Table of Contents

[1. ABSTRACT 4](#_Toc386319883)

[2. PROJECT OVERVIEW 5](#_Toc386319884)

[2.1 Introduction and Motivation 5](#_Toc386319885)

[2.2 Problem Statement 5](#_Toc386319886)

[2.3 Scope 6](#_Toc386319887)

[3. LITERATURE SURVEYED 7](#_Toc386319888)

[3.1 Introduction 7](#_Toc386319889)

[3.2 EXAMPLE STUDY 9](#_Toc386319890)

[3.2.1 Abstract 9](#_Toc386319891)

[3.2.2 Actionable Knowledge Discovery For CRM 10](#_Toc386319892)

[4. PROJECT DESIGN 14](#_Toc386319893)

[5. IMPLEMENTATION DETAILS 16](#_Toc386319894)

[5.1 Data Set Information 16](#_Toc386319895)

[5.2 Feature standardization and Cross validation 18](#_Toc386319896)

[5.3 Logistic Regression 18](#_Toc386319897)

[5.3.1 Hypothesis representation 19](#_Toc386319898)

[5.3.2 Simplified Cost Function and Gradient Descent 19](#_Toc386319899)

[5.3.2 Drawback (Problem of overfitting and underfitting) 20](#_Toc386319900)

[5.3.3 Graphs and Observations 24](#_Toc386319901)

[5.4 Support Vector Machine (SVM) 30](#_Toc386319902)

[5.4.1 Notation 30](#_Toc386319903)

[5.4.2 Functional and Geometric Margin 30](#_Toc386319904)

[5.4.2 The optimal margin classifier 32](#_Toc386319905)

[5.4.3 Kernels 33](#_Toc386319906)

[5.4.4 Sequential Minimal Optimization (SMO) 34](#_Toc386319907)

[5.4.5 Graphs and Observations 36](#_Toc386319908)

[5.5 Neural Networks 37](#_Toc386319909)

[5.5.1 Introduction 37](#_Toc386319910)

[5.5.2 Types of classification problems with NNs 37](#_Toc386319911)

[5.5.3 Cost function for neural networks 38](#_Toc386319912)

[5.5.4 Back Propogation Overview 38](#_Toc386319913)

[5.5.5 The Back Propagation Algorithm 39](#_Toc386319914)

[5.5.6 Graphs and observations 40](#_Toc386319915)

[6. TECHNOLOGIES USED 44](#_Toc386319916)

[7 PROJECT TIMELINE 45](#_Toc386319917)

[8. FUTURE WORK 46](#_Toc386319918)

[9. CONCLUSION 47](#_Toc386319919)

[10. REFERENCES 48](#_Toc386319920)

Figures

[Figure 1 : Traditional Data Mining Example 8](#_Toc386318745)

[Figure 2 : D3M Example 9](#_Toc386318746)

[Figure 3 : Lift Curve Example 13](#_Toc386318747)

[Figure 4 : Naïve Profit Curve Example 14](#_Toc386318748)

[Figure 5 : Sigmoid Function 15](#_Toc386318749)

[Figure 6 : Sigmoid Function 20](#_Toc386318750)

[Figure 7 : Underfitting and Overfitting 22](#_Toc386318751)

[Figure 8 : High Variance 23](#_Toc386318752)

[Figure 9 : High Bias 23](#_Toc386318753)

[Figure 10 : Iterations vs Cost(a) 25](#_Toc386318754)

[Figure 11 : Iterations vs Cost(b) 26](#_Toc386318755)

[Figure 12 : Iterations vs Cost(c) 27](#_Toc386318756)

[Figure 13 : Learning Curve 28](#_Toc386318757)

[Figure 14 : Lift Curve 29](#_Toc386318758)

[Figure 15 : Naïve Profit Curve 30](#_Toc386318759)

[Figure 16 : Max Margin Classifier 33](#_Toc386318760)

[Figure 17 : Coordinate Ascent 36](#_Toc386318761)

[Figure 18 : Naïve Profit Curve 38](#_Toc386318762)

[Figure 19 : Neural Networks Diagram 39](#_Toc386318763)

[Figure 20 : Iterations vs Cost 43](#_Toc386318764)

[Figure 21 : Lift Curve 44](#_Toc386318765)

[Figure 22 : Naïve Profit Curve 45](#_Toc386318766)

# 1. ABSTRACT

The concept of direct marketing has been an effective strategy to reach out to the customer. Data mining techniques do help to generate profits, but a more concrete methodology is desired to target the customer base effectively. This paper focuses on a Domain-Driven Data Mining(D3M) approach to optimize the Return on Investment (ROI). Here, we are considering the domain of a Portuguese banking institution, obtained from the UCI machine learning repository. This is one-to-one marketing where our objective is to target the customer base such that we can elicit a positive response from maximum by contacting minimum. That is, we contact only those customers who are most likely to respond. This implementation is done in two parts. The first part is to assign scores to customers (which is also the probability of responding) that is done using classification techniques. The second part is to suggest an action that would enable us to determine which and how many customers to contact in order to maximize profit or minimize loss. This is done using the concept of ‘lift’. Thus, the entire process is tantamount to Actionable Knowledge Discovery (AKD) wherein a viable action is suggested eventually.

# 2. PROJECT OVERVIEW

## 2.1 Introduction and Motivation

* Apart from marketing, our methodology can be extended to fit other scenarios as well: Electronic Health Records(EHRs) and medical information technology have resulted in large data warehouses which contain a detailed account of patients, symptoms, medication, visits, procedures etc. Modeling and understanding this data will help improve computer aided diagnosis(CAD) and quality of healthcare in the domains of disease prediction, non-invasive diagnosis etc. This is because the underlying mathematical chassis and computational methods, for any domain will largely remain the same, what will change is the dataset.
* Business Intelligence: It is financially relevant to the company where it hypothetically would be employed for a greater Return On Investment(ROI).
* Medical Informatics: Improvement in healthcare systems and CAD for better diagnoses and decision making on part of medical personnel.

## 2.2 Problem Statement

Real-world business problems are often buried in complicated environments and factors. The environmental elements are often filtered or largely simplified in traditional data mining research. As a result, there is a big gap between a syntactic system and its actual target problem. The identified patterns cannot be used for problem solving. Existing work often stops at pattern discovery, which is mainly based on technical significance and interestingness. Business concerns are not considered in assessing patterns. Consequently, the identified patterns are predominantly of technical interest. Traditional data mining is a data-driven trial-and-error process. It stops at discovered pattern/rule, either views data mining as an autonomous process, or only analyzes the issues in an isolated and case-by-case manner. As a result, the knowledge discovered is not interesting and actionable to constrained business. Often algorithms are delivered, but they are not executable and operable in the business system. No effective tools are provided to convert models to executables that can be integrated into production systems. Domain Driven Data Mining(D3M) aims at bridging this gap using specific domain factors.

## 2.3 Scope

Traditional data mining, as will be illustrated later on, makes use of methods such as regression analysis, exponential smoothening or classification to say, predict or forecast a certain value, eg: sales forecasting or customer classification in the business intelligence domain. Domain driven data mining is an extension of traditional data mining in the sense that, not only will a prediction, classification etcetera be made, but also a strategy will be suggested, based on the rules mined, to improve overall business profits. Thus, here an action is suggested.

Considering the context of CRM(Customer Relationship Management) mining is performed on all the relevant information(datasets) and patterns are extracted which are then fine-tuned by the domain expert to pitch in domain factors. This consultation and refinement are done in a cyclic fashion, that is closed-loop iterative refinement. Finally, after mutual consent, an action or a method is suggested that is desirable to the company, say, to augment the net profit or minimize the expenses. If appropriate datsets are not available, they can be constructed artificially.

# 3. LITERATURE SURVEYED

## 3.1 Introduction

Conventional data mining applications face serious difficulties in solving complex real-life business decision making problems when practically deployed.Traditional data mining stops at discovered pattern/rule, either views data mining as an autonomous process, or only analyzes the issues in an isolated manner. As a result, the knowledge discovered is not interesting and actionable to constrained business. D3M attempts to provide information and abilities to fill the existing gap between academic researches and real-world business problems

Traditional Data Mining Example:

Decision Tree

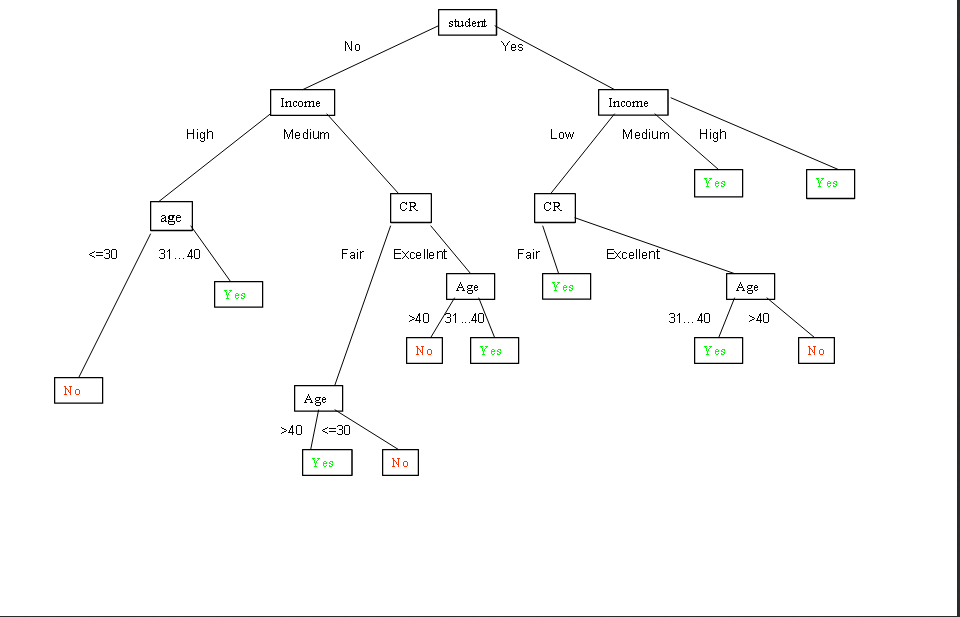


Figure : Traditional Data Mining Example

The rules generated from the above decision tree are as follows:

1. student(no)^income(high)^age(<=30) => buys\_computer(no)

2. student(no)^income(high)^age(31...40) => buys\_computer(yes)

3. student(no)^income(medium)^CR(fair)^age(>40) => buys\_computer(yes)

4. student(no)^income(medium)^CR(fair)^age(<=30) => buys\_computer(no)

etc.

Upshot:

We are only finding hidden trends. These trends may or may not be of significance to a business organization

( in the above figure, answer/result/output is only in the form of “yes” or “no”)

Domain-Driven Data Mining Example:



Figure : D3M Example

Context:

Optimization of the Return On Investment for a company by selecting the most effective communication channels for attracting those customers who satisfy a certain threshold value(probability that the customer is a buyer).

The above graph gives the results of an actionable knowledge discovery methodology, for one-to-one marketing, which allows to contact the right customer through the right communication channel(communication channel maybe e-mail,fax,phone,sms etc)

y-axis: Profit rate

x-axis: % of customers

Upshot:

The black dot(max. of the validation curve) indicates the optimum proportion of customers to contact: 50.3 % with 88.14 % of the profit rate.

( Result/answer/output suggests an action to be taken for maximizing profit of the company. This is tantamount to **“actionable knowledge”** which is explained later)

**Thus, one important difference between traditional data mining and D3M is that in D3M, an action is suggested which has the potential to influence a business scenario(such as increase in profit).**

## 3.2 EXAMPLE STUDY

### 3.2.1 Abstract

Data mining methods could be interesting to generate substantial profits for decision makers and to optimize the choice of different marketing activities. In this case study, an actionable knowledge discovery methodology is proposed, for one-to-one marketing, which allows to contact the right customer through the right communication channel. This methodology first requires a measurement of the tendency for the customers to purchase a given item, and second requires an optimization of the Return On Investment by selecting the most effective communication channels for attracting these customers. A methodology has been applied to the VM Mat´eriaux company. Thanks to the collaboration between data miners and decision makers, a domain-driven view of knowledge discovery has been presented satisfying real business needs to improve the efficiency and outcome of several promotional marketing campaigns.

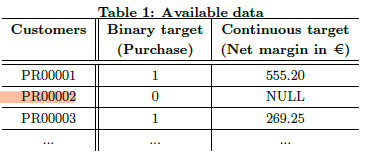
Actions, such as emailing, direct mailing and salespersons’ visits cost the company money. To overcome this drawback, marketing managers need to acquire relevant knowledge on a one-to-one basis to decide which are the most effective communication channels to use for each customer, while avoiding flooding customers with messages.

This methodology for CRM requires firstly to measure the tendency of customers to purchase an item, and secondly to optimize the Return On Investment (ROI) by selecting the most effective communication channels for attracting these customers.

### 3.2.2 Actionable Knowledge Discovery For CRM

The dataset is divided into three subsets for training, validation and the test to measure the performance of the final model. The proposed methodology consists of four steps: scoring customers for purchases , choosing marketing channels and the Return On Investment model.

Let us consider an array of customers for whom the binary target variable means the purchase in a promotional campaign (1 for purchase, 0 otherwise) and the continuous target variable means net margin generated by the visiting customer.

****

**Scoring customers for purchases**

*Ridge Regression*

Ridge regression can provide the contribution, i.e. a polynome weight Wx, that is used to show the relative importance Cx of a variable x in the model.

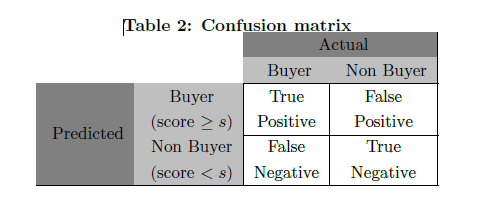
1.png

Given a threshold s, we predicted that a customer i is a buyer if the score si calculated by the model is greater than s (Table 2). Let u(s) be the proportion of customers whose score calculated by the model is greater than s:

1.png

Let v(s) be the real proportion of buyers identified by the model:

1.png



Finally, ridge regression allows us to obtain a score function is in decreasing order for each customer i reflecting the probability pi (obtained by normalization of is) to purchase during the promotional campaign.

*Accuracy and robustness*

In order that decision makers can graphically visualize the accuracy and robustness of our models, we use lift curves (Curves C3 and C4 on Figure 1 ). A lift curve (variation of ROC curve) is a parametric curve representing the proportion of buyers detected v(s) with relation to the proportion of customers selected u(s) .The accuracy and robustness of a model can be measured by comparing the lift curve to random and ideal curves (Curves C2 and C1 on Figure 1). The random curve is the curve y = x (we detect a% buyers selecting a% customers). The ideal curve is the one in which all buyers are selected first. From the lift curve, two indexes can be calculated. The first index is the Gini index , named KI. It corresponds to the area between the validation curve and the random curve, and measures the accuracy of the model, i.e. the ability of input variables to explain the target. The second indicator, named KR, corresponds to the difference in area between the estimation and the validation lift curves. It measures the robustness of the model, i.e. its ability to provide the same level of quality on a new dataset, typically the validation dataset.

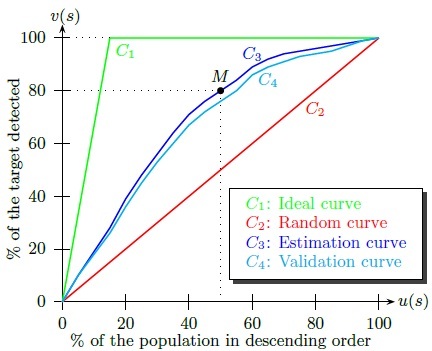
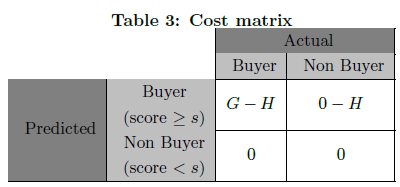


Figure : Lift Curve Example

*Naive profit curve*

A naïve profit curve is the transformation of a lift curve with a cost matrix (Figure 3) defined by the decision makers. The naïve profit of the promotional campaign can therefore be defined as follows: the net margin achieved by contacting u(s) % customers. Let N be the number of customers in the sample studied, G the average net margin per customer and H the average spending communication per customer (Table 3).



1.png

The theoretical maximum profit, profitMAX, is obtained with the model where all buyers are selected first. Thus, a naive profit curve (Figure 2) is a parametric curve representing the profit rate (naiveProfit(s)/profitMAX) according to the proportion of u(s) selected customers. This curve presents a different Y-axis of the lift curve with the percentage of maximum profit in order to graphically measure the ROI of the promotional campaign. For example, the point N (Figure 2) means that on the validation dataset, we contact 48 % of the population to achieve a maximum profit equal to 82 % of maximum theoretical profit. Therefore, this part of our methodology allows us to obtain the optimal point (maximum Y-axis) on the curve indicating the proportion of the population to be contacted.

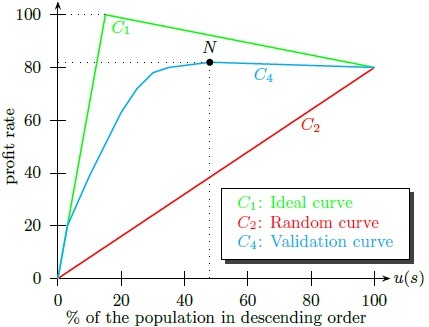


Figure : Naïve Profit Curve Example

# 4. PROJECT DESIGN

* Our scenario deals with optimum promotional campaign return on investment in bank direct marketing.
* Classification of customers using Logistic Regression (and assignment of score/probability to each customer) using gradient descent optimization. Plotting of learning and validation curves for diagnostic purposes. Testing for regularization (to determine high bias/high variance or underfit/overfit).
* Using more sophisticated algorithms- Support vector machine(SVM) and Neural Networks(NN). Compared to logistic regression and NN, SVM sometimes gives a cleaner way of learning non-linear functions.
* Construction of profit curve(for all three techniques - logistic regression , SVM and NN) using concept of ‘lift'.
* Classification used in our case is binary i.e. whether a customer subscribes to the bank or not. Thus, logistic regression is employed for assignment of probabilities(i.e. probability of subscribing) and binary classification.
* To evaluate probabilities, we need a hypothesis. We use a sigmoid function/logistic function for this.

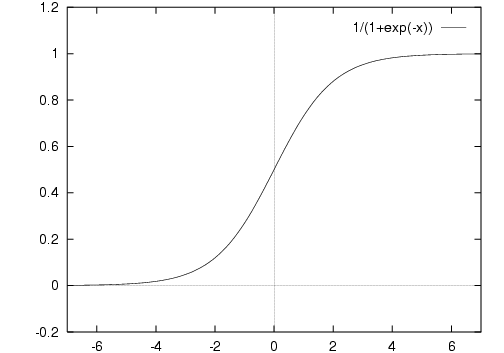


Figure : Sigmoid Function

Sigmoid function : Outputs values between 0 and 1

* SVM also used for classification because it is a large margin classifier. There are three main advantages of SVM

1. It has a regularisation parameter, which makes the user think about avoiding over-fitting.

2. It uses the kernel trick, so you can build in expert knowledge about the problem via engineering the kernel.

3. An SVM is defined by a convex optimisation problem (no local minima) for which there are efficient methods (e.g. SMO).

* NNs - one of the most powerful learning algorithms. It is a learning algorithm for fitting the derived parameters given a training set.

# 5. IMPLEMENTATION DETAILS

Implementation in two parts:

1. Customer Scoring (Using Logistic Regression - Classification)
2. Profit Estimation (Using concept of ‘lift’)

## 5.1 Data Set Information

The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be (or not) subscribed.

Input variables:

# bank client data:

1 - age (numeric)

2 - job : type of job

(categorical: 'admin.','unknown','unemployed','management','housemaid','entrepreneur','student',  
'blue-collar','self-employed','retired','technician','services')

3 - marital : marital status (categorical: 'married','divorced','single'; note: 'divorced' means divorced or widowed)

4 - education (categorical: 'unknown','secondary','primary','tertiary')

5 - default: has credit in default? (binary: 'yes','no')

6 - balance: average yearly balance, in euros (numeric)

7 - housing: has housing loan? (binary: 'yes','no')

8 - loan: has personal loan? (binary: 'yes','no')  
# related with the last contact of the current campaign:

9 - contact: contact communication type (categorical: 'unknown','telephone','cellular')

10 - day: last contact day of the month (numeric)

11 - month: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')

12 - duration: last contact duration, in seconds (numeric)

# other attributes:  
13 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)

14 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric, -1 means client was not previously contacted)

15 - previous: number of contacts performed before this campaign and for this client (numeric)

16 - poutcome: outcome of the previous marketing campaign (categorical: 'unknown','other','failure','success')

Output variable (desired target):  
17 - y - has the client subscribed a term deposit? (binary: 'yes','no')

## 5.2 Feature standardization and Cross validation

* We use feature standardization , that is, all features/attributes have zero mean and unit variance. Since we are employing gradient descent algorithm for optimization of training parameters, a preprocessing step such as feature scaling or standardization would improve the convergence speed of the algorithm.
* Since the original attribute values are skewed , we are avoiding skinny contour plots (which would have led to slower convergence). This step is very common and is used in several machine learning algorithms such as neural networks, support vector machine(svm), logistic regression etc.
* Before beginning with the first phase of classification, the data set is divided into three parts – training, cross-validation and testing set. In short, training set is used to train the data, cross validation for diagnostics and test set to test the model.
* We are predicting y in the data set with y=1 implying the customer will subscribe. Thus, we are dealing with binary prediction here i.e. 0 is negative class and 1 is positive class.

## 5.3 Logistic Regression

Logistic Regression important points:

🡪 y is a discrete value. Develop the logistic regression algorithm to determine what class a new input should fall into.

🡪 y is either 0 or 1 : 0 = negative class (absence of something) 1 = positive class (presence of something).

Logistic regression is convenient to our desired solution as it produces probabilities. Since it is likely that direct marketing problems deal with constrained budgets, it would be logically and financially potent to contact only those customers with the highest probabilities of subscribing. The customer probabilities are then arranged in descending order; in order to maximize the ROI generated by our model on our promotional campaign, we use lift and also a transformation of lift which is achieved by incorporating costs and profit.

Threshold: We’ve used the default threshold value of 0.5 for a binary classification problem. Different threshold values can be used depending on the application. Training on non-stratified data might lead to a misleading threshold value, as even though the overall misclassification error might be low, we might end up with a poor model.

### 5.3.1 Hypothesis representation

We want our classifier to output values between 0 and 1

For classification hypothesis representation we do hθ(x) = g((θ*T* x))Where we define g(z)

* + z is a real number g(z) = 1/(1 + e*-z*)

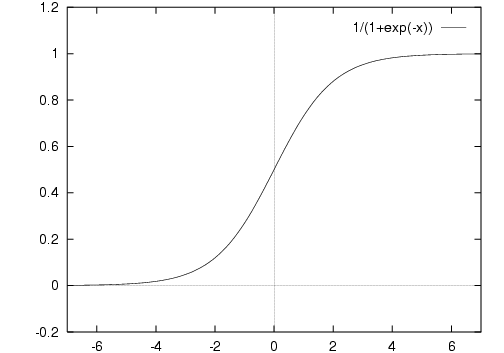
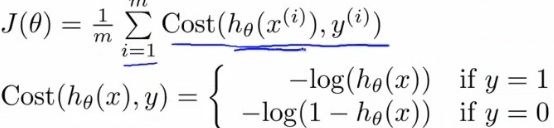


Figure : Sigmoid Function

Given this we need to fit θ to our data.

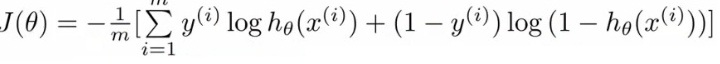
### 5.3.2 Simplified Cost Function and Gradient Descent

* Define a simpler way to write the cost function and apply gradient descent to the logistic regression. By the end should be able to implement a fully functional logistic regression function.
* Logistic regression cost function is as follows:



Eqn. 1

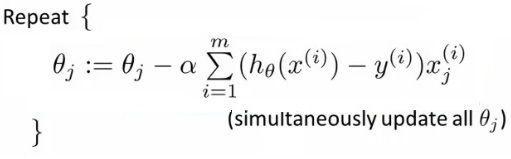
So, in summary, our cost function for the θ parameters can be defined as



Eqn. 2

Now we need to figure out how to minimize J(θ) . Use gradient descent.

Repeatedly update each parameter using a learning rate.

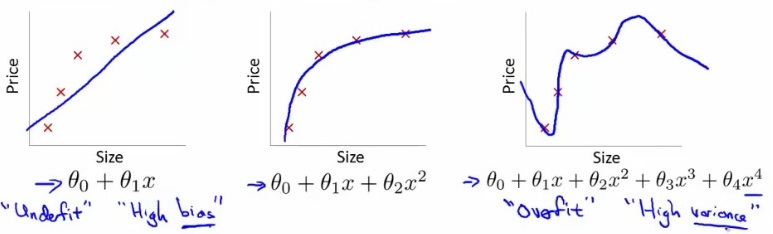


Eqn. 3

If you had *n*features, you would have an n+1 column vector for θ

### 5.3.2 Drawback (Problem of overfitting and underfitting)

Logistic regression uses a pre-determined model for binary classification by fitting data to a logistic curve. Since visualization is difficult in a 16 dimension attribute space, we make use of standard diagnostic measures such as learning curves.



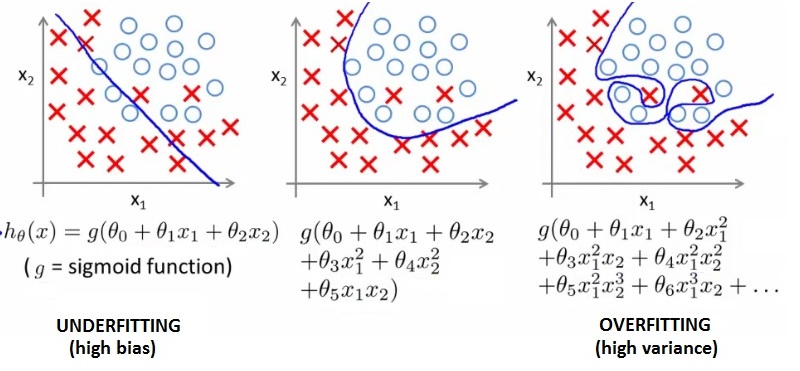


Figure : Underfitting and Overfitting

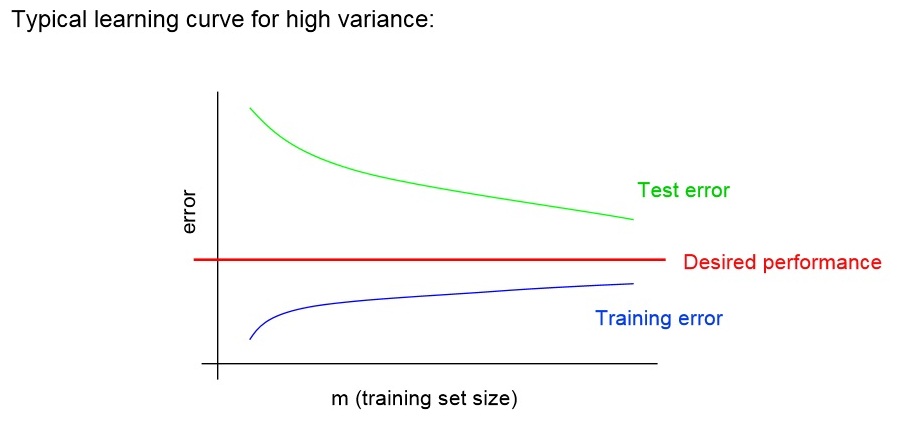


Figure : High Variance

Test error still decreasing as m increases. Suggests larger training set will help. Large gap between training and test error.

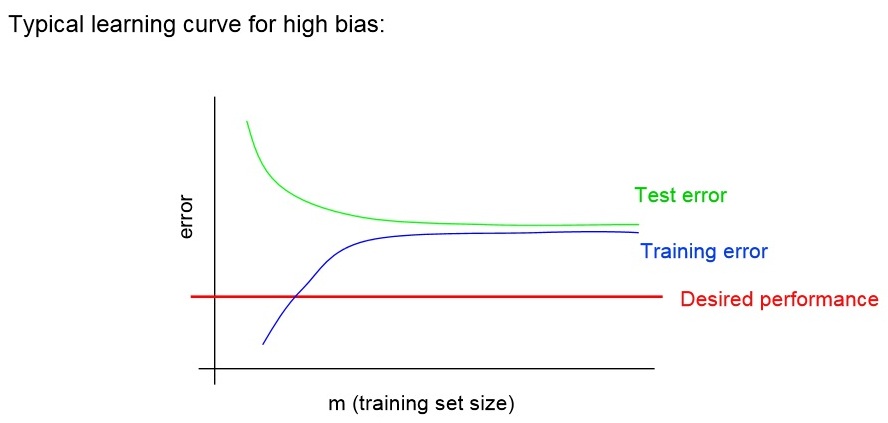


Figure : High Bias

Even training error is unacceptably high. Small gap between training and test error.

Addressing overfitting:

1) Reduce number of features

* + - Manually select which features to keep
    - Model selection algorithms are discussed later (good for reducing number of features)
    - But, in reducing the number of features we lose some information
      * Ideally select those features which minimize data loss, but even so, some info is lost

2) Regularization

* + - Keep all features, but reduce magnitude of parameters θ
    - Works well when we have a lot of features, each of which contributes a bit to predicting y

Small values for parameters corresponds to a simpler hypothesis (you effectively get rid of some of the terms). A simpler hypothesis is less prone to overfitting.

**λ**is the **regularization parameter**. Controls a tradeoff between our two goals

1) Want to fit the training set well

2) Want to keep parameters small

### 5.3.3 Graphs and Observations

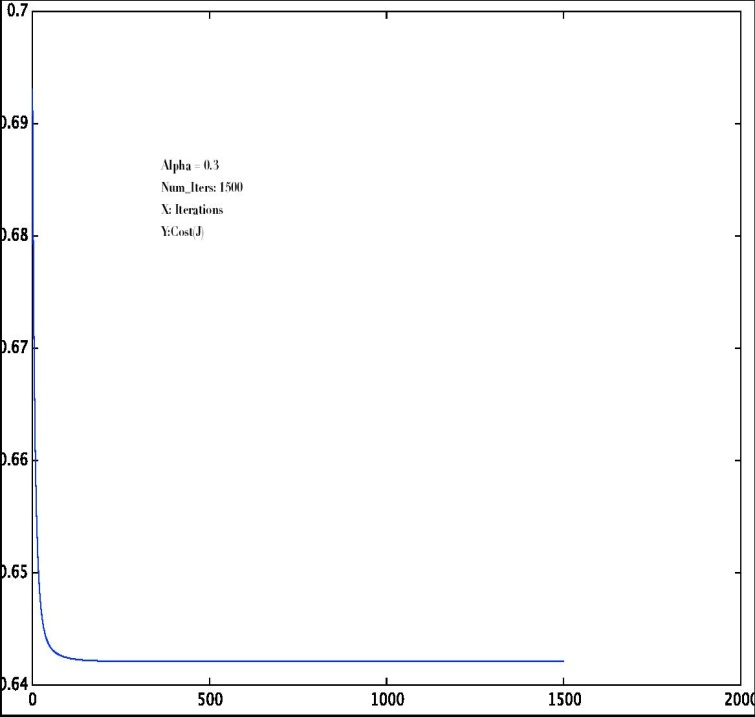


Figure : Iterations vs Cost(a)

X-axis represents number of iterations.

Y-axis represents error or cost.

As shown above, this is the plot we obtain for **alpha** = 0.3 (**alpha** is the learning rate)

Number of iterations = 1500

We observe that the error reaches minimum somewhere before 500 iterations and remains that way subsequently.

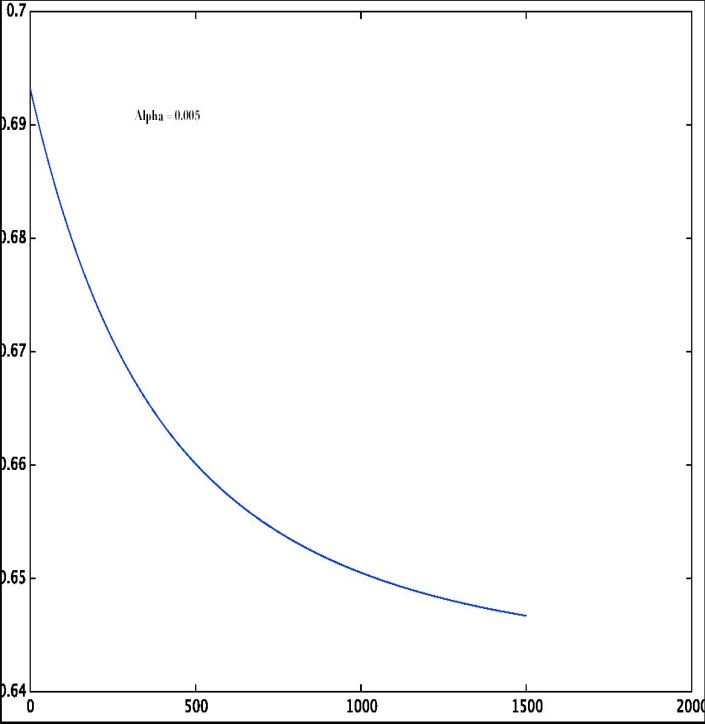


Figure : Iterations vs Cost(b)

In the above figure, **alpha** is0.005. Here, obviously as **alpha** is small convergence time is more.

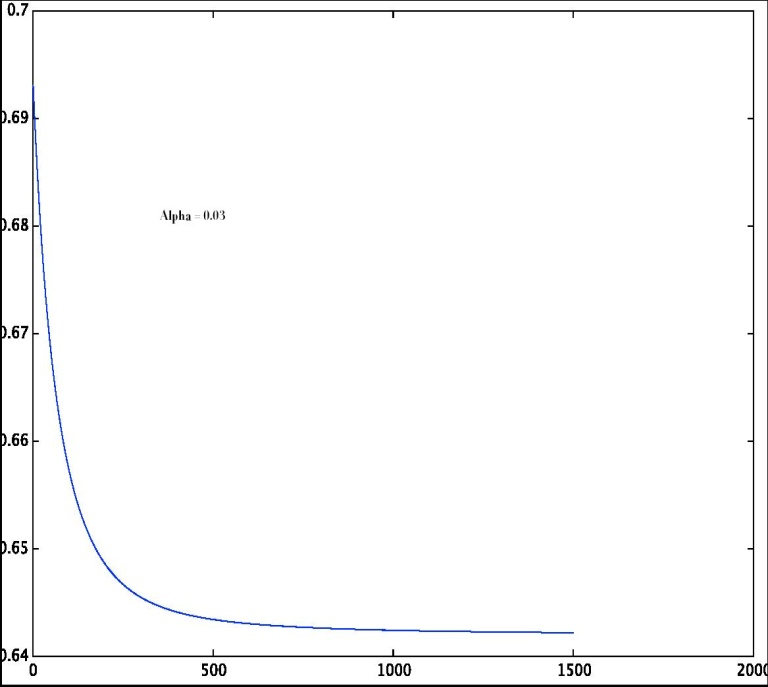


Figure : Iterations vs Cost(c)

In Fig. 12 , value of **alpha** is 0.03. We see that minimum error is obtained somewhere after 500 iterations.

The objective of this exercise is to find a suitable tradeoff between **alpha** and the number of iterations used for an optimization algorithm such as gradient descent.

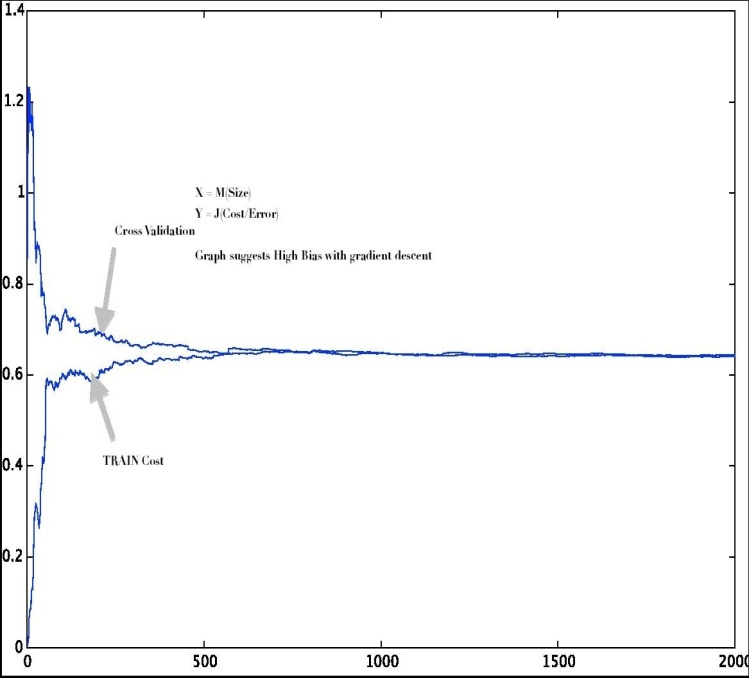


Figure : Learning Curve

X-axis : Number of samples

Y-axis : Error

We have taken 2000 tuples in the training set and 1000 tuples in the cross-validation(CV) set. As labeled in the figure, plots of training error v/s training size and CV error v/s CV size are shown. After a certain size, both plots eventually end up having line characteristics. Also, there is a very small gap between the two plots. This situation is indicative of a high bias scenario.

CV is mainly used for diagnostics. It is employed to give an idea of how well the model will confirm to different data sets.

The high bias scenario can be corrected using either a complicated model, adding more features or using a different optimization algorithm.

Obtained learning and validation curves, which indicated presence of an under fit model. Tested for regularization, but since a high bias scenario exists, the regularization parameter is set to zero. This is further asserted by the validation curve.

Second phase deals with plotting of lift and profit curves for the promotional campaign.

Lift: Lift, in the context of data mining, is a performance measure. We would like our classification model to sift through the records and sort them according to which ones are most likely to be subscribers, thereby making more informed decisions. The lift curve helps us to “skim the cream” i.e. select relatively small number of customers to obtain larger profit.

Different threshold values yield different confusion matrices; rather than calculating performance measures such as specificity, sensitivity etc it is more suited to look at the cumulative lift curve.

In the lift curve, we arrange the number of customers on the x-axis in descending order of probability. On the y-axis – number of customers who respond positively (i.e. true positives).

In our case, we have taken 750 tuples in the test set.

Number of true positives obtained were 73. We see from the curve that around first 300 customers need to be contacted to capture all the subscribers.

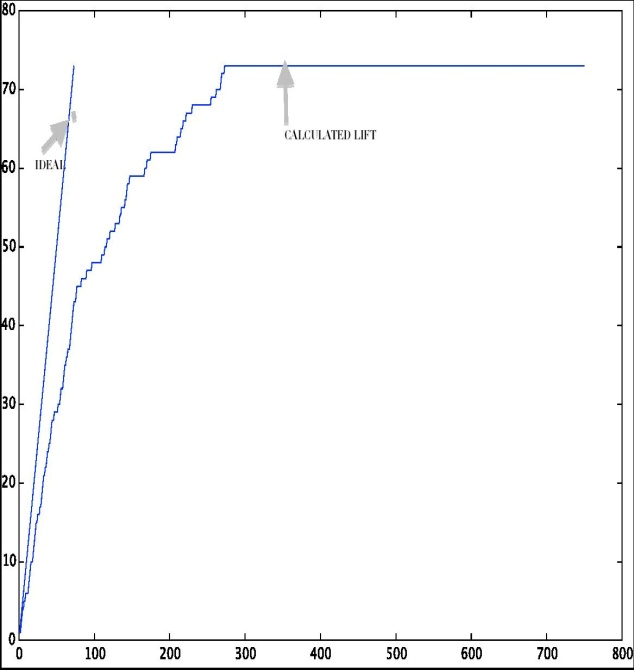


Figure : Lift Curve

Ideal case is when first 73 customers are contacted and all 73 subscribe. Thus, it is desirable that the area between the calculated and ideal lift curve be as small as possible.

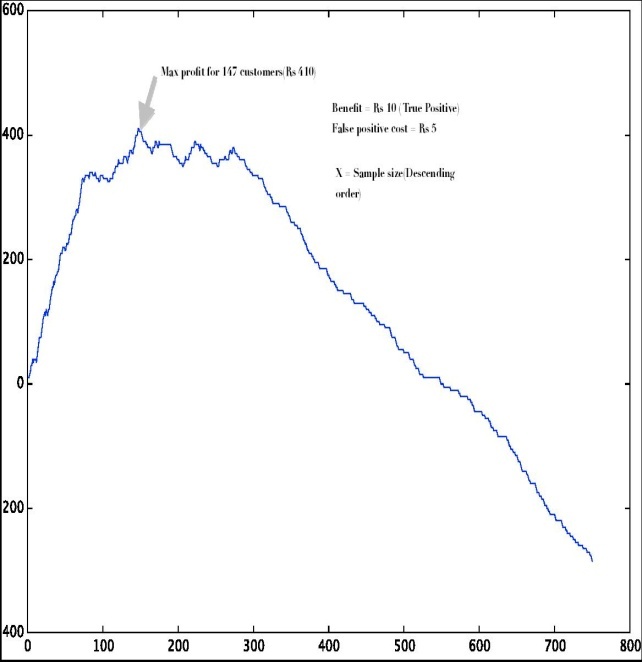


Figure : Naïve Profit Curve

The y-axis represents profit in Rupees. (to be written in graph)

This curve is the profit curve (also known as naïve profit curve). We have associated a benefit of Rs 10 for true positives and a penalty of Rs 5 for false positives. It can be seen that we need to contact only the first 147 customers to obtain maximum profit.

The under fit scenario is corrected by a more sophisticated algorithm such as Support vector machine(SVM). Regardless of high bias, we got precision as 93% and recall around 38%.

## 5.4 Support Vector Machine (SVM)

SVM is an optimal margin classifier which aims at finding a decision boundary that maximizes the margin.

### 5.4.1 Notation

We are considering a binary classification problem with labels y and features x.

We use y ∈ {−1, 1} (instead of {0, 1}) to denote the class labels. Also, rather than parameterizing our linear classifier with the vector θ, we will use parameters w, b, and write our classifier as

hw,b(x) = g(wTx + b)

Eqn. 4

Here, g(z) = 1 if z ≥ 0, and g(z) = −1 otherwise. This “w, b” notation allows us to explicitly treat the intercept term b separately from the other parameters.

Note also that, from our definition of g above, our classifier will directly predict either 1 or −1 (cf. the perceptron algorithm), without first going through the intermediate step of estimating the probability of y being 1(which was what logistic regression did).

### 5.4.2 Functional and Geometric Margin

Given a training example (x(i), y(i)), we define the functional margin of (w, b) with

respect to the training example

ˆγ(i) = y(i)(wT x + b).

Note that if y(i) = 1, then for the functional margin to be large (i.e., for our prediction to be confident and correct), we need wTx + b to be a large positive number. Conversely, if y(i) = −1, then for the functional margin to be large, we need wTx + b to be a large negative number. Moreover, if y(i)(wTx + b) > 0, then our prediction on this example is correct. Hence, a large functional margin represents a confident and a correct prediction.

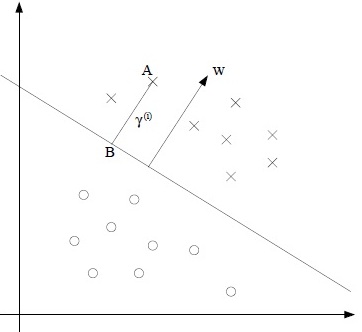


Figure : Max Margin Classifier

The decision boundary corresponding to (w, b) is shown, along with the vector w. Note that w is orthogonal (at 90◦) to the separating hyperplane. Consider the point at A, which represents the input x(i) of some training example with label y(i) = 1. Its distance to the decision boundary, γ(i), is given by the line segment AB.

w/||w|| is a unit-length vector pointing in the same direction as w. Since A represents x(i), we therefore find that the point B is given by x(i) − γ(i) ・ w/||w||. But this point lies on the decision boundary, and all points x on the decision boundary satisfy the equation wT x + b = 0.

γ(i) is the geometric margin and note that if ||w|| = 1, then the functional margin equals the geometric margin—this thus gives us a way of relating these two different notions of

margin. Also, the geometric margin is invariant to rescaling of the parameters; i.e., if we replace w with 2w and b with 2b, then the geometric margin does not change.

Specifically, because of this invariance to the scaling of the parameters, when trying to fit w and b to training data, we can impose an arbitrary scaling constraint on w without

changing anything important; for instance, we can demand that ||w|| = 1, or |w1| = 5, or |w1 + b| + |w2| = 2, and any of these can be satisfied simply by rescaling w and b.

### 5.4.2 The optimal margin classifier

We need to find a decision boundary that maximizes the (geometric) margin, since this would reflect a very confident set of predictions on the training set and a good “fit” to the training data. Specifically, this will result in a classifier that separates the positive and the negative training examples with a “gap” (geometric margin).

We can pose the following optimization problem:

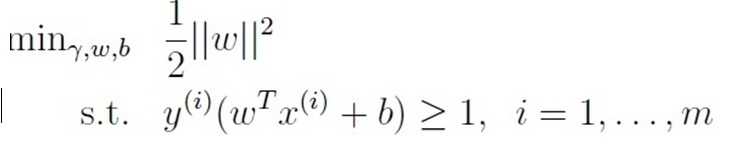
max,w,b γ

s.t. y(i)(wTx(i) + b) ≥ γ, i = 1, . . . ,m

||w|| = 1.

Eqn. 5

Through the above discussions and referring to the mathematics behind SVM, the following optimization problem can be posed.

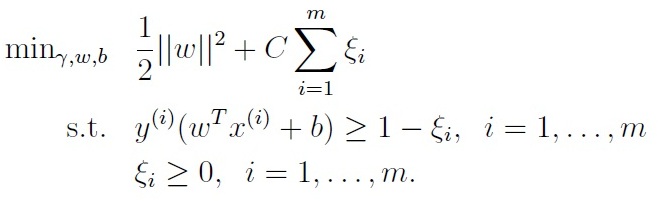


Eqn. 6

This can be solved using off the shelve QP software.

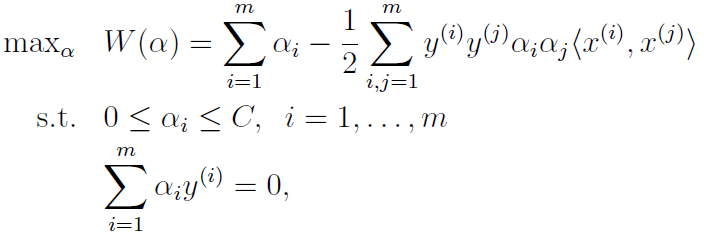
Here, we digress and construct Lagrangian for our optimization problem. We obtain the following dual optimization problem

Reformulated optimization problem taking outliers in consideration:



Eqn. 7

Reformulated dual optimization w.r.t alpha :

Eqn.8

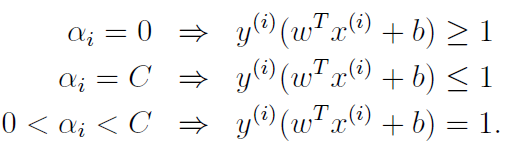
Where alphas are the lagrangian parameters.

### 5.4.3 Kernels

Using kernels, learning can happen in a high dimensional feature space. Here, a Gaussian Kernel is used (which corresponds to an infinite dimensional feature mapping). The linear kernel in our case yielded a comparatively poor result. Any other Kernel can be used; a valid kernel is one which satisfies the Mercer’s theorem.

Suppose for now that K is indeed a valid kernel corresponding to some feature mapping φ. Now, consider some finite set of m points (not necessarily the training set) {x(1),...,x(m)}, and let a square, m-by-m matrix K be defined so that its (i,j)-entry is given by Kij = K(x(i),x(j)). This matrix is called the Kernel matrix. for K to be a valid (Mercer) kernel, it is necessary and sufficient that for any {x(1),...,x(m)}, (m < ∞), the corresponding kernel matrix is symmetric positive semi-definite.

Alpha-i is optimal when is satisfies the KKT dual-complementarity conditions.



Eqn. 9

### 5.4.4 Sequential Minimal Optimization (SMO)

Optimization of alpha :

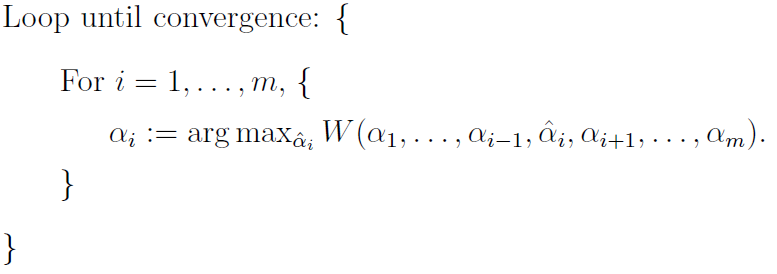
Coordinate ascent:

Trying to solve the unconstrained optimization problem:



Eqn. 10

Algorithm :



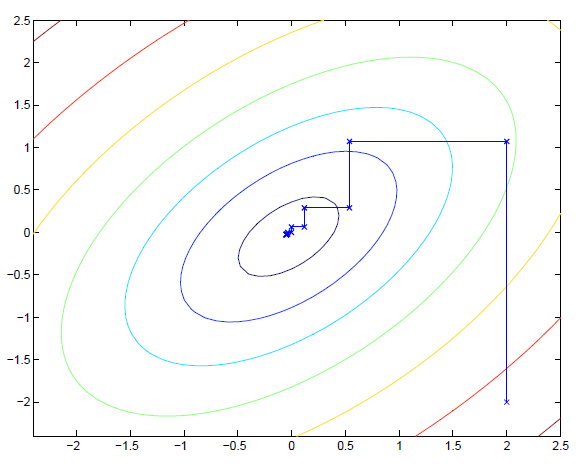


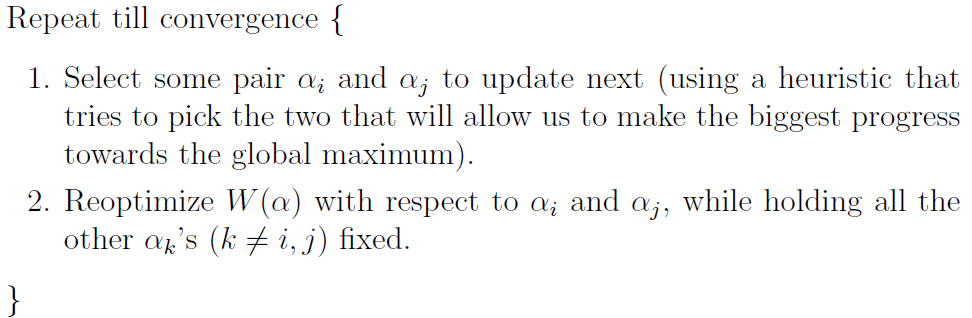
Figure : Coordinate Ascent

Can’t be used as may violate Summation(alpha\*y(1))= 0.

Sequential Minimal Optimization (SMO):

Optimizes 2 alphas at a time thus doesn’t violate the condition mentioned in the previous slide.

Algorithm :



The SMO algorithm allows to solve the dual problem originating from the derivation of SVM efficiently. There are essentially three aspects in the SMO algorithm : Selecting alpha parameters, optimizing alphas(Lagrangian multiplier) and computing the b threshold. Alpha is optimized when it satisfies the Karush-Kuhn-Tucker(KKT) conditions.

Several search heuristics can be employed to select the alpha parameters.

### 5.4.5 Graphs and Observations



Figure : Naïve Profit Curve

X-axis : Sample size (in descending order of probabilities)

Y-axis : Profit (here, in Rs)

Using SVM, we got number of true positives as 43. We see that these 43 are captured within the first 100 of sample size. The graph also suggests that we need to contact the first 100 customers to obtain max profit.

When compared to profit curve obtained from logistic regression, we see that approximately the same max. profit value is obtained by contacting fewer customers (100 in comparison to 300).

It can be said that although objectively similar profits are obtained by both methods, SVM has the upper hand because it contacts fewer customers and subsequently less communication is required on part of the communication (which may be deemed as subjective cost).

Other classification techniques can be used to address the asymmetric cost function such as Ada Boost algorithm or the software LibSVM.

# 5.5 Neural Networks

### 5.5.1 Introduction

NNs is one of the most powerful learning algorithms. It is a learning algorithm for fitting the derived parameters given a training set. The focus on application of NNs is for classification problems The notations used are as follows:

* Training set is {(x1, y1), (x2, y2), (x3, y3) ... (x*n*, y*m*)
* *L* = number of layers in the network.
* sl = number of units (not counting bias unit) in layer l

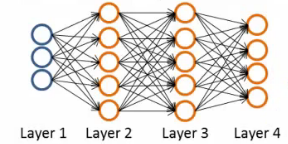


Figure : Neural Networks Diagram

* So here
  + l   = 4
  + s1 = 3
  + s2 = 5
  + s3 = 5
  + s4 = 4

### 5.5.2 Types of classification problems with NNs

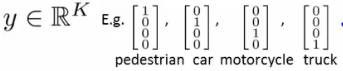
There are two types of classification that can be done using neural networks. They are as follows:

* **Binary classification**

It has only one output (0 or 1). So there is single output node thus the output value is going to be a real number. Notation k is number of units in output layer. In binary classification this is always 1.

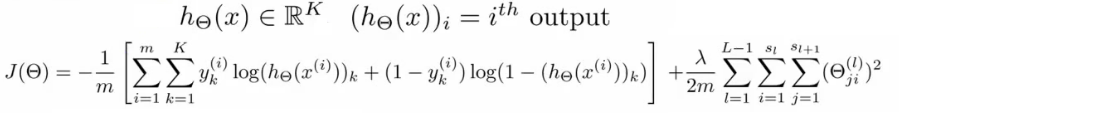
* **Multi-class classification**

It has k distinct classifications. Typically k is greater than or equal to three. If there only two then it is recommended to go for binary classification. The number of output units is equal to k. So y that is the output, is a k-dimensional vector of real numbers



### 5.5.3 Cost function for neural networks

For neural networks, cost function is a generalization of logistic regression equation above, so instead of one output we generate *k* outputs. Thus the cost function is as follows:



This cost function now outputs a *k* dimensional vector where hƟ(x) is a k dimensional vector, so hƟ(x)*i* refers to the ith value in that vector. Costfunction J(Ɵ) is [-1/m] times a sum of a similar term to which we had for logic regression. But now this is also a sum from k = 1 through to K (K is number of output nodes). Summation is a sum over the k output units that is for each of the possible classes. There is no need to sum over the bias terms (hence starting at 1 for the summation) Even if you do and end up regularizing the bias term this is not a big problem.

### 5.5.4 Back Propagation Overview

The backpropogation algorithm has two phases namely forward propogation and back propogation. They are described as follows:

* Forward Propagation

This is the algorithm which takes the neural network and the initial input into that network and pushes the input through the network. This leads to the generation of an output hypothesis, which may be a single real number, but can also be a vector

* Back Propagation

Back propagation basically takes the output you got from your network, compares it to the real value (y) and calculates how wrong the network was (i.e. how wrong the parameters were). It then, using the error just calculated, back-calculates the error associated with each unit from the preceding layer (i.e. layer *L -* 1). This goes on until you reach the input layer (where there is no error, as the activation is the input). These error measurements for each unit can be used to calculate the **partial derivatives.** Partial derivatives are very useful, because gradient descent needs them to minimize the cost function .We use the partial derivatives with gradient descent to try minimize the cost function and update all the Ɵ values. This repeats until gradient descent reports convergence.

There is a Ɵ matrix for each layer in the network. This has each node in layer l as one dimension and each node in l+1 as the other dimension. Similarly, there is going to be a Δ matrix for each layer.This has each node as one dimension and each training data example as the other.

### 5.5.5 The Back Propagation Algorithm

Let us consider a Training set of m examples as follows

http://www.holehouse.org/mlclass/09_Neural_Networks_Learning_files/Image%20%5b13%5d.png

First, set the delta values equal to 0  
http://www.holehouse.org/mlclass/09_Neural_Networks_Learning_files/Image%20%5b14%5d.png

Eventually these Δ values will be used to compute the partial derivative which will be used as accumulators for computing the partial derivatives.

**Next**, loop through the training set

http://www.holehouse.org/mlclass/09_Neural_Networks_Learning_files/Image%20%5b15%5d.png

i.e. for each example in the training set (dealing with each example as (x,y). Set a1(activation of input layer) = xi . Perform forward propagation to compute alfor each layer (l = 1,2, ... L) that is run forward propagation Then, use the output label for the specific example we're looking at to calculate δL where δL= aL- yi. So we initially calculate the delta value for the output layer.

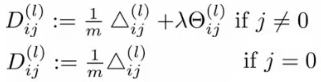
Then, using back propagation we move back through the network from layer L-1 down to layer .

Finally, use Δ to accumulate the partial derivative terms

http://www.holehouse.org/mlclass/09_Neural_Networks_Learning_files/Image%20%5b16%5d.png

Note here l = layer , j = node in that layer , i = the error of the affected node in the target layer You can vectorize the Δ expression too.

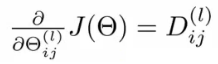
**Finally** After executing the body of the loop, exit the for loop and compute



When j = 0 we have no regularization term

At the end of ALL this All the *D* terms have been calculated using Δ. Each D term above is a real number

We can show that each D is equal to the following



We have calculated the partial derivative for each parameter. We can then use these in gradient descent or one of the advanced optimization algorithms

### 5.5.6 Graphs and observations

The various architectures tried and their observations are as follows

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Architecture | Hidden Layers | Hidden Units | True Positives | False Positives |
| 1 | 1 | 15 | 37 | 266 |
| 2 | 1 | 25 | 59 | 339 |
| 3 | 1 | 32 | 56 | 505 |
| 4 | 1with reg | 25 | 44 | 189 |
| 5 | 1 with reg | 32 | 56 | 243 |
| 6 | 2 | 20 | 38 | 298 |

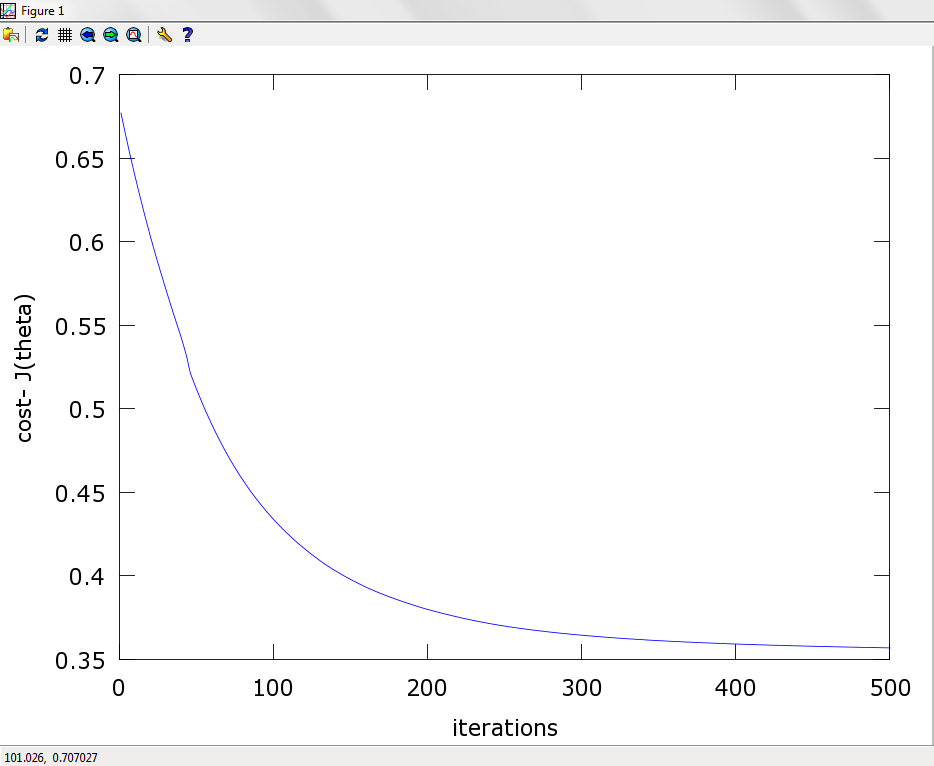


Figure : Iterations vs Cost

In Fig. , value of **alpha** is 0.01. We see that minimum error is obtained somewhere after 500 iterations. The objective of this exercise is to find a suitable tradeoff between **alpha** and the number of iterations used for an optimization algorithm such as gradient descent.

Lift Curve

Number of true positives obtained were 44. We see from the curve that around first 300 customers need to be contacted to capture all the subscribers.

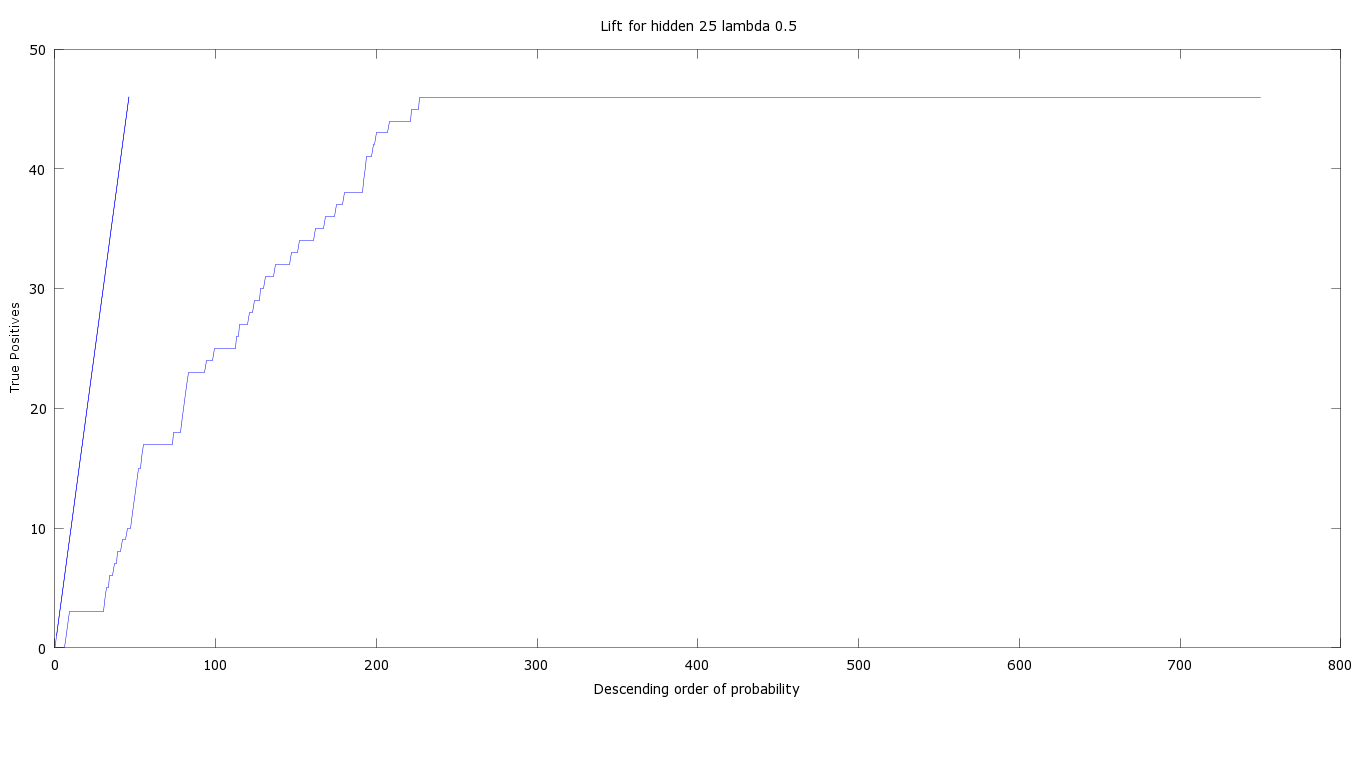


Figure : Lift Curve

Number of true positives obtained were 44. We see from the curve that around first 300 customers need to be contacted to capture all the subscribers.

Profit curve

The maximum profit obtained was Rs.215

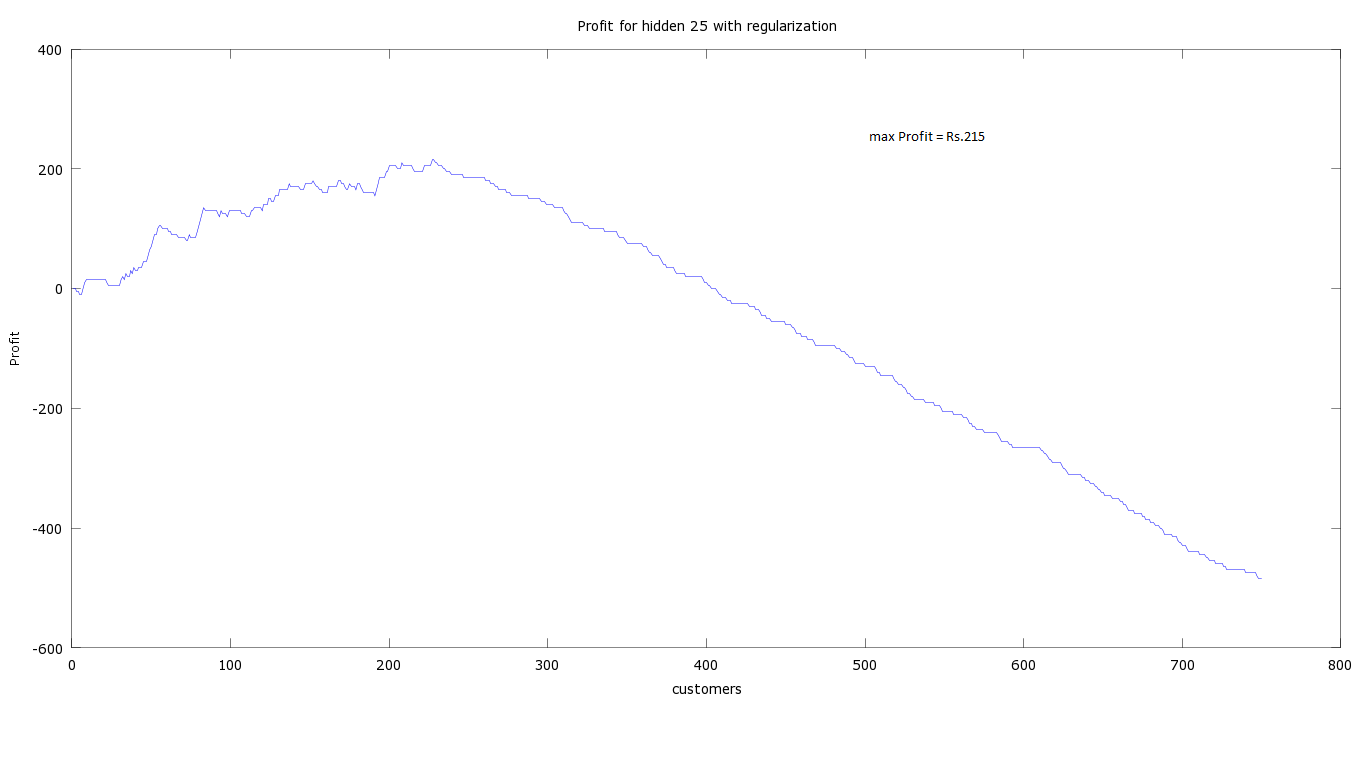


Figure : Naïve Profit Curve

# 6. TECHNOLOGIES USED

Octave, GNU Plot.

We've used octave, which has the same syntax as MATLAB, and provides a LOT more insight into the implementation of the algorithms and data analysis in general(As compared to softwares like Weka which only require passing parameters into inbuilt functions). Same was also done in R for comparing results.

# 7. PROJECT TIMELINE

1. Choice of Domain (Dec-Jan)
2. Data Set acquisition (Dec-Jan)
3. Selection of corresponding data mining algorithms (Jan-Feb)
4. Algorithms Implementation (Feb-Mar)
5. Refinement and Fine-Tuning (Feb-Mar)
6. Results (Feb-Mar)

# 8. FUTURE WORK

Dimensionality reduction and Principal component analysis, Optimization of algorithm parameters.

# 9. CONCLUSION

This study examined how data mining via domain driven data mining can be applied to businesses in order to yield more useful results. Three case studies were reviewed which show the effectiveness and efficacy of this method in the business domain. To improve customer relationship, the company must know what actions to take to optimize its communication with customers. A comparative analysis of different classification algorithms for bank direct marketing was done. We found that SVM would be the better choice among the chosen algorithms. Areas of future study could be the expansion of the scope to consider other fields such as agricultural, engineering and medical applications.

# 10. REFERENCES

[1] Toward Domain-Driven Data MiningZhengxiang Zhu1, Jifa Gu2, Wenxin Yang3, Xingsen Li4

[2] Mining Pattern of Supplier with the Methodology of Domain-Driven Data Mining Xu Xu, Jie Lin and Dongming Xu

[3] Domain Driven Data Mining to Improve Promotional Campaign ROI and Select Marketing Channels **-** Thomas Piton∗ , Julien Blanchard, Henri Briand and Fabrice Guillet

[4] Research Challenges for Data Mining in Science and Engineering - Jiawei Han and Jing Gao  
University of Illinois at Urbana-Champaign

[5] Evaluating Marketing Campaigns of Banking Using Neural Networks. Proceedings of the World Congress on Engineering 2013 Vol II, WCE 2013, July 3 - 5, 2013, London, U.K.Qeethara Kadhim Al-Shayea, *Member, IAENG*

[6]International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-2, Issue-6, August 2013

Bank Direct Marketing Based on Neural Network and C5.0 Models - Hany. A. Elsalamony, Alaa. M. Elsayad

[7] Data Mining In Excel: Lecture Notes and Cases Draft December 30, 2005

Galit Shmueli Nitin R. Patel Peter C. Bruce

[8] Using data mining for bank direct marketing: an application of the crisp-dm methodology Sérgio Moro and Raul M. S. Laureano, Paulo Cortez

[9] International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-3, Issue-4, September 2013

A Comparison of Different Classification Techniques for Bank Direct Marketing - K. Wisaeng

[10] Fast training for support vector machines using sequential minimal optimization

John C Platt, Microsoft Research.

[11] Probabilistic outputs for support vector machines and comparison to regularized likelihood methods, John C Platt, Microsoft Research.

[12] A Few Useful Things to Know about Machine Learning -Pedro Domingos  
Department of Computer Science and Engineering University of Washington

[13] Expert Systems with Applications 36 (2009) 2592–2602 Review

Application of data mining techniques in customer relationship management: A literature review and classification - E.W.T. Ngai, Li Xiu, D.C.K. Chau

[14] UCI Machine Learning Repository, "UCI Machine Learning Repository: Bank Marketing Data Set". Available:

<http://archive.ics.uci.edu/ml/datasets/Bank+Marketing>

[15] Platt, John. Fast Training of Support Vector Machines using Sequential Minimal Optimization

, in Advances in Kernel Methods – Support Vector Learning, B. Scholkopf, C. Burges,

A. Smola, eds., MIT Press (1998)