Control and Cybernetics

vol. **XX** (XXXX) No. X

Active Learning Using Pessimistic Expectation Estimators

by

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

[1] Department of Information System Engineering, Ben-Gurion University of the Negev, P.O.Box 653, Beer-Sheva 84105, Israel
[2] Deutsche Telekom Laboratories at Ben-Gurion University of the Negev,P.O.Box 653, Beer-Sheva 84105, Israel

Abstract: Active learning is the process in which unlabeled instances are dynamically selected for expert labelling, and then a classifier is trained on the labeled data. Active learning is particularly useful when there is a large set of unlabeled instances, and acquiring a label is costly. In business scenarios such as direct marketing, active learning can be used to indicate which customer to approach such that the potential benefit from the approached customer can cover the cost of approach. This paper presents a new algorithm for cost-sensitive active learning using a conditional expectation estimator. The new estimator focuses on acquisitions that are likely to improve the profit. Moreover, we investigate simulated annealing techniques to combine exploration with exploitation in the classifier construction. Using five evaluation metrics, we evaluated the algorithm on four benchmark datasets. The results demonstrate the superiority of the proposed method compared to other algorithms.

Keywords: cost-sensitive learning, active learning, direct marketing, decision trees

1. Introduction

In business scenarios such as direct marketing, it is not well understood which potential customers actually need the product or service and are inclined to purchase it. Data mining methods attempt to acquire knowledge from historical data about previous customers' behaviour to improve both the direct marketing learning rate (e.g., who are the best potential customers), as well as to estimate the probability of a positive response $\hat{p}_i$ from a potential customer x_i. Often, only a part of the data is labeled, i.e., the purchase behaviour is known for a minority of the potential customers, while the rest of the data is unlabeled and only the customers' attributes are known (e.g., demographic attributes such as age and gender). A classifier constructed only from the labeled instances may be used for classifying the rest of the unlabeled instances, i.e., predicting the probability of a potential customer actually buying a certain product or service. Additionally labeled data, generated from approaching potential customers, may be used to improve the quality of the original classifier. *Active learning* (Cohn *et al.*, 1996) is the process in which unlabeled instances (e.g., potential customers) are dynamically selected for expert labelling (e.g., a potential customer is approached in order to obtain his buying response to a marketing offer) and then a classifier is trained on the labeled data. Labelling the data can be costly; therefore the learner can actively choose the *specific* data to be labeled, attempting to reduce the need for large quantities of randomly labeled data. Once the training of the classifier is complete, the best policy is to approach only the potential customers with a predicted response rate above a certain threshold.

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. Tong and Koller (2000) focus on choosing good queries from the pool. Other approaches focus on cost-sensitive active learning and minimizing the misclassification costs (Elkan, 2001), the expected improvement of class entropy (Roy and McCallum, 2001), or minimizing both labelling and misclassification costs (Margineantu, 2005). Weiss and Tian (2006) suggest a method for identifying the optimal training set size for a given dataset based on analysing the effect of costs of acquiring new training examples in classification problems on the overall utility. 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. In general, most active learning methods work on a single *K-by-K loss matrix* (K is the number of classes) where, in the direct marketing scenario, labels may be $\{$*don't buy; buy small basket; buy large basket*$\}$ while the possible actions are $\{$*contact customer; do not contact customer*$\}$. Moreover, in most cases of cost-sensitive applications the diagonal elements in the misclassification loss matrix are usually set to zero, meaning correct classification has no cost, and all other elements are set to positive values, meaning that there are only costs and no profits (Hollmén *et al.*, 2000, Turney, 2000). Rather than trying to reduce the error or the costs, Saar-Tsechansky and Provost (2007) introduced a method that focuses on acquisitions that are more likely to affect decision-making. The loss (profit) function $\lambda(a_i\,|c_j)$ describes the loss incurred by taking action a_i when the state of nature is c_j. More specifically, instead of using misclassification costs, they use Bayesian decision theory framework in which actions other than merely instance labelling are allowed. This results in a more general loss (profit) function than the single *K-by-K* loss matrix.

In the targeted marketing context, an instance $x_i \in X$ is defined as the set of attributes (e.g., age, gender) of a unique potential customer. It is assumed that the records in the dataset are independent and behave according to some fixed and unknown joint probability distribution. For the sake of clarity we will assume a binary outcome for the target attribute y, specifically $y = \{"a", "r"\}$ standing for "accept" and "reject" respectively. The cost of approaching and suggesting a product to the customer x_i is denoted as C_i. If the customer x_i agrees to the offer, then the utility obtained from this customer is denoted as $U_i^a \in \Re$. If the customer rejects the offer, its utility is $U_i^r \in \Re$. Let the corresponding utility of inaction be Ψ_i. The notation $\hat{p}_i$ represents the estimate for p_i, the probability that customer x_i will respond positively to the proposal if approached. Note that all utility values are a function of the customer's attribute vector x_i. The targeted marketing problem is to select the best sequence of potential customers $\{i_1, i_2 \ldots i_n\}$ from the set of all potential customers that will be approached, such that the expected profit will be maximized.

In order to maximize the expected profit, the decision maker should approach customer x_i if the probability of a positive response is bigger than the costs of approach (Saar-Tsechansky and Provost, 2007). This is represented in the following equivalent equations:

$$\hat{p}_i \cdot U_i^a + (1 - \hat{p}_i) \cdot U_i^r - C_i > \Psi_i \tag{1}$$

$$\hat{p}_i > \frac{C_i + \Psi_i - U_i^r}{U_i^a - U_i^r} \equiv \frac{o_i}{r_i} \tag{2}$$

where o_i and r_i are merely shorthand for the numerator and denominator of the decision threshold ratio in Eq. 1. The classifier will be used to estimate $\hat{p}_i$.

In this paper we present a new active learning framework for the discrete choice targeted marketing problem: Active Cost sensitive learning with decision Trees (ACT). Specifically, the investigated problem is concerned with the decision as to which potential customer x_i we should approach with a new product offer. The decision is made according to the customer's own characteristics and the past history of purchasing by previously approached potential customers. While active learning strictly addresses improved exploration of the dataset, ACT selects the next customer (or batch of customers) to be approached by the marketing campaign considering the costs/profits of the exploration/exploitation tradeoff *during* the learning process. For this purpose we suggest measuring the utility using a new pessimistic approach. There are three contributions in ACT:

1. Pessimistic expectation: ACT uses a pessimistic expectation estimator for selecting the consequent data.
2. Working with batches: The training dataset is divided into a set of equal partitions (batches). We develop an approximation method to estimate the potential contribution of the n-th customer in the batch.
3. Exploration-exploitation trade-off: ACT balances the models needed to explore the data on one hand and to exploit the data on the other hand, using *simulated annealing* (Kirkpatrick *et al.*, 1983). Unlike most cost-sensitive active learning methods that try to optimize some testing set

measures (e.g., profit), in this study we are also interested in the training performance (i.e., profit or loss) *during* the training phase. Thus, there is no clear division between the training phase and the execution (validation) phase.

Here we use the *Decision Tree* induction for the classifier (Quinlan, 1993). Decision Trees are considered to be self-explanatory models and easy to follow when compacted (Rokach and Maimon, 2005). Pessimistic measures were used before for pruning decision trees (Rokach and Maimon, 2008). The proposed principles of ACT can be adjusted to other induction methods such as neural networks.

This paper extends the initial results of Rokach *et al.* (2007) with an expanded description of the algorithm and extensive experimentation of ACT on more datasets and evaluation metrics.

The rest of this paper is organized as follows: Section 2 presents the components of the new active learning algorithm for decision trees. Section 3 reports the experiments carried out on benchmark datasets. Finally, Section 4 concludes the work with a discussion.

2. The ACT Algorithm

A typical marketing database contains a huge dataset, with information on the company's potential customers. It can be expensive to label the data (e.g., we need to approach the potential customer and propose the new product to her). Starting with a small set of labeled examples, we search the unlabeled database for customers who may provide useful information for creating an accurate classifier. Once a customer is chosen, we approach her and propose the new product. According to the customer response, the newly labeled example is then put into the labeled pool. The learner trains on the labeled pool and outputs a classifier. Based upon the classifier, we search the unlabeled database, and repeat this process until triggering a kind of stopping criteria (e.g., running out of budget). Then, the final classifier is used to classify the rest of the potential customers.

If the classifier is a decision tree, then for estimating the probability p_i one should first locate the appropriate leaf k in the tree that refers to the given instance x_i. Following that, one should extract the frequency vector (how many instances relate to each possible value of the target feature). In the usual case of target marketing the frequency vector has the form: $(m_{k,a}, m_{k,r})$ where $m_{k,c}$ representing the number of instances in the training set that reach leaf k and are classified as "accept" or "reject" respectively. According to Laplace's law of succession, the probability p_i is estimated as:

$$\hat{p}_i = \frac{m_{k,a} + 1}{m_{k,a} + m_{k,r} + 2} \tag{3}$$

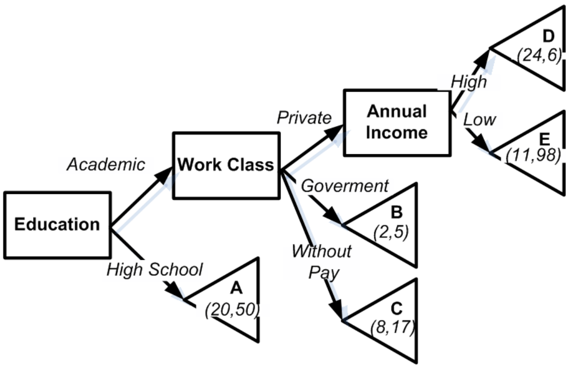

Figure 1. Decision Tree for Target Marketing

Besides estimating the point probability $\hat{p}_i$, we are interested in estimating the standard deviation $\hat{\sigma}_i$ for this probability. An approach to a customer can be considered as a Bernoulli trial. For the sake of simplicity, we approximate the standard deviation of the Bernoulli parameter with the normal approximation (see for instance Brown *et al.*, 2001):

$$\hat{\sigma}_i = \sqrt{\frac{\hat{p}_i(1-\hat{p}_i)}{m_{k,a}+m_{k,r}}} \tag{4}$$

To illustrate the importance of the standard deviation consider the simple decision tree classifier presented in Figure 1. Figure 1 demonstrates a simple decision tree with three input features: "*Education*", "*Work class*", and "*Annual income*". Each leaf display a vector indicating the number of customers in the training set that fit this path. Each customer is labeled as either "*accept*," indicating he accepted the proposed marketing offer, or "*reject*" indicating the opposite. For instance, there are twenty customers in the training set who have high school education and are classified as "*accept*" (leaf A). Note that in this decision tree both leaf A and leaf B have the same estimated probability of 0.4 for the "*accept*" class (for the moment ignoring the Laplace correction).

The potential contribution of having an additional instance for the second path (leaf B) is greater than that of having an additional instance for the first path (leaf A), because in the former case the additional labeling is crucial in order to clarify the actual value of estimated probability (i.e., shrinking the standard deviation). Moreover, the potential contribution of labeling the i-th instance in the same path and adding it to the training set decreases in i. Namely, the contribution of adding i instances to a certain path is lower than i times the contribution of adding the first instance to that path. Thus, the calculation of the potential contribution of each instance in the new selected batch depends on the other instances that are allotted to this batch.

In the following sub-sections we present the elements of a new cost-sensitive active learning algorithm – ACT: (i) a pessimistic profit estimator for selecting the consequent potential customer, or (ii) batch of potential customers, (iii) taking into consideration the tradeoff between exploration vs. exploitation.

2.1. Profit Evaluation using Pessimistic Expectation

In this sub-section we suggest a method for pessimistic evaluation of the profit. Suppose we approach some new potentially profitable customers whose features correspond with a specific leaf k in the decision tree – m_{new}. Following Eq. 1, the pessimistic probability that a single new potentially profitable customer will buy is:

$$\tilde{p} = \int_{-\infty}^{\infty} xf(x|x < \tfrac{o}{r})dx = \int_{-\infty}^{\frac{o}{r}} xf\left(x\right)dx \tag{5}$$

where $f\left(x\right)$ is the (unknown) true probability density function. We integrate the expected profit with respect to the condition that the decision is incorrect (i.e., the success probability is less than the decision threshold). The expected pessimistic profit (PP) from the new customers is:

$$PP = m_{new}(r \cdot \tilde{p} - o) \tag{6}$$

More specifically, the pessimistic profit is defined as follows:

$$PP = m_{new}(r \int_{-\infty}^{\frac{o}{r}} xf\left(x\right)dx - o) \tag{7}$$

For the normal approximation to the distribution $f\left(x\right)$ with the frequency vector $(m_{k,a}, m_{k,r})$ we can solve for the following analytic solution:

$$PP = m_{new}(r \int_{-\infty}^{\frac{o}{r}} \frac{xe^{\frac{-(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}} dx - o) = m_{new}(r\phi(\frac{\frac{o}{r} - \mu}{\sigma})(\mu - \tfrac{1}{2}) - o) \tag{8}$$

where μ, σ are replaced with their estimates Eq. 3 and Eq. 4 respectively and ϕ the cumulative normal distribution function.

To illustrate, we compute the expected pessimistic profit for the customers that belong to leaf E of the decision tree presented in Figure 1. This leaf represents 11 customers who previously agreed to the purchase proposal and 98 others who refused. For the sake of simplicity we assume that $o_i \equiv o = 1$ and that $r_i \equiv r = 10$. We also assume that there are an additional $m_{new} =$

1000 unlabeled customers who belong to the case represented by leaf E. $\hat{\mu} = \frac{11+1}{11+98+2} = 0.108108$ and $\hat{\sigma} = 0.029742$. Computing Eq. 8:

$$\begin{aligned} PP(m_{k,a}, m_{k,r}, m_{new}) &= 1000 \cdot 10 \cdot \phi(-0.272) \cdot (0.108 - 0.5) - 1000 \cdot 1 \\ &= -2538.48 \end{aligned}$$

The pessimistic profit after approaching customer x_i is weighted according to the estimated probability $\hat{p}_i$. There are two possible outcomes: If the customer buys the product, $m_{k,a}$ is increased by 1. If the customer does not buy the product, $m_{k,r}$ is increased by 1. In both cases $\hat{p}_i$ and $\tilde{p}_i$ are updated and m_{new} decreases by 1. The pessimistic *profit gain* (PPG) is the *difference* between the estimated pessimistic profit *before* and *after* approaching a customer. The leaf selection rule for the decision tree is to approach only customers from the leaf(s) with the highest PPG.

Consider the previous example. If we decide to approach one of the 1000 unlabeled customers (recall that for the sake of simplicity we ignore the additional branches in the decision tree), then the pessimistic profit can be one of the two options:

1. The customer buys the new product. The success ("accept") probability is updated to $\hat{\mu} = \frac{12+1}{12+99+2} = 0.116$ and $\hat{\sigma} = 0.0305$. Thus, using Eq. 8, the new pessimistic profit is (now that only 999 new customers are left): PP=-2147.18
2. The customer does not buy the new product. The success ("accept") probability is updated to $\hat{\mu} = \frac{11+1}{11+99+2} = 0.107$. Note that the decision rule to approach the customer has not been changed. Thus, the new pessimistic profit is: PP=-2585.76

Because we cannot predict the actual response of the customer, we weigh the above pessimistic profits according to the estimated probability and obtain: $-2147.18 \cdot 0.108108 - 2585.76 \cdot 0.891892 = -2538.26$. Thus, the pessimistic profit gain for approaching this customer is: $PPG_1 = -2538.26 - (-2538.48) = 0.2$.

Eq. 8 is used when the estimated success probability is greater than the threshold. In the case that this condition is not met, we use the following *optimistic loss* measure:

$$\begin{aligned} OL &= m_{new}(r \int_{-\infty}^{\infty} xf(x|x > \frac{o}{r})dx - o) \\ &= m_{new}(r\phi(\frac{\frac{o}{r} - \mu}{\sigma}) \cdot \frac{\sigma}{1 - \phi(\frac{\frac{o}{r} - \mu}{\sigma})} + r\mu - o) \end{aligned} \tag{9}$$

2.2. The Pessimistic Profit Gain for the n-th Customer

The previous section presented a method for calculating the pessimistic profit gain under two assumptions: (i) Customers are approached one at a time, and (ii) we wait for the response of one customer before approaching the next one. However, this situation is typically not the case in many targeted marketing applications, as multiple customers are contacted simultaneously by the different salespersons. Therefore, the targeting policy should be refined to approach a quota of customers simultaneously. The pessimistic profit for the first n customers of a certain node k is:

$$\begin{aligned} PPG_n\left(m_{k,a}, m_{k,r}, m_{new}\right) = \sum_{j=0}^{n} \left(\begin{array}{c} n \\ j \end{array} \right) p^j\left(m_{k,a}, m_{k,r}\right) \\ \cdot \left(1 - p(m_{k,a}, m_{k,r})\right)^{n-j} \cdot PP(m_{k,a} + j, m_{k,r} + n - j, m_{new} - n) \end{aligned} \quad (10)$$

Note that by setting $n = 0$ in Eq. 10 we obtain the current profit (before approaching any customer). Moreover, for the sake of simplicity Eq. 10 refers only to pessimistic profit. However, by introducing an appropriate indicator function, Eq. 10 can easily be generalized to cover the optimistic loss as well.

Following Eq. 10, we can define the gain obtained from selecting an additional n-th customer from node k:

$$\begin{aligned} G_n\left(m_{k,a}, m_{k,r}, m_{new}\right) = PPG_n\left(m_{k,a}, m_{k,r}, m_{new}\right) \\ - PPG_{n-1}\left(m_{k,a}, m_{k,r}, m_{new}\right) \end{aligned} \quad (11)$$

2.3. Next Batch Selection

Simulated annealing (Kirkpatrick *et al.*, 1983) is a generic probabilistic meta-algorithm for global optimization problems. Its key idea by default is to exploit, meaning, 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. In the context of ACT, each consecutive batch j, (of size N) is composed of the following proportion of randomly selected data instances:

$$T_{j+1} = 0.1^{\frac{\gamma \cdot j}{k}} \quad (12)$$

where j is the batch number, k is the number of batches, and γ is a positive constant. The remaining instances in the batch are selected using the pessimistic profit gain model. The exploration rate is decreased as T decreases: in the empirical study we used $\gamma = 2$, and the smallest k was 20. Therefore, in the second batch ($j = 1$) 79% of the instances were randomly selected, while in the last batch only 1% of the instances were randomly selected. Too small T values may result in inaccurate probability estimations. As T becomes smaller and the best instances are already exploited, the forthcoming batches will contain more instances located near the *border* $(r \cdot \tilde{p} - o)$ of the decision region. These points may have a great impact (as measured by how many unlabeled instances are in the same decision region) on the total profit. Note, however, that our main goal is not to improve the class probability estimations, but to improve the marketing decisions. A tradeoff between these two goals might exist (Saar-Tsechansky and Provost, 2007).

3. Experimental Study

The purpose of this section is to present the numerical experiments on a set of benchmark datasets that evaluate the efficiency of the ACT algorithm and each one of its components presented in the previous section. Thereafter, we compare the performance of ACT and the performance of (i) *ACT w/o P* – ACT without the pessimistic profit calculation (ii) *ACT w/o S* – ACT without the simulated annealing for explore/exploit control. Furthermore, we compare the performance of ACT to the performance of *random* instance selection and the performance of the *GOAL* algorithm (Saar-Tsechansky and Provost, 2007).

3.1. Experiment Setup

The algorithms were evaluated on four benchmark datasets. Each dataset was divided into a training set and a test set. Details can be found in the appendix. For *Donation*, *Adult* and *Insurance*, 60 equal-size batches were used. For the smaller *Credit*, 20 batches were used. In all cases we employed the C4.5 induction algorithm (Quinlan, 1993) with the unpruned option which enabled us to construct the decision tree. The Laplace correction Eq. 3 is used in order to estimate the success probability. The same 10 different randomizations of the training set were used to measure the generalized performance and compare the algorithm variations.

Note that in any real world application, the actual values of the cost and the revenue – o_i and r_i defined in Eq. 1 – should be estimated from the specific application. For the *Donation* data the solicitation cost is given and the positive response utility can be predicted (see for instance Saar-Tsechansky and Provost, 2007 for a detailed description on how these values can be appropriately estimated). For the other datasets we had to fabricate the values considering these 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 customer‘ positive response rate, the risk of loss becomes too high, and risky scenarios are unacceptable in most business applications. Therefore – avoiding risky scenarios – the maximum potential contribution (in percent) 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 around the customers' positive response rate.

3.2. Experimental Results

Our evaluation consists of five metrics, each showing a different possible scenario:

1. *Testing set profit*: The profit yielded by the algorithm using the testing set. That is the standard metric upon which cost-sensitive active learning methods are measured.
2. *Training set profit*: In a scenario where the marketing campaign is continuous, there is no clear division between the training phase and the test phase. We are interested in the profit achieved in the training phase as well
3. *Precision*: The accurate decision rate as a function of the percentage of acquired responses from the training pool. Precision is used to assess the profitability in the testing pool. Higher rates indicate higher gross profit margins and return of investment (ROI). In a scenario where the campaign is trying to improve efficiency, the ROI needs to be assessed.
4. *Gain Charts*: A scenario where the marketing budget is limited and the classifier is required to select a few top customers, and to approach only them. This is different from the rest of the scenarios, since in the other cases the classifier can approach any customer he predicts as profitable, while here the classifier is limited in the number of customers it can approach. Even if the classifier predicts more customers as profitable, it can't approach them due to budget restrictions.
5. *Campaign profit*: This metric simulates a real world situation where the campaign does not have separate data for the training phase and for the testing phase. The campaign starts directly from a single pool of data. In addition to measuring the profit, we can determine where the training phase and the campaign should end.

Since 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 possible dominance relationship among them. The reported values represent the mean AUC performance over ten random partitions of the data. All algorithms start at the same point, and converge at the end to the same point, so the AUC indicates differences only in the middle part of the algorithms. Using the AUC measure attenuates to some extent the differences between the algorithms, since this difference can only be seen in the middle part. Nevertheless, we found the AUC measure satisfactory for demonstrating the superiority of ACT.

In order to conclude which algorithm is superior from the ten different randomizations of each of the four datasets, we followed the robust non-parametric procedure that was proposed by Demsar (2006): First, we applied the adjusted

Friedman test in order to reject the null hypothesis (that neither algorithm is superior), then we applied the Bonferroni-Dunn test to examine if ACT performs significantly better than existing classifiers. We also computed the mean rank of each algorithm (over the four datasets), and the normalized mean. The statistical significance of the differences in performance between the ACT algorithm and the other algorithms was verified with the one-tailed paired t-test (pairing the ten randomizations of each dataset) with a confidence level of 90%.

We computed the Mean Rank of each algorithm (e.g., if an algorithm ranks 1st, 2nd, 1st, 3rd on the four datasets, respectively, then its Mean Rank is 7/4=1.75. We also computed the mean normalized performance, i.e., if the normalized performance of algorithm i on dataset j is defined as:

$$NAUC_{i,j} = \frac{AUC_{i,j} - \min_k AUC_{k,j}}{\max_k AUC_{k,j} - \min_k AUC_{k,j}} \tag{13}$$

The mean normalized performance of algorithm i is:

$$MNAUC_i = \sum_{j=1}^{n} \frac{NAUC_{i,j}}{n}. \tag{14}$$

We hereby present some tables for the AUC of the evaluation metric and some graphs for performance measures of the algorithm for the *Donation* dataset. The full set of graphs and tables is given elsewhere (Naamani, 2008).

3.2.1. Evaluation metric #1: Testing-set profit

Table 1 presents the AUC for the testing set profit. The adjusted Friedman test rejected the null hypothesis that all algorithms perform the same with a confidence level of 90%. The * sign in the boxes represents cases in which ACT is *significantly better* using the one tailed t-test with a confidence level of 90%. The # sign represents cases where the algorithm in question is better than ACT with a confidence level of 90% [1]. ACT is seen to be superior using ranking and normalized ranking. Using Eq. 13 indicates that pessimism alone contributes about 75% to the improvement of ACT, while simulated annealing alone contributes 62% [2].

3.2.2. Evaluation metric #2: Training-set profit

Training set profit graphs are presented in Figures 2 and 3. The profit increases as more instances become available. Naturally, when using the entire training set all the algorithms converge to the same profit. Algorithms that do not

[1] The high variance is caused by the ten-folds-cross validation procedure, yet the difference between the methods is statistically significant

[2] The two measures are partially correlated, therefore the sum of the individual components is over 100%

Table 1. Testing-set profit (AUC)

Dataset	ACT	GOAL	ACTw/oP	ACTw/oS	Random
Adult	16824±180	15999±160*	15450±203*	16185±152*	16011±166*
Credit	419.1±10.6	424.2±6.8#	410.4±16.3*	417.4±14.7*	423.8±8.1#
Donation	8647±64	7532±70*	7745±55*	7072±42*	7307±52*
Insurance	422.8±20.8	424.0±20.2	406.5±30.7*	404.8±27.7*	403.4±34.4*
Mean Rank	1.75	2.25	3.75	3.75	3.5
Mean Perf.	90%	67%	15%	28%	38%

employ simulated annealing (GOAL, random, ACT w/o S) have an almost linear behavior. Algorithms that employ simulated annealing (ACT, ACT w/o T, ACT w/o P) display a large unimodal peak, and an initial quadratic-like growth (e.g., up to about 30%). The positive affect of simulated annealing on the training profit (i.e., the maximum training profit) is observed until around 50% of the training data is selected. This phenomenon can be explained by the fact that a rather good classifier can be constructed with 50% of the training data, Yet, the remaining set of un-approached customers still contains many profitable customers.

The adjusted Friedman test rejected the null hypothesis that all algorithms perform the same with a confidence level of 90%. The Bonferroni-Dunn test concluded that ACT significantly outperforms Random and ACT w/o P at 90% confidence level.

As can be seen in Table 2, ACT is significantly better than GOAL and random for all datasets. Pessimism contributes 100% to the improvement of ACT, while simulated annealing contributes 78%.

Table 2. Training-set profit (AUC)

Dataset	ACT	GOAL	ACTw/oP	ACTw/oS	Random
Adult	477±174	-3697±68*	-4774±397*	-3695±76*	-3645±138*
Credit	125±24.8	-40±23*	-74±23*	-72±18*	-44±14*
Donation	454±68	208±67*	56±78*	198±57*	157±111*
Insurance	277±46	-59±41*	-285±63*	-105±53*	-107±46*
Mean Rank	1	2.5	5	3.25	3.25
Mean Perf.	100%	29%	0%	22%	23%

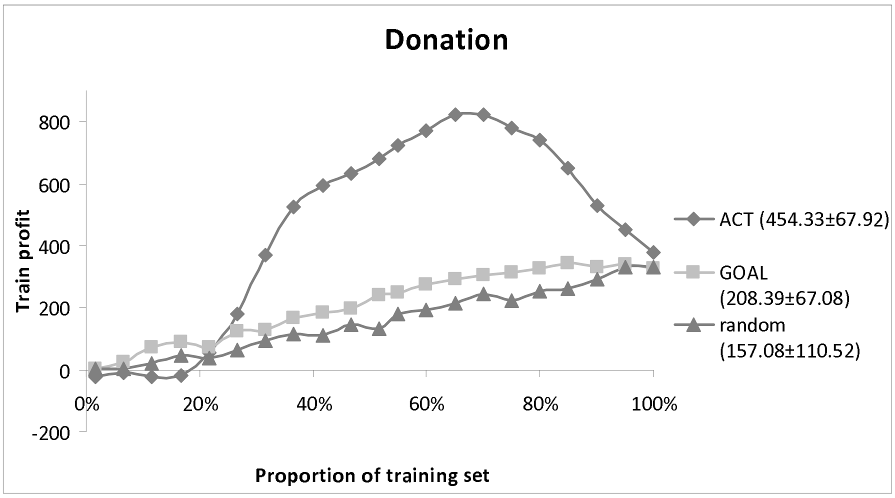

Figure 2. The training-set performance of ACT vs. GOAL and Random on the Donation dataset

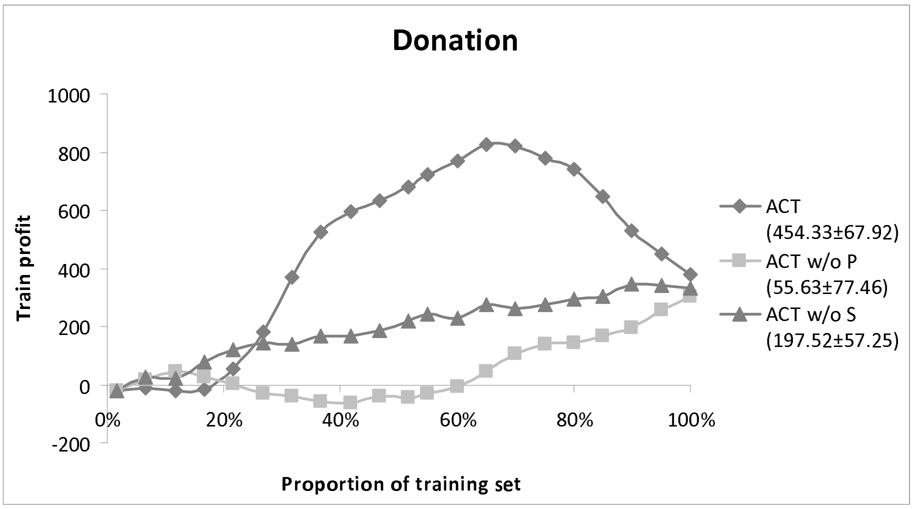

Figure 3. The training-set performance of ACT vs. components on the Donation dataset

3.2.3. Evaluation metric #3: Precision

The adjusted Friedman test rejected the null hypothesis that all algorithms perform the same with a confidence level of 90%. The Bonferroni-Dunn test concluded that ACT significantly outperforms only ACT w/o P at 90% confidence level. As presented in Table 3, the one tailed t-test shows that ACT is significantly better than GOAL and random for all datasets but one. Pessimism contributes about 76% to the improvement of ACT, while simulated annealing contributes about 45%.

Table 3. Precision (AUC)

Dataset	ACT	GOAL	ACTw/oP	ACTw/oS	Random
Adult	60.21±0.64	57.09±0.27*	54.84±0.54*	57.3±0.43*	57.12±0.65*
Credit	86.09±1.06	86.59±0.68	85.22±1.63*	85.92±1.47	86.55±0.81
Donation	6.41±0.48	6.14±0.75*	6.13±0.26*	6.17±0.52*	5.73±0.44*
Insurance	11.16±0.41	10.57±0.3*	10.27±0.3*	10.46±0.31*	10.33±0.41*
Mean Rank	1.5	2.5	4.75	2.75	3.5
Mean Perf.	91%	59%	15%	46%	36%

3.3. Evaluation metric #4: Gain charts

We investigate a scenario where the marketing budget is limited, we do not use all the training data and we approach only the top customers (those with the highest probability of a positive response). Figure 4 demonstrates the gain of the ACT algorithm: the top 20% of the customers generate almost 40% of the positive responses – a more than 10% improvement over the other two algorithms.

In Table 4 we present the AUC for the situations where only 50% of the training data is used, in order to approach the best 10% of the customers in the testing data. We took the middle batch of each experiment run - batch #30 for donation, adult and insurance datasets, and batch #10 for credit dataset. We then looked at the top 10% of customers approached. The adjusted Friedman test rejected the null hypothesis that all algorithms perform the same with a confidence level of 90%. However, the Bonferroni-Dunn test did not differ between the algorithms. Table 4 present cases in which ACT is significantly better using the one tailed t-test. ACT is significantly better than GOAL and random for all datasets. Pessimism contributes about 59% to the improvement of ACT, while simulated annealing contributes about 71%.

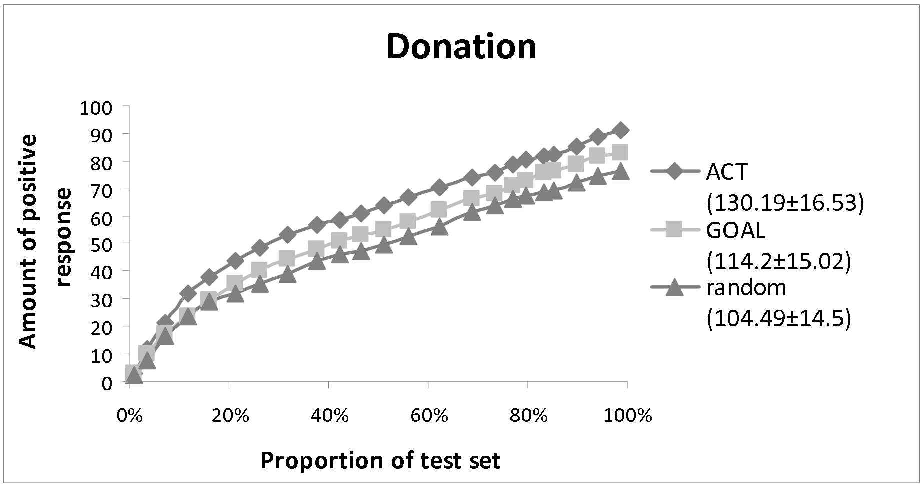

Figure 4. Gain chart for donation dataset, ACT vs. GOAL and random

3.4. Evaluation metric #5: Continuous profit

In the previous metrics we have demonstrated ACT's superiority by training it on the testing set and then testing it. Here, we do not separate the training phase from the testing phase. We show ACT running incrementally on all the available data without stopping the learning. As can be seen in Figures 5, 6, 7, and 8 ACT demonstrates a peak training profit at around 50%-60% of the dataset. This would suggest stopping the direct marketing campaign when the profit from each additional batch stops increasing. The curves start with a zero or even negative profit (e.g., Fig. 7, *adult*) initially (the initial 10%-25%) before the profits starts accumulating at a quadratic or linear rate. The initial flat section indicates that the prediction model is not yet effective and further learning is needed.

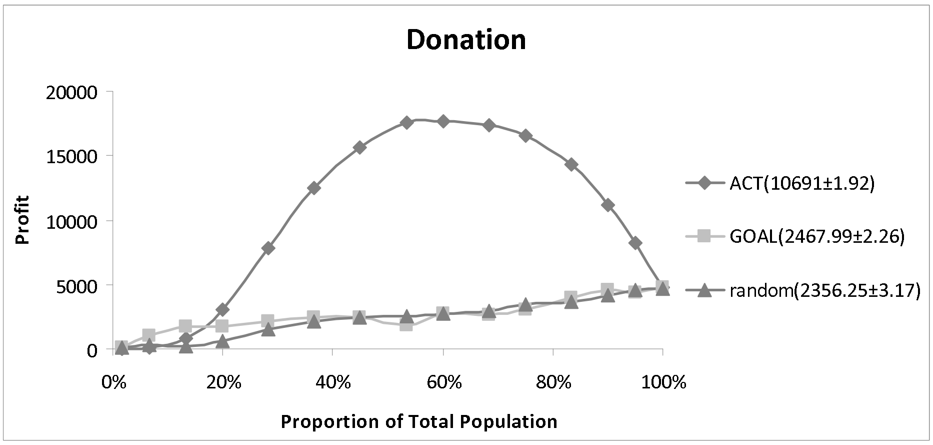

Figure 5. Donation Continuous Train Profit

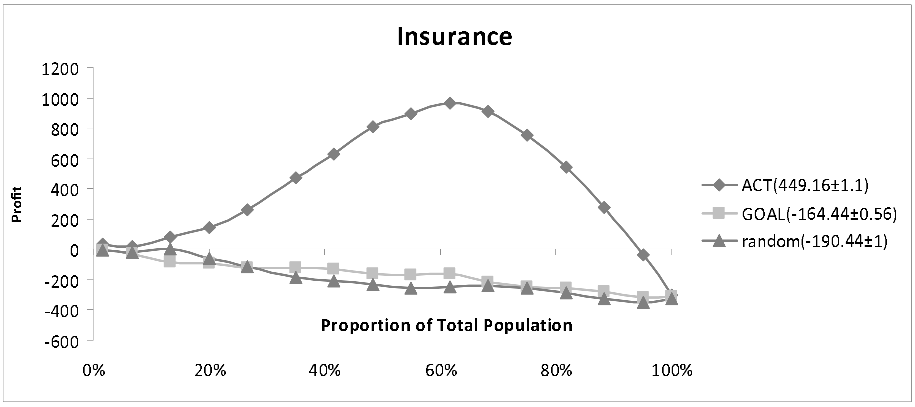

Figure 6. Insurance Continuous Train Profit

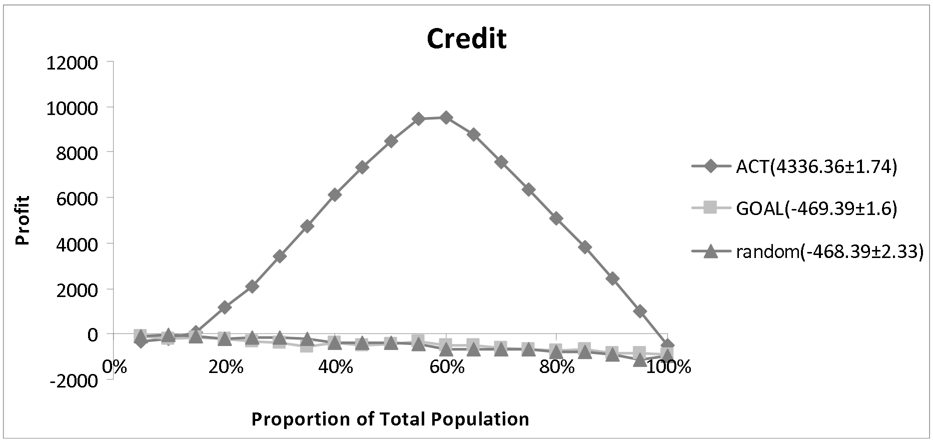

Figure 7. Credit Continuous Train Profit

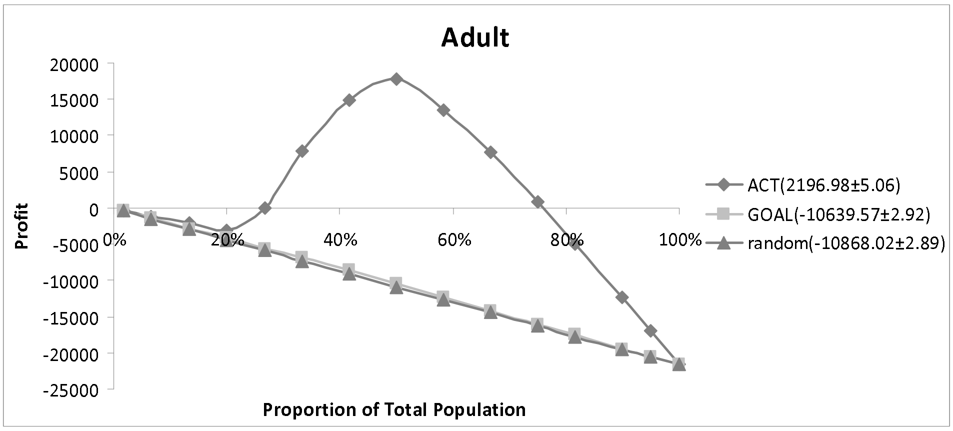

Figure 8. Adult Continuous Train Profit

Table 4. Gain summary (AUC)

Dataset	ACT	GOAL	ACTw/oP	ACTw/oS	Random
Adult	400±493	392±388*	389±316*	396±307*	396±359*
Credit	2.96±2.82	2.85±0.86*	2.99±1.24*	2.88±1.11*	2.92±0.61*
Donation	130.2±32.1	114.2±29.16*	113.5±18.2*	100.7±26*	104.5±28.2*
Insurance	52.6±2.82	46.88±0.86*	41.37±1.24*	43.3±1.11*	44.22±0.61*
Mean Rank	1.25	3.25	3.5	4	3
Mean Perf.	95%	30%	36%	24%	37%

4. Conclusions and Discussion

In this paper we presented a new method for cost-sensitive active learning with decision trees: ACT. Specifically, the investigated problem is concerned with the decision as to which potential customer we should approach with a new product offer. The decision is made according to the customer's own characteristics and the past history of purchasing by previously approached potential customers. While other active learning algorithms strictly address improved exploration of the dataset, ACT also considers the costs/profits of the exploration/exploitation tradeoff *during* the learning process. Thus, there is no need to separate the model training step from the actual exploitation of the dataset artificially.

We used four benchmark datasets to test ACT extensively. Using our devised performance metrics, ACT was found to outperform the newest active learning algorithm so far: GOAL. We extensively tested the contribution of each one of ACT's unique different contributions:

1. The pessimistic expectation estimator for selecting the consequent data seems to provide most of ACT's advantage for most performance measures.
2. The exploration-exploitation trade-off via simulated annealing is the second most contributory factor to ACT's performance. Unlike most studies of cost-sensitive active learning methods that try to optimize some testing set measures (e.g., profit), in this study we are also interested in the training performance (i.e., profit or loss) *during* the training phase. Thus, there is no clear cut between the training phase and the execution (validation) phase. Note that we did not attempt to optimize the simulated annealing algorithm, so further improvement is possible.
3. Training the dataset on sequential partitions (batches) is beneficial, since we can decide better when to stop the learning process, and it is more practical for the multi-marketing arena, where customers are processed in batches.

The proposed principles of ACT are not unique to decision trees and can be adjusted to other induction methods (e.g., such as neural networks) where we can solve approximately the integral expression in Eq. 7. Furthermore, the pessimistic expectation estimator is also not unique (Rokach *et al.*, 2008).

A. Appendix: Description of the Datasets Used in the Experiments

Table 5 presents the attributes of the datasets used in the experiments. Following is a description of each dataset.

1. The *Donation* dataset. This dataset represents a real-world case study and was previously used in the KDD cup 98 [3]. The original donation datasets contains 479 attributes. However, for the classification task we used only the following 15 input attributes: ODATEDW, INCOME, RAMNTALL, NGIFTALL, CARDGIFT, MINRAMNT, MINRDATE, MAXRAMNT, MAXRDATE, LASTGIFT, LASTDATE, FISTDATE,NEXTDATE, TIMELAG, AVGGIFT. The class refers to a real response of a person to contribute a donation. The A priori "success" response rate in the training set is almost 5%. The original dataset contained 95,413 training instances, of which we randomly selected only 10,000 training instances. The testing-set contains 96,357 instances, of which we randomly selected another 10,000 instances.
2. The *Adult* dataset. This dataset predicts whether income exceeds $50K/yr based on census data. It is also known as the "Census Income" dataset. It was taken from the UCI repository (Blake and Merz, 1998). The a priori "success" response rate in the training set is 23%. It contains 10,000 training instances and 20,000 instances for testing.
3. The *Insurance* dataset. The *insurance company* benchmark has been used in the CoIL challenge 2000 (Putten and Someren, 2000). The a priori "success" response rate in the training set is almost 6%. It contains 5822 training instances and 4000 instances for testing.
4. The *Credit* dataset. This dataset concerns credit card applications. It was taken from the UCI repository. The a priori "success" response rate in the training set is 43%. It contains 300 training instances and 370 instances for testing.

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

Table 5. Attributes of the Datasets Used in the Experiments

Dataset	# Attr.	Train Size	Test Size	# Batchs	Resp. Rate	o Value	r Value
Adult	14	10000	20000	60	23%	2.9	10
Insurance	85	5822	4001	20	6%	0.49	10
Credit	15	300	370	20	43%	3.5	10
Donation	15	10000	10000	60	5%	0.68 (Given)	Varied (mean 15)

References

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.

Brown, L. D., Cai, T. T., and DasGupta, A. (2001) Interval Estimation for a Binomial Proportion. Statistical Science 16(2): 101-117, 2001.

Cohn, D. A., Ghahramani, Z., Jordan, M. I. (1996), Active learning with statistical models, Journal of Artificial Intelligence Research, Vol. 4, pp. 129–145.

Demsar J. (2006) , Statistical Comparisons of Classifiers over Multiple Data Sets, Journal of Machine Learning Research, Vol. 7, pp.1-30.

Elkan, C. (2001) , The foundations of cost-sensitive learning. Proceedings of the 17th International Joint Conference on Artificial Intelligence, Morgan Kaufmann.

Hedayat A.S, N. J. A. Sloane N.J.A and Stufken J. (1999) , Orthogonal Arrays: Theory and Applications. Springer-Verlag, NY.

Hollmen J., Skubacz M. and Taniguchi M. (2000), Input dependent misclassification costs for cost-sensitive classifiers, In Proceedings of the Second International Conference on Data Mining, pp. 495-503.

Kirkpatrick S., Gelatt C.D., Vecchi M.P. (1983), Optimization by Simulated Annealing, Science, Vol. 220, No. 4598, pp. 671-680.

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.

Margineantu D. (2005) , Active Cost-Sensitive Learning, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, IJCAI-05.

Mayer U. F. and Sarkissian A. (2003) , Experimental design for solicitation campaigns, Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 717-722, Washington, D.C., ACM Press, New York, NY, USA.

Naamani, L. (2008) Cost Sensetive Active Learning for Target Marketing, a M.Sc. dissertation. Dept. Of Information Systems Engineering, Ben-Gurion University, Israel.

Putten P. and Someren M. (2000) , CoIL Challenge 2000: The Insurance Company Case. Published by Sentient Machine Research, Amsterdam. Also a Leiden Institute of Advanced Computer Science Technical Report 2000-09. June 22.

Quinlan, J. R. (1993), C4.5: Programs for Machine Learning, Morgan Kaufmann.

Rokach L. and Maimon O. (2005) , Top Down Induction of Decision Trees Classifiers: A Survey, IEEE SMC Transactions Part C. 35(4): 476- 487.

Rokach L. and Maimon O. (2008), Data Mining with Decision Trees: Theory and Applications, World Scientific Publishing Company.

Rokach L., Naamani L., Shmilovici A. (2007), Active Learning Using Conditional Expectation Estimators, In Morzy T., Morzy M., Nanopoulos A. (Eds.), proceeding of the 3rd ADBIS workshop on Data Mining and Knowledge Discovery ADMKD'2007, October 2, Varna, Bulgaria, pp. 83-95.

Rokach L., Naamani L., Shmilovici A. (2008) , Pessimistic Cost-sensitive Active Learning of Decision Trees for Profit Maximizing Targeting Campaigns", Data Mining and Knowledge Discovery, Vol. 17, No. 2, pp. 283-316.

Roy, N. and McCallum, A. (2001). Toward optimal active learning through sampling estimation of error reduction. Proceedings of the International Conference on Machine Learning.

Saar-Tsechansky M. and Provost F. (2007), Decision-Centric Active Learning of Binary-Outcome Models. Information systems research Vol. 18, No. 1, pp. 4-22.

Tong, S., Koller, D. (2000). Support vector machine active learning with applications to text classification. Proceedings of the 17th International Conference on Machine Learning, ICML-2000, July 2, Stanford, CA.

Turney P. (2000) , Types of Cost in Inductive Concept Learning. Proceedings of the Cost-Sensitive Learning Workshop at the 17th International Conference on Machine Learning, ICML-2000, July 2, Stanford, CA.

Weiss, G. M., Tian, Ye (2006) , Maximizing classifier utility when training information is costly, SIKDD exploration, Vol. 8 No. 2.

Zadrozny B. and Elkan C. (2001) , Obtaining calibrated probability estimates from decision trees and naďve Bayesian classifiers. In Proc. of the Eighteenth International. Conference on Machine Learning, pages 609–616. Morgan Kaufmann.