HOMEWORK ASSIGNMENT #4
B555- MACHINE LEARNING

FNU ANIRUDH
(aanirudh@umail.iu.edu)

Solution 1

Please refer solutions at the bottom of the document.

Solution 2

a) Since this question had to be implemented from scratch hence I have implemented entire question in R.

Part (a) asks us to implement linear kernel which is given by formula:

$$k(x,y)= x^T y + c$$

Linear kernel is inner product of x and y and c is optional constant.

Output:-

"Accuracy for Linear Kernel is = "

> acc
[1] 73.92593

[1] "Accuracy for Linear Kernel is = "

> acc
[1] 75.11111

"Accuracy for Linear Kernel is = "

> acc
[1] 76

"Accuracy for Linear Kernel is = "

> acc
[1] 74.96296

"Accuracy for Linear Kernel is = "

> acc
[1] 73.48148

b) I have implemented program to select 2700 rows randomly and then split it into training and test set in the ratio 3:1 and added columns of 1's to both xtrain and xtest. I have set regression weight as 0.01 since it is able to give accuracy more than 70%.

I have taken centers randomly from xtrain with 40 rows and then calculated K as K= xtrain* c^T.

Weights can be calculated as $w= (K^T K + \lambda I)^{-1} K^T y$

ytest can be calculated as ytest= Ktest* w

"Accuracy for Linear Kernel is = "

> acc
[1] 75.11111

Please refer attached program and output mentioned above.

c) I have implemented following kernel functions:-

1) <u>Polynomial Kernel function</u>:- The polynomial kernel function is given by

$$k(x,y)= (\gamma\, x^T y + c_0)^d$$

Where:

- x and y are input vectors
- d is the degree
- γ and c_0 are adjustable parameters

In my implementation I have used degree as 5, c_0= 0 which means that kernel is homogeneous.

The polynomial kernel represents the similarity between two vectors. Conceptually, the polynomial kernels considers not only the similarity between vectors under the same dimension, but also across dimensions. When used in machine learning algorithms, this allows to account for feature interaction. Polynomial kernel allows us to model feature conjunctions up to the order of the polynomial.

"Accuracy for Quadratic Kernel Function is "

> acc
[1] 60.2963

"Accuracy for Quadratic Kernel Function is "

> acc
[1] 61.92593

"Accuracy for Quadratic Kernel Function is "

> acc
[1] 64.14815

2) **Sigmoid or Hyperbolic Tangent Kernel:-** Sigmoid kernel is also known as hyperbolic tangent or Multilayer Perceptron (because , in the neural field, it is often used as neuron activation function) it is defined as:

$$k(x,y)= \tanh(\alpha\, x^T\, y + c_0)$$

Where:

- x and y are input vectors
- α is the slope
- c_0 is the intercept

There are two adjustable parameters in the sigmoid kernel, the slope alpha and the intercept constant c. A common value for alpha is 1/N, where N is the data dimension.

In my implementation I have chosen $\alpha= 0.05$ since I was getting very low accuracy on taking alpha= 1/(N)= 1/2025=0.0005

> print("Accuracy for Sigmoid Kernel is = ")
[1] "Accuracy for Sigmoid Kernel is = "

> acc
[1] 76.88889

print("Accuracy for Sigmoid Kernel is = ")
[1] "Accuracy for Sigmoid Kernel is = "

> acc
[1] 73.48148

[1] "Accuracy for Sigmoid Kernel is = "

> acc
[1] 77.77778

3) Quadratic Kernel Function:- The function is given by

$$k(x, z) = (x^T z)^2 \text{ or } (1 + x^T z)^2$$

I have implemented equation $(1 + x^T z)^2$ in my program since I was getting low accuracy without 1.

Quadratic Kernel function is similar to Polynomial function with degree=2.

"Accuracy for Quadratic Kernel Function is "

> acc
[1] 75.40741

"Accuracy for Quadratic Kernel Function is "

> acc
[1] 75.25926

"Accuracy for Quadratic Kernel Function is "

> acc
[1] 78.37037

My Observations

- Polynomial kernel accuracy is most of the times less than quadratic and sigmoid kernel.
- Quadratic kernel accuracy increased after adding 1 to (x^T c).
- Sigmoid Kernel performs consistently when alpha value is selected properly, performs poorly if alpha is chosen as 1/N.

Solution 3:-

The purpose of cross-validation is to identify learning parameters that generalize well across the population samples we learn from in each fold.

More specifically: We globally search over the space over learning parameters, but within each fold, we fix learning parameters and learn model parameters. The outcome should be learning parameters that produce on average the best performance in all folds. We can then use these to train a model on the entire dataset.

Statistical Significance Test

We will perform Student t-test or Welch's t-test to check if our Null Hypothesis is true or not. In our case , We will be comparing two Algorithm's performance hence it is two sample problem and Welch's t-test will be apt or suitable. Welch's test is better since it doesn't assume equal variance like Student's t- test.

Null Hypothesis

Let $\mu 1$ be the Mean of Algorithm1 and $\mu 2$ be the Mean of Algorithm2
Our Null Hypothesis is that both Algorithms have same mean.

$$\mu 1 = \mu 2$$
$$\mu 1 - \mu 2 = 0$$

OR

$$\mathbf{H_o: -\ \Delta = 0} \quad \text{(Null Hypothesis)}$$

Our Alternate Hypothesis will be that means are not equal.

H_a: - $\Delta \neq 0$ (Alternate Hypothesis)

Confidence Interval

I am assuming 95% Confidence Interval for our Calculation hence

$1 - \alpha = 0.95$
$\alpha = 0.05$
$\alpha/2 = 0.025$

Hence,

q= qnorm (1- (α/2)) = qnorm (1- 0.025) = qnorm (0.975)
Calculated in R

q≈ 1.96

Welch's t-test is given by formula

$$t = \frac{\overline{X}_1 - \overline{X}_2}{\sqrt{\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}}}$$

And Degree of freedom (v) is given by

$$\nu \approx \frac{\left(\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}\right)^2}{\frac{s_1^4}{N_1^2\nu_1} + \frac{s_2^4}{N_2^2\nu_2}}$$

Where $V_1 = N_1 - 1$ associated with 1st Variance estimate and $V_2 = N_2 - 1$ associated with 2nd Variance estimate.

Once t- value and v-value's are calculated we can calculate the p-value using the formula

Pvalue = 2*(1-**pt**(**abs**(t),df=v)) (For skewed data)

This can be done directly also by using t.test (s1, s2)

I tried applying cross validation to logistic regression which works fine without cross validation but throws error with cross validation hence I have applied statistical significance test for Random and Liner regression. (To show the procedure)

[50.2, 50.23, 49.84, 50.14999999999999, 49.49, 50.029999999999994, 49.78, 49.72, 49.220000000000006, 49.95499549954995] 49.86149955
[59.58, 60.36, 59.5, 60.160000000000004, 60.46, 60.209999999999994, 60.67, 59.89, 60.08, 60.3060306030603] 60.1216030603
statistic = -65.65543 and pvalue = 0.0000000000
statistic = -65.65543 and pvalue = 0.0000000000

p-value comes as 0 which shows that our Null Hypothesis is wrong i.e Means accuracy of two algorithms is not same. Hence we reject our Null Hypothesis.

Refernces

1. Pattern Recognition and Machine Learning :- Christopher M. Bishop
2. Machine Learning Notes:- Predrag Radivojac and Martha White
3. http://crsouza.com/2010/03/kernel-functions-for-machine-learning-applications/
4. http://scikit-learn.org/stable/modules/metrics.html
5. http://stats.stackexchange.com/questions/43131/cross-validation-and-parameter-optimization
6. http://www.graphpad.com/guides/prism/6/statistics/index.htm?stat_the_unequal_variance_welch_t_t.htm