DEPARTMENT OF ENGINEERING MANAGEMENT

Attributing Value in a Data Pooling Setting for Predictive Modeling

Julie Moeyersoms, Brian d'Alessandro, Foster Provost & David Martens

UNIVERSITY OF ANTWERP
Faculty of Applied Economics
City Campus
Prinsstraat 13, B.226
B-2000 Antwerp
Tel. +32 (0)3 265 40 32
Fax +32 (0)3 265 47 99
www.uantwerpen.be

FACULTY OF APPLIED ECONOMICS

DEPARTMENT OF ENGINEERING MANAGEMENT

Attributing Value in a Data Pooling Setting for Predictive Modeling

Julie Moeyersoms, Brian d'Alessandro, Foster Provost & David Martens

RESEARCH PAPER 2017-009
SEPTEMBER 2017

University of Antwerp, City Campus, Prinsstraat 13, B-2000 Antwerp, Belgium
Research Administration – room B.226
phone: (32) 3 265 40 32
fax: (32) 3 265 47 99
e-mail: joeri.nys@uantwerpen.be

D/2017/1169/009

Attributing Value in a Data Pooling Setting for Predictive Modeling

Julie Moeyersoms, Brian d'Alessandro, Foster Provost, David Martens

September 20, 2017

Abstract

The rapid growth of data sources comes with numerous challenges. One of them is the determination of its value. That is, when building prediction models based on different data sources, it is interesting to know how much each of the features has contributed to that specific prediction. As such, we get an idea on how the benefits created by the prediction model could be divided over the features responsible for it. The goal of this paper is to define, solve and evaluate a data attribution scheme for predictive modeling that is "fair", which is defined by using concepts from game theory. We use two methods from various research fields in order to distribute the value both on an instance level and ultimately on a feature level: The (approximate) Shapley value and an explanation approach for high-dimensional data. By using a high-dimensional and sparse data set, consisting of website visits for each user, we show that: (i) the proposed methods allow to create a fair value distribution among a very large number of data sources (websites in this case) in a prediction model, and (i) are able to obtain a double amount of instances that are explained for a given number of features as compared to just looking at the high-coefficient features. Interestingly, (iii) although the proposed methods come from different sources and motivations, the two new alternatives provide strikingly similar rankings of important features and division of the revenues.

1 Introduction

The increasingly widespread collection and processing of data enables companies to use these data assets to improve decision making. A popular way to do so is by using predictive modeling. For example, in the banking sector it has been shown that payment data is very predictive for fraud detection [Junqué de Fortuny et al., 2013] and bankruptcy prediction [Tobback and Martens, 2017]. Payment data is an example of fine-grained behavioral data, others are phone call records [Fawcett and Provost, 1997, Verbeke et al., 2012], location data [Provost et al., 2015] and website visits [Dalessandro et al., 2014b]. Such data contains all the little "crumbs" that one leaves behind in the digital world and are indicative of ones interest. An important challenge for working with such behavioral data, is that it has huge dimensions and is very sparse: for example, all unique accounts that one can pay to are features, while one person will typically only pay to a few dozen/hundreds of accounts.

Previous work has demonstrated that the more data is included in the training of a predictive model, the higher the predictive performance [Junqué de Fortuny et al., 2013, Martens et al., 2016]. For our banking example this implies that having more payment data would improve the accuracy of a fraud detection or bankruptcy prediction model and therefore would reduce costs. Collecting or purchasing data is one way to increase the available data, another would be for banks to pool data between them. Imagine the opportunities for banks if they would be able to pool their data together in order to reduce the overall losses due to fraud or bankruptcy in the banking sector. The question then becomes: how to attribute the reduced costs to the different banks?

Another example can be found in the telecommunications (telco) sector, where again behavioral data (call detail records) can be used to predict fraud [Fawcett and Provost, 1997]. If telco operators would pool their records to improve their fraud detection, overall losses due to fraud would be reduced. How to attribute the reduced costs to the different telco operators? Or think of an online advertising company that tries to predict product interest, based on browsing data over a large number of websites. Viewing the behavior of users across more websites will increase the accuracy of the predictions and hence will lead to more clicks; which in turn will lead to more revenues [Dalessandro et al., 2014b]. Websites that are very predictive for product interest should be compensated more for their data. Or: how to attribute the advertising revenue to the different websites? Yet another example can be found in the world of "Internet of Things" (IoT). The rise of IoT devices will lead to the generation of massive amounts of behavioral data, which can be used to predict certain events: is a machine failure arising, is a customer on vacation or at home, etc. In any predictive application using such data, the pooling of data from devices from different manufacturers might prove to be very useful. How to divide value of the improved results across the manufacturers is what we focus on.

These examples show that companies in a wide range of industries can benefit from cooperating by pooling data. The open research question that is addressed in this paper is:

How should the profits of the use of a predictive model using a data pooling coalition be divided?

This soon leads to subjective discussions as "my data is more valuable than yours". Due to the lack of a proper data sharing design, companies are often very reluctant to share data even if it would be beneficial, both for themselves as for society as a whole. And this simply because they don't want to give away value to the data coalition partners that they are not reimbursed for. This paper attempts to answer this call for a scientific and fair distribution method to enable data pooling for predictive modeling.

Although the application areas are wide-spread, as a running example we use the targeted online adver-

tising application: a prediction model can be built on a data set containing websites (features) visited by users (instances) in order to predict which users should be shown an advertisement. Let's take the example of a high quality shopping mall wishing to do targeted advertising. The predictive modeling problem then becomes: predicting who will be interested in this mall[1] based on the websites the person has visited. The features are hence all the websites on which one has data (typically hundreds to thousands), with a 1 value indicating the user has visited the website. Visiting sites such as katespade.com or brooksbrothers.com could be considered predictive for being interested in the high quality mall. Finding these patterns is what the predictive algorithm does (and is beyond the scope of this paper, we assume the predictive model is given). Then how do we know how much of the total advertising revenue each of these features/websites should receive? Of course, the weights (βs) of a linear prediction model provide the predictive power for every website but they do not incorporate information on feature presence, which is key when analysing sparse data. Therefore, we propose to work instance-based. So for this example let's say that user A has visited website x, y and z and therefore the model predicts the user will be interested and is shown an ad. If we assume that the revenue of showing this ad to user A is 1\$, then the value attribution method should allow to divide this 1\$ over website x, y and z in a fair manner. By doing so for every instance and aggregating these scores, a value per website is obtained.

The distribution schemes proposed in this work stem from different research fields, namely game theory on the one hand, and document classification on the other. Although these techniques were designed for different purposes, similarities between their applications and ours can be seen. The first method approaches the problem as a coalition game where each website (data source) is a player and the total ad revenue should be divided among the players. More specifically, we use an adapted version of the Shapley method [Fatima et al., 2008, Shapley, 1988] to divide the gains. The Shapley value distributes the gain of a coalition game in relation to the expected marginal contribution of each player. This approach is particularly useful because it provides a solution that is unique and fair, where fair is defined by game theoretical concepts. A drawback of this method is that it requires the calculation of all sub-coalitions, making it computationally infeasible for coalitions with more than 10 players. In all the listed applications, the the number of potential players in a coalition goes in the thousands, sometimes even millions, requiring a more scalable solution. There exist around 5.000 banks in the United States[2], 8.000 in Europe[3], there are more than 800 telco operators

[1] Interest in a product is often measured by observing which persons click on the corresponding ad, or which persons visited the product page on the advertiser's website.

[2] https://fred.stlouisfed.org/series/USNUM

[3] https://www.ecb.europa.eu/pub/pdf/other/mfilist-200702en.pdf

world-wide[4], and the number of websites in a typical ad network run in the hundreds or thousands. The second method, called Explaining Document Classification method (EDC) [Martens and Provost, 2014], has been introduced for explaining the reasons behind instance-level document classifications. Specifically, what is the minimal set of evidence such that if it had not been present, the inference would not have been drawn? Chen et al. [2015] call this an *evidence counterfactual*.

The contributions of this paper are the following:

1. Defining the problem of value attribution in a predictive setting.
2. Introducing two methods to come to a solution.
3. Providing an evaluation framework for predictive data value sharing solutions.
4. Validating our solutions with a real-life application in targeted online advertising.
5. Showing how game theory and data-based explanation methods come together and providing a new scalable value attribution scheme for other game theoretical problems.

2 Related work

Determining the economic value of data is a challenging problem and has received relatively little attention in the predictive modeling literature [Dalessandro et al., 2014b]. A study by the OECD [2013] reviews several methods to determine the monetary value of personal data. These methodologies do not capture the economic benefits of personal data but rather the prices that markets are assigning to the data in different contexts. These are based on for example the market price for data, the cost of data breach or the willingness of customers to pay in order to protect their privacy. These values, however, do not reflect the monetary value of data for a specific application or campaign but rather give an idea about market prices of data in general. That is, a certain attribute could be very valuable in one setting but not have any significant impact when used in another.

In a non-monetary sense of the word, value can be interpreted in different ways that have been studied in multiple disciplines: First, feature selection can be seen as a way to value data [Forman, 2003]. Certain selection procedures consider the impact of features by removing them from the modeling. This can be either for making large problems computationally efficient or because well-chosen features can improve classification accuracy substantially or reduce the amount of training data needed to obtain a desired level

[4]http://www.gsma.com/membership/who-are-our-gsma-members/full-membership/

of performance. Either way, feature selection assesses the value of existing features. Secondly, many studies have considered whether adding more data points actually improves the model performance [Junqué de Fortuny et al., 2013]. One way to look at this is by learning curve analysis [Junqué de Fortuny et al., 2013, Perlich et al., 2003]. Another discipline where the value of data is considered is active information acquisition, which focuses on the selective purchase of features during training and model use, by maximizing performance gains and minimizing acquisition costs, which is based on active learning [Provost et al., 2007, Settles, 2010].

In most of the non-monetary studies, the value of data is evaluated in terms of evaluation metrics or the impact of the data on model performance. Little has been said about the monetary consequences of feature acquisition. Dalessandro et al. [2014b] present a methodology for estimating the economic value of adding data sources in predictive modeling applications, based on expected value. The effect that incremental data has on model performance is assessed in terms of common classification evaluation metrics, which is then translated into economic units.

3 Problem setting

The value attribution problem in a predictive setting van be defined as follows:

Given:

- A high-dimensional, sparse data set
- A predictive model for that data set
- A set of instances that generated a certain value through the predictive model

Provide:

- A value per feature that is fair
- That can be computed for up to millions of features

In this research, we focus on the case of online advertising where the goal of the advertiser is to predict a user's preferences and to serve the user ads that are most likely of interest to him. The features of the prediction model are the different data sources (websites) and the instances denote customers. So for each customer the feature vector shows the websites visited by that user. This data is characterized by a large dimensionality and sparseness. Next, a prediction model is built from these features, that yields a score that ranks cases (customers) by their likelihood of belonging to the class of interest (in this case the positive

class would be "likely interested in the product"). By determining a threshold, we obtain the predicted labels and thus the advertiser decides which users are served an advertisement. By tracking which users are actually taking the action, the real labels are obtained. A confusion matrix can then be built by comparing these predicted and true labels. This is a contingency table that distinguishes between the correct and false predictions and allows to deal with different sort of errors separately. For each (predicted, actual) pair in the confusion matrix, the cost or benefit of making such decision can be assigned as well. This information can be stored in a cost-benefit matrix. Based on these targeting decisions as well as the corresponding cost-benefit matrix, a certain value is obtained for the advertiser. The goal of this research is to divide this value among the data sources (websites) that are used in order to make the targeting decisions or predictions. Before going into more detail on the requirements of the solution, a couple of clarifications on the predictive model and the online advertising context.

We assume that the prediction model is given to us and that we do not have an influence on the model itself or the way the predictions are made. In this way, we focus only on the distribution of value among the features rather than the model building itself. Moreover it is important to notice that the model of consideration likely is the best model that the data science team has found for this specific problem and data set. The best model is defined as the model that has the most predictive power for this specific case study and data set.

Secondly, the prediction models built using such data are most often linear models [Raeder et al., 2012, Langford et al., 2007, Dalessandro et al., 2014a]. This makes the problem slightly easier, but is not a requirement. The differences for the attribution problem for linear versus non-linear models will be mentioned throughout the text.

Thirdly, for the online advertising setting, multiple revenue models are used and tested. In the first case, CPM ("Cost Per Mille") is used, which stands for the amount that the advertiser pays for every thousand impressions of the ad to publishers. Whether or not this actually leads to the customer clicking on the ad, is not taken into account. This revenue model is commonly used in online advertising. This implies that for our setting, the decision of showing the ad has a certain value for the publisher, rather than the action taken by the customer. More specifically, this implies that value is created when a positive prediction is made rather than when it becomes a true positive. The advertiser of course cares about the effect of the advertisement on conversions or brand awareness, so if the model performs badly (makes many incorrect predictions), even though this won't have a direct influence on the cost of that campaign, the advertiser will simply not renew the contract. Although the CPM revenue model is often used in online advertising, it seems logical to treat a false and correct prediction differently. Therefore, CPA is used in the second case. CPA stands for "Cost

Per Action" and implies that the publisher is paid for every correct positive prediction in a sense that the targeted customer will take the desired action (e.g. click or purchase). A correct positive prediction has a certain positive value whereas a false positive prediction entails a cost (for showing the ad). In both cases, we assume that a negative prediction (both correct or falsely classified as negative) does not have any value for the publisher. This is because when no advertisement is shown, no revenue is created. Similar setups can be thought off for the other applications that use predictive modeling using behavioral data.

3.1 Requirements for distribution scheme

The main challenge remains to divide the value created by this process, over the features that are responsible for it. Put differently we would like to know how much each of the features contributes to the fact that an instance is (correctly) classified as positive and thus value has been created by (effectively) showing that instance an impression. One way to think of this is to just look at the weights (β) of the features in the linear classification model and use this as a way to divide value over that instance's feature vector. This is definitely an interesting point of view as the βs of a linear prediction model present the predictive performance of the features. This however, might be insufficient as the model coefficients do not incorporate information on feature presence, which is key when analysing sparse data: a website that is very predictive but is visited by only a couple of users, should surely not be given the most value. This, together with other requirements which we posit are essential to a good value distribution, are listed below.

1. **Requirement 1: The distribution is fair**: Fairness is a very broad concept that can be defined in many different ways. In order to have a theoretically sound way of describing fairness, we base ourselves on the field of cooperative game theory where the focus is on dividing gains among players in the game. Here, fairness is defined as having some desirable properties such as symmetry, efficiency, additivity and the dummy player principle. We will discuss this concept further in Section 4.1.

2. **Requirement 2: The calculation is scalable**: The number of features in a predictive model that uses behavioral data is typically in the thousands up to even millions. In a first step the value for each feature is calculated. Afterwards, this needs be aggregated to a data source level: for example take the sum of the values of all accounts belonging to the same bank, or all the devices belonging to the same manufacturer, or all the sites belonging to the same publisher group. So the number of features is typically much higher than the number of data sources, making the dimensionality issue even stronger. The attribution solution needs to scale to such settings, with a computationally feasible solution that

is provided in a reasonable amount of time.

3. **Requirement 3: Both instance and aggregate level**: The methods should be able to both operate on an instance and aggregate level. The instance level is important since the reason why one user is identified as a positive can be different from the reason why a different user is identified as positive. This is because the data under consideration has a sparse, high-dimensional nature. The aggregate level on the other hand is interesting as well since this gives an insight in feature importance and shows how to divide the total value across all players. Therefore, the methods described are always calculated on an instance level, then summed and normalized over the whole data set in order to create scores on an aggregate level, which results in a score per feature. In this way, the coverage (m) which measures the number of times a feature appears in the data set, is also taken into account (next to the predictive power of a feature). This is also a valid requirement for a good value distribution as a website that has many visitors in the data set, should get more value as compared to a website that only has a few (given equal predictive power).

These requirements will be elaborated upon in the next sections.

4 Methods

4.1 Value distribution: Shapley value

As stated before, we use concepts from cooperative game theory to find a proper distribution among the different features. A cooperative game is one in which a set of N players engage in a game that results in some non-negative payoff v for the set. So one way to look at our problem is to approach it as a game where the features, websites in our case, are the players. A coalition can be defined as the process of bringing together different players to achieve goals that they cannot achieve on their own or at least less efficiently [A. Mas-Colell, 1995, Osborne and Rubinstein, 1994, Fatima et al., 2008]. To form a good coalition, the gains that arrive from forming the coalition should be split between the different players. Cooperative game theory provides a number of solutions and concepts that have some desirable properties such as stability, uniqueness and fair division. Different solution concepts are defined in the literature such as the Core, Kernel and Shapley value [Osborne and Rubinstein, 1994, Shapley, 1988, Davis and Maschler, 1965]. We choose to use the Shapley value in this case, because it represents a distribution scheme for cooperative games that satisfies several axioms related to the fairness and uniqueness of the distribution. Let us first define some core concepts before we present the definition:

1. N is the complete set of players, with cardinality $\|N\| = n$.

2. S is a subset of players, with $\|S\| = s$ and $S \subseteq N$

3. $v(S)$ is a value function that represents the total utility (money, points, etc.) the set S generates when playing the game

4. $\Delta_i v(S) = v(S \cup \{i\}) - v(S)$, is the marginal utility of adding player i to a set S.

The Shapley value (SV) is defined on an individual player i, and represents how much of the total value $v(N)$ it should be allocated upon realization of the game. Formally, the SV (ϕ_i) is the expected marginal utility $E[\Delta_i v(S)]$, of adding player i to a set S, where S is the first s players taken from each random permutation of N. Or, in other words, the Shapley value(ϕ_i) of the game $\langle N, v \rangle$ for player i is the average of its marginal contribution to all possible coalitions. This can be expressed as such [Shapley, 1988]:

$$\phi_i = E[\Delta_i v(S)] = \sum_{S \subset N - \{i\}} P(S)\, \Delta_i v(S) \tag{1}$$

$$= \sum_{S \subset N - \{i\}} \frac{s!(n-s-1)!}{n!} \Delta_i v(S) \tag{2}$$

The reason why we chose to use the Shapley value is because unlike the Core and Kernel, it provides a solution that is both unique and fair [Fatima et al., 2008]. By unique we mean that there is only one solution for a game and the players know exactly what they gain from it. Fair relates to the way gains are divided among the coalition members. In order to create a fair and unique distribution, four axioms need to be satisfied [Nagarajan and Soi, 2008]:

1. Symmetry principle: Two players are said to be symmetric with regard to a game v if they make the same marginal contribution to any coalition. Symmetric players should be assigned the same value. $v(S \cup i) = v(S \cup j) \rightarrow \phi_i(v) = \phi_j(v)$ for every S of N that does not contain i or j.

2. Dummy principle: If i is a dummy player, i.e. $v(S \cup i) - v(S) = 0$ for every $S \subset N$, then $\phi_i(v) = 0$. This implies that zero payoffs are assigned to players whose marginal contribution is zero for every coalition.

3. Efficiency principle: The total gain is distributed among the players $\sum_{i \in N} \phi_i(v) = v(N)$

4. Additivity principle: When two gain functions v and w are combined, the distributed gains should correspond to the gains derived from v and gains derived from w: $\phi_i(v+w) = \phi_i(v) + \phi_i(w)$ for any pair of cooperative games (N, v) and (N, w)

One extra axiom, which is a logical consequence from the previous four axioms, is added to the list as it is important to point out in our setting.

5. Reversed-dummy principle: A player that does not have a zero contribution, receives a non-zero value: $v(S \cup i) - v(S) \neq 0$ for every $S \subset N$, then $\phi_i(v) \neq 0$

The method for finding a player's Shapley value depends on the definition of the gain function (v), which is different depending on the type of game. In this case we will approach this problem as a weighted voting game. A weighted voting game is a type of game consisting of n players, where each player is defined by a weight w_i. The payoff for any weighted voting game played by a subset S is defined as [Banasri et al., 2010]:

$$v(S) = \begin{cases} 1 & \text{if } \sum_{i \in S} w_i > q, \quad \text{for some specified } q \\ \\ 0 & \text{otherwise} \end{cases}$$

In words, if we think of each player as a voter whose vote is worth w_i, a game is won if the total sum of weighted votes is greater than some threshold. From this definition, we can define $\Delta_i v(S)$.

$$\Delta_i v(S) = \begin{cases} 1 & \text{if } \sum_{j \in S, i \notin S} w_j + w_i \geq q > \sum_{j \in S, i \notin S} w_j \\ \\ 0 & \text{otherwise} \end{cases}$$

A player's marginal utility is 1 if that player's vote swung the total above the threshold (q). If the threshold was not met, or the value was already above the threshold, player i adds no marginal value.

As can be seen, Shapley has a lot of desirable properties for our setting. It gives a distribution scheme that is both very intuitive as well as theoretically sound. The main drawback of the method is the computational complexity. Every set of n players has 2^n possible subsets (including the null subset). So calculating the Shapley value for any set larger than 10 becomes a computational burden. In our setting, the set of data sources can be up to tens or even hundreds of thousands, thereby requiring methods that can approximate this real Shapley value. The realistic application of the Shapley value depends heavily on computationally feasible approximations. One such approximation is presented by Fatima et al. [2008]. In the next section, we present an intuitive derivation of the cited method (which is not done in the original paper), and we also provide a few modifications that should reduce the approximation error (without proof for now).

4.1.1 Approximation of the Shapley value for a Weighted Voting Game

We can rewrite the Shapley value as:

$$\phi_i = \sum_{k=1}^{n-1} \frac{1}{n} \sum_{s_k \in S_k} P(s_k)\,\Delta_i v(s_k) = \frac{1}{n} \sum_{k=1}^{n-1} E[\Delta_i v(S_k)] \tag{3}$$

Where $P(s_k)$ is the likelihood of the exact set s_k being drawn from all possible sets S of size k. The term $E[\Delta_i v(S_k)]$ can be though of as the expected marginal contribution of player i to any random subset of size k. We now focus on this term for the weighted voted game.

$$\begin{aligned} E[\Delta_i v(S_k)] &= \sum_{s_k \in S_k} P(S_k)\, I\Big(\sum_{j \in s_k} w_j + w_i \geq q > \sum_{j \in s_k} w_j\Big) && (4) \\ &= \frac{1}{Z_k} \sum_{s_k \in S_k} I\Big(\frac{q - w_i}{k} \leq \hat{w}_{S_k} < \frac{q}{k}\Big) && (5) \end{aligned}$$

The second equality can be obtained algebraically and using the fact that all subsets of k are equally likely (i.e., $P(S_k) = 1/Z_k$, where Z_k is the sum of subsets of size k). The term $\hat{w}_{S_k}$ is the mean of w_j for those players in S_k: $\sum_{j \in s_k} w_j / k$. We can then rewrite this last term as:

$$E[\Delta_i v(S_k)] = P\Big(\frac{q - w_i}{k} \leq \hat{w}_{S_k} < \frac{q}{k}\Big) \tag{6}$$

One way of thinking about this last representation, is what is the probability that a random subset of size k will have an average in the interval $[\frac{q-w_i}{k}, \frac{q}{k}]$. If we let $D(k)$ be the probability density function of $\hat{w}_{S_k}$, the value we are looking for is simply the integral of $D(k)$ in the specified interval. At this, point, we have defined an exact computation of the Shapley value, if $D(k)$ is computed as the exact empirical distribution of $\hat{w}_{S_k}$. We approximate $D(k)$ instead with a normal distribution, i.e.:

$$D(k) \approx N(\mu_{/i}, SE^k_{/i}) \tag{7}$$

Where $\mu_{/i} = E[\hat{w}_{S_k}] = \frac{1}{n-1} \sum_{j \neq i} w_j$, and $SE^k_{/i} = SE(\hat{w}_{S_k}) = c * \sigma_{/i}$ is the standard error of $\hat{w}_{S_k}$, c is some constant and $\sigma_{/i}$ is the sample standard deviation of all $w_j\, s.t. j \neq i$. So putting these together, we can now approximate $E[\Delta_i v(S_k)]$ using the normal distribution $N(\mu_{/i}, SE^k_{/i})$ for each k. This replaces the combinatorial complexity of the problem $N-1$ separate estimations of $E[\Delta_i v(S_k)]$. The accuracy of the method ultimately depends on how well $N(\mu_{/i}, SE^k_{/i})$ approximates the true empirical distribution of $\hat{w}_{S_k}$.

Adjustments to the approximation method

In this section we make a few adjustments to the algorithm used in Fatima et al. [2008]. Specifically, in their paper, they use the distribution $D(k) \approx N(\mu, \frac{\sigma}{\sqrt{k}})$. We make the following adjustments to improve the approximation:

1. We replace μ with $\mu_{/i}$, which is the mean of the N players excluding player i
2. σ is replaced with $\sigma_{/i} = \sqrt{\frac{1}{N-1}\sum_{j \neq i}(w_j - \mu_{/i})^2}$, the *population* standard deviation of w_j for the set of players excluding player i.
3. We adapt the standard error of $\hat{w}_{S_k}$ so that it accounts for the fact that S_k is being drawn from a finite sample. We apply a standard correction factor for finite samples, which is

$$SE^k_{/i} = \frac{\sigma_{/i}}{\sqrt{k}} \sqrt{\frac{n-k}{n-1}} \tag{8}$$

The importance of the above adjustments is most notable in small n situations. For instance, the influence of any particular player's weight w_i has magnitude n^{-1}. Clearly, with lower n, the bias from leaving in a player from the computation of the distribution statistics will be greater. The adjustment to the standard error is even more significant. In particular, as k increases, the adjustment factor approaches 0. Without the adjustment factor to the standard error, the normal distribution used to approximate $E[\hat{w}_{S_k}]$ is wider than it should be. This would lead to an overestimate of the total contribution of the player at that level of k. Putting this all together the method can be summarized as shown in Algorithm 1.

4.1.2 Calculation

The Shapley value is calculated per instance and thus per binary website vector (i.e. for all websites that appear in the user's browser history). Afterwards the scores are summed over all instances and normalized (by the total sum). In this way, the coverage (m) per feature is taken into account. The fact that both β and coverage should be taken into account is already briefly discussed in 3.1. Let's further argue why this is important for the value distribution problem. First, note that a feature with a larger β does not necessarily gets a higher value as compared to a feature with a lower β value on *a feature level*. This is because the value depends on the coalitions (i.e. feature vectors) in which the feature appears on *an instance level*. As an (extreme) example, let's say we have two features x_1 and x_2 where $\beta_{x_1} > \beta_{x_2}$ and $m_{x_1} = m_{x_2}$ but the feature with β_{x_1} always appears in feature vectors with other βs that are all larger than β_{x_1} and β_{x_2} for example is always part of feature vectors with all βs lower than β_{x_2}, then $\phi_{x_2} > \phi_{x_1}$. If in this case $m_{x_1} \neq m_{x_2}$, we can no

longer say anything about the value on a feature level with certainty. Clearly, there is a relation between the value that a feature receives on *an instance level*, which is based on both its own β and the βs of the other features in the coalition (i.e. feature vector), and the value on *a feature (aggregate) level*, which depends on both the values for the feature on an instance level and the coverage. This example shows more clearly why both metrics should be taken into account: even if a feature has a high β, this does not necessarily say anything about its contribution to the different coalitions that he is part of. The same principle applies to the coverage where a feature with a high coverage does not necessarily receive a higher value on a feature level as this depends on the values that this website received at every instance it is a part of.

For every instance that was correctly classified as positive, a value of 1 is divided over the features[5]. Depending on the revenue scheme that is applied, in case of a falsely classified positive: (a) a negative value (or cost) of -0.1 is divided over the features when CPA is used or (b) a positive value (or revenue) of +1 is divided over the features when CPM is used. The latter implies that there is no distinction between a true or false positive prediction and thus any positive prediction implies a revenue. In case of CPA on the other hand, a negative value is thus divided over the features. One could interpret this as follows: when a false prediction was made, there is a cost for the publisher. Just as with revenue (in case of a correct prediction), the cost is also divided over the features that were responsible for that prediction. By finally summing the scores over all instances, it is clear that a feature that is often involved in false predictions, will have a lot of negative values as well, thereby lowering its total value.

While there is a clear similarity between linear classification and a weighted voting game, there is one setback. The classification weights are not strictly non-negative, which is required for weighted voting games [Banasri et al., 2010]. A negative prediction, either false or correct, does not have a revenue and thus positive value in our setting. This implies that it is reasonable to only focus on the positive predictions. This in turn implies that only features with a positive β have value because a feature with negative weight can never be responsible for a positive prediction. In other words, a negative feature should never be rewarded for its contribution to a positive prediction (both correctly or incorrectly). Yet, it might be that thanks to the negative features, some instances are correctly pushed under the threshold and thus are correctly classified as negatives but again, this has no value in our setting. Following this reasoning, the value is only divided among the positive features for every instance. The features with a negative β receive a zero value.

[5] A more correct value would be CPM/1000, but this is just a scaling factor that can be added.

4.2 EDC method

The second method stems from the domain of document classification. This method tackles the problem of explaining classifier classifications when the data is characterized by very high dimensionality [Martens and Provost, 2014]. The most common approach to understanding the classifications made by predictive models is to use "global explanations" by looking at the variables with the highest weight in the case of linear models, or by using rule extraction techniques for example. Global explanations however, are not satisfactory in this setting because of the massive dimensionality on the one hand (i.e. classification trees would not be readable for example) but also because one is interested in the reasons why a specific document is classified as positive (eg containing objectionable content) rather than having a global explanation of how the model works in general. Therefore, their paper focuses on instance based explanations. The explanations can be defined as a minimal set of features, such that removing all the features (words in their case) within this set from the document changes the predicted class from the class of interest. Only when all the words in the explanation are removed does the class change (the set is minimal). The definition used in the paper is as follows [Martens and Provost, 2014]:

DEFINITION 1. *Consider a document D consisting of m_D unique words W_D from the vocabulary of m words: $W_D = w_i, i = 1,2,...,m_D$, which is classified by classifier C_M: $D \rightarrow 1,2,...,k$ as class c. An explanation for document D's classification is a set E of words such that removing all words in E from the document leads C_M to produce a different classification. Further, an explanation E is minimal in the sense that removing any subset of E does not yield a change in class. Specifically:*
E is an explanation for $C_M(D) \iff$
1. $E \subseteq W_D$ (the words are in the document),
2. $C_M(D \backslash E) \neq c$ (the class changes), and
3. $\nexists E' \subset E : C_M(D \backslash E') \neq c$ (E is minimal).
$D \backslash E$ denotes the result of removing the words in E from document D.

Although the focus in their paper is clearly on providing instance-based explanations, one can also see that this method could provide an alternative way of defining which specific features were responsible for a certain classification. Instead of using these "explanations" (which we will call "combinations" or "subsets" in this work) in an explanatory way, one can interpret them as the features that caused the prediction to be made. We then could apply this method to our problem, where the goal is to divide value according to feature importance. An example could be; *"Client A is targeted (i.e. shown the advertisement) because he visited web page x, y and z"* (because if client A would *not* have visited websites x, y and z, then the client would *not* be classified as interested and would not have been targeted with an ad). So, one can argue that

web pages x, y and z should get value as they are responsible for that classification. That is, the *combination* of those specific web page visits has led to the model classifying that user as "potentially interested in the product". When the client would not have visited one of these pages in the smallest subset, he would not have been targeted. So that specific set of web pages should get value because this information has led to an impression.

4.2.1 Implementation

When a linear model is being used (as was assumed earlier), one could argue simply to list the top k websites with the highest positive weights that appear in the browser history of a certain user as an explanation for the class. k would then be the minimal number of top websites such that removing these k websites leads to a class change. EDC produces minimum-size combinations for linear models by ranking all websites visited by the customer according to the product $\beta_j x_{ij}$ where β_j is the linear model coefficient and x_{ij} a binary vector that denotes whether or not a website was visited by user i. The combinations with the top-ranked websites is a combination of smallest size (the proof can be found in [Martens and Provost, 2014]). This simple version of the algorithm is described in Algorithm 2.

However, it could be interesting to find alternative combinations next to the minimum-size subset. A straightforward approach would be to conduct a complete search through the space of all website combinations, starting with one website, and increasing the number of websites until a subset is found. The algorithm starts by checking whether removing one website from the customer's browsing history, would cause a change in class label. If so, an irreducible subset is found (in the linear case). If the class does not change based on only one website, the algorithm considers all combinations of size 2, 3 and so on. Not that for a browsing history of m_A websites, a combination of u websites requires $m_A!/(m_A - u)!$ evaluations. This complete search scales exponentially with the number of websites in the browsing history. For data sets with a high dimensionality, this is impracticable if one wants to find multiple combinations.

Note that if the goal is to find "a" subset, Algorithm 2 ("Simple EDC method") can be used and the search will be very fast. When one wants to find all subsets, this complete search is needed (let's call this the "Full EDC" method). However, since this will be too time consuming seen the dimensionality of browsing histories, a heuristic approach will be used that is proposed in Martens and Provost [2014]. The heuristic approach is designed to find one or more irreducible subsets of websites but is not guaranteed to find all irreducible subsets. Note that "irreducible" implies a subset so that only when *all* those features in that subset are removed, the class label changes. The algorithm is based on two important steps [Martens and Provost, 2014]:

1. **Heuristic search guided by local improvement**: In a first step, the algorithm starts by listing all possible subsets of one feature (website) and calculating the score and class change for each. When a combination is found in this way, it is added to the list of "irreducible subsets". Next, the algorithm proceeds as a straightforward heuristic best-first search. Specifically, each time the algorithm will expand the partial subsets for which the output score changes the most in the direction of the class change. This entails creating a set of new, candidate subsets, comprising all combinations with one additional feature (website) from the browing history (that is not yet included in the partial combination).

2. **Search-space pruning**: For combinations of size l, we do not need to check combinations of size $l+1$ with these same websites. So this part of the search area does not have to be further searched.

The algorithm describing this heuristic search, or what we will call the "Extended EDC" method, is formulated in Algorithm 3. Note that with the current, heuristic implementation this method is also an approximation (of the Full EDC method) since it does not perform an entire search through the subset space.

4.2.2 Calculations

The combinations are calculated per instance and afterwards summed and normalized (by the total sum) over all instances. At the instance level, a value of 1 is divided equally among all the features per subset when CPM is applied. This is because each of the features is believed to contribute equally as the label only changes when leaving *all* these features out. When multiple subsets are found (in case of the Extended EDC method), the score of 1 is divided among each subset and the scores per feature are summed and normalized by the total sum (i.e. the total number of subsets for that instance). Afterwards, in order to create a score per feature, the instance-based scores per feature are summed over all instances and normalized (by the total sum). By doing so, the coverage is taken into account as a feature that appears in many browsing histories has more chance to be included in the subsets. Again, this is important for the value distribution as already discussed in 4.1.2. In case CPA is used, a value of 1 is divided in case of a correct prediction and a negative value (or cost) of -0.1 is divided among the features of the falsely classified positive instances in the same way as described.

5 Experimental setup

By applying the proposed methods on an online advertising data set, we aim to show the characteristics and scoring strategies of the different techniques. Moreover, we benchmark the methods with the case where

value would be divided on a feature level based on the βs of the model or based on the coverage of the features. The advertising example that we are using is one from a high quality shopping mall. The data set consists of several 100.000s of binary features which represent websites visited by each customer. As explained before, the model is given and linear. Moreover, we assume that this is the best model that we have found for this specific data set. In this case, a logistic regression model is used.

The question of interest in this research is how to distribute the value of the predictions made by the model, among all the features that are responsible for that prediction. In this experiment, the linear model scored all instances. Next, we define the top segment as the highest 5 percentile of the ranked data set. This segment will be served an impression and thus will be targeted. Given the assumption that the model used is the best model we could find for this setting, it is reasonable to state that the value should be divided over the features that are responsible for these positive predictions made by the model. The label in this data set shows whether the user visited the homepage of the shopping mall. Visiting the homepage is thus what we define as the action taken by the user.

As explained before in Section 3, we use two different revenue models. The first one is CPM, implying that the ad serving company is paid, and thus value is created, per impression served (rather than per homepage visit by the customer). This implies that the instances that are ranked on top by the model (i.e. the first 5 percentile), create value for the company. Therefore, this value should be distributed among the features that are responsible for those instances to be ranked in the top. For every impression, we assume a revenue of 1 is created. Therefore, both the Extended EDC method and Approximate Shapley method will be calculated for every instance in that segment, thereby each time dividing this revenue over the features in that feature vector. In case of CPA on the other hand, a revenue of 1 is only created and divided when a correct prediction was made (meaning that the user is shown an impression and visited the home page of the product). A false prediction yields a cost. That is, these features can be seen as responsible for a false prediction, and thus are assigned a cost for that instance. In these experiments we use a cost of 0.1 and thus a value of -0.1 is divided over the features. Afterwards, for both CPM and CPA, these scores are summed and normalized (by the total sum) to create a score per feature.

6 Evaluation and results

6.1 Theoretical evaluation

The methods described above assign value to the features that are responsible for a (correct) positive classification. They do so by dividing value on an instance level first and aggregate these values to the feature

level. Another possibility to calculate the feature score would be to divide the value directly on a feature level, proportionally to the β or coverage of the features.

First, these different methods or approaches are compared in a more theoretical way as shown in Table 1. In order to do so, we refer once more to the requirements listed in Section 3.1. These were defined in order to create a fair and feasible value distribution among the data sources. The extra requirement of "computationally feasible" is also added in the table to ensure that the calculations can be done practically (e.g. on an available computer). As can be seen from the table, neither the normal Shapley method nor the Full EDC comply with this extra requirement and therefore cannot be calculated, which is why the approximation methods were described. Yet, we have included these methods in the table to show that they would comply with all requirements if they were computationally feasible.

The first requirement states that the distribution needs to be fair, where fair is defined by different concepts. As can be seen from the table, all methods comply with the efficiency, dummy and linearity principle. In terms of symmetry it can be noticed from Algorithm 1 that the Approximate Shapley method will assign the same value to features with the same contribution (i.e. β on an instance level). The Simple EDC and Extended EDC method on the other hand, cannot guarantee symmetry on an instance level. That is, depending on the implementation, only one out of two features with the same β will be included in the explanation. Because of the heuristic approach, not all feasible explanations are found and so it is possible that the algorithm has the tendency to always choose one over the other.

For the same reason, Simple EDC and Extended EDC cannot ensure that a player who has made a contribution, will receive a value and therefore the reverse-dummy principle is not met. For the Approximate Shapley method, we use an approximation of the real probability density function of $\hat{w}_{S_k}$. Therefore we cannot be sure that this reverse-dummy principle holds when approximating the real Shapley method. When we would not use approximate methods but rather use the full Shapley and EDC methods, the reverse-dummy requirement would be guaranteed. Lastly, dividing the value proportional to the βs or coverage (as show in the last two columns) is only done on a feature level and so we cannot assess the fairness on an instance level.

The second requirement states that on an aggregate level the compensation for each website is proportional to its contribution, implying that the predictive value (β) is accounted for. For the approximation methods however it cannot be guaranteed that on an instance level every feature that made a contribution, actually receives value (see reverse-dummy principle). Therefore, on an aggregate level some features might not be entirely compensated proportionally to their contribution.

Requirement 3 ensures that the methods are calculated on both an instance and feature level. This is the

Algorithm 1 Approximate Shapley Value

Input: $W =< w_1, \ldots, w_n >, q$

Precompute: $T = \sum_j w_j$, $T^2 \leftarrow \sum_j w_j^2$

for $i = 1$ to n **do**

$\mu_{/i} = \frac{T - w_i}{n-1}$

$\sigma_{/i} = \sqrt{\frac{1}{n-1}[T^2 - w_i^2 - \mu_{/i}(T - w_i) + (n-1)\mu_{/i}^2]}$

set: $\phi_i = 0$

for $k = 1$ to $n-1$ **do**

$SE^{k}{}_{/i} = \frac{\sigma_{/i}}{\sqrt{k}} \sqrt{\frac{n-k}{n-1}}$

$\delta_{ik} = \int_a^b N(\mu_{/i}, SE^k_{/i})$, $a = \frac{q - w_i}{k}$, $b = \frac{q}{k}$.

set: $\phi_i = \phi_i + \delta_{ik}/n$

end for

end for

Algorithm 2 EDC (linear case, simple implementation) for feature importance in online advertising.

Inputs:

$W_A = \{w_i, i = 1, 2, \ldots, m_A\}$ % Customer A to classify, with m_A website visits

$C_M : \mathscr{A} \rightarrow \{\infty, \in, \ldots, \|\}$ % Trained classifier C_M with scoring function f_{C_M}

Sort coefficients w_A in descending order as $1...m_A$

Output:

R = Smallest subset of features

$c = C_M(A)$ % The class predicted by the trained classifier

$p = f_{C_M}(A)$ % Corresponding probability or score

$R = \{\}$ % The smallest feature combination found

$i = 1$

$c_{new} = c$

while $c_{new} = c$ **do**

$W_A = (W_A \backslash w_i)$

$c_{new} = C_M(W_A)$ % The class predicted by the trained classifier if website w_i did not appear in the browsing history of customer A

$p_{new} = f_{C_M}(W_A)$ % The probability or score predicted by the trained classifier if website w_i did not appear in the browsing history of customer A

$R = R \cup w_i$

$i = i + 1$

end while

Algorithm 3 Extended EDC (via Best-first Search with Pruning) for feature importance in online advertising

Inputs:

$W_A = \{w_i, i = 1, 2, \ldots, m_A\}$ % Customer A to classify, with m_A website visits

$C_M : \mathscr{A} \to \{\infty, \in, \ldots, \|\}$ % Trained classifier C_M with scoring function f_{C_M}

$max\_iteration = 30$ % Maximum number of iterations

Output:

R = list of combinations

$c = C_M(A)$ % The class predicted by the trained classifier

$p = f_{C_M}(A)$ % Corresponding probability or score

$R = \{\}$ % The list of feature combinations that is gradually constructed

$sets\_to\_expand\_on = \{\}$

$P\_sets\_to\_expand\_on = \{\}$

for $i = 1 \to m_A$ **do**

  $c_{new} = C_M(W_A \setminus w_i)$ % The class predicted by the trained classifier if website w_i did not appear in the browsing history of customer A

  $p_{new} = f_{C_M}(D \setminus w_i)$ % The probability or score predicted by the trained classifier if website w_i did not appear in the browsing history of customer A

  if $c_{new} \neq c$ **then**

    $R = R \cup w_i$

  else

    $sets\_to\_expand\_on = sets\_to\_expand\_on \cup w_i$

    $P\_sets\_to\_expand\_on$ = $P\_sets\_to\_expand\_on \cup p_{new}$

  end if

end for

for $iteration = 1 \to max\_iteration$ **do**

  combo = website set in $sets\_to\_expand\_on$ for which $(p - p\_sets\_to\_expand\_on)$ is maximal % The best first

  combo_set = create all expansions of *combo* with one website

  *combo_set*2 = remove sets containing already found combinations of R from *combo_set* % The pruning step

  for all combos C_o in *combo_set*2 **do**

    $c_{new} = C_M(W_A \setminus C_o)$ % The class predicted by the trained classifier if the websites in C_o did not appear in the browsing history of customer A

    $p_{new} = f_{C_M}(W_A \setminus C_o)$ % The probability or score predicted by the trained classifier if the websites in C_o did not appear in the browsing history of customer A

    if $c_{new} \neq c$ **then**

      $R = R \cup C_o$

    else

      $sets\_to\_expand\_on = sets\_to\_expand\_on \cup C_o$

      $P\_sets\_to\_expand\_on$ = $P\_sets\_to\_expand\_on \cup p_{new}$

    end if

  end for

end for

Table 1: Comparison of the different methods according to the requirements

	INSTANCE TO FEATURE LEVEL				FEATURE LEVEL	
	Shapley	**Approx. Shapley**	**Full EDC**	**Simple / Extended EDC**	**Coverage**	β
1. "Fairness" (instance level)						
Efficiency	✓	✓	✓	✓	-	-
Symmetry	✓	✓	✓	X	-	-
Dummy	✓	✓	✓	✓	-	-
Additivity	✓	✓	✓	✓	-	-
Reverse-dummy	✓	X	✓	X	-	-
2. Compensation proportional to contribution (aggregate level)	✓	X	✓	X	X	✓
3. Both instance and aggregated level	✓	✓	✓	✓	X	X
Computationally feasible	X	✓	X	✓	✓	✓

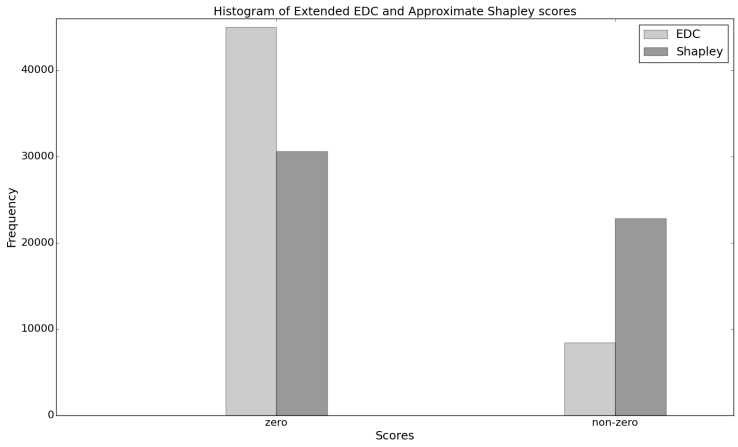


Figure 1: Histograms of both Approximate Shapley values and Extended EDC scores

case for the Shapley and EDC methods. When dividing the value based on β or coverage, this is only done on an aggregated level.

6.2 Empirical evaluation

6.2.1 Correlations and ranking

In order to empirically evaluate the distribution schemes, we will first look at the rank correlation and the distribution strategies. First, the comparison is made between the different distribution methods and secondly the two revenue schemes (CPM and CPA) are compared to one another.

DISTRIBUTION METHODS

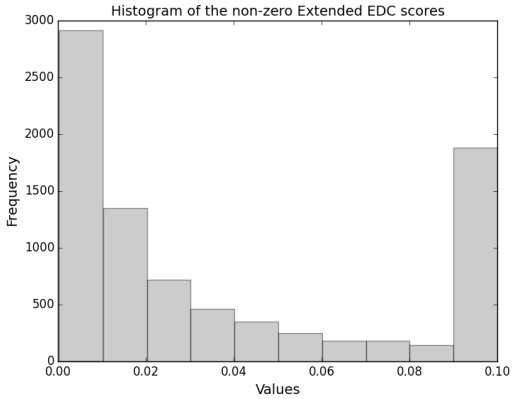
Table 2: Spearman's rank correlation coefficients between the different methods. The features are ranked according to the method as listed in the first column and the top 1000 of this ranking is considered. Notation in bold font implies a positive rank correlation of more than 0,80, normal script denotes a correlation between 0,50 and 0,80 whereas correlations lower than 0,50 are shown in italics.

	Approx Shap	Extended EDC	β	Coverage
Approx Shap	\	**0,97**	0,60	0,55
Extended EDC	**0,97**	\	0,62	*0,50*
β	0,74	0,75	\	*0,39*
Coverage	*0,16*	*0,16*	*0,05*	\

The histogram in Figure 1 gives an overview on the scoring strategy of both the Extended EDC and

Table 3: Scores for the top 20 highest ranked features according to the Extended EDC method using CPM. The EDC, extended EDC and Approximate Shapley scores (all with CPM) as well as the coverage were normalized (dividing by the sum) and represent a percentage. The βs in this table are the original weights.

Feature	Feature name	β	Coverage	Simple EDC	Shapley	Extended EDC
140677	www.ebay.com	0,497	0,680	8,759	4,968	7,256
30091	www.katespade.com	0,989	0,085	14,117	5,305	6,342
22487	www.bcbg.com	1,647	0,063	17,300	8,094	5,927
90667	www.stuartweitzman.com	1,264	0,041	8,816	4,161	4,210
54304	www.restorationhardware.com	0,702	0,060	6,933	2,473	3,950
22625	www.gilt.com	0,603	0,047	4,781	1,994	3,268
61142	www.huffingtonpost.com	0,252	0,598	0,292	2,113	2,126
37089	www.forever21.com	0,674	0,051	3,252	1,398	2,116
41888	www.colehaan.com	0,529	0,043	2,556	1,370	2,092
26399	www.talbots.com	0,828	0,035	4,158	1,440	1,890
28359	www.brooksbrothers.com	0,659	0,034	2,983	1,284	1,807
28847	www.wayfair.com	0,298	0,213	0,602	1,586	1,718
99148	www.harpersbazaar.com	0,458	0,059	1,884	1,218	1,589
108034	www.kayak.com	0,299	0,187	0,531	1,255	1,575
69916	www.elle.com	0,426	0,078	1,485	1,285	1,513
99243	www.menupages.com	0,451	0,055	1,549	1,040	1,376
48200	www.jetblue.com	0,332	0,088	0,747	0,973	1,335
87736	www.dior.com	0,705	0,016	1,915	0,832	1,281
35242	www.evite.com	0,367	0,118	0,678	0,904	1,275
73054	www.surlatable.com	0,561	0,025	1,795	0,717	1,165



(a) Histograms of the non-zero Extended EDC scores. The last bin includes all features with a score larger than 0,10.

(b) Histograms of the non-zero Shapley scores. The last bin includes all features with a score larger than 0,01.

Figure 2: Histograms for both non-zero scores of Shapley and Extended EDC

Table 4: Scores for the top 20 highest ranked features according to the Shapley method using CPM. The EDC, Extended EDC and Approximate Shapley scores (all with CPM) as well as the coverage were normalized (dividing by the sum) and represent a percentage. The βs in this table are the original weights.

Feature	Feature name	β	Coverage	Simple EDC	Shapley	Extended EDC
22487	www.bcbg.com	1,647	0,063	17,300	8,094	5,927
30091	www.katespade.com	0,989	0,085	14,117	5,305	6,342
140677	www.ebay.com	0,497	0,680	8,759	4,968	7,256
90667	www.stuartweitzman.com	1,264	0,041	8,816	4,161	4,210
54304	www.restorationhardware.com	0,702	0,060	6,933	2,473	3,950
61142	www.huffingtonpost.com	0,252	0,598	0,292	2,113	2,126
22625	www.gilt.com	0,603	0,047	4,781	1,994	3,268
28847	www.wayfair.com	0,298	0,213	0,602	1,586	1,718
26399	www.talbots.com	0,828	0,035	4,158	1,440	1,890
37089	www.forever21.com	0,674	0,051	3,252	1,398	2,116
41888	www.colehaan.com	0,529	0,043	2,556	1,370	2,092
69916	www.elle.com	0,426	0,078	1,485	1,285	1,513
28359	www.brooksbrothers.com	0,659	0,034	2,983	1,284	1,807
108034	www.kayak.com	0,299	0,187	0,531	1,255	1,575
99148	www.harpersbazaar.com	0,458	0,059	1,884	1,218	1,589
99243	www.menupages.com	0,451	0,055	1,549	1,040	1,376
48200	www.jetblue.com	0,332	0,088	0,747	0,973	1,335
36668	www.weddingwire.com	0,495	0,055	1,305	0,942	1,073
35242	www.evite.com	0,367	0,118	0,678	0,904	1,275
35907	www.allrecipes.com	0,203	0,281	0,022	0,890	0,728

Shapley method. It can be seen that the Extended EDC method gives the majority of the features a zero value. That is because only the features that appear in the irreducible subsets will receive a value implying that only the most informative features, according to this method, will receive a score. With the Approximate Shapley method on the other hand, a feature receives less often a zero value, except for those features who never contribute according to the definition of a weighted voting game (i.e. dummy principle). Figure 2 shows more detailed histograms for both methods. Note that these graphs only plot the non-zero values and that the last bin includes all values larger than the maximum value on the x-axis. The Approximate Shapley method typically divides the value over more features on an instance level and consequently, the values on a feature level are mostly smaller per feature as compared to the Extended EDC method.

Table 2 shows the Spearman's rank correlation between the values of the top 1000 features according to the different methods: Extended EDC, Approximate Shapley values, βs and coverage. The Spearman coefficient indicates how well the relation between two variables can be described as monotonic. A perfect Spearman's value of 1 or -1 implies a perfect monotonic positive or negative relationship. The first observation that can be made from these results is that there exists a very high rank correlation between the Extended EDC and Approximate Shapley scores. Moreover, both the Extended EDC and Shapley scores are positively related to the β of the features and to a lesser extent to the coverage as well. This latter observation can be explained by the fact that both methods were implemented on an instance level before being summed over all instances. Lastly, the rank correlations between the features scored proportionally to their β and coverage are only slightly positive.

Tables 3 and 4 show the β values, coverage, Simple EDC and Extended EDC scores as well as the Approximate Shapley values for the top 20 highest ranked features according to the Extended EDC and the Approximate Shapely method respectively. In the current implementation, the values per feature are normalized by dividing by the sum. Therefore, these values could be interpreted as percentages and so a specified amount could be divided over these features in terms of percentage. Some websites, like for example "www.stuartweitzman.com" have a large weight, and thus a high informative value, but a lower coverage, meaning that it is less visited. Still, both Extended EDC and Approximate Shapley show a high value for this feature. The website of "Wayfair.com" also receives a high score from both methods but is, on the contrary, often visited (high coverage) and has a lower β value. Although Table 2 already showed that both methods have a similar way of scoring the features, one can see that there are some differences in the ranking of the features. Nevertheless, most of the features that appear in the top 20 of the Extended EDC method, will also appear in the top 20 of the Approximate Shapley method, yet with a slightly different ranking. As an example take the website of "Surlatable.com" that appears in the top 20 of Table 3, but will

Table 5: Spearman's rank correlation coefficients between the different cost schemes. Notation in bold font implies a positive rank correlation of more than 0,80, normal script denotes a correlation between 0,50 and 0,80 whereas correlations lower than 0,50 are shown in italics.

	Approx Shap (CPA)	Extended EDC (CPA)
Approx Shap (CPM)	**0,99**	**0,97**
Extended EDC (CPM)	**0,97**	**0,99**
β	0,62	0,63
Coverage	0,53	*0,48*

only appear on the 36th place with the Shapley method.

REVENUE SCHEMES

When comparing the cost scheme CPM with CPA, where a cost of 0.1 is divided over the features in case of a false positive prediction, several observations can be made.

First, let's have a look at the Spearman's rank correlations between the top 1000 features according to the Extended EDC and Approximate Shapley method when using a cost of 0.1 (CPA), and the scores when CPM is applied. This is shown in Table 5. These results show that, for both Extended EDC and Approximate Shapley, there is almost a perfect ranking correlation between the CPA and CPM case. Also when comparing the two methods, the correlation is still 0.97, which demonstrates only a marginal difference in ranking. Again, a positive correlation can be detected with both β and coverage values. These findings already demonstrate that for this specific data set and case study, the cost scheme does not change the results substantially. Of course, this can only be concluded for this specific case study as for some data sets false predictions might have a larger influence.

Next, the same tables that show the top 20 highest ranked features according to the Extended EDC and Approximate Shapley method are constructed in the case of the cost of 0.1 (CPA) and are shown in Table 6 and Table 7. When comparing the rankings in case of CPM with those in case of a cost of 0.1, one can see that there is little difference, especially for the Shapley method (see Table 4 and 7). Also the scores of the different methods are very similar, regardless of the cost structure that was used. For example, when the Shapley method would be used, "bcbg.com" gets $8,094\%$ of the revenue in case of CPM and $8,199\%$ in case a cost of 0.1 is charged for false positive predictions. Although the top 20 rankings of features are similar for the Shapley and Extended EDC method, it can be seen that there might be some differences in scores still. For example, "ebay.com" would receive $4,968\%$ of the revenue in case Shapley (with CPM)

Table 6: Scores for the top 20 highest ranked features according to the extended EDC method using a cost of 0.1. The EDC, extended EDC and Approximate Shapley (all calculated with a cost of 0.1) scores as well as the coverage were normalized (dividing by the sum) and represent a percentage. The βs in this table are the original weights.

Feature	Feature name	β	Coverage	Simple EDC	Shapley	Extended EDC
140677	www.ebay.com	0,497	0,680	8,926	4,900	7,291
30091	www.katespade.com	0,989	0,085	13,092	5,178	6,258
22487	www.bcbg.com	1,647	0,063	16,876	8,199	6,107
90667	www.stuartweitzman.com	1,264	0,041	9,098	4,407	4,409
54304	www.restorationhardware.com	0,702	0,060	6,692	2,459	3,954
22625	www.gilt.com	0,603	0,047	4,893	2,033	3,395
41888	www.colehaan.com	0,529	0,043	2,649	1,365	2,127
37089	www.forever21.com	0,674	0,051	3,150	1,338	2,039
61142	www.huffingtonpost.com	0,252	0,598	0,319	2,066	1,999
28359	www.brooksbrothers.com	0,659	0,034	2,988	1,314	1,911
26399	www.talbots.com	0,828	0,035	3,905	1,395	1,869
28847	www.wayfair.com	0,298	0,213	0,653	1,547	1,694
99148	www.harpersbazaar.com	0,458	0,059	1,947	1,221	1,638
69916	www.elle.com	0,426	0,078	1,568	1,280	1,530
108034	www.kayak.com	0,299	0,187	0,571	1,209	1,476
99243	www.menupages.com	0,451	0,055	1,595	1,024	1,390
87736	www.dior.com	0,705	0,016	1,972	0,894	1,382
48200	www.jetblue.com	0,332	0,088	0,815	0,958	1,330
15691	us.christianlouboutin.com	0,755	0,017	1,926	0,848	1,236
35242	www.evite.com	0,367	0,118	0,711	0,866	1,218

is used but would receive $7,256\%$ when the Extended EDC method is used. This might still imply quite a difference when a large revenue is divided over the features.

Overall, we can conclude that both the EDC and Shapley scores are positively correlated with β and to a lesser extent to the coverage of the features. Moreover, the Extended EDC as well as the Approximate Shapley method divide the value in a similar way (which results in a notable similar feature ranking) except for the fact that the Extended EDC methods will, in the current implementation, only assign a value to the most informative features whereas Shapley will distribute the value over more features, also the less informative ones. Of course, for the Extended EDC method this entirely depends on the implementation. When more iterations are allowed, less features will receive a zero-value. It is an interesting and surprising result to discover that although these methods have a very different approach, background and application, they end up showing very similar scoring and ranking results. The main differences between the two mehods

Table 7: Scores for the top 20 highest ranked features according to the Shapley method using a cost of 0.1. The EDC, Extended EDC and Approximate Shapley scores (all calculated with a cost of 0.1) as well as the coverage were normalized (dividing by the sum) and represent a percentage. The βs in this table are the original weights.

Feature	Feature name	β	Coverage	Simple EDC	Shapley	Extended EDC
22487	www.bcbg.com	1,647	0,063	16,876	8,199	6,107
30091	www.katespade.com	0,989	0,085	13,092	5,178	6,258
140677	www.ebay.com	0,497	0,680	8,926	4,900	7,291
90667	www.stuartweitzman.com	1,264	0,041	9,098	4,407	4,409
54304	www.restorationhardware.com	0,702	0,060	6,692	2,459	3,954
61142	www.huffingtonpost.com	0,252	0,598	0,319	2,066	1,999
22625	www.gilt.com	0,603	0,047	4,893	2,033	3,395
28847	www.wayfair.com	0,298	0,213	0,653	1,547	1,694
26399	www.talbots.com	0,828	0,035	3,905	1,395	1,869
41888	www.colehaan.com	0,529	0,043	2,649	1,365	2,127
37089	www.forever21.com	0,674	0,051	3,150	1,338	2,039
28359	www.brooksbrothers.com	0,659	0,034	2,988	1,314	1,911
69916	www.elle.com	0,426	0,078	1,568	1,280	1,530
99148	www.harpersbazaar.com	0,458	0,059	1,947	1,221	1,638
108034	www.kayak.com	0,299	0,187	0,571	1,209	1,476
99243	www.menupages.com	0,451	0,055	1,595	1,024	1,390
48200	www.jetblue.com	0,332	0,088	0,815	0,958	1,330
36668	www.weddingwire.com	0,495	0,055	1,352	0,929	1,126
87736	www.dior.com	0,705	0,016	1,972	0,894	1,382
35907	www.allrecipes.com	0,203	0,281	0,024	0,867	0,684

are the following:

1. Whether or not to take into account all permutations of all players.
2. Whether or not to take into account all permutations of the players of the winning coalition.
3. What value to assign to a player that is part of a winning coalition.

The conditions under which the different approximations would converge are left as an interesting issue for future research.

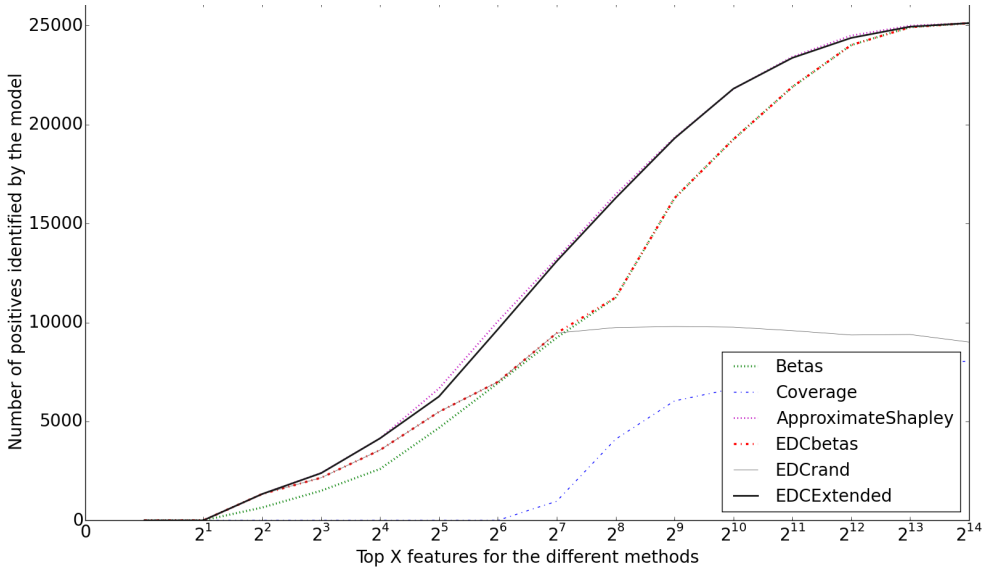
These results however, only describe the methods in terms of correlation and feature ranking. It would be interesting to compare them in terms of revenue that they would generate when using their ranking strategy. This will be shown in the next section.

6.2.2 Revenue curves

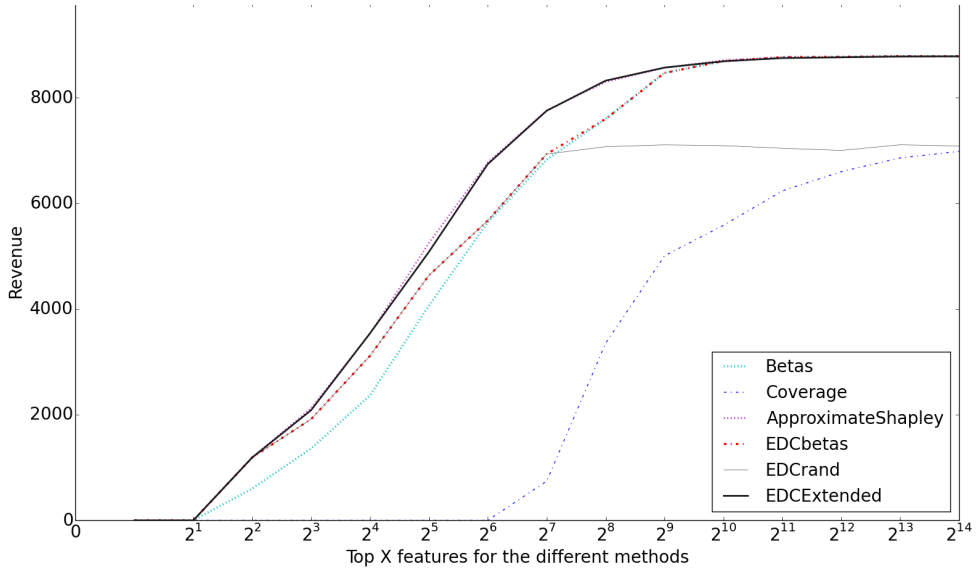
In this section, we propose an alternative way of evaluating the methods. Revenue curves are plotted which allow to select the data sources that are most useful. Or, put differently, which allow to find the combination of data sources or features that will lead to a certain revenue. This information could be valuable when data sources should be acquired. First, the different distribution schemes are compared in terms of revenue. Afterwards, we look at the differences between the two cost schemes (CPM and CPA).

DISTRIBUTION METHODS

Figure 3 plots these curves for the case of CPM, implying that features get rewarded both when they appear in a correct and false positive prediction. The X-axis denotes the top X number of features according to that method (in log scale), the Y-axis of Figure 3(a) shows the number of instances that would be classified as positive (i.e. get a score larger than the threshold) when only those X features are used and all other website visits are put to zero. That is, if we would only use the top X most valuable and informative features according to that method, how many positives would be identified. Figure 3(b) on the other hand plots the revenue, which is calculated by assigning a value of $+1$ for a correct and a cost of -0.1 for a false prediction. In other words, the curves in Figure 3(b) can be interpreted as follows: When the top X ranked features for a specific method are used, there are A true and B false predictions and so the revenue equals ((1*A) - (0.1*B)). This implies that an incorrect prediction denotes a cost and thus less revenue. Multiple plots were built in this way, one for each method. First, the curves for ***Coverage*** and ***Betas*** are shown. The beta curve for example could be interpreted as follows: If only the top X features with the largest β values are

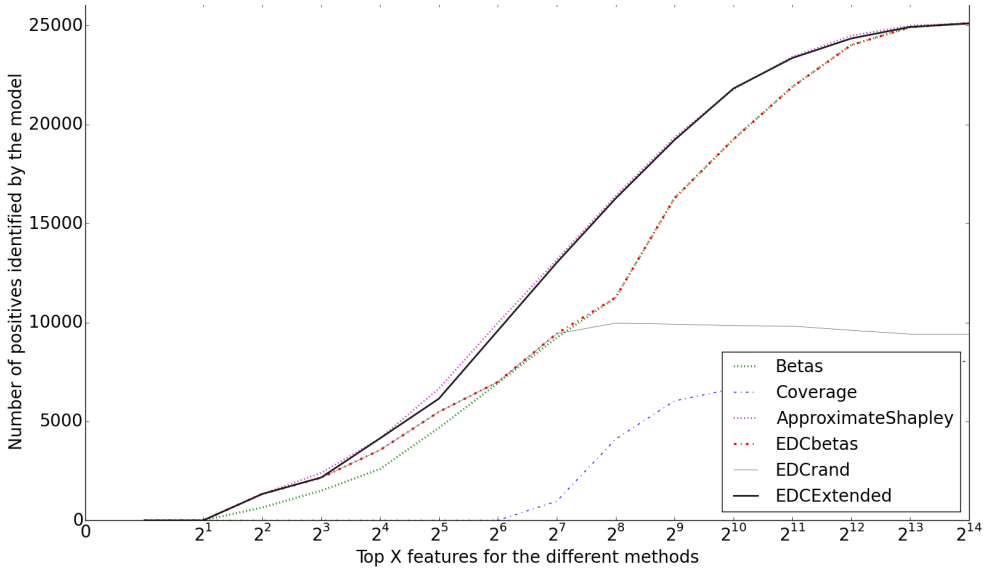


(a) Revenue curve showing the positives identified by the top X features as ranked by that method

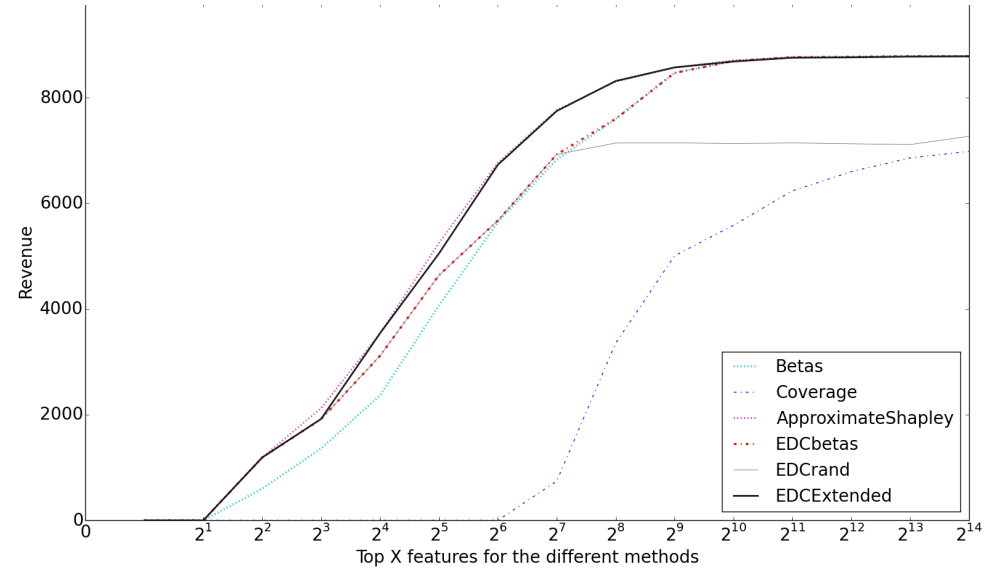


(b) Revenue curve showing the revenue (where a correct prediction yields a value of 1 and a false prediction a value of -0.1) when the top X features as ranked by that method are used

Figure 3: Revenue curve when CPM is used to calculate the value per feature



(a) Revenue curve showing the positives identified by the top X features as ranked by that method



(b) Revenue curve showing the revenue (where a correct prediction yields a value of 1 and a false prediction a value of -0.1) when the top X features as ranked by that method are used

Figure 4: Revenue curve when CPA (cost=0.1) is used to calculate the value per feature

included, how many positive predictions (Figure 3(a)) or revenue (Figure 3(b)) would be produced. For the EDC method, the simple implementation approach is shown where for each instance, only one explanation is used. Because the algorithm only looks for one explanation, fewer features will receive a score. In a first plot, the zero-value features that result from this simple EDC implementation, are sorted randomly. This curve is named ***EDCrand***. A second plot ranks these zero-value features according to their weight (descending order). We named this plot ***EDCbetas***. Next, the more extended version of EDC is plotted. That is, the method where multiple explanations are generated, by using a heuristic search (allows for 100 iterations). Again, there are still many zero-value features and those are again ranked according to their βs. This curve is called ***EDCExtended***. Next, the revenue curve as a result of the Approximate Shapley method is plotted and is referred to as ***ApproximateShapley***.[6] The graph shows that for the Simple EDC method, only about 100 features get a non-zero score. Until that point, EDCbetas and EDCrandom will thus be identical. After that, the graph for EDCbetas will follow the beta curve and the EDCrandom curve will continue without any (significant) raise in positive predictions. Surprisingly, one can see that only taking into account the weights (βs) of the prediction model is not the best way to check for the most informative features. The Extended EDC method as well as the Approximate Shapley method outperform the beta curve in a sense that using the combination of the most important features according to those methods leads to more instances classified as positives as compared to just using the top X βs of the linear prediction model (Figure 3(a)). That is, it might be that a feature has a very high weight in the prediction model but almost never appears. The coverage accounts for this effect and is also taken into account by both the Extended EDC and Approximate Shapley value method. Also in terms of revenue (Figure 3(b)), the beta curve is outperformed.

REVENUE SCHEMES

Figure 4(a) and 4(b) plot the same graphs but this time the feature scores are calculated by using a cost of 0.1 for a false positive prediction. That is, when a positive is correctly classified, a revenue of 1 is divided over the features. In case of a falsely classified positive, a negative value of -0.1 is divided over the features. The total score per feature is obtained by summing these positive and negative scores over all instances. When comparing Figure 3(b) and 4(b), it can be seen that these are very similar. Again, assigning a cost for false predictions does not have a large effect for this specific case study.

[6] Although it occurs less often, the Shapley method also assigns zero-values. Yet, these are not visible in the graph as the top 2^{14} features ranked by the Shapley values, have a non-zero value.

7 Discussion and limitations

In this work, we investigated how to divide value among features in a prediction model. Firstly, we defined the problem and gave several applications in which predictive models using high-dimensional sparse data is useful and the problem of assigning value among data sources needs to be tackled. These range from banking and telco to Internet of Things applications. The managerial implications of a solution to the attribution of value among players in a data pool are hence quite extensive.

The specific case study stems from an online advertising context where a customer is served an ad when a certain combination of websites has been visited. This implies that those websites or features have created value for the company. The question we tried to tackle is: *"How to reward each of those data sources (websites in this case) accordingly?"* Two methods from divergent research fields (game theory and document classification) were used to create a scoring mechanism that meets all requirements that were suggested, both qualitatively: (1) The data sources should be compensated proportional to the added value they create, (2) The distribution among the features should be fair and (3) both instance and feature level are important, and quantitaively with our newly introduced concept of revenue curves.

The first method, Shapley value, is a well-established method in the literature. The EDC method on the other hand is a more intuitive way of distributing the value and is easier to calculate. Unfortunately, both methods are impossible to calculate for high-dimensional data and so approximation methods are used. Surprisingly, the results show that even though at first sight it seems that these methods differ greatly, both schemes are performing fairly similar. With similar we mean that they will give a higher score to the same set of features, yet the exact ranking will be slightly different. The revenue curves show that both methods are able to identify combinations of features so that when only using those websites in the prediction model, more positives will be identified (or more revenue is created) as compared to dividing the value proportionally to the βs or coverage of the features. Furthermore, the results show that for this case study, the influence of the revenue scheme (CPM vs CPA) is rather low. Hence, assigning a cost for a false prediction does not have a large effect for this data set. It is nevertheless important to emphasize that the results are only indicative for this specific data set and generalization of these findings to other applications is to be seen.

An interesting finding beyond the scope of this paper is that we introduced a method for cooperative game theory that is applicable to non-linear settings. If in a voting game certain combinations get an additional surplus vote, the approximation methods that exist for linear settings will no longer be valid. Our EDC method on the other hand is able to deal with such non-linearities. More broadly, we have described

the differences between the EDC and Shapley method, both empirically and qualitatively (see also the Appendix). Although these differences seem small for our application, they do exist. The EDC method is a new method for cooperative game theory that might be of use in certain settings. Hence, we hope that this work opens up two research domains: (1) the area of value attribution in a data pooling scheme with predictive modeling, ranging from proposing new methods to applications in domains beyond online advertising; and (2) the link between the EDC method and cooperative game theory, ranging from the application of the EDC method in game theoretical settings, and more formal investigations of how and when these converge and when they are substantially different.

8 Future work

Since this was a first, preliminary research, many additional improvements and ideas for further research can be thought of.

Models First of all, other schemes that are able to take into account the requirements as listed above might be interesting for future research. Secondly, when looking differently at these methods, one could notice that ultimately this can be seen as feature selection as well. Thus, another interesting topic to further research is the comparison between these methods and some existing feature selection methods. Also, in the current implementation of the Extended EDC method, the value is divided equally among the features in a subset. Alternatively, one could divide this value proportional to βx for example. More research about the differences between these approaches is needed. Lastly, it is not realistic to calculate the value per website as was done in this paper. In the real world, data sources are used that include the information of several websites. How to bundle these values or use these methods on the level of data sources is an interesting research topic as well.

Requirements and assumptions In the current set-up, only positive predictions are taken into account. We only distribute value to the features that contribute to a (true or false) positive prediction, so in this case to features with a positive weight. It can be interesting though to expand this idea to the negative case as well. That is, features with negative βs do not contribute to a positive prediction, but might contribute to a correct prediction. In this case, negative βs should also be rewarded. For example, it is possible that some features with a negative β cause the probability score (for a specific instance) to be below the threshold. Therefore, this instance will be classified as negative. If this prediction is correct, it would make sense to assign value to these negative features as well. This is a very interesting topic for future research. Also, the

extension of these methods to non-linear models is an interesting and challenging topic for future research. This is quite straightforward for the EDC method but will be more challenging for the Approximate Shapley method since this one assumes the linear case and does not use model evaluation for every combination.

Evaluation The evaluation method is an interesting point that can be looked at. One problem that might arise when CPA is applied is that some features have a negative score. That is, features that are often present in a false positive prediction, might have an overall negative value on a feature level. How these negative weights should be interpreted is also something to think about. That is, when converting this to a monetary value, this would imply that some data sources should pay instead of receive money. Since this is not a valid solution, this is also interesting to look at for future research. Also, it would be useful to include the negative predictions and to build similar revenue plots which evaluate the AUC or lift. In this way, the correctness of the predictions, based on the top ranked features as proposed by the models, is evaluated as well. Lastly, we focused on assigning a value to each of the features. Another interesting research question in order to enhance applicability might be how to convert these to monetary values.

References

J.R. Green A. MasColell, M. Whinston. *Microeconomic Theory*. Oxford University Press, 1995.

Basu Banasri, Bikas K. Chakrabarti, Satya R. Chakravarty, and Kausik Gangopadhyay. *Econophysics and Economics of Games, Social Choices and Quantitative Techniques*. Springer, 2010. ISBN 978-88-470-1500-5.

Daizhuo Chen, Samuel P. Fraiberger, Robert Moakler, and Foster Provost. Enhancing transparency and control when drawing data-driven inferences about individuals. Working Paper 2451/33969, NYU, May 2015.

Brian Dalessandro, Daizhuo Chen, Troy Raeder, Claudia Perlich, Melinda Han Williams, and Foster Provost. Scalable hands-free transfer learning for online advertising. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1573–1582. ACM, 2014a.

Brian Dalessandro, Claudia Perlich, and Troy Reader. Bigger is better, but at what cost? *Big Data Journal*, 2(2):87–96, 2014b.

Morton Davis and Michael Maschler. The kernel of a cooperative game. *Naval Research Logistics Quarterly*, 12(3):223–259, 1965. ISSN 1931-9193.

Shaheen S. Fatima, Michael Wooldridge, and Nicholas R. Jennings. A linear approximation method for the shapley value. *Artificial Intelligence*, 172(14):1673 – 1699, 2008. ISSN 0004-3702.

Tom Fawcett and Foster Provost. Adaptive fraud detection. *Data mining and knowledge discovery*, 1(3): 291–316, 1997.

George Forman. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.*, 3:1289–1305, March 2003. ISSN 1532-4435.

Enric Junqué de Fortuny, David Martens, and Foster Provost. Predictive modeling with big data: Is bigger really better? *Big Data*, 1(4):215–226, 2013.

John Langford, Lihong Li, and Alex Strehl. Vowpal wabbit online learning project, 2007.

David Martens and Foster Provost. Explaining data driven document classification. *MIS Quarterly*, 38(1): 73–99, 2014.

David Martens, Foster Provost, Jessica Clark, and Enric Junqué de Fortuny. Mining massive fine-grained behavior data to improve predictive analytics. *MIS Quarterly*, 40(4):869–888, 2016.

M. Nagarajan and G. Soi. Game-theoretic analysis of cooperation among supply chain agents: Review and extensions. *European Journal of Operational Research*, 187(3):719 745, 2008. ISSN ISSN 0377-2217.

OECD. Exploring the economics of personal data. OECD Publishing, Paris, 2013.

M.J. Osborne and A. Rubinstein. *A Course in Game Theory*. The MIT Press, 1994.

Claudia Perlich, Foster Provost, and Jeffrey S. Simonoff. Tree induction vs. logistic regression: A learning-curve analysis. *J. Mach. Learn. Res.*, 4:211–255, December 2003. ISSN 1532-4435. doi: 10.1162/153244304322972694.

Foster Provost, Prem Melville, and Maytal Saar-Tsechansky. Data acquisition and cost-effective predictive modeling: Targeting offers for electronic commerce. In *Proceedings of the Ninth International Conference on Electronic Commerce*, ICEC '07, pages 389–398, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-700-1.

Foster Provost, David Martens, and Alan Murray. Finding similar mobile consumers with a privacy-friendly geosocial design. *MIS Quarterly*, 26(2):243–265, 2015.

Troy Raeder, Ori Stitelman, Brian Dalessandro, Claudia Perlich, and Foster Provost. Design principles of massive, robust prediction systems. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1357–1365. ACM, 2012.

Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010.

L.S. Shapley. A value for n person games. *University of Cambridge Press*, pages 31–40, 1988.

Ellen Tobback and David Martens. Bankruptcy predicting using fine-grained payment data. Technical report, Working Paper Universiteit Antwerpen, 2017.

Wouter Verbeke, Karel Dejaeger, D. Martens, J. Hur, and Bart Baesens. New insights into churn prediction in the telco sector: A profit driven data mining approach. *European Journal of Operational Research*, 218(1):211–229, 2012.