Pessimistic Cost-sensitive Active Learning of Decision Trees for Profit Maximizing Targeting Campaigns

Lior Rokach[1], Lihi Naamani[2], Armin Shmilovici[1]

[1]Department of Information System Engineering
[2]Deutsche Telekom Laboratories at Ben-Gurion University

Ben-Gurion University of the Negev, P.O. Box 653

Beer-Sheva 84105, Israel
{liorrk,ln,armin}@bgu.ac.il

Keywords
Cost-sensitive learning, Reinforcement learning, Active learning, Direct marketing, Decision trees, Design of experiments

Abstract
In business applications such as direct marketing, decision-makers are required to choose the action which best maximizes a utility function. Cost-sensitive learning methods can help them achieve this goal. In this paper, we introduce Pessimistic Active Learning (PAL). PAL employs a novel pessimistic measure, which relies on confidence intervals and is used to balance the exploration/exploitation trade-off. In order to acquire an initial sample of labeled data, PAL applies orthogonal arrays of fractional factorial design. PAL was tested on ten datasets using a decision tree inducer. A comparison of these results to those of other methods indicates PAL's superiority.

1. Introduction and Motivation

When marketing a service or a product, firms increasingly use predictive models to estimate the customers' interest in their offer. A predictive model estimates the response probability of the potential customers in question, and helps the decision-maker assess the profitability of the different customers. Predictive models assist a target marketing strategy: offering the right product to the right customer at the right time using the proper distribution channel. The firm approaches the customers estimated as the most interested and proposes a marketing offer. A customer that accepts the offer and conducts a purchase adds to the firms' profits. This strategy affords better efficiency than a mass marketing strategy, in which a firm offers a product to all known potential customers, usually resulting in low positive response rates. For example, a mail marketing response rate of 2% or a phone marketing response of 10% are considered good.

Predictive models can be built using data mining methods. These methods are applied to detect useful patterns in the information available about the customers purchasing behaviors (e.g., Zahavi and Levin, 1997; Buchner and Mulvenna, 1998; Ling and Li, 1998; Viaene et al., 2001; Yinghui, 2004; Domingos, 2005). Data for the models is available, as firms typically maintain databases that contain massive amounts of information about their existing and potential customers such as the customer's demographic characteristics and past purchase history.

Active learning (Cohn et al., 1994) refers to data mining policies which actively select unlabeled instances for labeling. It has been previously used for facilitating direct marketing campaigns (Saar-Tsechansky and Provost, 2007): during an *exploration phase* some potential customers are approached with a marketing offer. Based on their response, the learner actively selects the next customers to be approached, and so forth. Exploration does not come without a cost. Direct costs might involve hiring special personnel for calling customers and collecting their characteristics and responses to the campaign. Indirect costs may be incurred from contacting potential customers who would normally not be approached due to their low buying power or low interest in the product or service offer.

An aspect involved in marketing campaigns is the well-known concept of exploration/exploitation trade-off (Kyriakopoulos and Moorman, 2004). Exploration strategies interact with customers to explore their behaviors, while exploitation strategies operate on a firm’s existing marketing model. In the exploration phase, a concentrated effort is made to build an accurate model. In this phase, the firm may, for example, acquire any available information which characterizes the customer. During this phase, the results are analysed in depth and the best modus operandi is chosen. In the *exploitation phase* the firm simply applies the induced model – with no intention of improving the model – to classify new potential customers and identify the best ones. Thus, the model evolves during the exploration phase and is fixed during the exploitation phase. Given the tension between these two objectives, research has suggested that firms first explore customer behaviors and then follow with an exploitation strategy (Rothaermel and Deeds, 2004; Clarke, 2006). The result of the exploration phase is a marketing model that is then used in the exploitation phase.

The problem we address in this paper is which potential customers a firm should approach with a new product offer in order to maximize the net profit. Specifically, our objective is not only to minimize the net acquisition cost during the exploration phase, but also to maximize the net profit obtained during the exploitation phase. Our problem formulation takes into consideration the direct cost of offering a product to the customer, the utility associated with the customer's response, and the alternative utility of inaction.

We focus on a *binary discrete choice problem*, where the customer's response is binary, such as *acceptance* or *rejection* of a marketing offer. Discrete choice tasks may involve several specific problems, such as unbalanced class distribution. Typically, most customers considered for the exploration phase reject the offer, leading to a low positive response rate. However, an overly-simple classifier may predict that *all* customers in questions will reject the offer.

Another problem is that the predictive accuracy of a classifier alone is insufficient as an evaluation criterion. One reason is that different classification errors must be dealt with differently: mistaking acceptance for rejection, is particularly undesirable. Moreover, predictive accuracy alone does not provide enough flexibility when selecting a target for a marketing offer, or when choosing how an offer should be promoted. For example, the marketing personnel may want to approach 30% of the available potential customers, but the model predicts that only 6% of them will accept the offer (Ling and Li, 1996); or, they may want to personally call the first 100 most likely to accept, and send a personal mailing to the next 1000 most likely to accept. In order to solve some of these problems, learning algorithms for target marketing are required not only to classify but to produce a *probability* estimation as well. This enables ranking the predicted customers by order of their estimated positive response probability.

Active learning merely aims to minimize the cost of acquisition, and does not consider the exploration/exploitation tradeoff. Active learning techniques do not aim to improve online exploitation. Nevertheless, occasional income is a byproduct of the acquisition process. We propose that the calculation of the acquisition cost performed in active learning algorithms should take this into consideration.

Most existing active learning methods assume that the first batch of labeled instances is selected randomly or given as an input to the algorithm. Mayer and Sarkissian (2003) illustrated the usefulness of applying Design of Experiment (DoE) to active learning. We suggest using DoE in the initial sample, followed by a different strategy for selecting the subsequent unlabeled instances for labeling.

In this paper, we present a new learning framework for the discrete choice target marketing problem: Pessimistic Active Learning (PAL). When selecting the next batch of customers to be courted by a marketing campaign, active learning strictly addresses improved exploration.

However, PAL, like reinforcement learning, also considers a secondary criterion: the costs/profits of the exploration/exploitation trade-off during the exploration phase. PAL applies a novel incremental pessimistic measure, which relies on confidence intervals. According to this measure, during the exploration phase, PAL selects which customers are to be approached. PAL also employs a known simulated annealing model, so that the ratio between exploration and exploitation is traded dynamically, and thus exploration fades over time.

PAL offers four main innovations:

1) *Pessimism*: The selection of instances to be acquired during the exploration phase is based on the change in the lower bound of the confidence interval of the success probability rather than on the probability itself. There have been several successful attempts to use the pessimistic approach in machine learning (e.g., Quinlan, 1993 and Saar-Tsechansky and Provost, 2004). However, to the best of our knowledge, no cost-sensitive method considers the effect of the confidence level of the estimated probability in classifier learning problems.

2) *Working with batches*: Our assumption is that a marketing campaign is carried out in batches. In other words, given a trained classifier, the campaign manager selects a batch of customers to solicit. Only after obtaining the responses of all customers in this batch is a new classifier trained. Given this, the decision of whether to include a certain customer in the next batch should take into account its contribution to the entire batch. In this study, we develop an approximation method to estimate the potential contribution of the n^{th} customer in the batch.

3) *Design of Experiments*: The proposed algorithm employs well-known statistical design of experiment (DoE) methods in order to select the first batch of labeled customers, which are needed for the construction of the initial classifier in a non-random way. Specifically, we integrate an orthogonal array of fractional factorial designs.

4) *Exploration-exploitation trade-off*: While most cost-sensitive active learning methods try to optimize some testing set measures, such as profit, in this study we are also interested in training performance, namely the profit or loss incurred during the exploration phase. We adopt and incorporate a well-known simulated annealing technique to gradually increase exploitation during the exploration phase.

The remainder of this paper is organized as follows: Section 2 introduces the problem formulation. Section 3 presents the components of a new active learning algorithm for decision trees. Section 4 reports the experiments carried out on benchmark datasets. Section 5 presents related work, and Section 6 concludes the work.

2. Problem Formulation

The main objective of a marketing campaign is to select which potential customers a firm should approach with a new product offer, in order to maximize the net profit. In the marketing problem presented in this paper, we assume that the firm holds an initial dataset of potential customers that can be used during the exploration phase. This initial dataset does not, however, cover all potential customers. We also assume that while acquiring the customers' response is costly, some of the courted customers will respond positively to the offer and the income from their purchase will offset the cost. Thus, Mayer and Sarkissian (2003) proposed referring to the net acquisition cost, which is the total cost of acquiring customer response, less the income generated if the courted customers purchase the products.

We also assume that during a marketing campaign, a firm will not approach its customers one by one, but it will rather approach a batch of customers simultaneously, so that the firm can concentrate its exploitation of resources, such as marketing personnel and equipment. After a campaign session is over and a batch of customers has been courted, the firm can analyze the results and proceed to the next stage of the campaign. We assume a fixed batch size.

In our targeted marketing context, an instance $x_i \in X$ is defined as the set of attributes[1], such as age and gender, of a unique potential customer *i*. For the sake of clarity, we will assume a binary outcome for the target attribute *y*, specifically *y* = {"accept", "reject"}. Unlabeled instances are defined as instances with an unknown target attribute. A set *S* of *M* unlabeled instances from the set *X* is obtained. The instances in *S* are independent and behave according to some fixed and unknown joint probability distribution *D* of *X* and *Y*. The cost of approaching customer *i* with an offer is denoted as $C_i \in \Re$. The probability that customer *i* will respond positively to the offer is denoted as p_i. If customer *i* with some unknown probability p_i agrees

[1] In this research, we assume that the attributes are independent and that there are no missing values. If that is not the case, we assume that data *pre-processing* methods are used to complete the missing information, remove the dependent attributes and scale the numbers.

to the offer, the utility obtained from this customer is denoted as $U_i^S \in \Re$. If the customer rejects the offer, the utility is denoted as $U_i^F \in \Re$. Thus, the net acquisition cost of customer *i* is defined as:

$$NAC_i = \begin{cases} C_i - U_i^S & \text{if customer } i \text{ accepts the offer} \\ C_i - U_i^F & \text{if customer } i \text{ rejects the offer} \end{cases} \tag{1}$$

Note that all utility values are a function of the customer's attribute vector ($\boldsymbol{x_i}$).

Let the corresponding utility of inaction with respect to customer *i* be denoted as Ψ_i. In order to maximize the expected profit, the decision-maker should court customer *i* if the probability of a positive response is higher than the cost of approach (Saar-Tsechansky and Provost, 2007). This is represented in the following equivalent equations:

$$\hat{p}_i \cdot U_i^S + (1-\hat{p}_i)\cdot U_i^F - C_i > \Psi_i \text{ or } \hat{p}_i > \frac{C_i + \Psi_i - U_i^F}{U_i^S - U_i^F} \equiv \frac{o_i}{r_i} \tag{2}$$

where o_i and r_i are merely shorthand for the numerator and denominator of the decision threshold ratio. The notation $\hat{p}_i$ represents the classifier's estimation for p_i.

A pseudo code for the active learning framework used for the target marketing process is presented in Figure 1. The received input includes: a pool of unlabeled instances (*S*), an inducer (*I*), and a stopping criterion (*CRIT*). The first step is to initiate the labeled pool (line 1). An initial set of labeled examples is selected in Line 3. Once the potential customers are selected, they are approached with a product offer (line 6). According to the customers' response, the newly labeled examples are added to the labeled pool (line 7). The labeled pool is then used for building the classifier (line 8). Based on the classifier, the next subset from the unlabeled pool is selected (line 10). This process is repeated until triggering some sort of stopping criteria (line 3), such as running out of budget. The final classifier (the output, line 12) is used to estimate the probability of positive response $\hat{p}_i$ of new customers. Customers with an estimated probability that exceeds the threshold in Eq. (2) are contacted.

Based on the active learning framework presented in Figure 1, the marketing learning problem can be defined as follows:

While keeping the total net acquisition cost to minimum, the goal is to actively acquire from *S* mutually exclusive subsets $S_1, S_2, \ldots, S_k$ of a given batch size *M*, such that the final classifier induced from $\bigcup_{i=1}^{k} S_i$ maximizes the profitability of the campaign. The subsets are acquired sequentially.

```
Active Learning Framework
Input:
       S - An unlabeled pool of instances
       I – An induction algorithm
       CRIT – A stopping criterion
Output:
       CL – Classifier for predicting customer response

1. L←∅   /* the labeled pool */
2. i←1
3. Si←Select initial set of instances from S
4. While CRIT is not met do
5.     Remove Si from S
6.     Acquire labels for examples in Si
7.     Add Si to L
8.     Apply I to L, resulting in a classifier CL
9.     i←i+1
10.    Select subset Si from S using CL
11. End While
12. Return CL
```

Figure 1: Pseudo Code for the Active Learning Framework

This is a Multiple Criteria Decision-Making (MCDM) problem. The first criterion is to improve the decisions of the campaign manager. The positive reactions rate can be used to assess the profitability during the exploitation phase. Higher rates indicate higher gross profit margins and return of investments (ROI). The second criterion is to acquire labeled instances with minimal net acquisition cost during the exploration phase. Both criteria deal with financial utilities. Still, the two criteria cannot be summed. We cannot represent the first criterion as total income during the exploitation phase, since we do not know in advance how many customers are going to be evaluated using the model. The only assumption we make is that the instances in the unlabeled instances set used during the training phase (*S*) and the instances examined during the operational phase are both distributed according to a fixed and unknown distribution *D*. In this paper, we consider the first criterion as primary and the second as secondary. Prioritization of these criteria agrees with the assumption that the exploitation phase is longer than the exploration phase.

In this paper, we use a *Decision Tree* classifier to estimate p_i. Decision trees are considered to be self-explanatory models and easy to follow when compacted (Rokach and Maimon, 2005). They have been previously used in marketing scenarios (e.g., Levin and Zahavi, 2005; Saar-Tsechansky and Provost, 2007). The principles underlying the proposed PAL approach can be adjusted to other induction methods, such as neural networks. Neural network classifiers have also been applied to target marketing (Zahavi and Levin, 1995; Zahavi and Levin, 1997; Potharst et al., 2002).

In order to estimate the probability p_i with the decision tree classifier, the appropriate leaf k in the tree that refers to the given instance $\boldsymbol{x_i}$ should first be located. The frequency vector of each leaf node captures the number of instances from each possible class. In the usual case of target marketing, the frequency vector has the form: $\left(m_{k,accept}, m_{k,reject}\right)$ where $m_{k,c}$ denotes the number of instances in the labeled pool that reach leaf k and satisfy $y = c$. According to Laplace's law of succession, the probability p_i is estimated as:

$$\hat{p}_i = p(m_{k,accept}, m_{k,reject}) = \frac{m_{k,accept} + 1}{m_{k,accept} + m_{k,reject} + 2} \quad . \tag{3}$$

Besides estimating the point probability $\hat{p}_i$, we are interested in estimating a confidence interval for this probability. An approach to a customer can be considered as a Bernoulli trial. For the sake of simplicity, we approximate the confidence interval of the Bernoulli parameter with the normal approximation to the binomial distribution:

$$\hat{p}_i - z_{1-\alpha/2}\hat{\sigma}_i < p_i < \hat{p}_i + z_{1-\alpha/2}\hat{\sigma}_i$$
$$\hat{\sigma}_i = \sigma(m_{k,accept}, m_{k,reject}) = \sqrt{\frac{\hat{p}_i(1-\hat{p}_i)}{m_{k,accept} + m_{k,reject}}} \tag{4}$$

where $\hat{\sigma}_i$ represents the estimated standard deviation and $z_{1-\alpha/2}$ denotes the value in the standard normal distribution table corresponding to the $1-\alpha/2$ percentile. For a small n we can use the actual binomial distribution to estimate the interval. Leemis and Trivedi (1996) proposed additional approximations.

To demonstrate the importance of a confidence level, consider two leaves: leaf *A* and leaf *B* in a classification tree. Each leaf holds the customers in the labeled pool that fit its path. These customers are labeled as either “accept” or “reject”. If the “accept”/ “reject” proportions are the same, then according to Eq. (3), both leaves have the same estimated probability. Given this, if leaf *A* has more customers than leaf *B*, then according to Eq. (4), leaf *B* has a larger confidence interval. Thus, acquiring an instance to leaf *B* will have a greater impact on the class distribution than adding an example to leaf *A*. In the initial iterations, when the data are limited and the confidence intervals are large, obtaining an additional instance to the correct leaf is especially important. Moreover, the potential contribution of labeling the i^{th} instance in the same leaf and adding it to the labeled pool decreases in i. Thus, the calculation of the potential contribution of each instance in the new batch depends on the other instances that are selected to this batch.

3. The Pessimistic Active Learning Method

Figure 2 presents the pseudo code of the PAL (Pessimistic Active Learning) method. The algorithm receives as input the unlabeled set (*S*), an inducer (*I*) which PAL uses for building the classifier, and a certain batch size (*M*). First, the orthogonal arrays (OA) approach to designing experiments (Hedayat et al. 1999) is used to select the first batch of instances (lines 2-3). The first batch is labeled and is used to initiate the labeled set. The algorithm actively selects the next batches of size *M* until a given stopping criterion is met (Lines 4-15). In order to select the next batch, first an inducer is trained on the labeled set (line 5) and a new classifier (CL) is induced. This classifier is then used to make a selection for the next batch. The selected batch is labeled and is added to the labeled set.

In the following subsections, we present the important elements of PAL: (i) the OA approach to design of experiments used to select the first batch of instances; (ii) selection of subsequent batches by combining random exploration and biased exploration, which is intended to improve future exploitation; and (iii) a pessimistic profit estimator that is used for selecting the instances to be explored.

3.1 Initial Sample Selection

Design of experiments seeks to minimize the number of experiments required to collect useful information about an unknown process (Montgomery, 1997). The collected data are typically used to construct a model for the unknown process. The model may be used to optimize the original process.

A *full factorial design* is a design of experiment in which the experimenter chooses *n* attributes that are believed to affect the target attribute. Then, all possible combinations of the selected input attributes are acquired (Montgomery 1997). Applying a full factorial design is impractical when many input attributes are given.

```
Pessimistic Active Learning
Input:
       S - an unlabeled set
       I – an induction algorithm which is capable of providing confidence intervals
       M – the batch size
       CRIT – a stopping criterion
       α - the confidence level
       γ - Simulated annealing decay factor
       T0 – the portion of instances randomely selected

Output:
       CL – Classifier
1. j←0
2. Apply revised OA method to select the subset L containing M instances from S
3. Remove L from S, and label L
4. While CRIT is not met do
5.     Apply I to L, resulting in a classifier CL
6.     For each instance in S calculate the pessimistic profit gain (Eq. 10) using
                                          the probability estimation obtained from CL.
7.     Z1←Select top (1-Tj)· M customers from S sorted according to pessimistic profit gain
8.     S←S-Z1
9.     Z2← Randomly select Tj· M customers from S.
10.    S←S-Z2
11.    Z←Z1∪Z2
12.    Label Z and add Z to L
13.    j←j+1
14.    Calculate Tj using a simulated annealing technique (Eq. 11)
15. End While
16. Return CL
```

Figure 2: The PAL Pseudo Code

A *fractional factorial design* is a design in which only a fraction of the combinations required for the complete factorial experiment is selected. One of the most practical forms of fractional factorial design is the orthogonal array (Hedayat et al. 1999). An orthogonal array $Lk(d^n)$ is a matrix of k rows and n columns, with every element being one of the d values. The array has strength t if, in every n by t submatrix, the d^t possible distinct rows all appear the same number of times. An example of a strength 2 OA is presented in Table 1. Any two attributes in this array have all possible combinations (“11”, “12”, “21”, “22”). Each of these combinations appears an equal number of times. In an orthogonal array of strength 3 we can see all combinations in any three attributes (Hedayat et al. 1999).

Constructing a new orthogonal array design for any number of attributes is not an easy task. In this paper, we used a ready-made design for a specific number of attributes. The orthogonal designs were taken from Sloane's library of orthogonal arrays (Sloane 2007). The number of columns n in the OA should be equal to the cardinality of the input attributes set. Moreover the domain size d is usually set according to the nominal attribute with the largest domain. There might be several suitable designs for a certain dataset. Thus, we selected the design with the smallest number of rows k, aiming to include at least one complete design in the initial subset S_1. For some datasets, we did not find a design with the exact column cardinality. In such cases, we chose a design with more attributes, removed the redundant columns and kept only the distinct rows.

The original domain of each input attribute should be transformed to a domain of d distinct values. With nominal attributes, each attribute type represents one of the d values. Discretization methods address this issue for numeric attributes by transforming their values into d ranges of values[2].

Most experimental design approaches aim at settings where instances can be generated, as is often the case in lab experiments. In a pool-based selection setting, such as our own, we cannot generate instances to fit the design because the instances values are set in advance. Therefore, we first derive the design and then identify for each row in the design the instance in the unlabeled pool that is most similar. This is done by measuring the normalized Euclidean distance with all attributes having the same weight, and selecting the unlabeled instance with the smallest distance.

Table 1: The L8(2^7) OA's design

Instances	**Attributes**						
	a_1	**a_2**	**a_3**	**a_4**	**a_5**	**a_6**	**a_7**
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**
2	**1**	**1**	**1**	**2**	**2**	**2**	**2**
3	**1**	**2**	**2**	**1**	**1**	**2**	**2**
4	**1**	**2**	**2**	**2**	**2**	**1**	**1**
5	**2**	**1**	**2**	**1**	**2**	**1**	**2**
6	**2**	**1**	**2**	**2**	**1**	**2**	**1**
7	**2**	**2**	**1**	**1**	**2**	**2**	**1**
8	**2**	**2**	**1**	**2**	**1**	**1**	**2**

3.2 Pessimistic Profit Using Confidence Bounds

[2] In this paper we used a simple unsupervised discretization of equal-width.

In this section, we propose a new measure termed Pessimistic Profit Gain. This measure is used for ranking the customers as part of the selection of the next batch. The proposed approach aims to improve decision-making by measuring the change in the profit gain when risky profitable customers are acquired. We define a risky profitable customer as a customer about whom the decision made according to the estimated probability $\hat{p}$ is not coherent with the decision made according to the lower bound of the confidence interval (the pessimistic probability). Specifically, if the estimated probability of response suggests the customer is profitable, but the lower bound of the confidence interval of $\hat{p}$ is below the threshold of Eq. (2), then there is a non-negligible likelihood that the customer is not profitable. Hence, suggesting an offer to this customer is considered risky.

Our main goal during the exploration phase is indeed to explore the space. However, by selecting customers who are expected to be profitable, yet are considered risky, the Pessimistic Profit Gain measure adds exploitive value to the exploration. By acquiring more information about risky customers, a later classifier may infer that these customers are indeed unprofitable. Yet, we could equally consider an opposite strategy: acquiring information about customers for which the estimated probability suggests the customer is not profitable, but the upper bound of the confidence interval is greater than the threshold value, suggesting there is likelihood that the customer is in fact profitable. Ignoring the latter strategy is motivated by the assumption that most customers are likely to be unprofitable because contacting a customer is costly. Hence, it may be more beneficial to reduce the monetary risk of contacting unprofitable customers than improving the estimation thereby identifying more profitable customers.

Approaching a new customer can improve the probability estimation and the current decision tree must be updated accordingly. The decision tree can be updated in various ways, for instance by adopting the incremental procedure of the ID5R algorithm introduced by Utgoff (1989). For the sake of simplicity, we assume that the updated tree is obtained by revising only the class probability distribution of the corresponding leaf, and that no new sub-branches are created. Obviously, this is not always precise; however, we use it as a low-cost approximation of the actual value.

3.2.1 Definition of Pessimistic Profit

Let us define the notion of *profit* for a given leaf in the decision tree. Leaf k has $m_{k,accept}$ and $m_{k,reject}$ customers who were courted and responded positively or negatively, respectively. Moreover, there are additional $m_{k,new}$ customers in the unlabeled pool which belong to leaf k. We assume that the decision rule presented in Eq. (2) is satisfied, and we decide to approach the $m_{k,new}$ customers. For the sake of clarity, we assume that all costs and utilities are identical for all customers (e.g., $C_i \equiv C$). If a portion of p customers responds positively, then the total profit is calculated as the sum of four terms:

1. The utility from the customers who have responded positively: $p \cdot m_{k,new} \cdot U^S$
2. The utility from the customers who have responded negatively: $(1-p) \cdot m_{k,new} \cdot U^F$
3. The cost of approaching the $m_{k,new}$ customers: $m_{k,new} \cdot C$
4. The alternative income we lose (when no action is performed): $m_{k,new} \cdot \Psi$.

Therefore, the profit is:

$$profit \equiv p \cdot m_{k,new} \cdot U^S + (1-p) \cdot m_{k,new} \cdot U^F - m_{k,new} \cdot C - m_{k,new} \cdot \Psi \ . \tag{5}$$

Simplifying the expression with the definitions of r and o – the numerator and denominator defined in Eq. (2) – the expected profit is:

$$profit \equiv m_{k,new} \cdot r \cdot p - m_{k,new} \cdot o \ . \tag{6}$$

We define the pessimistic probability as the lower bound of (1-α)% confidence interval of the success probability. When payoffs are Boolean, the normal approximation to the binomial distribution can be used to construct the confidence interval. The pessimistic probability is the lower limit of the confidence interval presented in Eq. (4):

$$\tilde{p}_i \equiv \hat{p}_i - z_{1-\alpha/2}\hat{\sigma}_i \ . \tag{7}$$

By incorporating the pessimistic probability in Eq. (6), we define the *pessimistic profit* (PP) for a given leaf in the decision tree.

$$\begin{aligned} PP(m_{k,accept}, m_{k,reject}, m_{k,new}) &\equiv m_{k,new} \cdot r \cdot (\tilde{p}_i) - m_{k,new} \cdot o \\ &= m_{k,new} \cdot r \cdot \left(p(m_{k,accept}, m_{k,reject}) - z_{1-\alpha/2}\sigma(m_{k,accept}, m_{k,reject}) \right) - m_{k,new} \cdot o \end{aligned} \ . \tag{8}$$

Eq. (8) generates negative values when probability $\hat{p}_i$ and the pessimistic probability $\tilde{p}_i$ are on the opposite sides of the threshold value. When both $\hat{p}_i$ and $\tilde{p}_i$ are higher than the threshold value $\frac{o_i}{r_i}$ (Eq. 2), *PP* has a positive value. This means that both $\hat{p}_i$ and $\tilde{p}_i$ suggest the same decision. Thus the reduction of the confidence interval due to acquiring new unlabeled instances will not improve the decision. The positive *PP* value is therefore replaced with a value of *0*.

3.2.2 Calculation of Pessimistic Profit

Calculating the pessimistic profit for each customer is done in three steps. First, in order to separate customers with a profitable probability $\hat{p}_i$ from others, the decision rule presented in Eq. (2) is applied to all leaves in the decision tree. Customers who correspond to leaves where the probability $\hat{p}_i$ is lower than the threshold receive a gain value of 0, thus placing them at the bottom of the candidate list. The rest of the customers are considered as potential candidates, and move on to the second step: Eq. (8) is used to calculate the leaf's pessimistic profit for these cases. The final step is to estimate how the pessimistic profit will change if the response of a new customer is acquired and added to the corresponding leaf. Since the actual response of the customer in not known prior to its acquisition, the pessimistic profit is recalculated for each possible outcome.

a) The customer accepts the new product offer: $m_{k,accept}$ is increased by 1.

b) The customer rejects the new product offer: $m_{k,reject}$ is increased by 1.

In both cases $\hat{p}_i$ and $\tilde{p}_i$ are updated and $m_{k,new}$ decreases by 1. The two possible pessimistic profits above are weighted according to the estimated probability $\hat{p}_i$.

The pessimistic profit gain is the *difference* between the estimated pessimistic profit *before* and *after* approaching a customer. The customers are ranked in descending order according to their gain, and those with the highest gain are chosen to be contacted.

3.3 Pessimistic Profit Gain for a Group of *n* Customers

The previous subsection presented a method for calculating the pessimistic profit gain under two assumptions: (1) the customers are courted one at a time; and (2) the next customer is approached only after receiving the previous customer's response. However, this situation is not typically the case in many targeted marketing applications, since several salespersons can simultaneously contact multiple potential customers. Therefore, the targeting policy should be refined to allow a quota of customers to be approached simultaneously. The pessimistic profit for the first *n* customers of a certain leaf *k* is:

$$\begin{aligned} &PPG_n\left(m_{k,accept}, m_{k,reject}, m_{k,new}\right) = \\ &\sum_{j=0}^{n}\binom{n}{j} p^j(m_{k,accept}, m_{k,reject})\left(1 - p(m_{k,accept}, m_{k,reject})\right)^{n-j} \cdot \left(PP(m_{k,accept} + j, m_{k,reject} + n - j, m_{k,new} - n) + j \cdot r - n \cdot o\right) \end{aligned} \tag{9}$$

Eq. (9) calculates the expected pessimistic profit by examining all possible outcomes of approaching n customers and weighting their corresponded profits. We refer to the process of approaching n customers as a sequence of n independent Bernoulli experiments, each of which yields an "accept" outcome with a probability of $p(m_{k,accept}, m_{k,reject})$. Note that by setting $n=0$ in Eq. (9) we obtain the current pessimistic profit of a leaf, before any new customer is courted. The gain obtained by the n^{th} customer in leaf k is defined as:

$$G_n\left(m_{k,accept}, m_{k,reject}, m_{k,new}\right) = PPG_n\left(m_{k,accept}, m_{k,reject}, m_{k,new}\right) - PPG_{n-1}\left(m_{k,accept}, m_{k,reject}, m_{k,new}\right) \tag{10}$$

The gain is decreasing in n, i.e., the contribution of adding n instances to a certain leaf is smaller than n times the contribution of adding the first instance to that leaf. Note that our measure focuses on massive leaves, leaves in which there are more unlabeled customers; Hence, if two leaves have the same confidence interval for the estimated probability and the only difference between them is the amount of unlabeled corresponding customers, then Eq. (10) increases the priority of leaves corresponding to a larger set of instances.

3.4 Selecting the Subsequent Batches

While active learning explicitly seeks only improved exploration, PAL selects the next batch of customers to be courted by considering the exploration/exploitation tradeoff explicitly, just as reinforcement learning does. We employ simulated annealing (Kirkpatrick et al. 1983) to determine the amount of instances in a batch courted for exploitation purposes. The rest of the instances in the same batch are courted for exploration purposes.

Simulated annealing is a generic randomized strategy for global optimization problems. Its key idea by default is to exploit, that is, to take the action with the best estimated reward. Yet, with some probability, exploration is performed by selecting an action at random. The ratio between exploration and exploitation is traded dynamically, so that exploration fades in time.

The parameter $0 \le \gamma \le 1$ controls the rate of the decay. The parameter $0 \le T_j \le 1$ denotes the proportion of customers to be courted (explored) in batch j. T_j decreases over time to decrease exploration. We used the following simple and common exponential schedule:

$$T_j = \gamma T_{j-1} \tag{11}$$

The outcome T_j is multiplied by the batch size M in order to determine the amount of customers in the batch that are randomly selected. The complementary proportion $\left(1 - T_j\right)$ is again multiplied by M to determine the amount of customers that are selected according to their Pessimistic Profit Gain.

4. Experimental Study

In this section we present empirical evaluations of our approach for a set of benchmark datasets. These evaluations also examine the benefit from each of the algorithms components presented in the previous section.

4.1 Experimental Setup

4.1.1 The Benchmark Datasets Used in the Experiments

Because the proposed method is designed for binary domains, we selected ten publicly available binary class datasets with an unbalanced class distribution, so as reflect as much as possible the characteristics of the direct marketing domains addressed here. Specifically, we have used the *donation* dataset, which has been used in the KDD cup 98[3], and the *insurance company* benchmark, which has been used in CoIL challenge 2000 (Putten and Someren, 2000). In these two datasets, the class refers to a real response of the person to buy a policy or contribute a donation. The remaining datasets were obtained from the UCI repository (Blake and Merz, 1998). In these datasets we selected the less frequent class to represent the positive response.

Table 2 presents the characteristics of each dataset: the number of attributes, the selected training set size, the test set size, and the number of equally sized batches. The large datasets were partitioned into 60 batches, while the small datasets were partitioned into 20 batches.

In real world applications, the actual values of o and r, as defined in Eq. (2), are estimated from the specific application. In the donation problem domain, the cost of approach $(o=C_i)$ is given and the positive response utility ($r = U_i^S$) can be predicted (for instance, see Saar-Tsechansky and Provost, 2007 for a detailed description of how these values can be appropriately estimated). We had to fabricate the values for the other datasets considering the following arguments: (i) for values of o/r much lower than the customers' positive response rate, a positive profit is guaranteed and the relative contribution of an intelligent model is less significant; (ii) for values of o/r much higher than the customers' positive response rate, the risk

[3] http://kdd.ics.uci.edu//databases/kddcup98/kddcup98.html

of incurring losses becomes too high, and risky scenarios are unacceptable in most business applications. Therefore, avoiding risky scenarios, the maximum potential contribution of an intelligent model is manifested when the value of *o/r* is *equal* to the customers' positive response rate. Thus, we set the ratio of *o/r* at the proximity of the customers' positive response rate.

Table 2: Summary of the dataset characteristics used in the experimental study

Dataset	# Attributes	Training Size	Test Set Size	# Batches	Positive Response Rate	o Value	r Value
Adult	14	10000	20000	60	24%	2.9	10
Anneal	39	797	99	20	4.5%	0.49	10
Breast C.	10	500	199	20	34%	4.3	10
Credit	15	300	370	20	37%	3.5	10
Donation	15[4]	10000[5]	96,357	60	5%	0.68 (Given)	Varied (mean 15)
German	25	469	530	20	30%	3.2	10
Heart	14	124	145	20	44%	4.3	10
Insurance	85	5822	4000	60	6%	0.63	10
Mushroom	22	4062	4062	60	10%	1	10
Thyroid	30	2799	971	60	6%	0.61	10

4.1.2 Alternative Acquisition Algorithms

In order to evaluate the benefit of the PAL algorithm, we execute it with the following parameter values: γ=0.85 (the simulated annealing decay factor) and $\alpha = 5\%$ (the confidence level). We compared PAL to the following algorithms, which are later described in Section 5:

1. An algorithm which acquires new customers drawn uniformly at random.
2. Kaelbling's (1993) interval estimation algorithm: favors instances with high success probability estimation, and also focuses exploration on the most promising, but uncertain leaves. The confidence interval of the success probability is estimated for each leaf. Instances are selected from the leaf whose confidence interval has the highest upper bound.

[4] Originally the donation datasets contain 479 attributes. For the classification task we have used only the following input attributes: ODATEDW , INCOME ,RAMNTALL, NGIFTALL, CARDGIFT, MINRAMNT, MINRDATE, MAXRAMNT, MAXRDATE, LASTGIFT, LASTDATE, FISTDATE, NEXTDATE, TIMELAG , AVGGIFT

[5] The original dataset contained 95,413 training instances, of which we randomly selected only 10,000.

3. Randomized strategy using Boltzmann distribution (Kaelbling and Littman, 1996): in this case, an instance *i* is probabilistically chosen proportionally to $e^{-NAC_i/T}$, where NAC_i denotes the net acquisition cost of customer *i* and *T* is a temperature parameter that decreases over time to decrease exploration.
4. The GOAL algorithm (Saar-Tsechansky and Provost, 2007): GOAL, like PAL, aims to minimize the cost of acquisition to obtain a given performance. However, unlike PAL, GOAL considers only acquisition costs, but not revenue generated during the acquisition phase.

We also evaluated each one of PAL's four components presented in Section 3 by examining the following configurations:

1. *PAL without Simulated Annealing*: This variation of PAL includes the OA method for the initial sample with random acquisition on the subsequent batches. It does not include the simulated annealing module for trading exploration with exploitation (section 3.2). Any difference between the performance of this algorithm and PAL's can be attributed to the simulated annealing module.
2. *PAL without Pessimism*: Employing the OA method for the initial sample simulated annealing for trading exploration with exploitation, but instead of using the pessimistic estimate for the probability (sections 3.3 and 3.4) we use the probability estimation from Eq. (2). This PAL version aims to evaluate what pessimism buys us.
3. *PAL without Orthogonal Arrays*: Employing the PAL with random initialization, i.e., without OA. This PAL version is used to evaluate the value in the OA initialization.
4. *PAL with Optimism*: Employing the OA method for the initial sample simulated annealing for trading exploration with exploitation, but instead of using the pessimistic estimate for the probability (sections 3.3 and 3.4) we use the upper bound (optimistic). This PAL version aims to evaluate whether using the upper bound as other interval estimation techniques can produce preferable results.

The C4.5 induction algorithm (Quinlan, 1993) with the Laplace correction (Cestnik 1990) was employed in all the experiments to estimate the probability of success.

4.1.3 Evaluation Methodology

Each dataset was divided into two subsets. The first subset was used as the unlabeled pool for the iterative selection of the training instance. A fixed number of instances M (the batch size) was chosen in each iteration. The second subset is a test set of instances for which we compare the profits generated by each approach after each acquisition phase. In order to provide reliable estimates of the algorithms' performance and analyze if the differences between reported performances are statistically significant, we generated ten stratified random partitions onto training and testing datasets. To reduce the experimental variance, the same data partitions were used by all methods. Moreover, methods that did not use OA for the initial sample (i.e., Random, GOAL, PAL without Orthogonal Arrays) were started from the exact same initial random sample. Similarly, methods that employed OA were also started from the exact same OA sample.

We evaluated four performance measures for each algorithm and dataset: (i) training profit; (ii) test set positive reaction rate; (iii) test set profit; and (iv) gain charts. The first two measures represent the two criteria that were defined in the problem formulation. The last two measures are used to obtain an additional assessment of PAL's contribution.

Because the curves of the compared algorithms might intersect, we used the AUC (Area Under the Curve) measure as a single value metric to compare algorithms and establish a dominance relationship among them. The reported values represent the mean AUC performance over the ten random partitions of the data. The confidence interval of the AUC was estimated using the Student's t distribution. The statistical significance of the differences in performance between the PAL algorithm and the other algorithms was verified by the one-tailed paired t-test, with a confidence level of 95%.

Additionally, we provide the mean rank of each algorithm across data sets. For this purpose, we rank the algorithms for each dataset separately and provide the average rank of each algorithm across data sets. The best performing algorithm is ranked 1.

To compute the mean normalized performance of each method, we use simple linear scaling within the dataset *minimum and maximum performance values*. The normalized values are used to quantify the differences across all datasets. Formally, the normalized performance of algorithm i on dataset j is defined as

$$NAUC_{i,j} = \frac{AUC_{i,j} - \min_k AUC_{k,j}}{\max_k AUC_{k,j} - \min_k AUC_{k,j}} \quad (12)$$

Thus, the mean normalized performance of algorithm *i* is

$$MNAUC_i = \sum_{j=1}^{n} \frac{NAUC_{i,j}}{n} \quad (13)$$

In order to conclude which algorithm performs best over multiple datasets, we followed the robust non-parametric procedure that was proposed by Demsar (2006). In case of multiple classifiers, we first used the adjusted Friedman test in order to reject the null hypothesis, followed by the Bonferroni-Dunn test to find whether PAL performs significantly better than existing algorithms.

4.2 Experimental Results

In the following subsections, we report the evaluation results of the four performance measures. Tables 3, 4, 5, and 6 report a 95% confidence interval of the mean AUC for each algorithm and dataset combination. The shaded boxes represent cases where the difference between PAL and the corresponding algorithm is statistically significant with 95% confidence. Also, a mean rank and a mean normalized AUC are presented for each algorithm.

4.2.1 Comparing the Training Profit

Consider the three typical training profit graphs in Figure 3[6]. Methods that do not employ simulated annealing (Random, GOAL) have an almost linear behavior: the line either increases linearly, if *o/r* is smaller than customers' positive response rate in the training set, as *Donation* and *Credit*, decreases linearly, if *o/r* is greater than customers' positive response rate, as in *Adult*, or oscillates around the x-axis, if *o/r* is equal to the customers' positive response rate. When 100% of the training data is used, all methods converge to the same training profit because all methods eventually acquire all the examples in the unlabeled pool.

[6] In this section we provide the results summary on all datasets, but detailed graphs are provided for only three selected datasets. A complete and detailed report of the results of the remaining datasets is available from the first author.

Methods that attempt to balance the exploration and exploitation trade-off (PAL, Boltzmann, and Kaelbling's algorithm) display a large unimodal peak and an initial quadratic-like growth. The positive affect of simulated annealing on the training profit is observed until around 50% of the training data is selected. While a relatively accurate classifier can be constructed with 50% of the training data, there are many profitable customers among the remaining 50% of customers.

Table 3 presents the 95% confidence interval of the mean AUC of the training profit graphs. The highlighted values represent cases where the difference between PAL and the corresponding algorithm is statistically significant with 95% confidence. As can be clearly seen from the results, the simulated annealing feature in PAL significantly improves the training profit. The OA method, on the other hand, did not seem to influence the performance of PAL. Using optimistic upper bound undermines performance.

The train profit ranking indicates that PAL is the second best to Boltzman. However, as we will see later, the superiority of the Boltzman approach comes at the expense of its test set positive reaction rate performance. The adjusted Friedman test with a confidence level of 95% rejected the null-hypothesis that all classifiers perform the same. The Bonferroni-Dunn test concluded that PAL significantly outperforms Random, GOAL and Kaelbling at a 95% confidence level. However, we could not conclude that Boltzman significantly outperforms PAL.

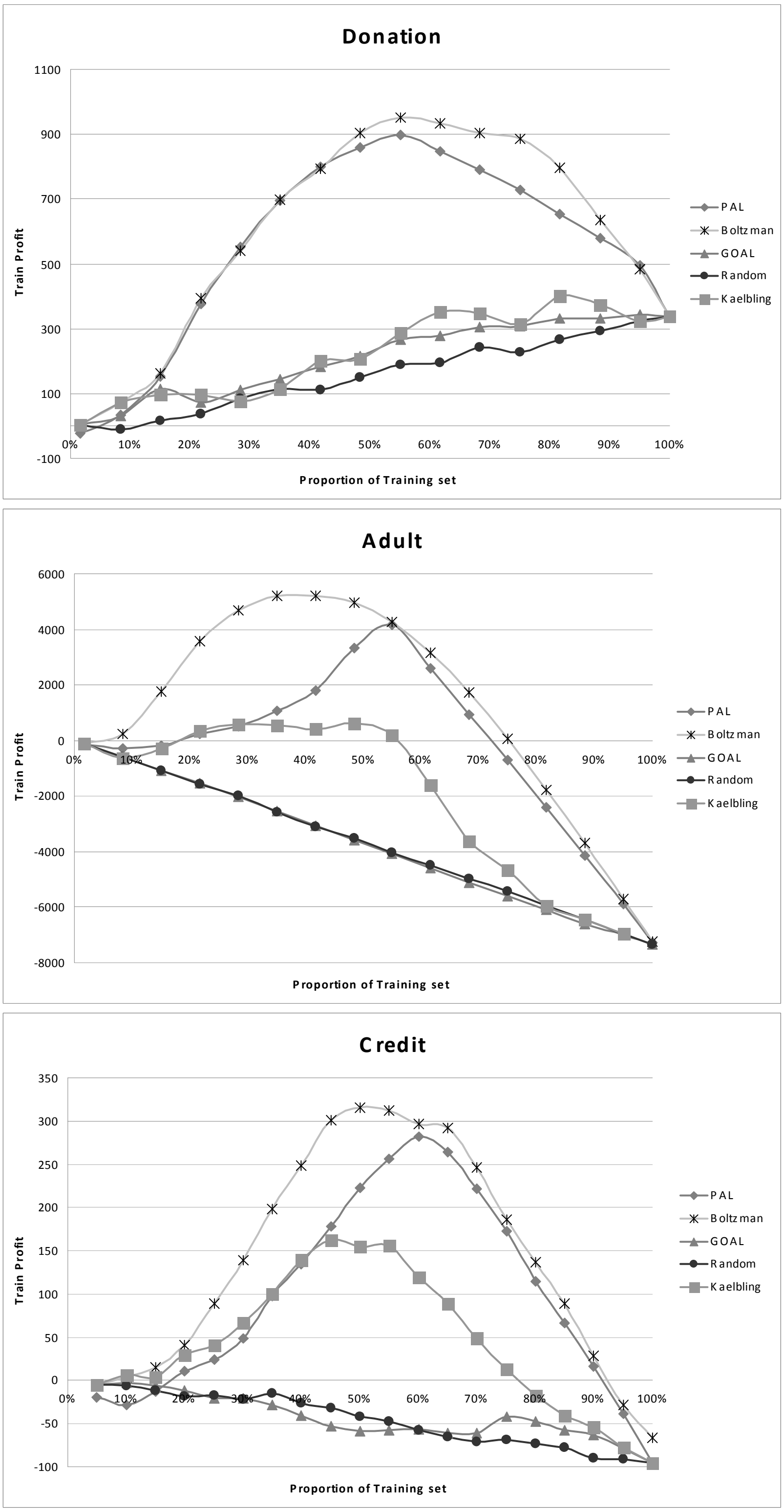

Figure 3: Illustration of Training Profit Graphs

Table 3: 95% confidence interval of the mean AUC of the training profit graphs. Highlighted values indicate that the PAL performance is significantly different from the performance of the corresponding algorithm at a confidence level of 95%.

Dataset	Random	GOAL	Boltzman	Kaelbling	PAL	PAL without Orthogonal Arrays	PAL without Simulated Annealing	PAL without Pessimism	PAL with Optimism
Adult	16015.8±581	16169.17±554	17445 ±103	15857±1025	16850.34±339	16875.66±455	16051.02±787	16425.17±407	15727 ±209
Anneal	-182.1±20	-192.72±26	-67 ±16	-173 ±72	-53.62±2	-58.72±12	-180.92±19	-54.02±3	-82 ±16
Breast C.	51.2±16	156.03±59	278 ±27	87 ±305	201.57±37	209.32±27	24.45±26	270.15±17	196 ±51
Credit	161.24±73	199.09±20	344 ±1	182 ±9	362.26±22	362.44±13	172.61±13	390.71±24	195 ±2
Donation	175.7±69	208.59±77	616 ±81	183 ±141	583.78±95	568.01±94	169.39±98	632.29±88	440 ±80
German	-106.21±28	-105.01±16	4 ±23	-105 ±107	-0.01±20	12.89±36	-118.84±21	14.84±28	-125 ±29
Heart	9.97±9	21.15±23	77 ±10	17 ±58	58.47±9	59.5±13	25.6±22	76.45±9	-8 ±16
Insurance	-94.87±56	-59.03±48	340 ±40	229 ±177	413.31±41	374.03±47	-80.87±57	400.11±47	319 ±29
Mushroom	52.63±6	-1388.28±56	1579 ±8	1564 ±227	1690.63±8	1659.93±78	207.15±5	1712.16±50	557 ±16
Thyroid	31.16±16	16.72±66	635 ±17	308 ±278	678.61±8	668.48±54	10.92±6	682.35±49	191 ±65
Mean Rank	7.4	3.9	5.4	7.3	1.7	3.7	6.8	4.9	3.9
Mean Normalized AUC	45%	79%	42%	31%	96%	84%	55%	64%	75%

4.2.2 Comparing the Test Set Positive Reaction Rate (Profit Margin)

Cost-sensitive active learning methods are typically measured on their test set performance. In this subsection, we examine the positive reaction rate as a function of the percentage of acquired responses from the training pool.

Figure 4 illustrates three typical positive reaction rate graphs. The positive reaction rate increases as more instances become available. Naturally, all methods converge to the same rate when using the entire training set.

Table 4 presents the 95% confidence interval of the mean AUC of the positive reaction rate graphs. The highlighted values represent cases where the difference between PAL and the corresponding algorithm is statistically significant with 95% confidence. As before, adding the simulated annealing feature, which also takes into consideration the fact that customers are acquired in batches, but without pessimism, improves Random's results. Adding pessimism further improves performance. Nevertheless, if we remove OA from PAL, we obtain almost similar results. This implies that the OA method improves the positive reaction rate only slightly, if at all.

PAL obtained the best rank among all algorithms. The adjusted Friedman test with a confidence level of 95% rejected the null-hypothesis that all classifiers perform the same. The Bonferroni-Dunn test concluded that PAL significantly outperforms Random, Kaelbling, Boltzmann and GOAL at a 95% confidence level. Nevertheless, we could not reject the null hypothesis that PAL and PAL without Pessimism perform the same. One might conclude that pessimism does not significantly improve results. However, when we set PAL without Pessimism as the control classifier, the Bonferroni-Dunn test indicates that it does not significantly outperform GOAL. Thus, the pessimism feature is required in order to obtain significant superiority to existing methods.

Recall that our problem is a Multiple Criteria Decision Making, and thus, we are interested in maximizing both the Positive Reaction Rate and the Train Profit which were evaluated in the previous section. Figure 5 presents the Positive Reaction Rate vs. Train Profit. Note that higher values are preferred to lower values in both axes. Random, Kaelbling and GOAL are not on the Pareto Frontier because they are dominated by both PAL and Boltzmann.

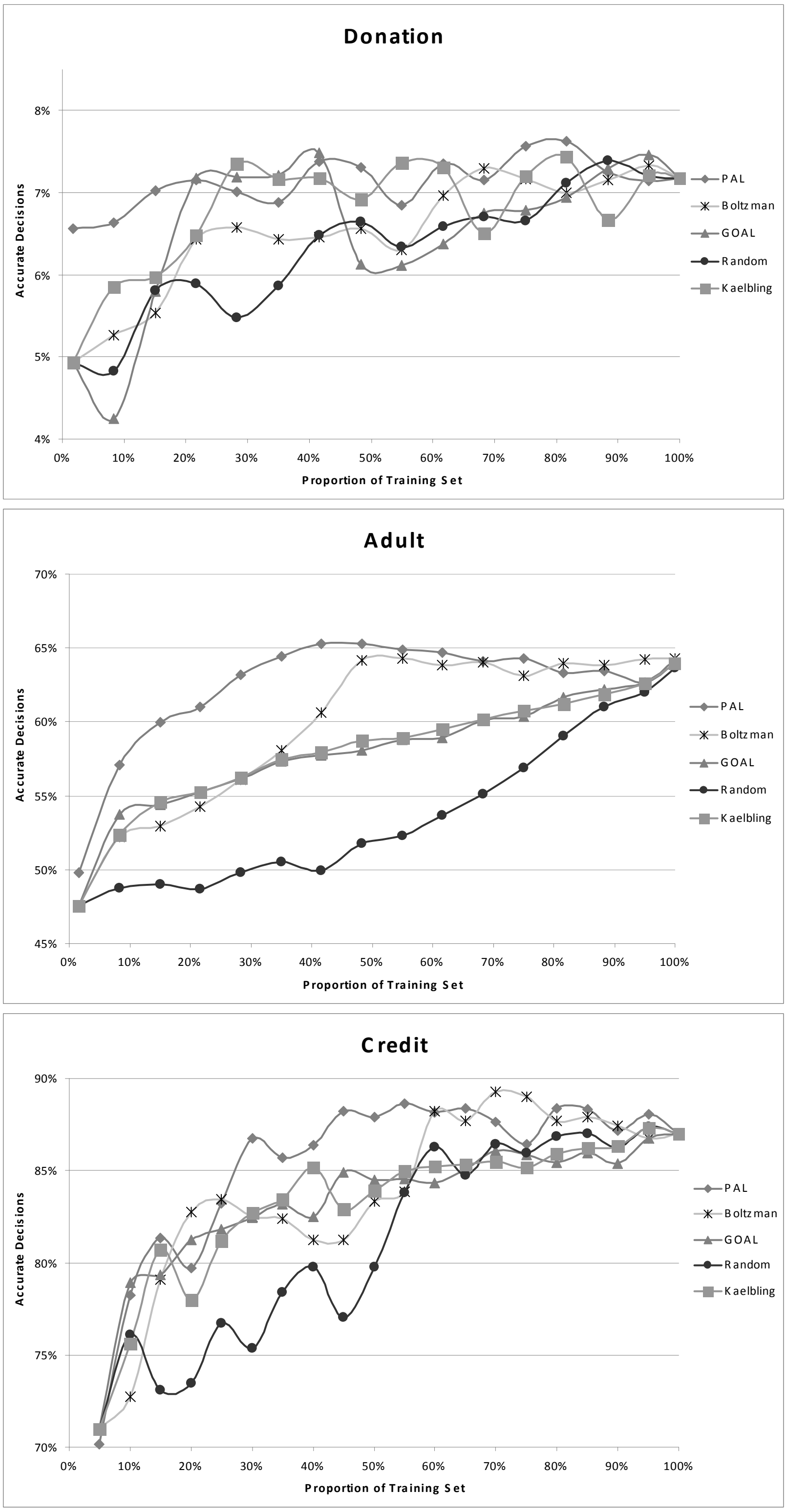

Figure 4: Illustration of Test Set Positive Reaction Rate Graphs

Table 4: 95% confidence interval of the mean AUC of the Test Set Positive reaction Rate Graphs. Highlighted values indicate that the PAL performance is significantly different from the performance of the corresponding algorithm at a confidence level of 95%.

Dataset	**Random**	**GOAL**	**Boltzman**	**Kaelbling**	**PAL**	**PAL without Orthogonal Arrays**	**PAL without Simulated Annealing**	**PAL without Pessimism**	**PAL with Optimism**
Adult	57.11%±0.75	57.85%±0.31	59%±0.92	57%±2.54	60.88%±0.61	60.86%±0.6	57.11%±0.51	59%±0.53	57.6%±0.92
Anneal	77.7%±6.68	76.23%±9.41	80%±5.89	74%±6.12	82.04%±0.74	80.73%±5.79	73.97%±8.53	80.95%±2.02	78%±10.96
Breast C.	76.26%±2.55	74.62%±2.17	77.5%±3.46	75.2%±6.93	82.42%±1.89	80.88%±1.48	76.57%±3.36	79.76%±2.27	69.3%±3.46
Credit	73.63%±6.65	75.69%±2.9	79.8%±2.42	74.2%±3.92	79.09%±2.27	78.49%±3.13	74.9%±2.97	77.16%±12.35	78.7%±2.42
Donation	5.86%±0.44	6.26%±0.87	5.97%±0.58	6.26%±0.23	6.36%±0.33	6.39%±0.35	5.89%±0.43	6.19%±0.33	6.22%±0.35
German	47.03%±2.25	47%±1.81	48%±2.77	46.4%±2.19	48.28%±1.65	48.86%±2.12	45.77%±1.97	46.98%±1.07	46.3%±1.5
Heart	66.29%±10.39	66.82%±5.25	68.8%±2.31	67%±3.46	68.14%±4.78	68.11%±3.19	63.55%±4.17	67.39%±4.17	67.9%±3.46
Insurance	10.32%±0.61	11.27%±0.35	11.3%±0.46	10.8%±0.46	12.26%±0.65	11.19%±0.54	10.65%±0.42	11.25%±0.47	10.7%±0.35
Mushroom	19.14%±0.2	18.54%±0.39	18.6%±0.46	18.2%±2.31	18.56%±0.21	18.55%±0.45	19.3%±0.33	18.79%±0.39	18.4%±0.12
Thyroid	48.69%±2.52	49.19%±1.21	46 ±3.69	48.95%±5.89	51.23%±1.21	50.84%±1.22	48.77%±2.55	50.32%±0.92	48.2%±7.04
Mean Rank	6.8	5.6	3	6.9	1.9	3	7	4.1	6.4
Mean Normalized AUC	33%	45%	69%	34%	78%	73%	30%	56%	40%

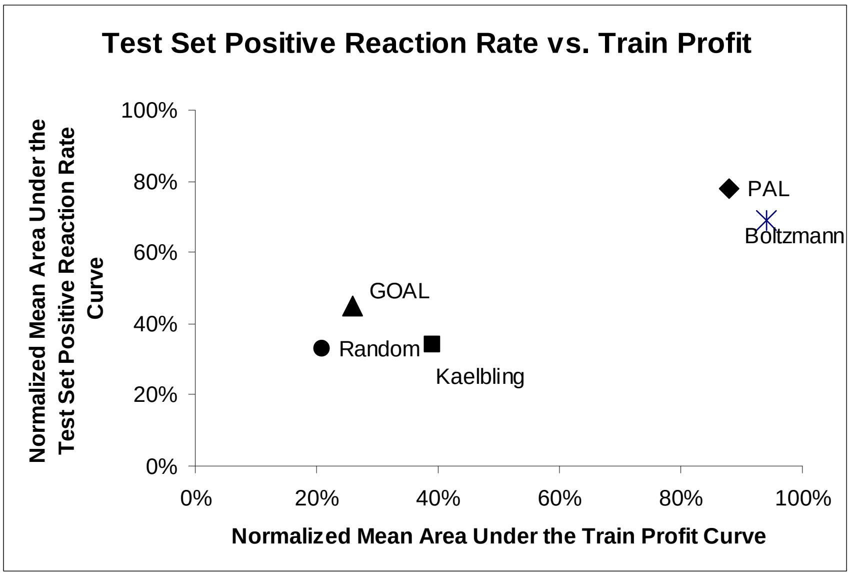

Figure 5: Positive Reaction Rate vs. Train Profit

4.2.3 Comparing the Test Set Profit

In the previous section we evaluated the test set performance by measuring the positive reaction rate. However, a superior response rate might also be obtained at the expense of approaching fewer customers (low recall). In this section, we will examine the actual test set profit.

Table 5 presents the 95% confidence interval of the mean AUC of the test set profit graphs. The highlighted values represent cases where the difference between PAL and the corresponding algorithm is statistically significant with 95% confidence. As expected, the Random method often yields the most inferior results, while PAL and GOAL often yield the best results. The adjusted Friedman test, with a confidence level of 95%, rejected the null-hypothesis that all classifiers perform the same. The Bonferroni-Dunn test concluded that PAL significantly outperforms Random, Boltzman, Kaelbling, PAL without Pessimism and PAL without simulated annealing at a confidence level of 95%. Nevertheless, we could not reject the null hypothesis that PAL and GOAL perform the same at a confidence level of 95%. We conclude that pessimism and simulated annealing significantly improve the test set profit. OA's contribution is not substantial.

Table 5: 95% confidence interval of the mean AUC of the Test Profit Graphs. Highlighted values indicate that the PAL performance is significantly different from the performance of the corresponding algorithm at a confidence level of 95%.

Dataset	**Random**	**GOAL**	**Boltzman**	**Kaelbling**	**PAL**	**PAL without Orthogonal Arrays**	**PAL without Simulated Annealing**	**PAL without Pessimism**	**PAL with Optimism**
Adult	16015.8±581	16169.17±554	16328±234	16564±1063	16850.34±340	16875.66±456	16051.02±787	16425.17±408	16150±327
Anneal	46.7±3	46.87±3	49 ±3	48 ±2	49.73±0	48.78±3	46.4±0	49.83±3	47 ±5
Breast C.	463.97±18	502.78±22	498 ±33	442 ±81	513.73±25	502.46±23	486.8±21	461.38±20	499 ±38
Credit	857.33±62	870.86±35	589 ±2	548 ±3	883.47±30	872.82±35	855.88±32	839.21±131	879 ±2
Donation	7399.45±69	9462.37±80	7980 ±122	7620 ±40	10557.61±54	10477.28±76	7450.01±106	8440.5±88	7720 ±55
German	2720.21±30	2816.37±38	269 ±40	261 ±44	2818.07±26	2675.83±28	2584.71±37	2599.92±34	2819 ±32
Heart	148.61±14	164.81±29	165.1 ±12	157.3 ±15	167.2±18	158.03±12	152.36±14	150.51±14	172.3 ±15
Insurance	404.77±28	424.01±23	431 ±23	373 ±37	457.81±39	415.73±28	428.54±39	417.95±38	423 ±14
Mushroom	1746.77±26	1884.68±68	1803 ±44	1772 ±463	1868.57±26	1858.27±68	1776.7±42	1879.8±55	1871 ±31
Thyroid	486.06±3	489.73±10	341 ±14	470 ±67	494.98±2	494.26±10	483.52±10	492.99±3	492 ±9
Mean Rank	7.4	3.9	5.4	7.3	1.7	3.7	6.8	4.9	3.9
Mean Normalized AUC	45%	79%	42%	31%	96%	84%	55%	64%	75%

4.2.4 Comparing the Gain Charts (Market Share)

In this section we examine a scenario in which the marketing budget is limited and the classifier is used to select a subset of customers. This scenario occurs, for example, when a corporation aims to increase its market share, perhaps at the expense of immediate profitability. Thus, in this scenario we are interested in reaching a pre-specified quota (e.g., 75%) of potential respondents. In these cases it is useful to use a *Gain Chart*. A gain chart presents the cumulative gains (e.g., profitability or response) accrued when using a predictive model versus those obtained via a default approach, which assumes that all customers are identical. The cumulative proportion of the population being targeted, $x_i = 100 \cdot i / n$ (where *n* is the size of the audience, *i* – customer index), is shown on the x-axis. The cumulative positive response rate, $Y_i = 100 * \sum_{j=1}^{i} y_j \Big/ \sum_{j=1}^{n} y_j$, is shown on the y-axis.

Table 6 presents the 95% confidence interval of the mean AUC of the gain charts. The highlighted values represent cases where the difference between PAL and the corresponding algorithm is statistically significant with 95% confidence. The gain chart is calculated when 50% of the training data is selected by each algorithm. As in the previous measures, PAL is the dominant algorithm. The adjusted Friedman test, with a confidence level of 95%, rejected the null-hypothesis that all classifiers perform the same. The Bonferroni-Dunn test concluded that PAL significantly outperforms Random, Boltzman, Kaelbling at a confidence level of 95%. PAL significantly outperforms GOAL at a confidence level of 90%. Finally, as shown, all of PAL's elements contribute to its performance.

4.2.5 Confidence Intervals

The purpose of this subsection is to examine if PAL can provide a tighter confidence interval for risky decisions, i.e., when the mean and the lower bounds are located on opposite sides of the threshold value defined in Eq. (2). The learning process shrinks the confidence interval, and thus, as the learning progresses, less risky decisions should be made. Table 7 presents the percentage of risky decisions after acquiring 50% of the Donation dataset. Note that in this case, lower values are considered better. As can be seen from the table, GOAL, PAL and Kaelbling perform similarly, but better than Random and Boltzmann. A similar behavior has been revealed in all the other datasets.

Table 6: 95% confidence interval of the mean AUC of the Lift Charts. Highlighted values indicate that the PAL performance is significantly different from the performance of the corresponding algorithm at a confidence level of 95%.

Dataset	Random	GOAL	Boltzman	Kaelbling	PAL	PAL without Orthogonal	PAL without Simulated	PAL without Pessimism	PAL with Optimism
Adult	63.33±4.7	78.69±2.4	69 ±9.2	73 ±10.4	79.82±3.7	79.54±5.6	77.62±5.5	79.11±2.9	739.5
Anneal	84±20	85±26	80±1.2	80±1.4	85±2.3	85±11.9	85±19.1	85±2.7	80±0.1
Breast C.	62.85±11.1	66.98±9.6	63.4±3.5	61.8±2.9	70.91±12.7	68.65±13.5	67.48±7.2	67.6±9.6	63.2±0.3
Credit	70.89±1.6	72.72±1.6	72.94±0.5	75.94±0.8	74.06±1.2	73.51±1.2	71.28±0.6	72.86±1.9	72.88±0.5
Donation	50.36±1.2	56.15±1	61±2.2	54±17.3	59.58±1.2	58.78±1	54.88±0.8	59.31±1	51±1.5
German	64.76±2.6	64.5±1.1	66.8±1.7	65.3±1	65.98±1.3	62.67±0.4	58.62±1.1	58.62±1.1	62±1.2
Heart	17.19±3	18.32±2.6	20.2 ±4.6	20.3 ±0.3	19.4±3.5	18.78±3.7	18.46±2	18.5±2.6	17.9 ±0.2
Insurance	61.84±0.1	64.16±0.1	66.2±0.1	60.1±5.9	68.18±0.1	66.2±0.1	63.64±0.1	66.79±0.1	63.5±0.1
Mushroom	85.43±0.7	93.97±0.1	89 ±2.1	86 ±0.5	93.97±0.1	93.98±0.4	93.98±0.1	93.98±0.7	88±0.2
Thyroid	94±1.8	93.87±1.8	96.4 ±0.3	91 ±1.3	93.85±1.8	93.85±1.8	93.85±1.8	93.85±1.8	95.9 ±1
Mean Rank	7.4	4.8	3.9	6.1	2.3	3.2	5.1	3.8	5.9
Mean Normalized AUC	26%	56%	67%	42%	77%	64%	45%	57%	49%

Table 7: Percentage of Risky Decision in Donation Dataset

Random	GOAL	Boltzman	Kaelbling	PAL
19.53%	11.17%	14.33%	8.13%	10.05%

4.2.6 Simulated Annealing

As demonstrated in the previous subsections, some of PAL's capabilities are due to the simulated annealing algorithm (Eq. 11). We examined different values of the simulated annealing decay factor, and found that the best results are obtained when γ is set to a value in the range [0.7,1]; however, the best value depends on the dataset. Figure 6 illustrates the train profit and positive reaction rate as a function of γ in the donation dataset. In this dataset the best value is obtained for $\gamma = 0.88$. Note that for the previous experiments we used a fixed value of $\gamma = 0.85$

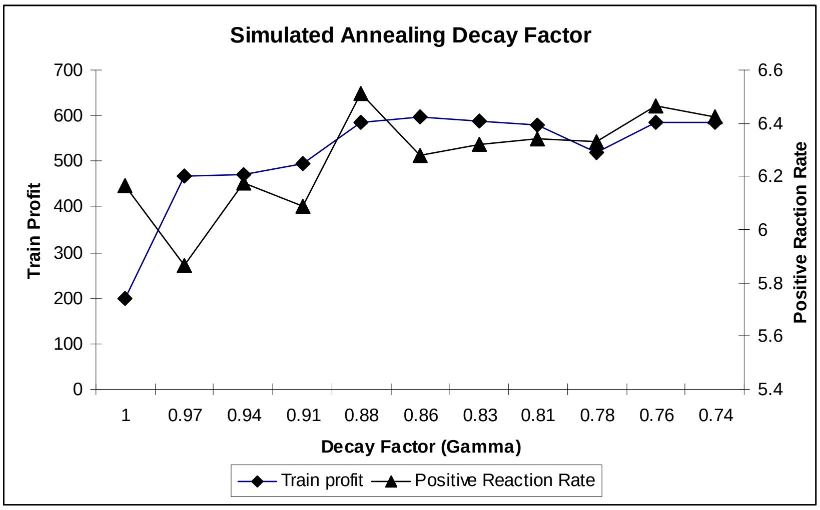

Figure 6: The train profit and positive reaction rate as a function of γ in the donation dataset

4.2.7 The Effect of Batch Size

We explored the effect of the batch size on PAL's performance. We tried using batches which are 1%, 5% or 10% of the total training set. Table 8 presents the mean net profit per approached customer in the donation dataset using different batch sizes. The highlighted values represent cases where the difference between PAL and the corresponding algorithm is statistically significant with 95% confidence. We found that the smaller the batch is, the better the total profit will be. This is expected, since a smaller batch means the classifier is rebuilt more often and is thus finer tuned to the obtained information. Although it gives the best results, in real-world campaign we can not approach the customers one at a time.

Table 8: The effect of the batch size on the mean profit per approached customer in the donation dataset. Highlighted values indicate that the PAL performance is significantly different from the performance of the corresponding algorithm at a confidence level of 95%.

	1%	5%	10%
PAL	0.594	0.558	0.532
GOAL	0.552	0.522	0.468
Random	0.434	0.428	0.424

5. Related Work

In this section we discuss how our method relates to existing work. The underlined problem and the proposed solution share some common properties with Reinforcement Learning and Active Learning. In the following sections, we present these methods and discuss how our proposed solution differs from the existing methods.

5.1 Reinforcement Learning

Reinforcement learning (Kaelbling and Littman, 1996, Sutton and Barto, 1998) is concerned with how an *agent* ought to take *actions* in an *environment* so as to maximize some notion of a long-term *reward*. At each step of the interaction between an agent and its environment, the agent receives as input some indication of the current *state* of the environment $s \in S$. For example, a sales agent with a set of potential customers' receives as input the willingness of each potential customer to buy. The agent then chooses an *action* $a \in A$ from a discrete set of actions to generate as output. The action changes the state of the environment. The value of this state transition is communicated to the agent through a scalar reinforcement signal *r*. For example, a sales agent chooses to make an offer to a potential customer, and the reinforcement signal is the accept/reject decision of the potential customer, or the revenue from its purchase. Reinforcement learning algorithms attempt to find a policy to map states to actions $\pi : S \rightarrow A$ that maximizes some long-run measure of reinforcement, e.g., some kind of discounted revenue stream. We expect the environment to be non-deterministic, taking the same action in the same state on two different occasions. For example, approaching two similar potential customers may result in different next states and/or different reinforcement signals. However, we assume that the environment is stationary, that is, the probabilities of making state transitions or receiving specific reinforcement signals do not change over time. The main difficulty is that the agent is not told if the immediate reward after taking an action means that he took the best action for its long-term interests. For example, high revenue from approaching a particular potential customer does not necessarily mean that approaching the same type of potential customers is always profitable. Pednault et al. (2002) modeled the continuous relationship between a retailer and its customer as a Markov Decision Process. The sequential offers of the retailer can probabilistically alter the state of the customer, and potentially generate a reward to the retailer. They used reinforcement learning to find the policy of the retailer toward its customers.

Reinforcement learning is considered a difficult problem in its most general formulation, and it typically requires thousands of learning steps to find a good policy. A practical difficulty in the context of target marketing is that this approach requires that the past history of purchases and promotions for each customer be available. This is true for some datasets. For example, the *donation* dataset from the KDD Cup 1998 competition contains approximately two years of direct-mail promotional history for each donor. However, in many other datasets, this information is not available.

Alternatively, when no information regarding previous purchases is available, a reinforcement learning problem is considered to be a problem of only one state i.e., there are no state transitions, but multiple actions, such as contacting different prospective customers, can take place, and exploration/exploitation tradeoff exists. An example is the classic k-armed bandit problem (Robbins, 1952).

The problem of choosing which leaf in the decision tree to explore is similar to the k-armed bandit. Each leaf can be treated as a one-armed bandit and the batch size can be referred to as the allowed number of pulls. Still, our problem has some unique properties. First, the classic k-armed bandit does not enforce any restriction on how many times we can pull each arm. It assumes that each arm can be selected as many times as we want as long as the agent does not exceed the fixed number of pulls, *h*. However, in our problem each customer can be selected only once. Thus, the number of times we can select each leaf is bound by the number of customers in the unlabeled training set who correspond to this leaf. On the one hand, this sets a restriction on selecting the next action; on the other, it also implies which arms (leaves) should be explored better. Specifically, we should carefully look into the leaves with which many *unlabeled* instances are associated. Assuming that the unlabeled training set represents the overall distribution D, making wrong decisions about these leaves may result in low quality decisions in the long run exploitation phase for future customers who are currently not included in the training set.

Additionally, the k-armed bandit assumes a purely online environment, namely the algorithm makes a decision and it is told the outcome before it makes the next decision. In a direct marketing campaign, it is more reasonable to assume that we approach a batch of customers at a time, and only then are we told the outcome. Finally the k-armed bandit assumes that there is no cost associated with each pull, other than the opportunity cost from playing a suboptimal machine.

Nevertheless, the resemblance of the k-armed bandit problem to our problem encouraged us to look into known solutions. Kaelbling and Littman (1996) divide the techniques for solving the single-state case such as the k-armed bandit into two types: Formally Justified Techniques and ad-hoc Techniques. In this paper we focus on the ad-hoc Techniques which are considered to be computationally tractable heuristics. Interval estimation techniques are ad-hoc Techniques that use second-order information about the certainty or variance of the estimated values of actions, such as the Kaelbling's interval estimation algorithm. While it is not formally justified, this interval estimation algorithm has been widely used and can be applied to our problem. It stores statistics for each action: the number of successes and the number of trials. For each arm, a confidence interval centred on the sample mean is calculated. The arm whose confidence interval has the highest tail is chosen. Larger intervals encourage greater exploration. Fong (1995) introduced the γ -IE strategy, which is a generalization of Kaelbling's IE (Interval Estimation) Strategy. The parameter γ affects the size of the confidence intervals and often improves the performance of the algorithm.

Model-based Interval Estimation (MBIE) is another learning algorithm that builds a model to construct an exploration policy (Wiering, and Schmidhuber, 1998). More recently, Strehl and Littman (2005) introduced a version of MBIE, which not only combines Interval Estimation with model-based reinforcement learning, but also comes with a formal PAC-like learning-time guarantee. Nevertheless, to the best of our knowledge, no cost-sensitive method considers the effect of the confidence level of the estimated probability in classifier learning problems. Previous research studies, such as Cohn et al. (1996), which tried to minimize the variance, have not been developed in a full cost-sensitive context. Schein (2005) uses A-optimality, a strategy for minimizing the variance to examine which pool-based active learning methods will work well with logistic regression. He considers the labeling cost, but not the utility costs.

There are four issues which differentiate our method from existing interval estimation methods:

1. *Interval Change:* In existing interval estimation methods, instances are selected according to the interval bound value. In this research, we propose to select instances according to the estimated *change* in the interval bound if exploration is indeed made in that interval. The change indicates to what extent the interval has been decreased. Large intervals indicate that additional exploration is required. Thus, the anticipated reduction in the interval size that a certain instance provides is indicative of the explorative contribution of that instance.
2. *Profit Interval:* Existing methods use the probability interval. Here we use the profit interval instead. The profit interval takes into consideration how many unlabeled instances are associated with the interval. Thus, the same probability interval reduction is obtained for different leaves, regardless of the number of instances corresponding to each leaf. This approach is conceivable because change in the probability estimation of a massive interval has an effect on further decisions to be made in the future. Note that our main goal is not to improve the class probability estimations, but to improve the marketing decisions. A trade-off between the two goals might exist (Saar-Tsechansky and Provost, 2007). Assuming fixed distribution of the instances, prioritizing large leaves has a positive effect on larger number of examples in the population.
3. *Using a Lower Bound (Pessimistic) of the Confidence Interval:* Existing interval estimation methods prefer instances with a higher upper bound. The idea is that a higher upper bound indicates both a high success probability, which is good for exploitation, and a wide confidence interval, which is good for exploration. Thus, the fact that we are using the lower bound seems odd. However, we are not using the value itself, but the anticipated change in the value. Recall that we aim to improve the decision-making for risky customers whose estimated probabilities suggest that they are profitable. Thus, selecting customers according to the change in the lower bound does not prioritize profitless customers, but does prioritize customers who are potentially profitable and can add the most benefit by additional exploration.
4. *Using Batches:* Our new measure, as discussed in Section 3.4, takes into consideration the fact that in every iteration, we select a batch of customers to approach. Thus, the explorative contribution of acquiring the first customer in a certain interval is greater than

the contribution of the subsequent customers from the same interval. The reason for this is that the reduction in the interval confidence size is not linear. Considering the normal approximation to the binomial, the confidence interval shrinks at the approximate rate of $\sqrt{\frac{\hat{p}(1-\hat{p})}{m}} \leq \sqrt{\frac{1}{4m}}$, where *m* is the number of acquired customers for a leaf in the decision tree.

Randomized Strategies are another group of ad-hoc techniques that are widely used to trade exploration and exploitation in practice. The idea is to choose the action with the highest estimated expected reward by default, but with probability *p*, to choose an action at random. Some versions of this strategy begin with a large value of *p* to encourage exploration, which is gradually decreased. (Kaelbling and Littman, 1996). In particular, Boltzmann distribution has been frequently used in order to ensure sufficient exploration while still favoring actions with higher value estimates. The Boltzmann policy chooses actions according to a stochastic function of their associated expected rewards. The expected reward from an action is used to choose an action probabilistically according to the Boltzmann distribution. The likelihood of picking an unlabeled customer is exponentially weighted by its utility via the Boltzmann distribution. The ratio between exploration and exploitation is traded dynamically so that exploration fades in time. A "temperature" parameter controls the rate of the convergence. The temperature parameter can be decreased over time to decrease exploration. This strategy may suffer when the values of the actions are close (Kaelbling and Littman, 1996).

In PAL, as well, we dynamically reduce the relative part of the exploration in every new batch. However, we do it directly using γ, and do not consider the utility of the customer as in Boltzmann. In this sense, our approach is simpler.

5.2 Active Learning

Several active learning frameworks are presented in the literature. In pool-based active learning (Lewis and Gale, 1994) the learner has access to a pool of unlabeled data and can request the true class label for a certain number of instances in the pool. Other approaches focus on the expected improvement of class entropy (Roy and McCallum, 2001), or minimizing both labelling and misclassification costs (Margineantu, 2005). Zadrozny (2005) examined a variation in which instead of having the correct label for each training example, there is one possible label (not necessarily the correct one) and the utility associated with that label. Most active learning methods aim to reduce the generalization accuracy of the model learned from the labeled data. They assume uniform error costs, and do not consider benefits that may accrue from correct classifications. They also do not consider the benefits that may be accrued from label acquisition (Hollmén *et al*., 2000, Turney, 2000).

Rather than trying to reduce the error or the costs, Saar-Tsechansky and Provost (2007) introduced the GOAL method that focuses on acquisitions that are more likely to affect decision-making. GOAL acquires instances which are related to decisions for which a relatively small change in the estimation can change the preference order of choice. In each iteration, GOAL selects a batch of instances based on their effectiveness score. The score is inversely proportional to the minimum absolute change in the probability estimation that would result in a decision different from the decision implied by the current estimation. Instead of selecting the instances with the highest scores GOAL uses a sampling distribution in which the selection probability of a certain instance is proportional to its score.

6. Limitations and Conclusions

Target marketing is a multi-million dollar industry. This paper presented the PAL active learning algorithm, which considers the long-term need to increase the positive reactions rate and the short term need to decrease the total net acquisition costs.

The experimental study indicates that PAL can improve the profit of a marketing campaign. PAL achieved the best rank for the Test Set Positive Reaction Rate criterion, the Test Profit criterion, and the *Lift Charts* criterion. It was second only to Boltzmann for the Train Profit criterion. This indicates that PAL provides a good balance of the exploration/exploitation tradeoff.

Closer examination indicates that PAL is particularly advantageous when the estimation confidence intervals are relatively large. Once the decision tree obtains sufficient evidence for each leaf, the relative advantage of PAL is diminished.

Our results suggest that on average, all PAL's elements contribute to its performance. The PAL configurations with simulated annealing demonstrate a significant improvement in the train profit. It seems that of the four key elements of the PAL algorithm, the simulated annealing element was responsible for most of the improvement. This is not surprising, as simulated annealing is an effective, generic method for global optimization.

The second most beneficial element of PAL is the pessimistic learning mechanism. The proposed method is differentiated from existing interval estimation algorithms in that it takes into consideration interval change, as opposed to the actual bounds values. The experimental study shows that this approach is preferable for the problem addressed here.

The OA DoE methodology is another element used in the PAL algorithm. It is a generic element that can easily be integrated into any other active learning algorithm. To the best of our knowledge, this is the first time that the potential contribution of this methodology was measured *explicitly*.

Boltzmann and GOAL's algorithms require that the base classifier be probabilistic, i.e., that given the input attributes, it provides an estimate to the conditional success probability. The Kaelbling algorithm requires an estimate of the confidence interval. This is a drawback, as currently not all classifiers can provide this measure explicitly. The confidence interval can be estimated for certain induction algorithms, such as neural networks (Hwang and Ding, 1997) and Logistic Regression (Sofroniou and Hutcheson, 2002) which are used in marketing problems. Additionally, the PAL algorithm requires an estimate of how a new acquired instance changes the confidence interval of the success probability. This can be done, for example, by calculating the difference between the confidence interval before and after acquiring an instance. However, it might be computationally expensive, since it requires rebuilding or updating the classifier for each candidate instance. Therefore, it is recommended to develop a low-cost approximation for classification trees, as we did in this paper.

Another limitation of the PAL algorithm is the need to fit an orthogonal array to the examined dataset. Selecting an orthogonal array is performed according to the cardinality of the input attribute set. However, the available libraries for orthogonal arrays do not include an array for any cardinality. Therefore, if no suitable array can be found, a larger array must be used and adjusted to the dataset.

Current PAL implementation uses a static batch size. There may be a benefit to using dynamic batch sizes which evolve along the exploration process (Weiss and Tian, 2008). Moreover, we assume that the number of batches bounds the exploration phase. Alternatively, one may consider any other constraint which bounds the exploration phase, such as total gross acquisition cost, total net acquisition cost or time.

While this paper focuses on direct marketing applications, similar settings may arise in other domains, such as medicine (Percus and Percus, 1984; Petkau, 1978), where the pessimism mechanism can be useful. Consider, for example, cancer patients volunteering to participate in a medical experiment, such as the study of the efficiency of a new chemotherapy drug. Only some of the patients would benefit from the experimental drug, while the rest may suffer from severe side effects. Typically, volunteer patients are recruited over time. Therefore, the statistics accumulated from the first volunteers, together with the pessimistic measure can be used as an additional criterion to decide which additional volunteers may be particularly beneficial for the study while reducing the cost.

Acknowledgments

The authors gratefully thank the action editors and the anonymous reviewers whose constructive comments significantly helped in improving the quality and accuracy of this paper.

References

1. Blake, C.L. & Merz, C.J. (1998), UCI Repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science, http://www.ics.uci.edu/~mlearn/MLRepository.html.
2. Buchner A. and Mulvenna M. (1998), Discovering internet marketing intelligence through online analytical web usage mining. SIGMOD Record, 27(4):54-61.
3. Cestnik, B. (1990), Estimating probabilities: A crucial task in machine learning. In Proceedings of the Ninth European Conference on Artificial Intelligence, 147-149.
4. Cohn, D. A., Ghahramani, Z., Jordan, M. I. (1996), Active learning with statistical models. Journal of Artificial Intelligence Research, 4, 129–145.
5. Clarke P. (2006), Christmas gift giving involvement, Journal of Consumer Marketing, Volume 23, Number 5, 2006 , pp. 283-291(9)
6. Demsar J., Statistical Comparisons of Classifiers over Multiple Data Sets, Journal of Machine Learning Research, 7 (2006):1-30.
7. Domingos P. (2005), Mining Social Networks for Viral Marketing (2005). IEEE Intelligent Systems, 20(1): 80-82.
8. Fong, P. W.L. (1995). A Quantitative Study of Hypothesis Selection. Twelfth International Conference on Machine Learning, 226-234, Tahoe City, California.
9. Hedayat A.S, N. J. A. Sloane N.J.A and Stufken J. 1999. Orthogonal Arrays: Theory and Applications. Springer-Verlag, NY.
10. Hwang J. and Ding, A. (1997), Prediction intervals for artificial neural networks, J. of the American Statistical Association, 92: 748-757.
11. Kaelbling L.P. (1993), Learning in Embedded Systems. MIT Press.
12. Kaelbling L.P., Littman M. (1996), Reinforcement Learning: A survey, Journal of Artificial Intelligence Research, 4: 237-285.
13. Kirkpatrick S., Gelatt C.D., Vecchi M.P. (1983). Optimization by Simulated Annealing, Science, Vol. 220(4598): 671-680.
14. Kyriakopoulos K., Moorman C., Tradeoffs in marketing exploitation and exploration strategies: The overlooked role of market orientatio, Intern. J. of Research in Marketing 21 (2004) 219–240
15. Leemis L. M., and Trivedi, K. S., (1996), A comparison of approximate interval estimators for the Bernoulli parameter. The American Statistician, 50: 63-68.

16. Levin N., and Zahavi, J. (2005), “Data Mining for Target Marketing” in The Data Mining and Knowledge Discovery Handbook, (Editors: O. Maimon and L. Rokach), 2005, pp. 1261-1301.
17. Lewis, D., & Gale,W. (1994). A sequential algorithm for training text classifiers. Proceedings of the International ACM-SIGIR Conference on Research and Development in Information Retrieval, pp. 3-12.
18. Ling C. X. and Li C. (1998), Data mining for direct marketing: Problems and solutions. In Proceedings 4th International Conference on Knowledge Discovery in Databases (KDD-98), New York, pp. 73-79.
19. Margineantu D. (2005): Active Cost-Sensitive Learning, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, IJCAI-05.
20. Mayer U. F. and Sarkissian A. (2003), Experimental design for solicitation campaigns, Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 717-722, Washington, D.C., ACM Press, New York, NY, USA.
21. Montgomery, D. C. (1997) Design and Analysis of Experiments, 4th edition, Wiley, New York.
22. Pednault E., Abe N. and Zadrozny B. (2002), "Sequential cost-sensitive decision making with reinforcement learning", Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 259-268.
23. Percus O. E. and Percus J. K., “Modified Bayes technique in sequential clinical trials,” Comput. Biol. Med., vol. 14, pp. 127–134, 1984.
24. Petkau A. J., “Sequential medical trials for comparing an experimental with a standard treatment,” J. Amer. Statist. Assoc., vol. 73, pp. 328–338, 1978.
25. Potharst R., Kaymak U. and Pijls W.(2002), "Neural Networks for Target Selection in Direct Marketing", in: Kate A. Smith and Jatinder N. D. Gupta (eds.), Neural Networks in Business: Techniques and Applications, Idea Group Publishing, pp. 89-110.
26. Putten P. and Someren M. (eds) (2000), CoIL Challenge 2000: The Insurance ` Case. Published by Sentient Machine Research, Amsterdam. Also a Leiden Institute of Advanced Computer Science Technical Report 2000-09. June 22.
27. Quinlan, J. R. (1993), "C4.5: Programs for Machine Learning", Morgan Kaufmann.
28. Robbins, H. (1952). Some aspects of the sequential design of experiments. Bulletin American Mathematical Society, 55:527-535.
29. Rokach L. and Maimon O. (2005), "Top Down Induction of Decision Trees Classifiers: A Survey", IEEE SMC Transactions Part C. 35(4): 476- 487.
30. Rothaermel FT, Deeds DL. Exploration and exploitation alliances in biotechnology: a system of new product development. Strateg Manage J 2004;25(3):201–17.
31. Sofroniou N. and Hutcheson G. D. (2002), Confidence Intervals for the Predictions of Logistic Regression in the Presence and Absence of a Variance-Covariance Matrix, Understanding Statistics, 1(1):3–18.
32. Roy, N. and McCallum, A. (2001). Toward optimal active learning through sampling estimation of error reduction. In Proceedings of the International Conference on Machine Learning.
33. Saar-Tsechansky, M., and Provost, F. (2004). Active sampling for class probability estimation and ranking. Machine Learning 54(2): 153-178.
34. Saar-Tsechansky M. and Provost F. (2007), Decision-Centric Active Learning of Binary-Outcome Models. Information systems research, 18(1): 4-22.

35. Sloane,N.J.A. A library of orthogonal arrays. http://www.research.att.com/~njas/oadir/index.html
36. Strehl A. L., Littman M. L. (2005), A theoretical analysis of Model-Based Interval Estimation, Proceedings of the 22nd international conference on Machine learning, August 07-11, 2005, Bonn, Germany, pp.856-863,
37. Sutton R.S., Barto A.G., (1998). *Reinforcement Learning: An Introduction*. MIT Press. http://www.cs.ualberta.ca/~sutton/book/ebook/the-book.html
38. Turney P. (2000): Types of Cost in Inductive Concept Learning. Workshop on Cost Sensitive Learning at the 17th ICML, Stanford, CA
39. Utgoff P. E. (1989), Incremental Induction of Decision Trees, Machine Learning 4(2):161-186.
40. Viaene S., B. Baesens and B. De Moor and D. Van Den Poel and G. Dedene and J. A. K. Suykens and J. Vanthienen and S. Viaene and T. Van Gestel (2001), Knowledge Discovery using Least Squares Support Vector Machine Classifiers: a Direct Marketing Case, International Journal of Intelligent Systems, 16 (9): 1023-1036.
41. Weiss G., Tian T., Maximizing classifier utility when there are data acquisition and modeling costs, Data Min Knowl Disc (2008)
42. Wiering, M., and Schmidhuber, J. (1998). Efficient model-based exploration. Proceedings of the 5th International Conference on Simulation of Adaptive Behavior , pp. 223–228.
43. Yinghui Y. (2004), New data mining and marketing approaches for customer segmentation and promotion planning on the Internet, Phd Dissertation, University of Pennsylvania, ISBN 0-496-73213-1.
44. Zahavi, J. and N. Levin (1995), Issues and Problems in Applying Neural Computing to Target Marketing" (with N. Levin), Journal of Direct Marketing, 9(1):7-12.
45. Zahavi, J. and N. Levin (1997). Applying neural computing to target marketing, Journal of Direct Marketing, 11 (1): 5-22.
46. Zadrozny B. (2005), One-Benefit Learning: Cost-Sensitive Learning with Restricted Cost Information. In Proceedings of the Workshop on Utility-Based Data Mining at the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.