LCC- SJCE presents

Machine Learning 1.0
Logistic regression

A CLASSIFICATION ALGORITHM

- Shashank R

Recap: Regression problem

- In the last session you understood what a regression problem was with respect to housing price prediction.
- That is, given a set of features, say { Number of rooms, Area of the house} you would predict at what price would the house be sold?
- Here you had the price to be a continuous output.

- **Now if we want to understand the same problem in terms of classification here is how you can think of it as.**

Classification problem

- From the last slide you saw predicting prices of house was a regression problem. Now what if the problem statement is slightly changed, and it states:

Given features {Number_of_rooms, Area_of_house} *classify* houses based on their pricing as follows,

If price < $200K : cheap (class-0)

If $200K< price < $500K : moderate (class-1)

If price > $500K: costly (class-2)

- Now the problem became classification. So classification is basically mapping set of features to discrete classes

Algorithms for classification problem

- There are many algorithms that aid as tools for solving classification problems. Some of them are,
  - Naïve Bayes classifier
  - K Nearest Neighbor classification
  - Decision Trees
  - **Logistic regression**
  - Neural networks etc.,

Binary classification

- In this session we will talk about only binary classification.
- In binary classification there are only two output classes. Class-0 and Class-1.
- **Examples:**
  - Given a mail classify it as a spam (class-1) or ham (class-0).
  - Given a picture, classify it as either a cat picture (class-1) or not (class-0).
  - In the example of housing prices, there are only two classes we are interested in, costly (class-1) or not costly (class-0)

Logistic Regression

- Logistic regression is a classification algorithm (here regression does not imply regression problem)
- **The goal:**

  Given an X, find out the probability that Y=1.

  In other words, we want to find P(Y=1|X).

  The training set contains **(X,Y)** pairs. Number of examples are **m**

  Each example **x** is **(1,n)** vector and corresponding **y = 0/1**

What to do?

- Fix up set of parameters: **W** (n,1) and **b**
- Compute a hypothesis, **Z = WX + b**
- To bring **Z** in the range [0,1] use activation function.
- Example of an activation function:
  - **Sigmoid function,**

$$\textbf{sigmoid (z)} = 1 / 1 + e^{-z}$$

  - Key points:

    i) It is a monotonic function

    ii) It is continuously differentiable

    iii) Output always in the range (0, 1)

Logistic regression cost function

- We are given (X,Y) = {($x^{(1)}$,$y^{(1)}$), ($x^{(2)}$,$y^{(2)}$),….., ($x^{(m)}$,$y^{(m)}$)}, m training examples
- For a single example, (x,y) the loss is computed as follows:

z = wx+b

a = sigmoid(z)

L(a,y) = -(y log(a) + (1-y) log(1-a))

- Now to compute the cost, take average Loss over training set.

$$J(w,b) = \frac{-1}{m} \sum_{i=1}^{m} L(a^{(i)}, y^{(i)})$$

- Now that will be what we will try to minimize using gradient descent

Gradient descent

- We need to minimize the cost function J(w,b) using gradient descent optimization.

$$J(w,b) = \frac{-1}{m}\sum_{i=1}^{m} L(a^{(i)}, y^{(i)})$$

- We do it as follows.

Repeat{

w = w – alpha * dw here dw = $\frac{dJ(w,b)}{dw}$

b = b – alpha * db here db = $\frac{dJ(w,b)}{db}$

}until convergence

Computation graph

- Computation graphs make it easy for us to understand what is going on in logistic regression.
- From computation graph for logistic regression we found out the following:
- For a single example,

$dz = a - y$

$dw = x\, dz$

$db = dz$

Now for m training examples

```
$J = 0, dw_1 = 0, dw_2 = 0, db=0$
For i = 1 to m do,
    $z^{(i)} = w x^{(i)} + b$                #forward propagation
    $a^{(i)} = sigmoid(z^{(i)})$
    J += compute cost(a , y) #homework
    $dz^{(i)} = a^{(i)} - y^{(i)}$                #back propagation
    $dw_1 += x_1^{(i)} * dz^{(i)}$
    $dw_2 += x_2^{(i)} * dz^{(i)}$
    $db += dz^{(i)}$
$dw_1$ /= m, $dw_2$ /= m, db /= m, J /= m
```

Vectorization

- In the previous slide, we saw how for loops were used, when there are large number of computations, it is better that we limit the usage of for loops, this is called vectorization.

- Vectorization provides two advantages:
  - It reduces usage of for loops and hence minimizes the code lines
  - It internally uses parallel instruction there by making computation fast.

Vectorized logistic regression

```
Logistic regression(X,Y){
    Initialize w, b to 0
    for i in range(iteration = 1500){
        #forward propagation
        Z = np.dot(W,X)+b
        A = sigmoid(Z)

        #compute cost
        J = computeCostVectorized(A,Y)

        #back propagation
        dZ = A - Y
        dW = 1/m *  (np.dot(X,dZ).T)
        db = 1/m * np.sum(dZ)

        #update parameters
        W = W - alpha * dW
        b = b - alpha * db
    }
    return W,b
}
```

Thank you

Any questions?

HAVE A NICE DAY

- Shashank R