Project Assignment 2 - Winter 2011
Deciding the Optimal Marketing Strategy by using Logistic and Real-Valued Regression

Andrew Heiberg
aheiberg@ucsd.edu

André Christoffer Andersen
andre@andersen.im

Fabian Siddiqi
fsiddiqi@ucsd.edu

February 8, 2011

Abstract

The aim of this paper is to investigate the potential revenue benefit of applying machine learning techniques to historical marketing data. Specifically, logistic and real-valued regression are applied to data in an attempt to correlate customer attributes, marketing strategy, and the value of their purchases. First, the probabilistic foundation for this approach is discussed. Next, the data pre-processing steps, which turn it into a more amenable or performant form with respect to the logistical and real-valued regression algorithms, are enumerated. A description of these algorithms follows, detailing the parameter selection process and other techniques found useful in practice. Also included is our attempts at fitting real-valued regression to the data.

Finally, the results of applying the trained classifiers are analyzed. This includes both a cross-validated test of model accuracy and predicted revenue gains from applying the model to future data. Overall the results are encouraging, with an average estimated spend error of +4%. When the optimal treatment per customer is used, a 60.4% increase in revenue is seen. While these numbers are encouraging, the relationship, or lack thereof, between the model accuracy and final revenue prediction is considered.

1 Introduction

Machine learning techniques can be used to determine how much money a customer will spend if he or she visits an online retailer. The data set used can be found at Kevin Hillstroms website [1] and contains 64,000 entries, each of which describes a different customer. Multiple properties are considered, such as the amount of money the customer has spent previously or whether or not the customer has visited the website (for a full explanation of the data, please refer to [1]).

The 64,000 customers are divided into three groups (each with approximately 33% of the whole set). The first segment received an e-mail featuring mens merchandise, the second an e-mail featuring womens merchandise and the third no e-mail at all.

The goal of this paper is to propose a method for determining the amount of money a new customer will spend given a data vector describing said customer. Once this initial task is finished, it is possible to determine which e-mail should be sent out to maximize profits.

The problem can be formally described as follows.

$$\begin{aligned}\mathbb{E}[\text{spend}|x, \text{treatment}] = \mathbb{E}[\text{spend}|\text{purchase}, x, \text{treatment}] \times \\ p(\text{purchase}|\text{visit}, x, \text{treatment}) \times \\ p(\text{visit}|x, \text{treatment})\end{aligned}$$

Where *spend* is the amount of money spent by a customer (real-valued and non-negative), x is the customer's attribute vector, *purchase* represents whether a purchase has been made or not (binary), *visit* is the visit variable (binary) and *treatment* is the e-mail which he or she has received.

The derivation of the previous equation can be found below, where each variable has been substituted for its initial, and binary variables have a bar over them if they are false (e.g. $\bar{\text{v}}$ is equivalent to $\text{v} = 0$).

$$\begin{aligned}\mathbb{E}[\text{s}|x, \text{t}] &= \mathbb{E}[\text{s}|\text{p}, x, \text{t}] \times p(\text{p}|x, \text{t}) + \mathbb{E}[\text{s}|\bar{\text{p}}, x, \text{t}] \times p(\bar{\text{p}}|x, \text{t}) \\ &= \mathbb{E}[\text{s}|\text{p}, x, \text{t}] \times p(\text{p}|x, \text{t}) + 0 \times p(\bar{\text{p}}|x, \text{t}) \\ &= \mathbb{E}[\text{s}|\text{p}, x, \text{t}] \times \Big(p(\text{p}|\text{v}, x, \text{t}) \times p(\text{v}|x, \text{t}) + p(\text{p}|\bar{\text{v}}, x, \text{t}) \times p(\bar{\text{v}}|x, \text{t})\Big) \\ &= \mathbb{E}[\text{s}|\text{p}, x, \text{t}] \times \Big(p(\text{p}|\text{v}, x, \text{t}) \times p(\text{v}|x, \text{t}) + 0 \times p(\bar{\text{v}}|x, \text{t})\Big) \\ &= \mathbb{E}[\text{s}|\text{p}, x, \text{t}] \times \Big(p(\text{p}|\text{v}, x, \text{t}) \times p(\text{v}|x, \text{t})\Big)\end{aligned}$$

Our goal is to determine the expected value of money spent given a customer attribute vector and the treatment used. This can be expanded using conditional probabilities and further simplified by noting that the money spent will be equal to zero if the customer does not purchase anything or does not visit the website. These modifications to the original expression allow for a more focused use of the given data since only relevant customers will be considered when generating the real-valued regression classifier.

This leaves us having to predict the probability of an individual making a purchase. Logistic regression can be used, but by applying the same trick of conditioning on another variable (in this case *visit*), we can split this into two logistic regression problems, each with a refined set of data to be trained on. As before, one of these logistic regressions in unnecessary, because we know that the probability of a purchase with no visit is zero. In the final analysis, we are left needing two logistic and one real-valued regression classifiers for each treatment.

The data was divided into three segments according to their treatment and were analyzed independently, although the same methods were used for each model. A detailed explanation of their implementation and testing can be found in Section 2.

2 Design of Algorithms

Three main of algorithms are considered in this section. The three submodels are created using two algorithms: logistic regression for the two probability terms and real-valued non-linear regression for the expectation term. The original dataset is parsed and pre-processed using the methods described below.

2.1 Data Preparation and Preprocessing

2.1.1 Feature Preparation and Identification

The raw data is represented in a comma separated (csv) data file where the feature space can be divided in to three groups. The first eight features represent the historical customer data which we are trying to learn from. The next feature "segment", also called treatment, determines what campaign, if any, the customer was exposed to. The last three features are the results

of the campaign and thus different types of labels. Refer to Figure 1 for a detailed explanation of different features and their preprocessing.

Original features		Manipulated features				
Name	Data type	Name	Data type	Expanded	Reduced	Training
Recency	integer	Recency	integer	X	X	X
History_Segment	integer	History_Segment	integer	X	X	
History	real	History	real	X	X	X
Mens	binary	Mens	binary	X	X	X
Womens	binary	Womens	binary	X	X	X
Zip_Code	{urban,	Zip_Code: Urban	binary	X	X	X
	suburban,	Zip_Code: Suburban	binary	X	X	X
	rural}	Zip_Code: Rural	binary	X		
Newbie	binary	Newbie	binary	X	X	X
Channel	{phone,	Channel: Phone	binary	X	X	X
	mail,	Channel: Mail	binary	X	X	X
	multi}	Channel: Multi	binary	X		
Segment (treatment)	{men,	Segment: Men	binary	X	X	
	women,	Segment: Women	binary	X	X	
	none}	Segment : None	binary	X	X	
Visit	binary	Visit	binary	X	X	
Conversion (purchase)	binary	Conversion	binary	X	X	
Spend	real	Spend	real	X	X	

Figure 1: List of features present in the original data set, and the modifications which have beeen applied.

Initial Parsing and Cleaning Initial parsing of the raw data was done in Python. We transformed all data to a numerical or integer equivalent.

Binary Transformation of Category features. All features that were represented as discreet categories were split in to as many binary features as there were categories. To see why this is necessary consider the following case. The variable 'zip_code' can take on three different values. If they were coded as 0,1,2, it would imply that the value that got coded to 2 would be twice as important as the value that was coded to 1. We do not know this *a priori*, so if we create three binary variables for each of the three values we avoid biasing the algorithm.

Feature Redundancy Elimination The binary transformation of discreet categories create redundant features. One of the now binary categories

can be pushed to the intercept β_0 and thus eliminated. This reduces the computational complexity and convergence speed of learning at the expense of model readability.

Data Selection Because each model is given a treatment we remove all data points that does not pertain to that categorization, and then eliminate the now uninformative treatment feature.

Label Elimination When training a model with a specific label we remove not only this label feature, but all labels from the training features. It does not make sense to train on any of the other potential labels since these will not be supplied when presented with new data to give predictions on.

2.1.2 Standardization

When feature preparation is performed we standardize all training data. This was done by transforming all data of each training feature to have a mean of zero and standard deviation of one. The benefit of standardization is that it increases learning convergence and speed. This happens because we use the same learning rate for all features. One learning rate can be good for one set of data but bad for another. Standardization circumvents this by transforming the data to a size that handles equally good.

2.2 Logistic Regression

Regularized stochastic logistic regression was employed. An outer loop iterated through the epochs while an inner loop iterated through the rows of the data set. Each time through the outer loop, the learning rate was set to: $\lambda = \frac{1}{\lambda_0 + epoch}$. In the inner loop (working on row i of the data), each β_j was updated with the following rule: $\beta_j = \beta_j + \lambda((y_i - p_i) - 2\mu\beta_j)$, where $p_i = \frac{1}{1+e^{-\beta * x_i}}$. Both λ_0 and μ are supplied as parameters.

The convergence of β was used as the stopping criteria. More specifically, $\Delta\beta$, defined as $||\Delta\beta||_2 < \Delta\beta_{thresh}$; we let $\Delta\beta_{thresh} = 10^{-4}$ based on empirical evidence. The final requirement was that β had to satisfy the 'sanity check', a consequence of $\frac{dLCL}{d\beta} = 0$, which says that the optimal β should satisfy $\sum_i y_i - \sum_i p_i = 0$. In practice, the right hand was replaced with $\leq \frac{\sum_i y_i}{200}$. This was necessary because often the β would converge for some arbitrary

threshold, but the sanity check would be far off. Finally, in some cases the β would converge, yet the sanity check would refuse to decrease by any significant amount. For this reason, the number of epochs was bounded to $E_{max} = 1000$.

The run-time of each outer loop (epoch) is $O(mn)$, if we define the size of the data matrix being iterated over to be $m \times n$. The sanity check, if performed, takes $O(mn)$ time, as it must compute p_i for m rows, each computation taking $O(n)$ time. The number of epochs is unknown, as an infintesimal learning rate would mean the β converge extremely slowly. Since the number of epochs was bounded to E_{max}, the total run time is $O(E_{max}mn)$.

In practice, one epoch took: 0.4421 seconds when running on a $21,306 \times 10$ matrix. Convergence and sanity check satisfaction ranged within 600-1,000 epochs.

2.3 Real Valued Regression

In order to predict spending using real valued regression several different approaches were tried: regular linear, polynomial and pairwise non-linear regression. In the later two cases, the training data was transformed into the desired form prior to applying linear regression on each of them.

Polynomial regression data transformation f is described below:

$$
\begin{aligned}
&f : X \to X_f \\
&X \in \{x_1, x_2, ..., x_n\} \\
&X_f \in \bigcup_{i=1}^{k} \{x_1^i, x_2^i, ..., x_n^i\}
\end{aligned}
$$

Where k is the degree of the polynomial.

Pairwise non-linear regression data transformation g was an extension of squared polynomial regression. It adds all pairwise multiplicative combinations:

$$
\begin{aligned}
&g : X \to X_g \\
&X \in \{x_1, x_2, ..., x_n\} \\
&X_g \in \bigcup_{i=0}^{n} \bigcup_{j=i\neq 0}^{n} x_i x_j
\end{aligned}
$$

where $x_0 = 1$.

3 Design of Experiments

3.1 Logisitic Regression Parameter selection: Finding the Optimal Learner

The regularized stochastic logistic regression learner has two parameters that must be selected: the learning rate λ and μ, used in the regularization term. To find the optimal combination, λ and μ were varied from 10^{-3} to 10^{-6} and 10^{-3} to 10^{-5}, respectively. For each (λ, μ) combination, 3-fold cross-validation was performed. For each fold, the stochastic regularized logistic regression learner was given $E_{MAX} = 600$ epochs on the training data, after which it would output the learned beta values. Scoring these beta values on the reserved test data was done via sanity check: $\left| \sum_i \left(y_i - f(x_i, \beta) \right) \right|$. The (λ, μ) combination that had the lowest average sanity check was chosen as the optimal parameters for the learner. This parameter selection was done twice: once to generate a learner for $p(\text{visit}|x, \text{treatment})$ term and separately to generate a learner for the $p(\text{purchase}|\text{visit}, x, \text{treatment})$ term.

3.2 Real-Valued Regression: Finding an Acceptable Regression Model

In order to test the real-valued regression models, the data set is reduced in terms of features and data points as described in Subsection 2.1. Thus, the remaining data points with positive spend amounts are divided in to three sub-models, each given a specific treatment. Depending on the regression model we applied any needed data transformation and then, in order to detect overfitting, divided each sub-model in to a training set of 70% and testing set of 30%. After running the MATLAB multinomial regression function on the training set we applied the resulting β weights on the testing data yielding predictions. The mean of all absolute prediction errors is then calculated as a measure of accuracy. As a benchmark we compared the predicting error with the standard deviation of the labels.

All experiments were averaged over several thousand runs each with randomized data ordering.

3.3 Accuracy of Estimated Spending

The previous two sections discussed constructing three learners. When trained, these learners produce classifiers, which can then be multiplied as described in Section 1 to predict the spend amounts of a customer.

Unfortunately, we were unable to find a suitable non-linear function between data dimensions and the expected spend amount. However, in order for the models to predict an expected spend amount, we need number. Our idea was to simply use the mean of the known spend amounts. Over a large enough number of test examples, and assuming our visit and purchase probabilities are correct, the predicted sum of these spend amounts using the mean should become arbitrarily close to the actual spend amounts. Plotting the spend amounts as a CDF, it was clear they could be sampled as coming from an exponential distribution. Since the exact connection between the data and this distribution eluded us, it was deemed the safest to stick with selecting the mean in lieu of sampling from this distribution.

Before we can place confidence in our predicted spend amounts, the accuracy of the classifiers must be established. For each of the three treatment datasets, 5-fold cross validation was used. For each fold, the two learners were trained on the appropriate subset of the training data: the entire treatment subset for the $p(\text{visit}|x, \text{treatment})$ logistic regression learner and only visit $= 1$ data for the $p(\text{purchase}|\text{visit}, x, \text{treatment})$ logistic regression learner.

With the classifiers now in hand, their accuracy is tested as follows:

$$\text{spend}_{\text{estimated}} = \sum diag(\text{data}_{\text{test}} \times \beta_{\text{visit}}^{T}) \times (\text{data}_{\text{test}} \times \beta_{\text{purchased}}) \times \text{spend}_{\text{mean}}$$

The percentage error was averaged over all folds to yield a measure of the expected degree of over/under estimation by the model.

3.4 Value of Machine Learning

Now that we have our three treatment models and a measure of their accuracy, we are ready to investigate their value. To do this, we again start with 5-fold cross validation. This time, however, the *entire dataset* is divided into training and test (as opposed to the previous section, where the divisions were made inside the treatment subsets). The learners are trained on the appropriate data, generating two out of the three necessary components for each treatment model. To model the other third, we will generate a spend

amount for each test example by simply using the mean of all the spend amounts. Using this spend amount and the two computed probabilities, we can calculate the expected spend amount for each treatment per test example. We will define choosing the maximizing treatment for each individual as the optimal treatment set.

There is an additional wrinkle here. We assume that the accuracy of a given model will not be perfect, e.g. it will have over- or under-estimated the true spending amount. We have computed this percent error in the previous experiment and can use it to adjust each models prediction. For example, if the 'men' model, on average, over-estimates by 15%, we can divide each of it's predictions by 1.15 to adjust for this error.

Comparing the sum of these maximum estimations to the known total spend amount, we can see how much additional value doing machine learning generates (if any).

4 Results of Experiments

4.1 Real-Valued Regression

The results from the real-valued regression experiments are listed in Figure 2. Testing was done both on the training data and the testing data in order to discover overfitting. The primary results are the error on testing data. The errors on the testing data were close to the standard deviation of all labels (105.77) which we deemed to be unsatisfactory for further use. The prediction means were similar to the label mean (125.22), thus not failing the sanity check for predictions. Linear and all polynomial regression rendered similar results and thus no benefit was observed in using the latter. Non-linear overfits predictions of training data yielding error of 66.4361 vs testing data with error of 100.8323.

4.2 Parameter Estimation

Refer to Tables 1 and 2.

4.3 Accuracy of Estimated Spending

5-fold CV of Percent Error $= \frac{\text{Spend}_{\text{Estimated}} - \text{Spend}_{\text{Actual}}}{\text{Spend}_{\text{Actual}}}$; refer to Table 3.

Visits			
	$\mu = 10^{-3}$	$\mu = 10^{-4}$	$\mu = 10^{-5}$
$\lambda = 10^{-3}$	20.3852	18.3317	28.8747
$\lambda = 10^{-4}$	17.5633	5.6825	24.5382
$\lambda = 10^{-5}$	-26.9635	6.2198	17.2792
$\lambda = 10^{-6}$	15.0545	13.6247	26.1522

Table 1: Average sanity check using (μ, λ) for training 'visit' logistic regressor.

Purchase			
	$\mu = 10^{-3}$	$\mu = 10^{-4}$	$\mu = 10^{-5}$
$\lambda = 10^{-3}$	5.0920	-1.2021	1.6440
$\lambda = 10^{-4}$	-4.0750	-3.4678	0.4468
$\lambda = 10^{-5}$	3.6765	-3.1723	14.1310
$\lambda = 10^{-6}$	2.2539	-2.7782	-6.6120

Table 2: Average sanity check using μ, λ for training 'purchase' logistic regressor.

	Error		Mean
	Training data	Test data	Prediction
Linear	78.4645	77.2772	120.5200
Squared	77.8106	78.3342	120.5214
Cubed	77.1043	79.5694	120.4215
Non-linear	66.4361	100.8323	119.9790

Figure 2: Results obtained for different real-valued regression schemes.

	μ	σ
Men	-0.08	0.12
Women	0.03	0.19
None	0.18	0.50

Table 3: Standard deviation (σ) and mean (μ) for each of the different treatment accuracies.

4.4 Value of Machine Learning

5-fold CV of Percent Improvement = $\frac{\text{Spend}_{\text{Estimated}} - \text{Spend}_{\text{Actual}}}{\text{Spend}_{\text{Actual}}}$; refer to Table 4.

5 Findings and Analysis

Training classifiers using the optimal λ_{base} and μ yielded good accuracies, with only the 'none' treatment outside of 10% range. This gives us confidence

Treatment	Percent Improvement
All women	-0.0011
All men	0.5583
All none	-0.4328
Optimal	0.6043

Table 4: Percentage of revenue improvement per treatment.

that our valuation of doing machine learning is a reasonable one, especially since we adjust for these accuracy errors. One cause for concern, however, is the variance of the 'none' model accuracy over five trials. Intuitively, one would expect the 'none' purchasers to be internally more dissimilar than their 'women' and 'men' counterparts. This would make training on the 'none' examples to yield more volatile results.

Obviously, the primary weakness of our model is selecting the spend mean for the predicted spend amount. In lieu of a good regression function though, this seemed like a reasonable alternative.

It is important to note that the accuracy of the models is only well-defined when applied to data of the same treatment. For instance, when we applied the three treatment models to a data point to pick the maximum spend amount, we have no actual spend amount with which to compare the estimation. We can never know how much customer x would have spent if only they had received another treatment.

The other limitation on our models is the value of real-valued prediction. Perhaps those treated with 'women', when they do spend, spend more than 'men'. Our model fails to capture subtleties such as this one, but on average we expect it to be a good stop-gap, assuming of course that future potential-customers come from the same distribution as our data. If new customers are spending significantly more or less, using this old mean will give poor spend predictions.

6 Conclusion

Qualitatively, we can say the following:

1. Doing machine learning will yield approximately 60% increase in expected revenue. We cannot be completely confident in this number, but on the performance measures we can evaluate we can confine the error to $\pm$ 10%.

2. Exclusively using men's advertisements will yield a nearly equivalent revenue boost. The data somewhat supports this: as 267/512, or 46%, of buyers were men.

In our opinion, there is very little downside to choosing the optimal treatment strategy. Once the model has been trained, the cost associated with

suggesting treatments for a new set of data is small and the predicted revenue increase is large.

However, as always, there must be a synthesis of human and machine intelligence. If the assumptions of the model change, i.e the target demographic changes or new products are being marketed, predictions based on the old data would likely be inaccurate, necessitating the training of a new models.

References

[1] The MineThatData E-Mail Analytics And Data Mining Challenge `http://blog.minethatdata.com/`