MAD Assignement 5

MAD 2018
Department of Computer Science
University of Copenhagen

Casper Lisager Frandsen <fsn483@alumni.ku.dk>

Version 1
Due: January 9th, 23:59

Contents

1. Exercise **3**
(a) . . . 3
(b) . . . 4
(c) . . . 6

2. Exercise **7**
(a) . . . 7

3. Exercise **10**
(a) . . . 10
(b) . . . 10
(c) . . . 10

In all of these exercises I have worked closely with Asger Kjeldsen, Noah Stonall and Emillie Burkal.

1. Exercise

(a)

Calculating the posterior:
First we have to look at the function $p(r|y_N)$:

$$p(r|y_N) = \frac{p(y_N|r)p(r)}{p(y_N)}$$

We are already given the prior, $p(r)$:

$$p(r) = 1$$

We will then find that the value of $p(y_N|r)$ looks as follows:

$$p(y_N|r) = \binom{N}{y_N} r^{y_N}(1-r)^{N-y_N}$$

And the final component of the function, $p(y_n)$:

$$p(y_N) = \int_{r=0}^{r=1} p(y_N|r)p(r)dr = \int_{r=0}^{r=1} 1 \cdot \binom{N}{y_N} r^{y_N}(1-r)^{N-y_N}dr$$

We know that for beta distributions the following is true, as per pg 110 in the Machine Learning textbook:

$$\int_{r=0}^{r=1} r^{\alpha-1}(1-r)^{\beta-1}dr = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$$

Therefore, if we rewrite the previous expression to something resembling that, we can rewrite is as gamma functions:
We first rewrite it by substituting in $k = y_N + 1$ and $t = N - y_N + 1$, and moving $\binom{N}{y_N}$ out of the integral:

$$p(y_N) = \int_{r=0}^{r=1} 1 \cdot \binom{N}{y_N} r^{y_N}(1-r)^{N-y_N}dr = \binom{N}{y_N} \int_{r=0}^{r=1} r^{k-1}(1-r)^{t-1}dr$$

We can now solve the integral:

$$p(y_N) = \binom{N}{y_N} \frac{\Gamma(k)\Gamma(t)}{\Gamma(k+t)}$$

$$p(y_N) = \binom{N}{y_N} \frac{\Gamma(y_N+1)\Gamma(N-y_N+1)}{\Gamma(y_N+1+N-y_N+1)} = \binom{N}{y_N} \frac{\Gamma(y_N+1)\Gamma(N-y_N+1)}{\Gamma(N+2)}$$

We can now put all of this into the first function:

$$p(r|y_N) = \frac{p(y_N|r)p(r)}{p(y_N)} = \frac{\binom{N}{y_N} r^{y_N}(1-r)^{N-y_N}}{\binom{N}{y_N} \frac{\Gamma(y_N+1)\Gamma(N-y_N+1)}{\Gamma(N+2)}} = \frac{r^{y_N}(1-r)^{N-y_N}}{\left(\frac{\Gamma(y_N+1)\Gamma(N-y_N+1)}{\Gamma(N+2)}\right)}$$

We rewrite the denominator so that we instead multiply by the inverse:

$$p(r|y_N) = \frac{\Gamma(N+2)}{\Gamma(y_N+1)\Gamma(N-y_N+1)} r^{y_N}(1-r)^{N-y_N}$$

We can now see that this is a beta density distribution.

(b)

First we will find the values of α and β. We don't have a proper method to do so, but will look at it piecewise and realise that for this value of $p(r)$ it is trivial. Whenever we write an expression regarding $p(r)$ it is assumed that it is the following segment: $p(r), (0 \leq r \leq 1)$

$$p(r) = 2r = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} r^{\alpha-1}(1-r)^{\beta-1}$$

We can realise that the following part will always be a constant multiplier on r:

$$\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}$$

This means we have to solve the following:

$$\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} = 2$$

We can also realise that the latter part will always result in an exponent to r:

$$r^{\alpha-1}(1-r)^{\beta-1}$$

This means we have to solve the following:

$$r^{\alpha-1}(1-r)^{\beta-1} = r$$

We find that the values $\alpha = 2, \beta = 1$ fit all these criteria:

$$\frac{\Gamma(2+1)}{\Gamma(2)\Gamma(1)} r^{2-1}(1-r)^{1-1} = 2r$$

For calculating the posterior, we can reuse much of the math from $1(a)$. We can realise that $p(y_N|r)$ is the same since it is not dependent on $p(r)$. The other two functions only have small changes:

$$p(r|y_N) = \frac{p(y_N|r)p(r)}{p(y_N)}$$

$$p(r) = 2r$$

$$p(y_N|r) = \binom{N}{y_N} r^{y_N}(1-r)^{N-y_N}$$

$$p(y_N) = \int_{r=0}^{r=1} p(y_N|r)p(r)dr = \int_{r=0}^{r=1} 2r \binom{N}{y_N} r^{y_N}(1-r)^{N-y_N}dr$$

We use the same assumption as in the last exercise:

$$\int_{r=0}^{r=1} r^{\alpha-1}(1-r)^{\beta-1}dr = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$$

And rewrite:

$$p(y_N) = 2\binom{N}{y_N}\int_{r=0}^{r=1} r * r^{y_N}(1-r)^{N-y_N}dr$$

$$p(y_N) = 2\binom{N}{y_N}\int_{r=0}^{r=1} r^{y_N+1}(1-r)^{N-y_N}dr$$

We substitute again, with $k = y_N + 2$ and $t = N - y_N + 1$

$$p(y_N) = 2\binom{N}{y_N}\int_{r=0}^{r=1} r^{k-1}(1-r)^{t-1}dr$$

We can now solve it as per the assumption above:

$$p(y_N) = 2\binom{N}{y_N}\frac{\Gamma(y_N+2)\Gamma(N-y_N+1)}{\Gamma(y_N+2+N-y_N+1)} = 2\binom{N}{y_N}\frac{\Gamma(y_N+2)\Gamma(N-y_N+1)}{\Gamma(N+3)}$$

We can now put everything into the original expression again:

$$p(r|y_N) = \frac{p(y_N|r)p(r)}{p(y_N)} = \frac{2r\binom{N}{y_N} r^{y_N}(1-r)^{N-y_N}}{2\binom{N}{y_N}\frac{\Gamma(y_N+2)\Gamma(N-y_N+1)}{\Gamma(N+3)}}$$

$$p(r|y_N) = \frac{p(y_N|r)p(r)}{p(y_N)} = \frac{2\binom{N}{y_N} r^{y_N+1}(1-r)^{N-y_N}}{2\binom{N}{y_N}\frac{\Gamma(y_N+2)\Gamma(N-y_N+1)}{\Gamma(N+3)}}$$

$$p(r|y_N) = \frac{p(y_N|r)p(r)}{p(y_N)} = \frac{r^{y_N+1}(1-r)^{N-y_N}}{\left(\frac{\Gamma(y_N+2)\Gamma(N-y_N+1)}{\Gamma(N+3)}\right)}$$

We rewrite the denominator so that we instead multiply by the inverse:

$$p(r|y_N) = \frac{\Gamma(N+3)}{\Gamma(y_N+2)\Gamma(N-y_N+1)} r^{y_N+1}(1-r)^{N-y_N}$$

We can now see that this is a beta density distribution.

(c)

First we will find the values of α and β. We don't have a proper method to do so, but will look at it piecewise and realise that for this value of $p(r)$ it is trivial. Whenever we write an expression regarding $p(r)$ it is assumed that it is the following segment: $p(r), (0 \leq r \leq 1)$

$$p(r) = 3r^2 = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} r^{\alpha-1}(1-r)^{\beta-1}$$

We can realise that the following part will always be a constant multiplier on r:

$$\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}$$

This means we have to solve the following:

$$\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} = 3$$

We can also realise that the latter part will always result in an exponent to r:

$$r^{\alpha-1}(1-r)^{\beta-1}$$

This means we have to solve the following:

$$r^{\alpha-1}(1-r)^{\beta-1} = r^2$$

We find that the values $\alpha = 3, \beta = 1$ fit all these criteria:

$$\frac{\Gamma(3+1)}{\Gamma(3)\Gamma(1)} r^{3-1}(1-r)^{1-1} = 3r^2$$

For calculating the posterior, we can reuse much of the math from $1(a)$ and $1(b)$. We can realise that $p(y_N|r)$ is the same since it is not dependent on $p(r)$. The other two functions only have small changes:

$$p(r|y_N) = \frac{p(y_N|r)p(r)}{p(y_N)}$$

$$p(r) = 3r^2$$

$$p(y_N|r) = \binom{N}{y_N} r^{y_N}(1-r)^{N-y_N}$$

$$p(y_N) = \int_{r=0}^{r=1} p(y_N|r)p(r)dr = \int_{r=0}^{r=1} 3r^2 \binom{N}{y_N} r^{y_N}(1-r)^{N-y_N} dr$$

We use the same assumption as in the last exercise:

$$\int_{r=0}^{r=1} r^{\alpha-1}(1-r)^{\beta-1} dr = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$$

And rewrite:

$$p(y_N) = 3\binom{N}{y_N}\int_{r=0}^{r=1} r^2 * r^{y_N}(1-r)^{N-y_N}dr$$

$$p(y_N) = 3\binom{N}{y_N}\int_{r=0}^{r=1} r^{y_N+2}(1-r)^{N-y_N}dr$$

We substitute again, with $k = y_N + 3$ and $t = N - y_N + 1$

$$p(y_N) = 3\binom{N}{y_N}\int_{r=0}^{r=1} r^{k-1}(1-r)^{t-1}dr$$

We can now solve it as per the assumption above:

$$p(y_N) = 3\binom{N}{y_N}\frac{\Gamma(y_N+3)\Gamma(N-y_N+1)}{\Gamma(y_N+3+N-y_N+1)} = 3\binom{N}{y_N}\frac{\Gamma(y_N+3)\Gamma(N-y_N+1)}{\Gamma(N+4)}$$

We can now put everything into the original expression again:

$$p(r|y_N) = \frac{p(y_N|r)p(r)}{p(y_N)} = \frac{3r^2\binom{N}{y_N}r^{y_N}(1-r)^{N-y_N}}{3\binom{N}{y_N}\frac{\Gamma(y_N+3)\Gamma(N-y_N+1)}{\Gamma(N+4)}}$$

$$p(r|y_N) = \frac{p(y_N|r)p(r)}{p(y_N)} = \frac{3\binom{N}{y_N}r^{y_N+2}(1-r)^{N-y_N}}{3\binom{N}{y_N}\frac{\Gamma(y_N+3)\Gamma(N-y_N+1)}{\Gamma(N+4)}}$$

$$p(r|y_N) = \frac{p(y_N|r)p(r)}{p(y_N)} = \frac{r^{y_N+2}(1-r)^{N-y_N}}{\left(\frac{\Gamma(y_N+3)\Gamma(N-y_N+1)}{\Gamma(N+4)}\right)}$$

We rewrite the denominator so that we instead multiply by the inverse:

$$p(r|y_N) = \frac{\Gamma(N+4)}{\Gamma(y_N+3)\Gamma(N-y_N+1)}r^{y_N+2}(1-r)^{N-y_N}$$

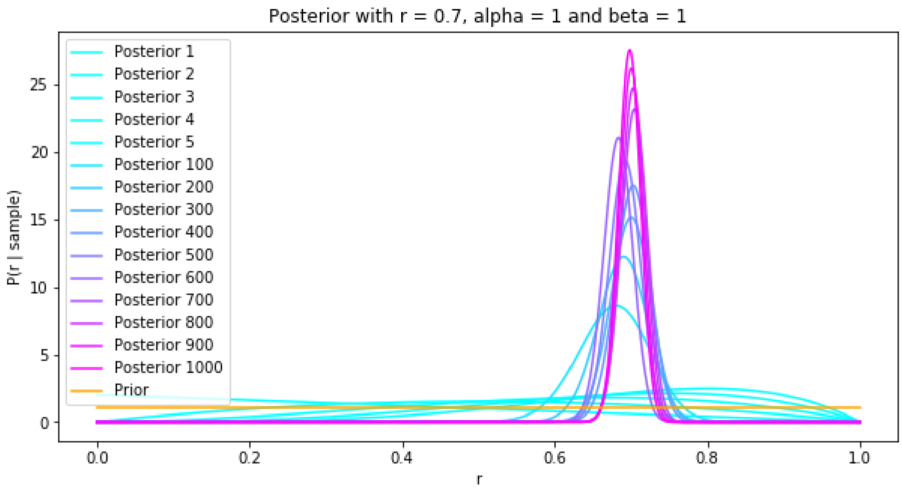
We can now see that this is a beta density distribution.

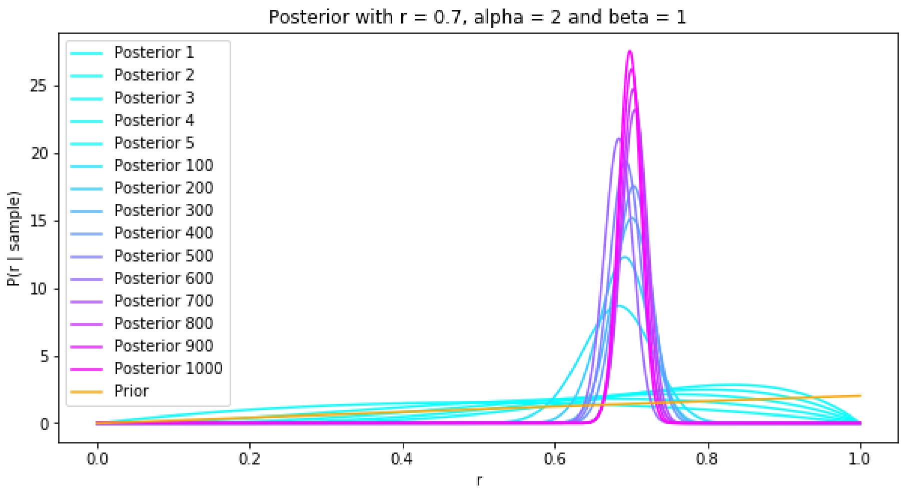
2. Exercise

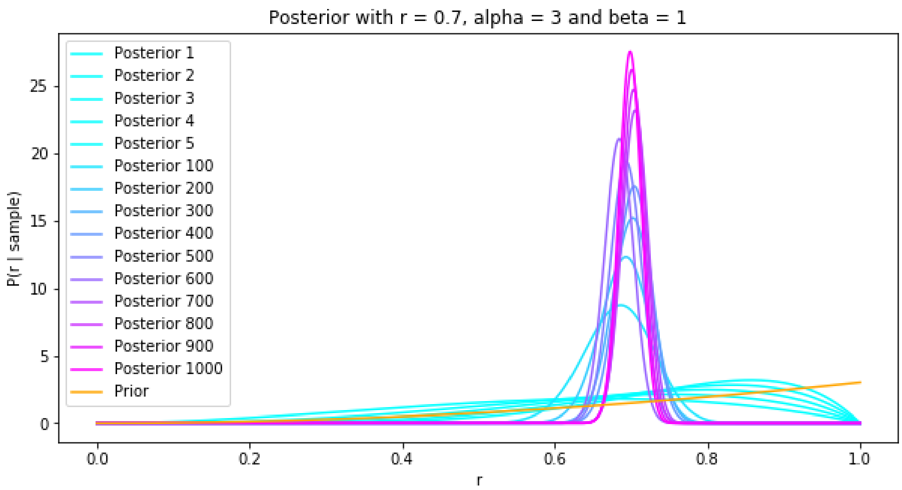
(a)

The most relevant parts of the source code and the resulting plots are written here, but the rest, and the code for plotting can be found in the attached Jupyter Notebook named "CLFA5.ipynb":

```
# This function has been taken from the lecture slides
def prior(x, prior_alpha, prior_beta):
    return beta.pdf(x, prior_alpha, prior_beta)

# This function has been taken from the lecture slides
def posterior(x, prior_alpha, prior_beta, sample):
    N = len(sample)
    y_N = sum(sample)
    delta = y_N + prior_alpha
    gamma = N - y_N + prior_beta
    return beta.pdf(x, delta, gamma)

# This function has been taken from the lecture slides
def draw_sample(r, num):
    return np.random.binomial(1,r,num)

# Parts of this code has been taken from
# lecture slides and adapted for my use
def plotPrior(alpha, pbeta, accuracy):
    xsamples = np.linspace(0,1,accuracy+1)
    PriorVals = prior(xsamples, alpha, pbeta)

    plt.plot(xsamples, PriorVals, color = 'orange', label="Prior")
    plt.legend()
    return PriorVals

# Parts of this code has been taken from
# lecture slides and adapted for my use
def plotPosterior(alpha, pbeta, r, iterations, sample, frequency):
    xsamples = np.linspace(0,1,1000)
    PosteriorVals = np.zeros([iterations, len(xsamples)])
    PlottedSampleSizes = np.arange(1,iterations+1,1)
    colors = colors = cm.cool(np.linspace(0,1,np.max(PlottedSampleSizes)+1))

    for i in range(iterations):
        PosteriorVals[i,:] = posterior(xsamples, alpha, pbeta, sample[:i+1])

    for i in PlottedSampleSizes:
        if (i%frequency == 0 or i <= 5):
            plt.plot(xsamples,PosteriorVals[i-1,:],color=colors[i],label=f'Posterior {i}')

    plt.xlabel('r')
    plt.ylabel('P(r | sample)')
    plt.title(f'Posterior with r = {r}, alpha = {alpha} and beta = {pbeta}')
    return PosteriorVals
```


Posterior with r = 0.7, alpha = 1 and beta = 1
Posterior 1
Posterior 2
Posterior 3
Posterior 4
Posterior 5
Posterior 100
Posterior 200
Posterior 300
Posterior 400
Posterior 500
Posterior 600
Posterior 700
Posterior 800
Posterior 900
Posterior 1000
Prior
P(r | sample)
r


Posterior with r = 0.7, alpha = 2 and beta = 1
Posterior 1
Posterior 2
Posterior 3
Posterior 4
Posterior 5
Posterior 100
Posterior 200
Posterior 300
Posterior 400
Posterior 500
Posterior 600
Posterior 700
Posterior 800
Posterior 900
Posterior 1000
Prior
P(r | sample)
r


Posterior with r = 0.7, alpha = 3 and beta = 1
Posterior 1
Posterior 2
Posterior 3
Posterior 4
Posterior 5
Posterior 100
Posterior 200
Posterior 300
Posterior 400
Posterior 500
Posterior 600
Posterior 700
Posterior 800
Posterior 900
Posterior 1000
Prior
P(r | sample)
r

3. Exercise

(a)

We know that with $p(t|w, X, \sigma^2)$, t is defined as follows, with $\sigma^2 = 10$:

$$t = Xw + \epsilon$$

Where $\epsilon \sim \mathcal{N}(Xw, \sigma^2 I)$. This is a Gaussian random variable with the constant added to the mean. This means our likelihood looks as follows:

$$p(t|w, X, \sigma^2) = \mathcal{N}(Xw, 10 \cdot I_N)$$

We know that our model has zero mean, so we can write the following:

$$p(t|w, X, \sigma^2) = \mathcal{N}(0, 10)$$

(b)

The posterior will look as follows, since we know the posterior will:

$$p(w|t, X, \sigma^2) = \mathcal{N}(\mu_w, \Sigma_w)$$

Where

$$\Sigma_w = \left(\frac{1}{\sigma^2} X^T X + \Sigma_0^{-1}\right)^{-1}$$

$$\mu_w = \Sigma_w \left(X^T t + \Sigma_0^{-1} \mu_0\right)$$

(c)

The code for this exercise has been written in the attached Jupyter Notebook named "CLFA5.ipynb", but the relevant code can be seen here as well:

```
def myposterior(X, y, mu0, Sigma0, sigmaSquared):
    sigmaw = np.linalg.inv(1/sigmaSquared*np.dot(X.T,X)+np.linalg.inv(Sigma0))
    muw = np.dot(sigmaw, np.dot(X.t,y) + np.linalg.inv(Sigma0)*mu0)
    return muw, sigmaw
```