Sayyed Mohsen Vazirizade

23398312

smvazirizade@email.arizona.edu

INFO 521

Introduction to Machine Learning

Assignment 4

Problem 1

$$p(\sigma^2|\, t, X, w, \alpha, \beta) = \frac{p(t|\, \sigma^2, X, w, \alpha, \beta)\, p(\sigma^2|\, \alpha, \beta)}{p(t|\, X, w, \alpha, \beta)}$$

$$Prior: p(\sigma^2|\, \alpha, \beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} \quad (\sigma^2)^{-1-\alpha}\, e^{-\frac{\beta}{\sigma^2}}$$

$$likelihood: p(t|\, \sigma^2, X, w, \alpha, \beta) = \frac{1}{(2\pi)^{N/2}\sqrt{\sigma^2}} e^{-\frac{\sum(t_i - xw_i)^2}{\sigma^2}}$$

$$p(\sigma^2|\, t, X, w, \alpha, \beta) = \frac{p(t|\, \sigma^2, X, w, \alpha, \beta)\, p(\sigma^2|\, \alpha, \beta)}{p(t|\, X, w, \alpha, \beta)}$$

$$= \frac{\beta^{\alpha}}{\Gamma(\alpha)} \quad (\sigma^2)^{-1-\alpha}\, e^{-\frac{\beta}{\sigma^2}} \times \frac{1}{(2\pi)^{N/2}\sqrt{\sigma^2}} e^{-\frac{\sum(t_i - xw_i)^2}{\sigma^2}}$$

$$\propto \quad (\sigma^2)^{-1-\alpha}\, e^{-\frac{\beta}{\sigma^2}} \times \frac{1}{\sqrt{\sigma^n}} e^{-\frac{\sum(t_i - xw_i)^2}{\sigma^2}} = (\sigma^2)^{-1-(\alpha+n/2)}\, e^{-\frac{\sum(t_i - xw_i)^2 + \beta}{\sigma^2}}$$

$$p(\sigma^2|\, t, X, w, \alpha, \beta) \propto (\sigma^2)^{-1-(\alpha+1/2)}\, e^{-\frac{\sum(t_i - xw_i)^2 + \beta}{\sigma^2}}$$

Therefore, the new value of $\alpha\ and\ \beta$ is calculated as follow:

$$\alpha_{new} = \alpha_{old} + n/2$$

$$\beta_{new} = \beta_{old} + 0.5 \sum (t_i - xw_i)^2 = \beta_{old} + 0.5\, (t - Xw)^T (t - Xw)$$

Problem 2

$posterior \propto \ prior \times likelihood$

$posterior\ (gamma) \propto \ prior\ (gamma) \times likelihood\ (binomial)$

From chapter 3 we know that

$\alpha_{new} = \alpha_{old} + y$

$\beta_{new} = \beta_{old} + y$

$$g(r; y, N, \alpha, \beta) = p(y|r, N) \times p(r|\alpha, \beta) = \binom{N}{y} r^y (1-r)^{N-y} \times \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} r^{\alpha-1}(1-r)^{\beta-1}$$
$$= \frac{\Gamma(N+1)}{\Gamma(N\ -\ y+1)\Gamma(y+1)} r^y (1-r)^{N-y} \times \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} r^{\alpha-1}(1-r)^{\beta-1}$$
$$\propto r^y (1-r)^{N-y} \times r^{\alpha-1}(1-r)^{\beta-1}$$

$$log(g(r; y, N, \alpha, \beta)) = log(\frac{\Gamma(N+1)}{\Gamma(N\ -\ y+1)\Gamma(y+1)}\ r^y (1-r)^{N-y} \times \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} r^{\alpha-1}(1-r)^{\beta-1})$$
$$= log(\frac{\Gamma(N+1)}{\Gamma(N\ -\ y+1)\Gamma(y+1)} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}) + log(r^y (1-r)^{N-y}) + log(r^{\alpha-1}(1$$
$$- r)^{\beta-1})$$
$$= log(\frac{\Gamma(N+1)}{\Gamma(N\ -\ y+1)\Gamma(y+1)} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}) + ylog(r) + (N-y)log(1-r) + (\alpha$$
$$- 1)log(r) + (\beta-1)log(1-r)$$

$$\frac{\partial log(g(r; y, N, \alpha, \beta))}{\partial r} = \frac{y}{r} - \frac{N\ -\ y}{1-r} + \frac{\alpha-1}{r} - \frac{\beta\ -\ 1}{1-r}$$

$$\frac{\partial^2 log(g(r;y,N,\alpha,\beta))}{\partial r^2} = -\frac{y}{r^2} + \frac{N-y}{(1-r)^2} - \frac{\alpha-1}{r^2} + \frac{\beta-1}{(1-r)^2} = -\frac{y+\alpha-1}{r^2} + \frac{N-y+\beta-1}{(1-r)^2} < 0$$

$$\frac{\partial log(g(r; y, N, \alpha, \beta))}{\partial r} = 0 \Rightarrow \frac{y+\alpha-1}{r} = \frac{N\ -\ y+\beta\ -\ 1}{1-r} \Rightarrow \frac{r}{y+\alpha-1} = \frac{r-1}{N\ -\ y+\beta\ -\ 1} \Rightarrow r$$
$$= \frac{y+\alpha-1}{\alpha+N+\beta-2}$$

$$log(g(r; y, N, \alpha, \beta)) \approx log(g(\hat{r}; y, N, \alpha, \beta))\ + 0.5\left(\frac{y+\alpha-1}{r^2} - \frac{N\ -\ y+\beta\ -\ 1}{(1-r)^2}\right)\ (r-\hat{r})^2$$

$$\mu = \hat{r} = r = \frac{y+\alpha-1}{\alpha+N+\beta-2}$$

$$\Sigma^{-1} = \ (\sigma^2)^{-1} = \frac{\partial^2 log(g(r; y, N, \alpha, \beta))}{\partial r^2} = \frac{y+\alpha-1}{\frac{y+\alpha-1}{\alpha+N+\beta-2}^2} - \frac{N\ -\ y+\beta\ -\ 1}{(1-\frac{y+\alpha-1}{\alpha+N+\beta-2})^2}$$
$$= \frac{(\alpha+N+\beta-2)^2}{y+\alpha-1} + \frac{(\alpha+N+\beta-2)^2}{N\ -\ y+\beta\ -\ 1}$$

$$\sigma^2 = 1/\left(\frac{(\alpha+N+\beta-2)^2}{y+\alpha-1} + \frac{(\alpha+N+\beta-2)^2}{N\ -\ y+\beta\ -\ 1}\right) = \frac{(y+\alpha-1)(N\ -\ y+\beta\ -\ 1)}{(\alpha+N+\beta-2)^2(\alpha+N+\beta\ -\ 2)}$$

Problem 3

The formulation for calculation of the required parameters for normal distribution are provided in the previous example. In this regard, having the values for each parameter of alpha, beta, N and y we can calculate sigma and mean. The following table summarized the calculated values for each state.

Table 1 Summarized calculated value

State	Alpha	Beta	N	y	Variance	Mean
1	5	5	20	10	0.008928571428571428	0.5
2	3	15	10	3	0.005974055530268548	0.19230769230769232
3	1	30	10	3	0.0018206645425580337	0.07692307692307693

The following figures compare Normal and Beta Dist. for posterior for state 1 through 3, respectively. As it can be seen, because Normal Dist. is symmetric inherently, it provides a better estimation when the real posterior function is symmetric. Furthermore, it is obvious that the estimated curve, normal pdf is not exactly as the true posterior function; however, it is a trade of between computing cost and precision.

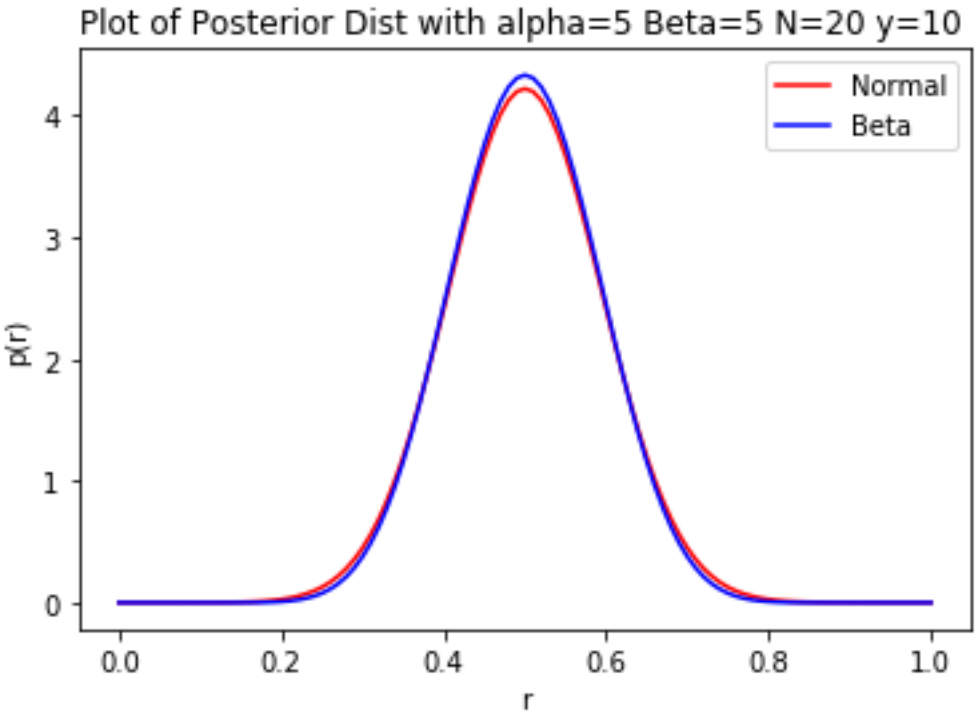

Figure 1Comaring Normal and Beta Dist. for posterior for state 1

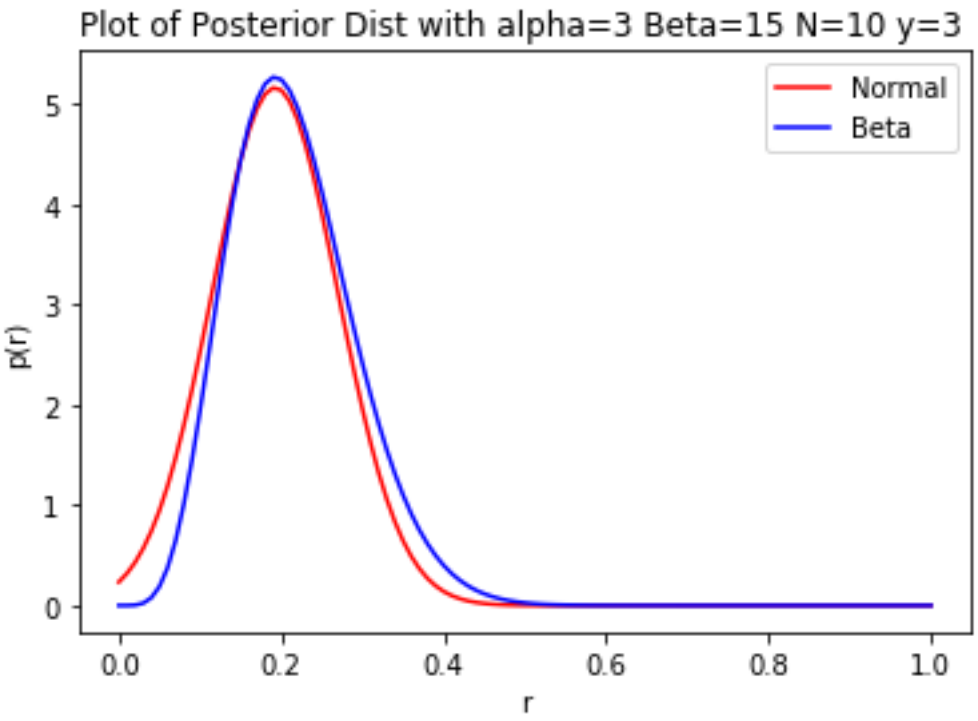

Figure 2 Comparing Normal and Beta Dist. for posterior for state 2

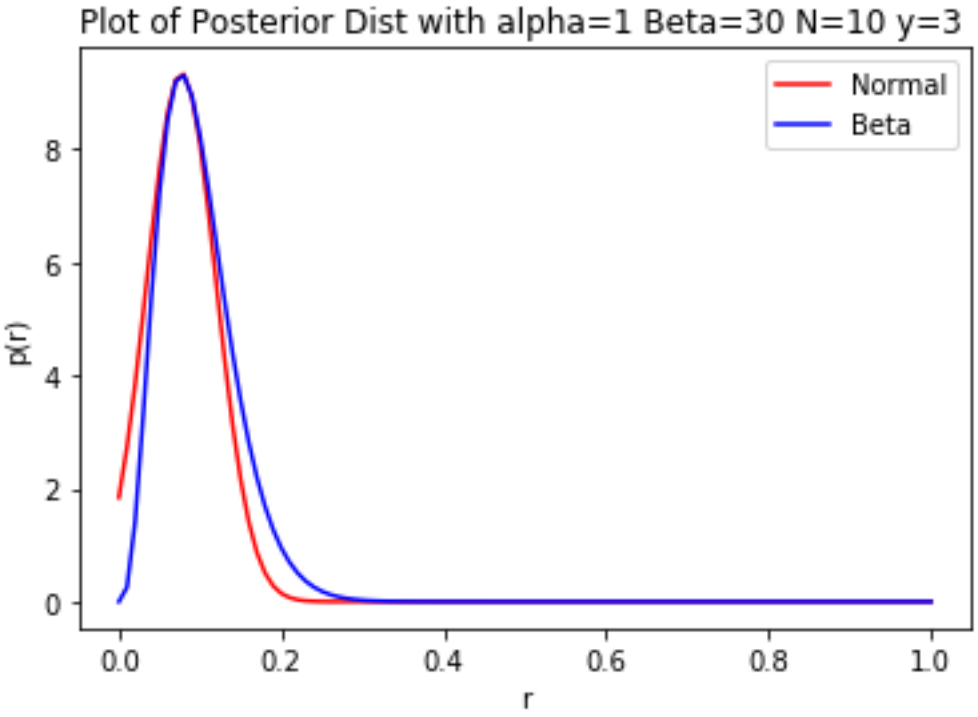

Figure 3 Comparing Normal and Beta Dist. for posterior for state 3

The developed script is as follow:

```
# -*- coding: utf-8 -*-
"""
Created on Sun Nov  5 09:53:08 2017

@author: smvaz
"""

def clear_all():
    """Clears all the variables from the workspace of the spyder application."""
    gl = globals().copy()
    for var in gl:
        if var[0] == '_': continue
        if 'func' in str(globals()[var]): continue
        if 'module' in str(globals()[var]): continue

        del globals()[var]


if __name__ == "__main__":
    clear_all()




import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
#defining each of three states
for i in range(0,3):
    if i==0:
        Alpha=5
        Beta=5
        N=20
        y=10
    if i==1:
        Alpha=3
        Beta=15
        N=10
        y=3
    if i==2:
        Alpha=1
        Beta=30
        N=10
        y=3
    #caclulating Variance (Standard Deviation) and mean based on previous problem
    Variance=(y+Alpha-1)*(N-y+Beta-1)/(Alpha+N+Beta-2)**2/(Alpha+N+Beta-2)
    print( 'Variance', Variance)
    SD=Variance**0.5
    r=(y+Alpha-1)/(Alpha+N+Beta-2)
    print('r',r)
    R=np.linspace(0, 1, 101)
    #drawing the figures
    plt.figure()
    plt.plot(R,stats.norm.pdf(R, loc=r, scale=SD),color='r',label='Normal')
    plt.plot(R,stats.beta.pdf(R, Alpha, Beta),color='b',label='Beta')
    plt.xlabel('r')
    plt.ylabel('p(r)')
    plt.legend()
    ti = 'Plot of Posterior Dist with alpha={} Beta={} N={} y={} '.format(Alpha,Beta,N,y)
    plt.title(ti)
    plt.show()
```

Problem 4

In order to estimate the value of π we generate a 2D random numbers. Actually, we fill a square with total area of 4 with dots. As we know the formula for distance is $\sqrt{(x_1 - x_2)^2 - (y_1 - y_2)^2}$ and if we consider the center as the second point it changes to $\sqrt{x^2 - y^2}$. If the distance of point to the center in that square is more than, it means it can be confined by a circle with radius of 1. As the following figure shows, the red dots are the dots with the distance to the center less than one, $\sqrt{x^2 - y^2}$<1.

$$\frac{\text{Area of the Circle with radius 1}}{\text{Area of the square with side of 2}} = \frac{\pi}{4} \Rightarrow \text{PI} = \frac{\text{Number of Inside dots}}{\text{Total number of dots}} \times 4$$

Table 2 Comparing the influence of different values of iterations

Total Number of the Points	Calculated PI
1e7	3.142661
1e6	3.142660
1e5	3.141640
1e4	3.143600
1e3	3.128000
1e2	2.920000

As it can be seen, increasing the number of the total points reduces the error compared to the exact number.

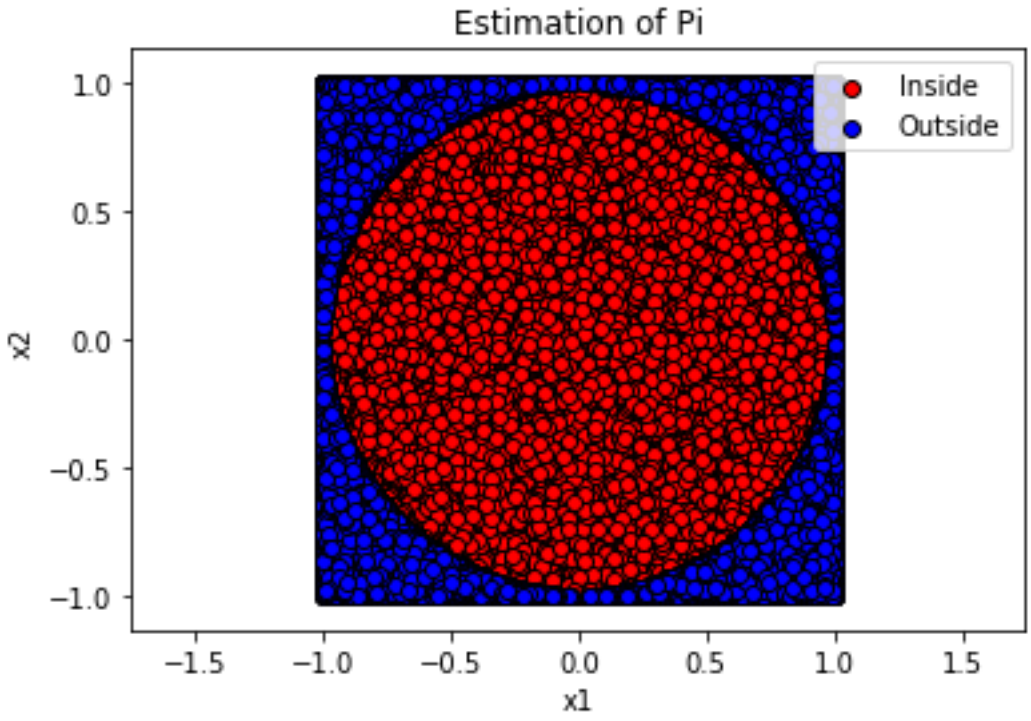

Figure 4Dots inside and outside of the circle

The script associated with this problem is:

```
# -*- coding: utf-8 -*-
"""
Created on Sat Nov  4 14:45:00 2017

@author: smvaz
"""

def clear_all():
    """Clears all the variables from the workspace of the spyder application."""
    gl = globals().copy()
    for var in gl:
        if var[0] == '_': continue
        if 'func' in str(globals()[var]): continue
        if 'module' in str(globals()[var]): continue

        del globals()[var]


if __name__ == "__main__":
    clear_all()




import numpy as np
import matplotlib.pyplot as plt


a=10000000

np.random.seed(seed=1)


Random=np.random.uniform(-1,1,(a,2))
Inside=[0,0]
Outside=[0,0]
print('All sample pairs \n{}'.format(Random))
for j in range(0,a):
    Radius=(Random[j,0]**2+Random[j,1]**2)
    if Radius<1:
        #print('Inside point \n {} {}'.format(Random[j,:],Radius ))
        Inside=np.vstack((Inside,Random[j,:]))
        #print('Inside \n {}'.format(Inside))

    else:
        #print('Outside point \n {} {}'.format(Random[j,:],a ))
        Outside=np.vstack((Outside,Random[j,:]))
        #Outside=(Random[j,:])

Inside=np.delete(Inside,0,0)
print('Inside \n {}'.format(Inside))
Outside=np.delete(Outside,0,0)
print('Outside \n {}'.format(Outside))
#print(Inside.shape[0])
NumberofInside=Inside.shape[0]
NumberofOutside=Outside.shape[0]
PI=NumberofInside/(NumberofInside+NumberofOutside)
print('Calculated value for Pi is {:01.6f}'.format(4*PI))


plt.figure()
plt.scatter(Inside[:,0], Inside[:,1], color='r', edgecolor='k',label='Inside')
plt.scatter(Outside[:,0], Outside[:,1], color='b', edgecolor='k',label='Outside')
plt.xlabel('x1')
plt.ylabel('x2')
```

```
plt.legend()
plt.title('Estimation of Pi')
plt.axis('equal')
plt.show()
```

Problem 5

$$P(t_n|w,x_n) = (W^T x_n)^{t_n} \exp(-W^T x_n)/t_n!$$

$$\text{P=}\prod_{n=1}^{n=N} P(t_n|w,x_n) = \prod_{n=1}^{n=N} (W^T x_n)^{t_n} \exp(-W^T x_n)/t_n!$$

$$g(w;X,t,\sigma^2) = p(T|W,X).p(w|\sigma^2) \Rightarrow \log(g) = \log(p(T|W,X)) + \log(p(w|\sigma^2))$$

$$\log(g(w;X,t,\sigma^2)) = -0.5D\log(2\pi) - 0.5D\log\sigma^2 - \frac{1}{2\sigma^2} w^T w + \sum_{n=1}^{N} -\log(t_n!) + t_n \log(w^T x_n) - w^T x_n$$

So, we have for first and second derivative to w:

First derivative:

$$\frac{\partial \log(g)}{\partial w} = -\frac{1}{\sigma^2} w + \sum_{n=1}^{N} \left(\frac{t_n x_n}{w^T x_n} x_n - x_n\right)$$

Second derivate (hessian Matrix)

$$\frac{\partial^2 \log(g)}{\partial w \partial w^T} = -\frac{1}{\sigma^2} I + \sum_{n=1}^{N} -t_n x_n (w^T x_n)^{-2} x_n^T$$

Newtom-Raphson:

$$w_{t+1} = w_t - \frac{f(w_t)'}{f(w_t)''}$$

$$w_{t+1} = w_t + \left(-\frac{1}{\sigma^2} I + \sum_{n=1}^{N} -t_n x_n (w^T x_n)^{-2} x_n^T\right)^{-1} \left(-\frac{1}{\sigma^2} w + \sum_{n=1}^{N} \left(\frac{t_n x_n}{w^T x_n} x_n - x_n\right)\right)$$

Widrow_hoff

$$: w_{t+1} = w_t + \left(-\frac{1}{\sigma^2} w + \sum_{n=1}^{N} \left(\frac{t_n x_n}{w^T x_n} x_n - x_n\right)\right)$$