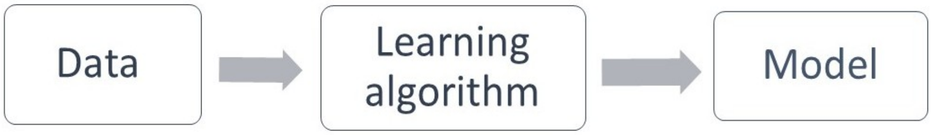
Machine Learning: An Introductory Survey

Mukund S

January 2015

1 Introduction

Machine learning is the science of understanding data by learning models which can be used to make good decisions on unseen data. It is a interdisciplinary area that combines ideas from probability theory, optimization, linear algebra etc to solve problems such as ranking various objects, classify the data, detect patterns in the data, predict answers with good amount of accuracy for different cases even before they are observed and so on. Since similar data types can be solved with similar methods, it is good to identify the data type and characterize the problems according to the type of data.



This is how a typical learning problem looks like. Given the training data we try to design a learning algorithm which learns a model that can output the desired value.

Definition: A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.[10]

Machine Learning algorithms can be classified into different types based on the tpye of problems.

2 Supervised Learning:

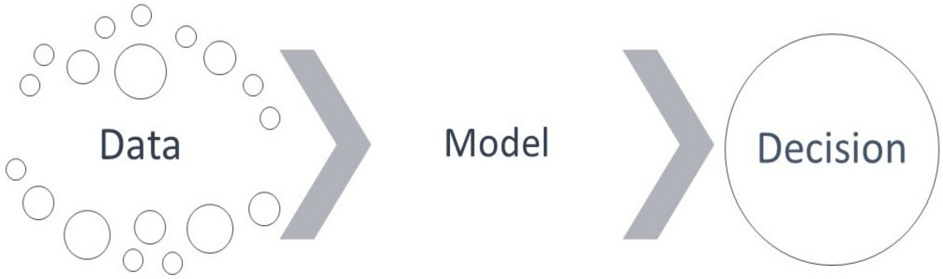
Supervised Learning refers to dealing with classification problems and also pattern recognition problems.

Supervised learning can be defined as the task of taking the labeled data sets, extracting information from it and labeling new data sets or mapping to desired output value using function approximation. Here the supervision refers to provision of desired output values(can be labels or real values) to the input data.

2.1 Classification Problem:

Classification is the process of taking some input and mapping it to some discrete label. Here we supervise the labeling of the input data. For example classifying emails as spam or not spam is a classification problem where the training set consists of previous emails which have been labelled spam or not spam and the classification algorithm labels the new incoming email spam or not spam. Assume there are 'n' different objects belonging to 'n' categories labelled with

’n’ different labels. Now object identification in images can also be thought of as an classification problem where training data consists of images where the object in image is labelled and given an new image the algorithm tries to identify an object and label it using one of the n labels if that object belongs to one among n categories otherwise labels it unknown.



A model is learnt using the training data. Now on giving new data as input the model decides the label of the data point.

2.1.1 Linear Classifiers:

Linear models for classification separate input vectors into classes using linear (hyperplane decision boundaries. The simple case would be the Two Class Discriminant Function.

2.1.2 Decision Trees:

Decision trees are very efficient and one among the widely used techniques for classification problems. Decision trees are built from set of decision rules such that each path from root to a leaf is a rule and leaf nodes are different classes. ID3 is a standard and popular algorithm where the features are ranked based on their information gain and trees are built using the features with information gain in decreasing order.[4, chapter 1]

2.1.3 Nearest Neighbour Algorithm:

In Nearest neighbour algorithm all the features are weighted equal importance. In K nearest neighbour algorithm, k nearest points to the unlabeled point are found and unlabeled point is classified based on majority class among the k nearest points. the main drawback is the complexity involved in finding the k nearest points and this method is effected due to feature scaling.[4, chapter 2]

2.1.4 Decision Boundaries:

D-dimensional space is sorted into 2 regions where each region represents each class. The drawback of this model is that all the features are assigned equal weights and only finitely many features need to be selected. [2]

2.1.5 Perceptron:

The goal of perceptron learning is to find a separating hyperplane by minimizing the distance of the misclassified points to the decision boundary. Each feature is assigned with a particular weight. Activation value is calculated and then bias is introduced. Later the weights and bias are updated for which many methods have been proposed one of which is stochastic gradient descent method. The algorithm is trained with the data over and over, each time permuting the data set. Improvised generalizations of this algorithm can be found. In Voted Perceptron, the hyperplane which has survived most number of times is selected and in Averaged Perceptron, all the survived hyperplanes are remembered and their weighted average is calculated. In perceptron decision boundaries cannot be non-linear.[4, chapter 3]

2.1.6 Logistic Regression:

Logistic Regression is same as Linear Regression(discussed under the section Regression) except that the hypothesis function is the estimated probability that output is 1 on input.

2.1.7 Neural Networks:

The idea of artificial neural networks arose from the functioning of neurons in our brains. These are generalization of perceptrons and are basically non linear decision boundaries. Neural networks can produce any Boolean function. Activations of the hidden units are computed based on the inputs and the input weights. Then the activations of the output unit are computed using the hidden unit activations and hidden layer weights. Hidden layers compute a non linear function of their inputs. Multiclass classification can also be computed using one vs all and other algorithms. [3]

Back propagation algorithm can be used for gradient computation and cost function minimization. During forward propagation, given one training example all the activations are calculated. The errors are propagated backwards. Gradients are numerically estimated using gradient checking. this is where actually the learning takes place, the weights are adjusted.

2.1.8 Support Vector Machines:

The goal of SVM is to design a hyperplane that classifies all training points into different classes. The best hyperplane will be the one that leaves the maximum margin from both classes. Hyperplanes are defined by the equations which have weight weight vector and these equations will deliver different values for all the input points belonging to different classes. Minimizing the weight vector will maximize the margin. Minimizing weight vector is a nonlinear optimization problem and can solved by Karush-Kuhn-Tucker conditions using Lagrange multipliers. [3]

2.1.9 Multiclass Classification:

Multiclass classification is classification problem where there are more than 2 classes. Most of the above mentioned algorithms can be used to classify into multiple classes with required improvements. Few popular models for multiclass classification are presented below. [2]

2.1.10 One vs All Algorithm:

In OVA algorithm, assuming there are K classes, K many binary classifiers are trained such that each classifier shows positive value for each class. Then the classifiers are trained using the data and the classifier which assigns the positive value to the data point is assigned the corresponding class.[2]

2.1.11 All vs All Algorithm:

In this algorithm assuming there are K classes $\binom{K}{2}$ binary classifiers(for every pair of classes) are trained and the class with more value for a particular data point is assigned the corresponding class.[2]

Feature Normalization and feature pruning are few techniques which are useful to remove irrelevant features and also help making comparisons easier across the data.[7]

2.2 Regression:

Generally in regression we deal with continued valued functions i.e., given an input we try to map it to a real value. Assume we are given the average child height at "n" different ages say for example at age 4 - 40inches, age 5 - 43inches, age 6 - 46inches,age 8 - 50inches,age 10 - 53inches, age 12 - 58inches. And now we would like to guess the height of the child when his age was 9 or when his age will be 16. This is an example of a regression problem where we predict height(ordered real values) by trying to map age to height through some function approximation. Tumors are classified benign or malignant based on tumor size. So inorder to determine whether the tumor is benign or malignant we would like to estimate the size of the tumor given his/her previous medical reports which contain the information age, gender, size of tumor, whether the tumor is benign or malignant etc,. This is another example of regression problem where we would like to map patient's past medical info to size of the tumor which can be either discrete or indiscrete real value. In classification the desired outputs are categorized while in regression the outputs are either continuous or discrete numerical real values.

2.2.1 Linear Regression with One Variable:

In this algorithm, we try to find a linear function that fits the given data. This can be done by minimizing the squared error i.e., squared deviation between linear function and the data points. This minimization can be done using Gradient Descent Algorithm. Gradient Descent algorithm doesn't always converge to global minimum but it minimizes to local minimum.

This method can also be extended to finding a polynomial function that can be fitted to the given data. In this case minimization of error can be done using Normal Equation method. Although the aim of the algorithm is to find the most precise output of the new input data rather than finding a polynomial that fits the given data with no errors. Hence over-fitting and under-fitting need to be taken care of while approximating a function. [7]

2.2.2 Bayesian Learning:

Bayesian learning is based on the Bayes rules. For each candidate hypothesis in hypothesis space, probability of the hypothesis is calculated given the data and output is the hypothesis which has the maximum probability. But we don't actually try to find the Maximum likelihood hypothesis or MAP hypothesis rather we compute the full posterior distribution over all possible parameter settings. This is extremely computationally intensive. And then we can make predictions by letting each different setting parameter make its own prediction and then averaging these predictions together weighting by their posterior probability.[3]

3 Unsupervised Learning:

In unsupervised learning we essentially have an input data set and and we try to derive some structure from them by looking at the relationships between the inputs.[6]

3.0.3 Clustering:

Clustering is a type of unsupervised learning that automatically forms clusters of similar things. This notion of similarity depends on the similarity measurement which inturn depends on the type of application and data set.[4]

3.0.4 K-Means Clustering:

In K-means Clustering algorithm k unique clusters are found and center of each cluster is the mean of the values of the elements in that cluster. Initially number of clusters(K) needed is chosen and K points are chosen in the data set which are assumed to represent the centroid of K clusters. Each point in the data set is assigned to one of the K points for which the distance from the centroid is minimum. After a point is assigned to a cluster, centroids are updated by calculating the mean of all the points in each cluster. This method converges at local minima but is slow on very large data sets. It doesn't necessarily need to converge to a global minimum.[4]

3.0.5 Bisecting K-means:

In this algorithm initially all the points are assigned to a single cluster. While the total number of clusters is less than K, total error is measured for each cluster. Then K-means clustering with K = 2 is performed on the cluster with largest total error and the process is repeated till desired number of clusters are formed.[4]

4 Online Learning:

Online learning helps us to make accurate predictions having the knowledge of correct answer to previous predictions. The algorithms are run in a sequence of consecutive rounds where in each round the algorithm needs to predict an answer for the question and then the true outcome is revealed. The errors are calculated and learning models are updated accordingly to improve the accuracy of prediction.

Regret:

Regret measures the performance of the online learning algorithm with respect to the experts used in the algorithm. Mathematically regret R$_{E,n}$ is defined as

$$R_{E,n} = \sum_{t=1}^{n} (l(\hat{p_t}, y_t) - l(f_{E,t}, \mathrm{y}_t))$$

where $l()$ is the loss function, n is the total number of examples trained so far, $\hat{p_t}$ is the prediction by the forecaster for t^{th} output, y_t is the true outcome and $f_{E,t}$ is the prediction by the i^{th} expert for t^{th} input.

4.1 Prediction with Expert Advice:

In this learning model there are finite number of Experts where each expert predicts an answer and the predicted answer is made available to the forecaster(learning model). In each round, the forecaster can make use of these expert's predictions for predicting the answer. Later the true answer is revealed. Forecaster suffers regret and experts suffer loss. The goal of this algorithm is to minimize the regret. Various models are proposed to solve this minimization problem.[1]

4.1.1 Weighted Majority Algorithm:

In this algorithm each expert is assigned a particular weight and forecaster predicts the weighted average of the expert's predictions. After the true outcome is revealed losses are calculated and weights are updated accordingly.[1]

4.1.2 Randomized Weighted Majority Algorithm:

In this algorithm initially all the experts are assigned equal weights(equal to 1) and the forecaster predicts the weighted average of the expert's advice. Experts may suffer loss and weights of experts who make a mistake is penalized by ratio β . At round t, the ratio of weight of an expert to total sum of weights of all experts can be thought of as probability of that expert predicting the right answer. [9]

4.1.3 Online Gradient Descent:

Generally in Gradient Descent algorithm we run through all the samples in the training set to do a single update for parameter in a particular iteration while in online gradient descent, only one training sample is used from the training set, to do the update for the parameter in a single iteration. so, if the training set is very large, online gradient descent will be faster since only one point is used to update the parameter. Online gradient descent converges faster than gradient descent algorithm but is not as well minimized as the gradient descent algorithm. This method can also be used for the minimization problem of Prediction with Expert Advice. [8]

4.2 Perceptron:

When there are only two possible outputs(say -1 or 1) this algorithm can be used to predict the output in an online learning problem. This method tries to find a hyperplane that divides the vector space where each part denotes one output. We assume there are N experts and each expert is assigned a weight. The forecaster predicts its output based on the sign of the weighted sum of expert's choices. after the true outcome is revealed the weights are updated accordingly. [9]

4.2.1 Follow the Leader:

Follow the leader algorithm can be used to minimize the regret over all of the previous time steps. This is done by selecting the hypothesis in every round for which the loss suffered by the forecaster is minimum over the previous steps. The main backdraw of this algorithm is that it sometimes doesn't guarantee the selection of hypothesis with minimal regret. [5]

4.2.2 Follow the Regularized Leader:

The objective of this algorithm is to improve on the mistakes made by the follow the leader algorithm. This is achieved by adding a penalty function to the function minimizing the regret. Better maximum bounds on the regret can be attained by choosing appropriate penalty function. [5] [1]

4.2.3 Follow the Perturbed Leader:

In Follow the Perturbed Leader algorithm a small random perturbation is added to the cumulative losses. Forecaster chooses the expert which has the least cumulative loss. Assume there are N experts. $Z_1, Z_2, ...$ be identically distributed N-random vectors with components $Z_{i,t}$. Z_t has absolutely continuous distribution with respect to N-dimensional Lebesgue measure. Forecaster chooses the expert which has the minimum value of $L_{i,(t-1)}$ + $\mathrm{Z}_{i,t}$

Mini Project

Task:

To predict the kind of ball a bowler bowls during a cricket match based on his previous bowling statistics.

Data:

Commentary during the bowling of the bowler Dayle Steyn during the test matches test1944, test1946, test1948 and test2049 between South Africa and England is downloaded from the website www.cricinfo.com and is semantically analysed and used as the training data.

Idea:

The line and length of the next ball being bowled can be predicted by training the machine with the data (statistics) of his previous balls that have been bowled. We have assumed that the number of variations in the bowling to be finite.

Algorithm Overview:

Each output (next ball) can be predicted using Prediction with Expert Advice where each expert is assumed to predict constantly single possible output. Since the total number of outputs that can be predicted are finite number of experts is also finite.

Variables in the output are line and length. Line of the delivery can be chosen in 5 different ways, In swinger, Out swinger, Reverse swing, Outside off, Straight. And similarly length can be chosen in 5 different ways, Very short, Short, Good length, Yorker, Fuller Delivery. So the total number of possible outcomes is 25. Each expert constantly predicts one possible outcome. Now the machine predicts the outcome using the predictions of the 'n' experts using the Randomized Weighted Majority algorithm. Initially the weights of each expert is adjusted to 1. Then the machine predicts the outcome and then true outcome is revealed. Then loss and regret are calculated. And if the prediction of a particular expert is true then its weight remains constant otherwise its weight is decreased by fraction β. This is iteratively run for large number of examples so that the machine gets trained and bounds the number of mistakes made when compared to the best expert (expert which makes least number of errors). Regrets and losses for each expert is calculated and best among the experts is found.

β = penalizing factor = $\frac{1}{1+\sqrt{\frac{2lnN}{M}}}$

f_i^t denotes the prediction by the i^{th} expert in round 't'.

W_i = Weight of the i^{th} expert predicting the outcome(assume that all possible outcomes are tabulated in some definite order).

W' = Total weight of all experts after 't' predictions by the machine.

N = total weight of the experts initially.

m = number of training data.

If i^{th} expert has made a wrong prediction, then its weight is updated as:

$$W_i \leftarrow W_i \times \beta$$

$$W' \leq N \times \tfrac{(1+\beta)^n}{2^n}$$

Expert i is expected to predict correctly with the probability $\frac{W_i}{W'}$ where W_i are updated weights.
If the best expert makes 'M' mistakes then the expected number of mistakes

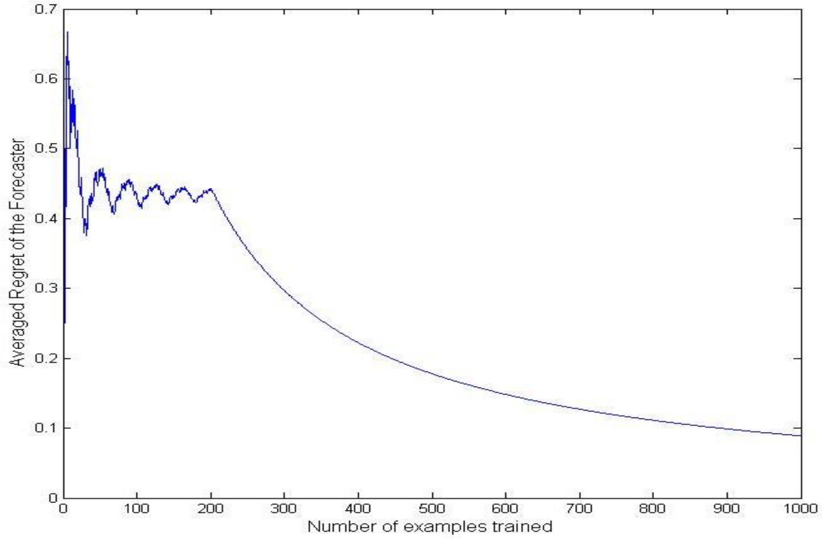
$$M' \leq \frac{\log N}{\beta} + M$$

Algorithm:

Algorithm 1 Predicting the output and updating the weights

```
**Step:1** Read the input from the text file which contains predictions of N experts and parameter $\beta$
**Step:2** Adjust weights of all the experts to 1.    $W_i \leftarrow 1$
**Step:3** Scan the input data.
**Step:4** for $i = 1 : m$
•$\widehat{y_t} \leftarrow round(\frac{\sum(W_i) \times f_i^t}{W'})$ prediction of the forecaster is computed.
•Output $y_t$ is revealed.
• If forecaster's prediction is wrong, then weights are updated by $W_i^{t+1} \leftarrow W_i^t \times \beta$
•Expected loss is calculated according to the 0,1 loss function for each expert and forecaster for every training point.
• Regrets with respect to each expert is calculated.
• Best expert is found and regret with respect to best expert is found.
End.
```

Figure 1: Averaged Regret of the forecaster with respect to the best expert over the entire data set with $\beta = 0.917$



The following plot shows the progress of the algorithm in learning to predict as good as the best expert. The averaged regret is decreasing with the number of examples trained. From this we can infer that our algorithm is learning to predict as good as the best expert.

The number of mistakes made by the best expert is found to be 730.

References

[1] Nicolo Bianchi. *Prediction, learning, and games.* Cambridge University Press, Cambridge New York, 2006.

[2] Christopher Bishop. *Pattern recognition and machine learning.* Springer, New York, 2006.

[3] Micheal Charles. Machine learning: Supervised learning, 2014. [Online, accessed June-2014].

[4] Hal Daumé III. *A Course in Mahcine Learning.* TODO, September 2013.

[5] Roni Khardon. *Lecture 17 , COMP236: Computational Learning Theory , Department of Computer Science , Tufts University.* 2013.

[6] Kevin Murphy. *Machine Learning: A Probabilistic Perspective.* The MIT Press, August 2012.

[7] Andrew Ng. Coursera - machine learning 2014.

[8] Andrew Ng. *CS229 Lecture Notes*.

[9] Robert Schaphire. *COS: 511 , Foundations of Machine Learning ,Lecture 14 , Princeton Univesity*. march 2006. (Visited on 01/10/2015).

[10] Wikipedia. Machine learning — wikipedia, the free encyclopedia, 2015. [Online; accessed 9-January-2015].