under admissible reference measure changes. Soochow J Math 18(1):35–45
35. Taneja IJ (2001) Generalized information measures and their applications. http://www.mtm.ufsc.br/~taneja/book/book.html (on-line book)
36. Toussaint GT (1978) Probability of error, expected divergence, and the affinity of several distributions. IEEE Trans Syst Man Cybern (SMC) 8:482–485
37. Wang T, Qin Z, Jin Z, Zhang S (2010) Handling over-fitting in test cost-sensitive decision tree learning by feature selection, smoothing and pruning. J Syst Softw 83(7):1137–1147
38. Witten IH, Frank E (2005) Data mining: practical machine learning tools and techniques. Morgan Kaufmann, Los Altos
39. Zhang R, Tran T (2011) An information gain-based approach for recommending useful product reviews. Knowl Inf Syst 26(3):419–434
40. Zhang S (2010) Cost-sensitive classification with respect to waiting cost. Knowl Based Syst 23(5): 369–378

Author Biographies

Piotr Rzepakowski received his M.Sc. degree in computer science from Warsaw University of Technology, Poland, in 2003. Currently, he is a Ph.D. student at the Faculty of Electronics and Information Technology at Warsaw University of Technology and a research assistant at the National Institute of Telecommunications in Warsaw, Poland. His research interests include data mining, data analysis, and decision support. He has 10 years of experience in leading industrial projects related to data warehousing and data analysis mostly in the area of telecommunications.

Szymon Jaroszewicz is currently an Associate Professor at the National Institute of Telecommunications, Warsaw, Poland and at the Institute of Computer Science of the Polish Academy of Sciences. Szymon received his Master's degree in Computer Science at the Department of Computer Science at the Szczecin University of Technology in 1998 and his Ph.D. at the University of Massachusetts Boston in 2003, where in 1998 and 1999, he was a Fulbright scholar. In 2010, he received his D.Sc. degree at the Institute of Computer Science, Polish Academy of Sciences. His research interests include data analysis, data mining, and probabilistic modeling; he is the author of several publications in those fields. He has served as a program committee member for major data mining conferences and is a member of the editorial board of Data Mining and Knowledge Discovery journal.