Assignment 4

Answer 1

1(a) $f(x,y) = x^2 - y^2$ subject to $x^2 + y^2 = 4$

Solution

given, $y^2 = 4 - x^2$

Putting this into $f(x,y)$

$$f(x,x) = 2x^2 - 4$$

$$f'(x) = 4x$$

Equating $f'(x) = 0$

$$4x = 0$$
$$x = 0$$

$$x = 0 \implies y = \pm 2 \quad \left[\because x^2 + y^2 = 4\right]$$

$$F''(x) = 4 > 0$$

Therefore, function max is 4 at $(2,0)$ & $(-2,0)$

1 (b) $f(x,y) = x^2 y - \log x$ subject to ~~$x^2 + y^2 = 4$~~

$x + 2y = 0$

Given,

$$y = -x/2$$

Putting this in $f(x,y)$

$$f(x, x/2) = x^2 \cdot \left(-\frac{x}{2}\right) - \log x$$

$$= \frac{-x^3}{2} - \log x$$

$f'(x)$ should be 0

$$f'(x) = \frac{-3x^2}{2} - \frac{1}{x} = 0$$

$$3x^3 = -2 \Rightarrow x = \left(\frac{-2}{3}\right)^{1/3}$$

$$f''(x) = \frac{-6x}{2} + \frac{1}{x^2}$$

$$= \frac{-3x^3 + 1}{x^2}$$

$$f''\left(-\left(\frac{2}{3}\right)^{1/3}\right) = 3.91 > 0$$

$\because$ x is negative, $\log x$ cannot be defined

Hence, their is no solution

1(c) $f(x,y) = x^2 + 2xy + y^2 - 2x$ subject to $x^2 - y^2 = -1$

Solution

$$\frac{\partial f}{\partial x} = 2x + 2y - 2 + 2\lambda x = 0 \quad \text{--- (1)}$$

$$\frac{\partial f}{\partial y} = 2x + 2y - 2\lambda y = 0 \quad \text{--- (2)}$$

$$\frac{\partial f}{\partial \lambda} = x^2 - y^2 + 1 = 0 \quad \text{--- (3)}$$

I solved the above equations on computer and solution is

$$x = 0, \; y = 1, \; \lambda = 1$$

Answer 2

(a) I have submitted the code as *"Kernel Linear.py"*.

(b) I tried implementing the given linear kernel with different kernel size in range (5,40) choosing different regularizer weight.
The best accuracy was at kernel size: 30, regularizer weight: 1 = **77.2**
Below are my observations

C:\Users\sohil\AppData\Local\Enthought\Canopy\User\python.exe D:/Downloads/Kernel1.py

Accuracy for kernel size: 5, regularizer weight: 0.001 = 76.6

Accuracy for kernel size: 5, regularizer weight: 0.01 = 76.6

Accuracy for kernel size: 5, regularizer weight: 0.1 = 76.0666666667

Accuracy for kernel size: 5, regularizer weight: 1 = 76.9333333333

Accuracy for kernel size: 10, regularizer weight: 0.001 = 76.6

Accuracy for kernel size: 10, regularizer weight: 0.01 = 76.6

Accuracy for kernel size: 10, regularizer weight: 0.1 = 76.4

Accuracy for kernel size: 10, regularizer weight: 1 = 76.6

Accuracy for kernel size: 15, regularizer weight: 0.001 = 76.6

Accuracy for kernel size: 15, regularizer weight: 0.01 = 76.6

Accuracy for kernel size: 15, regularizer weight: 0.1 = 76.7333333333

Accuracy for kernel size: 15, regularizer weight: 1 = 76.6666666667

Accuracy for kernel size: 20, regularizer weight: 0.001 = 76.6

Accuracy for kernel size: 20, regularizer weight: 0.01 = 76.6

Accuracy for kernel size: 20, regularizer weight: 0.1 = 76.5333333333

Accuracy for kernel size: 20, regularizer weight: 1 = 77.1333333333

Accuracy for kernel size: 25, regularizer weight: 0.001 = 76.6

Accuracy for kernel size: 25, regularizer weight: 0.01 = 76.6

Accuracy for kernel size: 25, regularizer weight: 0.1 = 76.4

Accuracy for kernel size: 25, regularizer weight: 1 = 77.0666666667

Accuracy for kernel size: 30, regularizer weight: 0.001 = 76.6

Accuracy for kernel size: 30, regularizer weight: 0.01 = 76.6

Accuracy for kernel size: 30, regularizer weight: 0.1 = 76.6666666667

Accuracy for kernel size: 30, regularizer weight: 1 = 77.2

Accuracy for kernel size: 35, regularizer weight: 0.001 = 76.6

Accuracy for kernel size: 35, regularizer weight: 0.01 = 76.6

Accuracy for kernel size: 35, regularizer weight: 0.1 = 76.6

Accuracy for kernel size: 35, regularizer weight: 1 = 76.8666666667

Accuracy for kernel size: 40, regularizer weight: 0.001 = 76.6

Accuracy for kernel size: 40, regularizer weight: 0.01 = 76.6

Accuracy for kernel size: 40, regularizer weight: 0.1 = 76.4666666667

Accuracy for kernel size: 40, regularizer weight: 1 = 76.9333333333

(C) I have tried three different kernels on the same dataset namely - Linear, Polynomial, Gaussian and written down the results in file '*Kernel Comparision Results.txt". *The choice of a Kernel depends on the problem at hand because it depends on what we are trying to model. My main observations when I ran all the three kernels for same dataset are –

- The results of accuracy was best for Gaussian, second for Linear and last for Polynomial.
- The Gaussian took noticeably more time for execution, Linear and Polynomial time execution was not noticeable
- I observed huge variation in the accuracies of Gaussian Kernel, the others gave quite consistent accuracies for different settings.

Upon running the kernels separately, I tried optimizing the accuracy for Gaussian kernel by different settings of the constant term sigma and noted my observations in file '*Gaussian Kernel.txt'*

- Performed very badly with very low and high values of sigma
- On the given dataset, it performed badly with higher regularizer weight of more than 0.1
- I received very good accuracy of **79.0** after some parameter tuning at regularizer weight: 0.001, sigma: 6

Other theoretical concepts I came across about the different kernels –

- A polynomial kernel allows us to model feature conjunctions up to the order of the polynomial.
- Radial basis functions allows to pick out circles (or hyperspheres) – in contrast with the Linear kernel, which allows only to pick out lines (or hyperplanes).
- The Rational Quadratic kernel is less computationally intensive than the Gaussian kernel and can be used as an alternative when using the Gaussian becomes too expensive

Answer 3

The code for this experiment has been attached in Assignment4 package, and I have made the required changes in "*script_classify.py"* file. The result has been attached in file *"Statistical Significance Tests.txt"*

To perform model comparision experiment, I picked up Linear and Logistic Regression and performed statistical significance tests on them using different mata parameter values. The metaparameter I have tried to tune here is the lambda value for regularization. I have performed t_test on the best accuracy value for Linear and Logistic regression for different splits of dataset.

Procedure:

- I shuffled my dataset
- Divided dataset into 7 external splits
- Inside every split, used 10-fold cross validation
- For every fold, applied both the models for different values of lamda between [e-5,e-4,...1]
- Picked up the best accuracy and best lambda value for every split.
- Used these 7 accuracy values to perform t_test
- The p_value was 0.74 which suggest that my hyposthesis that the best accuracy values for every split for both Logisitic and Linear regression was true.

Therer are total 10000 rows in the susysubset out of which I did my parameter tuning on the initial 8400 rows and the remaining for test.

Results:

Accuracy set : {'Logistic Regression': [81.66666666666667, 82.5, 82.5, 81.66666666666667, 84.16666666666667, 81.66666666666667, 86.66666666666667], 'Linear Regression': [82.5, 82.5, 82.5, 81.66666666666667, 85.0, 81.66666666666667, 87.5]}

Metaparameter set : {'Logistic Regression': [1e-05, 1e-05, 1e-05, 1e-05, 1e-05, 1, 1e-05], 'Linear Regression': [1e-05, 1e-05, 1e-05, 1e-05, 0.1, 1, 0.1]}

Mean Accuracy Linear : 83.3333333333

Mean Accuracy Logistic : 82.9761904762

Variance Linear : 3.96825396825

Variance Logistic : 2.94784580499

statistic = 0.33265 and pvalue = 0.7452612320

statistic = 0.33265 and pvalue = 0.7451380136

Theory:

Null Hypothesis H0 :- M1-M2=0 (Both mean for linear and logistic regression are equal)

Alternate Hypotheis Ha:- M1-M2!= 0

My approach is based on this article - https://www.oreilly.com/ideas/evaluating-machine-learning-models/page/5/hyperparameter-tuning

In order to check whether Null hypothesis is true or false, we need to calculate p-value

- Calculate t-value using Welch’s t-test formula

- Calculate degree of freedom using formula where df1=N1-1 and df2=N2-1
- Calculate confidence interval assume 95% so that q= pnorm(0.975)=1.96
- Calculate p-value using 1-pt(t,df=v)
- If p-value is less than 0.05 then we dismiss our null hypothesis
- On Implementing this in my program I got p-value= 0.74 which is greater than 0.05 hence we fail to reject our null hypothesis and we conclude that Mean accuracy of Linear and Logistic regression is same.

Welch's t-test is used when there are two samples and can be run even if variances are not same. Welch's t-test doesn't assume equal variance like Student t-test.

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

Degree's of freedom is given by

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