Machine Learning Notes

Machine Learning

1. Linear Regression(Supervised learning)

1.1 **Linear Algebra Tricks** :-

* If we have a linear function of a linear model which tries to predict price of houses in Florida based on distance from beach which is x1 and size of house which is x2

like f(x) = ⊖x0 + ⊖x1 + ⊖x2

Dataset schema = [x,x2,f(x) or y]

Dataset =([10,10,5], [1,1,5], [10,1,5], [1,10,5], [101,1,5], [110,110,15])

A neat trick which can be used if we have a large data sets we want to model dataset with the function

⊖0

⊖ = ⊖1

⊖2

x0 x1 x2 y

10 10 5

1 1 1 5

x = 1 10 1 5

1 1 10 5

1 101 1 5

1 110 110 15

The first column which contains ones is assumption that x0 to make it easier to be added, this trick will make it easier to calculate the cost function when summing the output of hypothesis function using the equation (Transpose(⊖) \* x).

* In case you want calculate normal function and matrix is not invertible take in consideration two factors Redundant Features, Too Many Features (if number of features greater than or equal number of rows in dataset)

1.2 **Optimisation tricks :**

In case you are not sure wether to use gradient descent(iterative approach) or normal function(one step approach) to decrease the cost function the factors you should look into is number of features compared to training example and efficiency, if number of features >= 10\*\*3 then Gradient Descent would make more sense than normal function because the big o notation of Normal function is O(n\*\*3) which is going to be remarkably slow if used with this number of features

2. Classification(Supervised learning)

**2.1 classification problem :-**

Classification problem deals with output of discrete nature not like Linear regression which deals with output of continuous nature like the above example of predicting house prices which output a continuous number(price), one of the simplest classifiers is binary classifier from it’s name you can infer that it’s either classify the input either as 1 or 0 a real world example for this can be spam detection the email you’re checking it’s either a spam or not there’s no a third case, another jargon in the classification problem is called threshold classifier output, depend on this threshold you classify input wether it belongs to class 0 or class 1 for example if it’s value is less than the threshold it belongs to 0 else it belongs to 1.

**2.1.1 Logistic regression :-**

The hypothesis or prediction function of a learning algorithm is how the algorithm predicts the output based on given input from the data-set and then comparing the predicted output of the hypothesis or prediction function with the real output from the data-set which is known as **cost function** to know how accurate our model is, the hypothesis function of Logistic regression is called Logistic function or Sigmoid function what a sigmoid function really is, it’s that it predict probability of a given value to happens or take place, so the output of this function is between 0 and 1 since it’s a probability value, so the notation of hypothesis can be something like h⊖(x) = g(⊖\*\*Tx)=P( y = 1 | x ; ⊖), which means the probability of y to be 1 given x has the parameters of ⊖ is whatever h⊖(x) gonna output.

Machine Learning Jargons and important definitions:

* A. **Novelty Detection** : this term mostly appears in Classification problems(un-supervised learning) where the model tries to classify the input into different classes based on their vector(input) but the model come across an input they have not seen before so they classify to the nearest class which is not correct, for example let’s say we are building a neural network for a vending machine which classifies German coins based on some features and you put in the vending a British coin it will classify the British in the class which is the most similar to it which is wrong(Example taken from Machine Learning (Algorithmic Perspective, Stephen MarsLand).
* B. **Fuzzy Classifier** : usually classification problem categorise the input into different classes/categories some input belongs to more than one class, such problem is handled by what is known as fuzzy classifier
* C. **Perceptron** : Perceptron is a type of a neural network which does have an input and the input map to an output, the input is fasten to the neurons through what is known as weights, this definition according to McCulloch and Pitts Neuron Model
* D.**Testing Data-set vs Training Data-set** : you will hear both these terms a lot when dealing with supervised learning problem because you test and train and tweak data to make it do whatever you want, you should also avoid using same dataset for both tasks, you’re simply testing literally the data you were training so it will create a bias problem
* E.**Confusion Matrix** : Confusion Matrix measure the percentage/number of retrieved documents by a classification model, so let’s say we are building a classifier which classifies people with blue eyes, our classified returned 5 persons(3 with blue eyes, 2 with brown eyes)out of 12 total persons 5 of which only have blue eyes, now we will classify the results into four categories, TP(True Positive), TN(True Negative), FN(False Negative), FP(False Positive), TP = 3, TN = 4, FN = 2, FP = 2, in a simpler words true positive means the number of persons classified correct, true negative means the number of persons classified as wrong and they are wrong, False Negative means the number of persons classified as Wrong and they are right, False positive means the number of persons classified as Right and they are wrong
* F.**Evaluation metrics (Precision vs Recall)** : Evaluation metrics is how you’re going to evaluate how efficient your algorithm was during training method so to know how efficient it was a lot of metrics and measures are used like : Precision, Accuracy, F-Score and different dozens of advanced evaluation metrics, I will talk about Precision and Accuracy but in order to understand it better refer to the Jargon with title confusion Matrix, Precision = TP / TP + FP, precision measure percentage of the results which is retrieved while Recall = (TP / TP + TN) measure percentage of correct results retrieved so Precision focus on quality of information retrieved while Recall focus on quantity of information retrieved.
* G.**Cost function** : cost function is used as an estimate for the prediction accuracy of a model, it shows how accurate is the prediction of a model in a supervised learning problem by subtracting the output of the prediction function minus the output of the real function, Cost function J(⊖) = 𝛴i h⊖((x\*\*i) - y\*\*i)\*\*2 where H⊖(x) = ⊖0 + ⊖1\*x1 + ⊖2\*x2 + …..
* H.**Feature scaling :** Features is the variables of a data-set, when building our prediction model in a classical supervised learning problem we have to take care that the value of features is skewed or reasonably in the same range for example having features with huge variance will cause a lot of problems like gradient descent is going to drastically slower in comparison with another dataset with closer ranges
* I. **Decision Boundary**: decision boundary is a graphical shape which separates different classes into regions in a classification problem where every output belongs to a class based on an input.
* J. **Over-fitting :** over-fitting implies that the hypothesis the machine learning model tries to reach lacks generalization, this does happens in supervised problems where the model fails to generalize a function after reading the data-set because of the noise in the data-set example it might contain too many features and some of those features are not representative or important in predicting the hypothesis function, when trying to control over-fitting we can try different approaches like Reducing number of features or Regularization.
* k. **under-fitting:** under-fitting implies that the data-set the model trained on is biased or not representative enough, it’s the opposite of over-fitting, in over-fitting the model learns the data with the noise in it, under-fitting the model doesn’t learn about the data in a proper way.
* L. **regularization**: regularization is one of the solutions/algorithms used to solve over-fitting problem by tuning
* M. **Gradient-Descent :** optimization algorithm which aim to reduce the cost function of algorithms (refer to point G for more information about the cost function), it achieves this through minimizing variables(⊖0) related to the features in the algorithm till it converge(the cost function become zero or near from zero).