Maximizing Return on Direct Marketing Campaigns in Commercial Banking

CS 229 Project: Final Report

Oleksandra Onosova

Introduction

Recent innovations in cloud computing and unified communications have made a technologically advanced call center a more viable option for marketing campaigns than ever. A well-executed direct advertising campaign can provide a positive return on investment by motivating customers to respond to a call-to-action. One of the preparatory stages ensuring a high return on investment consists of carefully querying a marketing database to generate a target list of respondents.

In this paper, a number of machine learning techniques were used to optimize the target list of consumers of a campaign. The data for the analysis (S. Moro R. L., 2011) was provided by a Portuguese banking institution seeking to sell the subscription to a bank deposit through direct calls. A prior research paper (S. Moro R. L., 2011) describes the analysis performed utilizing the same data. SVM methodology was used and call duration was recognized as the most relevant feature for classification.

To enhance the findings of the prior paper, call duration was excluded from the feature list and the objective was reformulated. The new goal was to select a target audience for the campaign by using machine learning techniques prior to making calls in order to maximize the net profit.

Data Preparation

The original data consist of 16 mixed (categorical and numeric) features with no missing values. They fall into three main groups:

- The demographical information (age, education, job, marital status)
- Information related to banking(balance, prior defaults, loans)
- Data about the current and prior campaigns (date of the call, days passed since last contact, outcome of the prior campaign for that client, etc.)

As a part of data analysis, first, additional features were added:

- the year feature (data description characterized it as being organized by date in an ascending order)
- day of the week

Next, the data were scrubbed. For example, the value of the variable "pdays", reflecting how many days before the current call, the customers had been contacted contained the value of "-1" signifying that the customer was never contacted before. It was substituted by a value slightly higher than the maximum value of that feature.

As the next step, the data were binned separately for various classification approaches. Binning was performed according to the distribution of the variables. For the variables that were almost uniformly distributed (age, "pdays"), an equally spaced binning was used, with the exception of the tail values that were binned together. For the approximately logarithmically distributed variables balance, "campaign", (where the latter signifies the number of contacts performed during this campaign and for this client), the log transformation was applied and then the

normal distribution was used for binning. The categorical variables were coded with levels for Naïve Bayes classifier and as matrices of 0s and 1's for SVM and regression classifiers.

Finally, the data were split into 80% for training and 20% for testing purposes. The training dataset was further analyzed using 3-fold validation and the final results (presented below) were generated on the testing set.

METHODS

The following machine learning approaches for performing binary classification were attempted: Naïve Bayes, Logistic and Probit Regression, and SVM. The algorithms split a set of potential customers into two classes: the ones to call (i.e. those that have a sufficiently high probability of subscription) and those not to call (i.e. those that are unlikely to make a purchase). The net profit was the metric used in optimizing classifier performance and was formulated as follows:

$$Profit = \sum_n [1\{classifier\ predicted + \cap sale\ done\} * (R - C1)] + \sum_n [1\{classifier\ predicted + \cap sale\ not\ done\} * (-C2)]$$

Where:
n – number of customers in the list
Customers belonging to "+" ("-") class – customers that were evaluated as likely (unlikely) to make a purchase
R – Expected revenue from a successful call;
C1 – Cost of a successful call;
C2 – Cost of a failed call, where on average, C2 < C1 due to shorter call duration.

An important additional task was adjusting the selected machine learning approaches to account for the asymmetry of the misclassification cost, i.e. taking into the account the ratio of the cost of a false positive and the cost of a false negative. Due to the disproportionately low cost of direct marketing calls in comparison with the high expected revenue in case of the successful outcome of the call, false positives were considerably "cheaper" than false negatives. Consequently, two techniques: 1) thresholding (H. Drucker, 1999) and 2) enhancing the SVM objective function to incorporate two cost parameters and then tuning all the parameters (cf. Osuna, Freund, & Girosi, 1997) were attempted to implement the asymmetric cost adjustment.

Thresholding consisted of finding an optimal threshold probability of a successful call. It helped identify the maximum probability of a successful sell that would make the call worthwhile. A range of thresholds from 0% to 100% was generated and the profit metric was calculated for each threshold in the range. The procedure was repeated for the three subsets of the training data as 3-fold cross-validation was used. The results were then applied to the testing set to estimate the resulting profit. The following thresholding approaches were used for various classifiers:

- For Naïve Bayes, Probit Regression and Logistic Regression the optimal threshold was calculated as follows:

$$t = \underset{t}{\mathrm{argmax}}(\sum_{i=1}^{3} Profit\ (CV_i))$$

Where: $t - Threshold$
CV_i - Cross-Validation set i out of 3 sets

For performing Naïve Bayes and Logistic/Probit regression, the MATLAB inbuilt functions were utilized.
- For SVM, the optimal threshold was tuned along with the rest of the parameters for Linear SVM and SVM with RBF Kernel:

$$(t, C, W_1) = \underset{t,C,W_1}{\mathrm{argmax}}(\sum_{i=1}^{3} Profit\ (CV_i))$$

(Linear)

2

$$(t, C, W_1, \gamma) = \underset{t,C,W_1,\gamma}{\mathrm{argmax}} \left(\sum_{i=1}^{3} Profit\ (CV_i) \right)$$

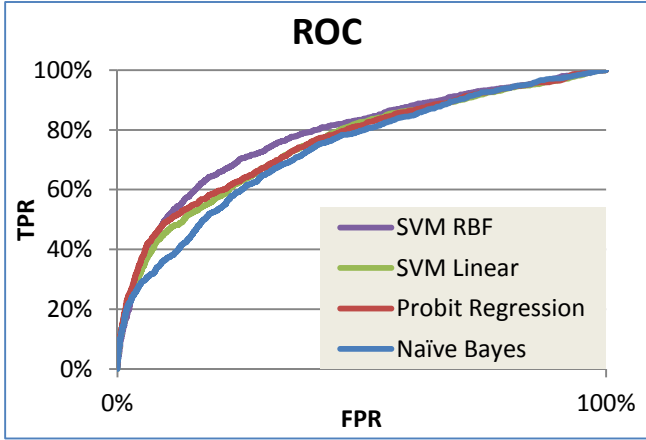(Gaussian kernel, where W_1 is a weight coefficient on cost of the "-" class)
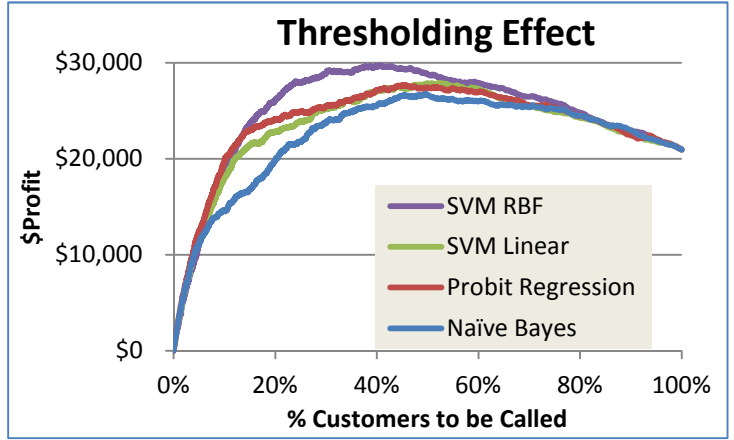

Fig. 1

Fig. 2

The ROC curve was generated on the test set by varying the threshold parameter and estimating the True Positive Rate and the False Positive Rate at each threshold level. The results showed that SVM RBF was clearly superior with respect of Naïve Bayes classifier, and most likely better that the other approaches used. The results on Figure 2 were generated by the same procedure of varying threshold level and on the same data as the ROC curves, but additionally, the Profit and '% Customers to call' metrics were estimated. The plots are intuitive in showing that, as the probability grows from 0% to about 50%, and, correspondingly, a higher percent of respondents is called, the net profit is rising: a slightly higher cost of the campaign is still lower than the marginal increase in the revenue. After the saturation point is reached (40%-50% probability threshold for various classifiers), the marginal cost of additional calls become higher than the expected revenue, as the probability of achieving a successful sell is lower. Only Probit regression result with is shown on the figure due to very similar results for Logit and Probit.

Since the goal was to maximize the profit by varying the threshold, the ultimate metric for the classification was:

$$Profit = \max_{\{t\}} \sum_n [1\{P(S) > t \cap Call\ successful\} * (R - C1)] + \sum_n [1\{P(S) > t \cap Call\ failed\} * (-C2)]$$

Where: t – Threshold; P(S) – Predicted probability of successful call

The second technique for adjusting the classifiers to account for the asymmetric cost function was enhancing the SVM objective function and tuning the parameters in the SVM algorithm. The SVM Linear objective function with two cost parameters was formulated as follows:

$$\min_{\omega,b,\xi} \frac{1}{2} \|\omega\|^2 + \sum_{y_i=1} C * \xi_i + \sum_{y_i=-1} CW_1 * \xi_i$$

s.t. $\xi \geq 0, i = 1, \dots, n$
$\xi \geq 1 - y_i(\omega^T x_i + b), i = 1, \dots, n$

where C is the cost for the "+" cases, corresponding to y=1
CW_1 is the cost for the "-" cases, corresponding to y=-1

With the above objective function, the SVM parameters were tuned using a grid-search and maximizing the targeted profit. The algorithm is implemented with the inbuilt features of LibSVM (Chih-Chung Chang).

In case of linear SVM, the optimal parameters C and W_1 were found with a grid search on ranges 2^{-3} to 2^2 and 2^{-3} to 2^6 respectively. The resulting maximum profit as a function of C and W_1 is shown on Figure 3. It is easily noticeable that for different values of cost C, the maximum is achieved near $W_1=8$. More localized grid search resulted in C=0.25 and $W_1=10$ as parameters maximizing cross-validation profit for linear SVM.

For RBF kernel SVM, the optimal parameters were found in two steps. First, by searching for optimal C and W_1 (while keeping γ default value for LIBSVM), it was found that $W_1=10$ is the optimal value for all values of C considered (C varied from 2^{-3} to 2^8 and W_1 from 2^3 to 2^6), as shown on Figure 4. The value $W_1=10$ was chosen for the second grid search step as the optimal CV (Cross-Validation) profit was found by varying C and γ (C varied from 2^{-3} to 2^4 and γ - from 2^{-9} to 2^2). The optimal parameters for RBF SVM model were found to be the following: C=0.25, γ=0.125, and $W_1=10$.
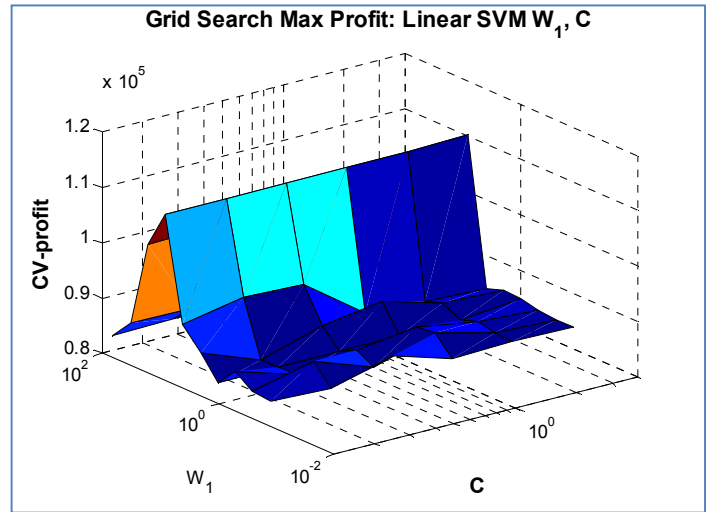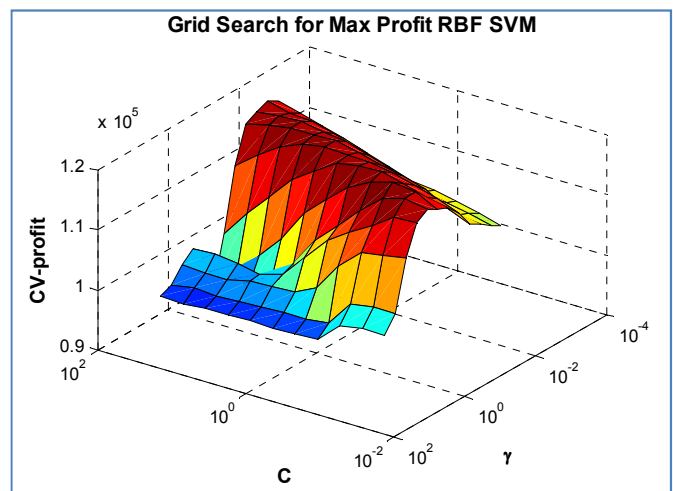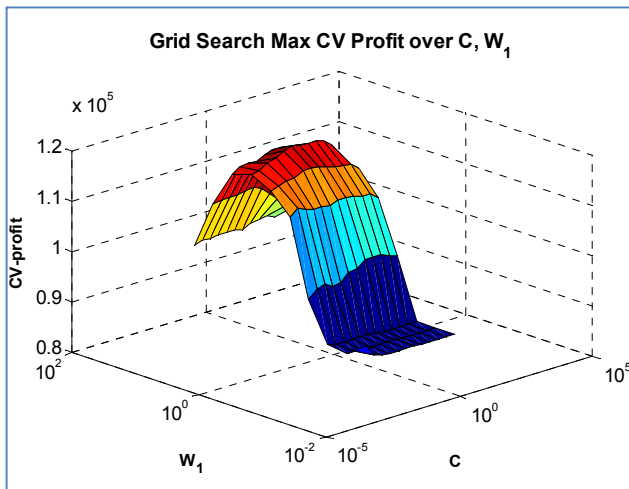

Fig. 3

Fig. 4

Fig. 5

QUANTIFICATION OF THE COST MATRIX

The average cost of 1 minute of a marketing call, performed by a call center is approximately $1; the average call duration for a successful call is 10 minutes, whereas the call duration for a failed marketing call is 4 minutes ("3 Reasons the Call Center Is Far From Dead", 2012). The costs used for the analysis were respectively C1 = $10 and C2 = $4. Lifetime value for an average customer relationship in retail banking is $150 (Milman, 2012). The value of the customer subscribing to the deposit account was assumed to be $60, comprising approximately 40% of the total value of a customer relationship. Thus, the net profit of a customer subscription is ($60 - $10 = $50).

The combined net profit matrix becomes:

	Unsuccessful call	*Successful Call*
Call is not made	0	0
Call is made	-$4	$50

RESULTS AND CONCLUSION

SVM was found to produce the best optimization in terms of the profit margin. In particular, SVM with the Gaussian Kernel and the parameters tuned in the grid search illustrated above was found to be the champion classifier among the attempted ones.

The chart on the right (blue bars) shows the profit on the test set, generated by applying the models with thresholds (and other parameters in the case of SVM) found to maximize CV-profit on the training set. The "base case" used for comparison shows the profit generated by calling 100% of the customers on the test set list. The red bars show the "Threshold Oracle View" estimated by using the optimal threshold with respect to the test set. The optimal threshold was found by grid search, with the goal of maximizing the test set profit. The minimal difference between the Test Profit and the "Threshold Oracle View" demonstrates robustness of threshold estimation on the training set.

Total Profit on Test Set vs Treshold Oracle Model View

In conclusion, using SVM with Gaussian kernel resulted in a 41% increase in marketing profit of the campaign. In the given dataset, the number of successful sells was uncharacteristically high – about a tenth of the calls. In the more typical marketing campaigns, the successful sells comprise an even smaller percentage of calls, leading to higher costs. In that case, the asymmetry of the cost function would be even more pronounced and thus the benefit of the analysis would be even higher. In addition to maximizing profit, other goals such as minimizing customer frustration could be achieved by trimming the target customer list.

Future direction for quantifying the effect of the campaign would be measuring the negative impact of unsuccessful marketing calls and assigning a negative cost to them to further diminish the number of calls to the customers. That would effectively change the cost function and leave the underlying problem unchanged. In addition, a more thorough grid-search of SVM parameters could be attempted, with all RBF parameters being varied along all the dimensions at the same time, if time permits. Finally, other classification techniques that are known to effectively address the asymmetric cost function could be attempted, such as Ada Boost algorithm.

REFERENCES

"3 Reasons the Call Center Is Far From Dead". (2012, April 24). Retrieved November 10, 2012, from Mashable: http://mashable.com/2012/04/24/call-center-death-exaggerated

Chih-Chung Chang, C.-J. L. (n.d.). *"LIBSVM -- A Library for Support Vector Machines"*. Retrieved from http://www.csie.ntu.edu.tw/~cjlin/libsvm/

H. Drucker, D. W. (1999). "Support vector machines for spam classification". (p. 10(5)). IEEE.

Milman, S. (2012). *"Bank and Credit Union Business Strategy and Customer Life Time Value"*. Retrieved November 10, 2012, from Optirate: http://bankblog.optirate.com/bank-and-credit-union-business-strategy-and-customer-life-time-value/#.UMbsUOhSTwh

S. Moro, R. L. (2011, October). *Bank Marketing Data Set*. Retrieved November 10, 2012, from UC Irvine Machine Learning Repository: http://archive.ics.uci.edu/ml/datasets/Bank+Marketing

S. Moro, R. L. (2011). Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology. *Proceedings of the European Simulation and Modelling Conference - ESM'2011* (pp. pp. 117-121). Guimarães, Portugal: EUROSIS.