Decision trees for uplift modeling

Piotr Rzepakowski
National Institute of Telecommunications
Warsaw, Poland
p.rzepakowski@itl.waw.pl

Faculty of Electronics and Information Technology
Warsaw University of Technology
Warsaw, Poland

Szymon Jaroszewicz
National Institute of Telecommunications
Warsaw, Poland
s.jaroszewicz@itl.waw.pl

Institute of Computer Science
Polish Academy of Sciences
Warsaw, Poland

***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. In the paper we present a tree based classifier tailored specifically to this task. To this end, we design new splitting criteria and pruning methods. The experiments confirm the usefulness of the proposed approach and show significant improvement over previous uplift modeling techniques.**

***Keywords*-uplift modeling; decision trees; information theory;**

I. Introduction and notation

In most practical problems involving classification, the aim of building models is to later use them to select a subset of cases to which some action is to be applied. A typical example is 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 marketer wants. They want to target people who will buy *because* they received the offer.

These two aims are clearly not equivalent, 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, there are however no easy means of identifying them. See [1], [2] 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 [1]–[3], 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 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, serving 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.

If the treatment group selection is completely random, this type of modeling has another advantage: it allows for modeling the effect *caused* by the action. Objects are often subject to other actions (such as competition's marketing campaigns) 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 to those groups equally. A more thorough motivation for uplift modeling can be found in [2].

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 literature [1], [2], [4], they are quite basic and use simple splitting criteria which maximize class differences directly. Also, no specific pruning methodology is described. The uplift decision trees we propose are more in the style of modern algorithms [5]–[7], which use splitting criteria based on information theory. Unlike [1], which only allows two class problems and binary splits, our algorithm can handle arbitrary number of classes and multiway splits.

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 [7] and C4.5 [6] approaches. That is, when the control group is empty, they behave identically to decision trees known in the literature. The advantages of this approach are twofold: first, when no control data is 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.

A. Notation

Let us now introduce the notation used throughout the paper. The situation considered is different from standard Machine Learning setting in that we now have *two* datasets (samples): treatment and control. This presence of double datasets necessitates a special notation.

Recall that nonleaf nodes of decision trees are labeled with *tests* [6]. A test may have a finite number of outcomes. We create a single test for each categorical attribute, 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 [6], [7].

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 probabilities estimated based on the treatment dataset will be denoted by P^T and 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 $N(a) = N^T(a) + N^C(a)$.

II. Related work

Despite its practical importance the problem of uplift modeling has received surprisingly little attention in literature. Most available publications are business whitepapers offering only vague descriptions of algorithms used [8], [9]. The details are typically omitted, probably not to disclose the inner workings of commercial products. 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 then the pattern of the probabilities themselves, so predicting treatment and control probabilities separately can result in poor model performance [1], [4], [10]. 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 [2]. It presents a thorough motivation including several use cases. A modified decision tree learning algorithm is also proposed, albeit with very few details given. It is only stated that the tree building algorithm favors splits for which on one side of the split the outcome rates in the treated group are higher than in the control group by more than in the whole population, and on the other this difference is respectively smaller. The actual splitting criterion used is probably similar to $\Delta\Delta P$ discussed below.

Hansotia and Rukstales [1] offer a detailed description of their uplift approach. They describe two ideas, one based on logistic regression, the other on decision trees. The decision tree part of [1] 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 [1] is defined as

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

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 Section IV we generalize the $\Delta\Delta P$ measure to multiway splits).

In [4], 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 [11], so all remaining splits are selected such that the class is well predicted, while our approach selects splits which lead to large differences between treatment and control distributions. In [12] logistic regression has been applied, along with a simple approach based on building two separate Naive Bayes classifiers.

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 [13]–[15] similar to regression models proposed in [12]. An overview with a list of related literature can be found in [16]. 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 [17] 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 which only deals with a single campaign. Otherwise the approach is completely different from ours.

Action rules discovery [18], [19] 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.

Ways of measuring performance of uplift models are discussed in [1], [3], [4], these include analogues of ROC and lift curves. See Section IV for more details.

III. 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 [1], [2] used directly the difference between response probabilities, i.e. the predicted quantity, we follow an approach more typical to decision trees, that is modeling the amount of *information* that a test gives about this difference.

We will now describe several postulates which a splitting criterion should satisfy, 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 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 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 discussion below further clarify the situation.

A. 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 [20]–[22]. 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 two distribution divergence measures, the Kullback-Leibler divergence [20], [22] and the squared Euclidean distance [21]. Those divergences, from a distribution $Q = (q_1, \dots, q_n)$ to a distribution $P = (p_1, \dots, p_n)$, are defined respectively as

$$\begin{aligned} KL(P : Q) &= \sum_i p_i \log \frac{p_i}{q_i}, \\ E(P : Q) &= \sum_i (p_i - q_i)^2. \end{aligned}$$

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 [21], [23], and applied for example to Schema Matching [24].

We will argue that the squared Euclidean distance has some important advantages which make it an attractive alternative to the Kullback-Leibler measure. First, it is symmetric, which will have consequences for tree learning when only control data is present. We note however, that the asymmetry of Kullback-Leibler divergence is not necessarily a problem in our application, as the control dataset is naturally a 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 divergence tends 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 [24]. Moreover, it is enough for just one of control group probabilities in one of the tree branches to have a small value for the KL-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

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

where $D\left(P^T(Y) : P^C(Y) | A\right)$ is the conditional divergence defined below. Substituting for D the KL-divergence and squared Euclidean distance we obtain our two proposed splitting criteria KL_{gain} and E_{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 [6] and Gini gain [7] are defined for standard decision trees. In fact we will show that the analogy goes deeper, and KL_{gain} reduces to entropy gain when the control set is missing, and E_{gain} reduces to Gini gain when either control or treatment set is missing. Recall that for both measures 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 [22] but the definition is not directly applicable to our case. The difficulty comes 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 $N(a) = N^T(a) + N^C(a)$):

$$\begin{aligned} &D(P^T(Y) : P^C(Y) | A) \\ &\qquad = \sum_a \frac{N(a)}{N} D\Big(P^T(Y|a) : P^C(Y|a)\Big), \end{aligned} \tag{1}$$

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 [22].

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

Theorem 3.1: The KL_{gain} and E_{gain} test selection criteria satisfy postulates 1–3. Moreover, if the control group is empty, KL_{gain} reduces to entropy gain [5], and when either the treatment or control set is empty, E_{gain} reduces to Gini gain [7].

The proof can be found in the Appendix. More properties of divergences can be found in [20], [22]. The $\Delta\Delta P$ splitting criterion used in [1] only satisfies the first two postulates.

Notice that the value of KL_{gain} and E_{gain} can be negative. Splitting a dataset can indeed lead to more similar treatment and control distributions in all leaves. This is a variant of the well known Simpson's paradox [25]. However, 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, both 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 both KL_{gain} and E_{gain} 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 [22].

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

In order to prevent a bias towards 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 [6]. 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 which 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 which 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 empty treatment dataset), but the ability to detect uplift will be completely lost.

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)

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

for the KL_{gain} criterion, and

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

for the E_{gain} criterion. For the sake of symmetry, we use entropy related measures for KL_{gain} and Gini index related measures for E_{gain}, although one can also imagine using 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 which 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}). 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 [6]. We use the sum of entropies (Gini indices) of the test's 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 [26] 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

$$\frac{KL_{gain}(A)}{I(A)} \qquad \text{and} \qquad \frac{E_{gain}(A)}{J(A)}.$$

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

C. 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 [1], [4] 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). 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

$$-c + \sum_y v_y \left(P^T(y|l) - P^C(y|l)\right). \qquad (2)$$

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

D. 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 [6], [26]–[28] for details.

We choose the simplest, but nevertheless effective pruning method based on using a separate validation set [27], [28]. 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 analogue 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 gain in profit obtained in the leaves is identical to that obtained in the root of a subtree. To see this sum (2) 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 which have the same sign on the training and validation sets contribute to the increase of our analogue 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, and also remember the sign of this difference $s(t) = \operatorname{sgn}(P^T(y^*|t) - P^C(y^*|t))$. 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:

$$\begin{aligned} d_1(r) &= \sum_{i=1}^{k} \frac{N(l_i)}{N(r)} s(l_i) \left(P^T(y^*(l_i)|l_i) - P^C(y^*(l_i)|l_i)\right), \\ d_2(r) &= s(r) \left(P^T(y^*(r)|r) - P^C(y^*(r)|r)\right), \end{aligned}$$

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 analogue 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).

IV. 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 [1] 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 be able to compare against the DeltaDeltaP method [1] we had to modify the $\Delta\Delta P$ criterion to work for tests with more than two outcomes. The modification is

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

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.

A. Methods of evaluating uplift classifiers

Discussions on assessing the quality of uplift models can be found in [1], [3]. In most classifier testing schemes, some amount of data is set aside while 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, 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

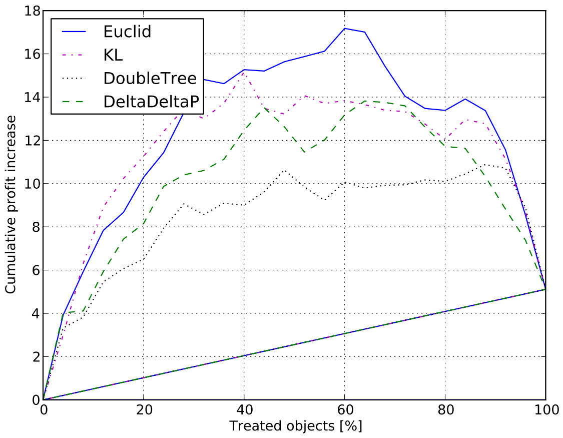

Figure 1. The uplift curve for the `splice` dataset.

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 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 [1], [3].

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 easy to compare numerical values we computed areas under 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.

B. Dataset preparation

The biggest problem we faced was 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 [17], where simulated data were used in experiments.

We resorted to another approach: using publicly available datasets from the UCI repository and splitting them into treatment and control groups. Table I 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. 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 which 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 to 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 I.

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.

C. Experimental results

To test the significance of differences between classifiers, we use the statistical testing methodology described in [29]. First, all classifiers are compared using Friedman's test, a nonparametric analogue 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 I.

Table II shows the results of applying the classifiers to the datasets in Table I. Each cell contains the AUUC (Area

Table I
DATASETS USED IN THE EXPERIMENTS.

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

Table II
AREA UNDER THE UPLIFT CURVE

dataset	DeltaDeltaP	DoubleTree	Euclid	KL
acute inflam.	-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**

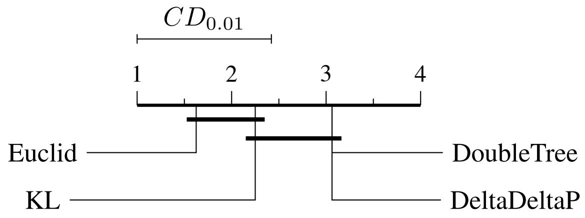

Figure 2. Comparison of all classifiers using the Nemenyi test at $p = 0.01$. Results for Area Under Uplift Curve.

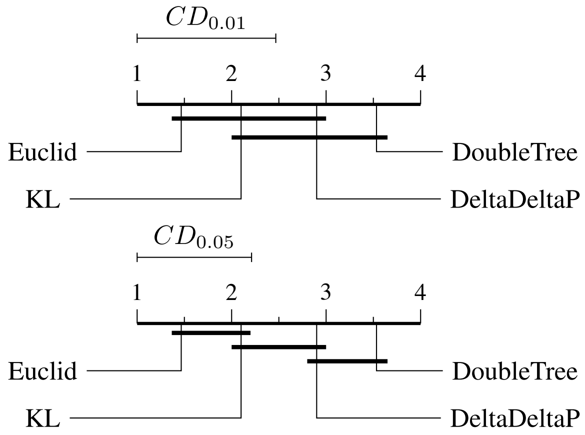

Figure 3. Comparison of all classifiers using the Nemenyi test at $p = 0.01$ and $p = 0.05$. Results for the height of the curve at the 40th percentile.

Under the Uplift Curve) obtained by 2×5 crossvalidation. 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 double tree 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 level of 0.01, that is the minimum difference between the average ranks of classifiers, which is deemed significant.

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 [1], [2]. 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 $5.4 \cdot 10^{-5}$), so we proceeded with the Nemenyi test to further investigate the differences. We only show the results graphically in Figure 3. The results are confirmed also in this case, although sometimes only at the significance level of 0.05.

V. Conclusions and Future Research

The paper presents a method for decision tree construction for uplift modeling. Two splitting criteria and a tree pruning method have been designed specifically for this purpose, and demonstrated experimentally to significantly outperform previous approaches to uplift modeling. 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. Future work will concentrate on adapting other classification methods to the problem of uplift modeling.

VI. Acknowledgments

This work was supported by Research Grant no. 4149/B/T02/2010/38 of the Polish Ministry of Science and Higher Education (Ministerstwo Nauki i Szkolnictwa Wyższego) from research funds for the period 2010–2012.

References

[1] B. Hansotia and B. Rukstales, "Incremental value modeling," *Journal of Interactive Marketing*, vol. 16, no. 3, pp. 35–46, 2002.

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

[3] N. J. Radcliffe, "Using control groups to target on predicted lift: Building and assessing uplift models," *Direct Marketing Journal, Direct Marketing Association Analytics Council*, vol. 1, pp. 14–21, 2007.

[4] D. M. Chickering and D. Heckerman, "A decision theoretic approach to targeted advertising," in *UAI*, Stanford, CA, 2000, pp. 82–88.

[5] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81–106, 1986.

[6] J. Quinlan, *C4.5: programs for machine learning*. Morgan Kauffman, 1992.

[7] L. Brieman, J. Friedman, R. Olshen, and C. Stone, "Classification and regression trees," *Wadsworth Inc*, 1984.

[8] "Generating incremental sales," White paper, Stochastic Solutions Limited, 2007.

[9] N. Radcliffe and R. Simpson, "Identifying who can be saved and who will be driven away by retention activity," White paper, Stochastic Solutions Limited, 2007.

[10] C. Manahan, "A proportional hazards approach to campaign list selection," in *SAS User Group International (SUGI) 30 Proceedings*, Philadelphia, PA, 2005.

[11] W. Buntine, "Learning classification trees," *Statistics and Computing*, vol. 2, no. 2, pp. 63–73, 1992.

[12] V. S. Y. Lo, "The true lift model - a novel data mining approach to response modeling in database marketing," *SIGKDD Explorations*, vol. 4, no. 2, pp. 78–86, 2002.

[13] J. Robins and A. Rotnitzky, "Estimation of treatment effects in randomised trials with non-compliance and a dichotomous outcome using structural mean models," *Biometrika*, vol. 91, no. 4, pp. 763–783, 2004.

[14] J. Robins, "Correcting for non-compliance in randomized trials using structural nested mean models," *Communications in Statistics - Theory and Methods*, vol. 23, no. 8, pp. 2379–2412, 1994.

[15] E. Goetghebeur and K. Lapp, "The effect of treatment compliance in a placebo-controlled trial: Regression with unpaired data," *Applied Statistics*, vol. 46, no. 3, pp. 351–364, 1997.

[16] S. Bellamy, J. Lin, and T. T. Have, "An introduction to causal modeling in clinical trials," *Clinical Trials*, vol. 4, no. 1, pp. 58–73, 2007.

[17] N. Abe, N. Verma, C. Apte, and R. Schroko, "Cross channel optimized marketing by reinforcement learning," in *KDD*, 2004, pp. 767–772.

[18] G. Adomavicius and A. Tuzhilin, "Discovery of actionable patterns in databases: The action hierarchy approach," in *KDD*, 1997, pp. 111–114.

[19] Z. Raś, E. Wyrzykowska, and L.-S. Tsay, "Action rules mining," in *Encyclopedia of Data Warehousing and Mining*. IGI Global, 2009, vol. 1, pp. 1–5.

[20] I. Csiszar and P. Shields, "Information theory and statistics: A tutorial," *Foundations and Trends in Communications and Information Theory*, vol. 1, no. 4, pp. 417–528, 2004.

[21] L. Lee, "Measures of distributional similarity," in *Proc. of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, 1999, pp. 25–32.

[22] T. Han and K. Kobayashi, *Mathematics of information and coding*. American Mathematical Society, 2001.

[23] M. Salicrú, "Divergence measures: Invariance under admissible reference measure changes," *Soochow Journal of Mathematics*, vol. 18, no. 1, pp. 35–45, Jan. 1992.

[24] S. Jaroszewicz, L. Ivantysynova, and T. Scheffer, "Schema matching on streams with accuracy guarantees," *Intelligent Data Analysis*, vol. 12, no. 3, pp. 253–270, 2008.

[25] J. Pearl, *Causality: models, reasoning, and inference*. Cambridge University Press, 2000.

[26] I. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2005.

[27] T. Mitchell, *Machine Learning*. McGraw Hill, 1997.

[28] J. Quinlan, "Simplifying decision trees," *Int. Journal of Man-Machine Studies*, vol. 27, no. 3, pp. 221–234, 1987.

[29] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

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 both Kullback-Leibler 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} 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,

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

giving

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

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

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

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

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

Similarly

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

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

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

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

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

A similar proof can be found in [20]. For the squared Euclidean distance, notice that for every class y

$$\begin{aligned} &\left(P^T(y) - P^C(y)\right)^2 \\ &= \left(\sum_a P(a)(P^T(y|a) - P^C(y|a))\right)^2 \\ &\leq \sum_a P(a)\left(P^T(y|a) - P^C(y|a)\right)^2, \end{aligned}$$

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

$$\begin{aligned} E(P^T(Y) : P^C(Y)) &= \sum_y \left(P^T(y) - P^C(y)\right)^2 \\ &\leq \sum_a P(a) \sum_y \left(P^T(y|a) - P^C(y|a)\right)^2 \\ &= KL(P^T(Y) : P^C(Y)|A). \end{aligned}$$

■