Università
della
Svizzera
italiana

Faculty of Informatics

April 10, 2018

Stochastic Methods

Assignment 2

Sumeet Gyanchandani

1 Bayesian Inference

We are given that prior $\pi(p) = Beta(\alpha, \beta)$, with $Beta(\alpha, \beta) = \frac{p^{\alpha-1}(1-p)^{\beta-1}}{B(\alpha,\beta)}$ where $B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$.

Hence, the prior $\mathbb{P}(p)$ is $\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\ p^{\alpha-1}(1-p)^{\beta-1}$

And the likelihood $\mathbb{P}(X|p)$ would be $\frac{\Gamma(H+T+1)}{\Gamma(H+1)\Gamma(T+1)}\ p^{H}(1-p)^{T}$, where H is the number of successes ($X_i = 1$) and T the number of failures ($X_i = 0$).

By Bayes Rule:

$$\begin{aligned}
\mathbb{P}(p|X) &= \frac{\mathbb{P}(X|p)\ \mathbb{P}(p)}{\mathbb{P}(X)} \\
&= \frac{\frac{\Gamma(H+T+1)}{\Gamma(H+1)\Gamma(T+1)}\ p^{H}(1-p)^{T} \times \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\ p^{\alpha-1}(1-p)^{\beta-1}}{\mathbb{P}(X)} \\
&= \frac{\frac{\Gamma(H+T+1)\Gamma(\alpha+\beta)}{\Gamma(H+1)\Gamma(T+1)\Gamma(\alpha)\Gamma(\beta)}\ p^{\alpha+H-1}(1-p)^{\beta+T-1}}{\mathbb{P}(X)} \\
&= \frac{\gamma\ p^{\alpha+H-1}(1-p)^{\beta+T-1}}{\mathbb{P}(X)}
\end{aligned}$$

Where, $\gamma = \frac{\Gamma(H+T+1)\Gamma(\alpha+\beta)}{\Gamma(H+1)\Gamma(T+1)\Gamma(\alpha)\Gamma(\beta)}$

$$\begin{aligned}
\mathbb{P}(X) &= \int_0^1 \mathbb{P}(X, p)\ \mathrm{d}p \\
&= \int_0^1 \gamma\ p^{\alpha+H-1}(1-p)^{\beta+T-1}\ \mathrm{d}p \\
&= \int_0^1 \gamma \times \frac{\Gamma(\alpha+H)\Gamma(\beta+T)}{\Gamma(\alpha+\beta+H+T)} \times \frac{\Gamma(\alpha+\beta+H+T)}{\Gamma(\alpha+H)\Gamma(\beta+T)} \times p^{\alpha+H-1}(1-p)^{\beta+T-1}\ \mathrm{d}p \\
&= \gamma \times \frac{\Gamma(\alpha+H)\Gamma(\beta+T)}{\Gamma(\alpha+\beta+H+T)} \int_0^1 \frac{\Gamma(\alpha+\beta+H+T)}{\Gamma(\alpha+H)\Gamma(\beta+T)} \times p^{\alpha+H-1}(1-p)^{\beta+T-1}\ \mathrm{d}p \\
&= \gamma \times \frac{\Gamma(\alpha+H)\Gamma(\beta+T)}{\Gamma(\alpha+\beta+H+T)} \times 1
\end{aligned}$$

Substituting this value to our previous result, we get:

$$
\begin{aligned}
\mathbb{P}(p|X) &= \frac{\gamma\ p^{\alpha+H-1}(1-p)^{\beta+T-1}}{\mathbb{P}(X)} \\
&= \frac{\gamma\ p^{\alpha+H-1}(1-p)^{\beta+T-1}}{\gamma \frac{\Gamma(\alpha+H)\Gamma(\beta+T)}{\Gamma(\alpha+\beta+H+T)}} \\
&= \frac{p^{\alpha+H-1}(1-p)^{\beta+T-1}}{\frac{\Gamma(\alpha+H)\Gamma(\beta+T)}{\Gamma(\alpha+\beta+H+T)}} \\
&= \frac{p^{\alpha+H-1}(1-p)^{\beta+T-1}}{B(\alpha+H,\beta+T)} \\
&= Beta(\alpha+H,\beta+T)
\end{aligned}
$$

The posterior is also a Beta distribution, hence we have proved that Beta distribution is a conjugate prior to the Bernoulli distribution.

Listing 1. mybi.m

```
close all;
clear all;

Beta(0.5,0.5);
Beta(1,1);
Beta(10,10);
Beta(100,100);

function Beta(a,b)
X = [0 1 0 1 0 0 0 0 0 0];
H = length(X(X==1));
T= length(X)-H;
p = 0.01 : 0.01 : 0.99;
y1 = betapdf(p, a, b);
y2 = betapdf(p, a+H, b+T);
y3 = binopdf(H,length(X),p);
y1 = y1/sum(y1);
y2 = y2/sum(y2);
y3 = y3/sum(y3);
figure;
plot(p,y1/sum(y1),'LineStyle','-.','Color','r','LineWidth',2);
hold on
plot(p,y2/sum(y2),'LineStyle',':','Color','b','LineWidth',2);
plot(p,y3/sum(y3),'LineStyle','--','Color','g','LineWidth',2);
legend({'Prior', 'Posterior','Likelihood'},'Location','NorthEast');
hold off
end
```

The main observation is that higher the values of α and β more the similarity between the prior and the posterior. Secondary observation is that the constant pdf in Figure 2 (red line) shows that the standard uniform distribution is a special case of the beta distribution, which occurs when $\alpha = \beta = 1$.

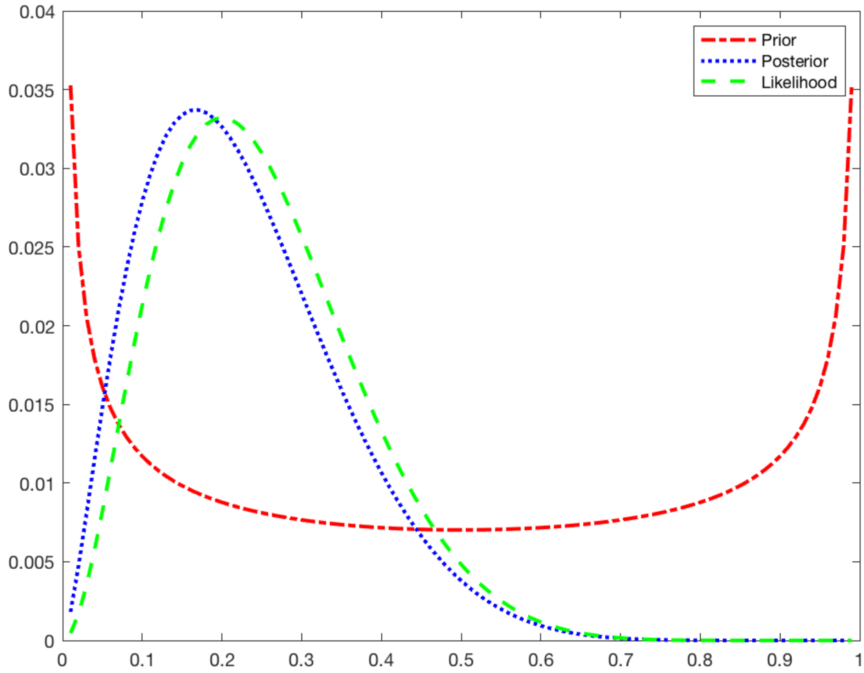

Figure 1. The prior, the posterior and the likelihood for $\alpha = 0.5$ and $\beta = 0.5$

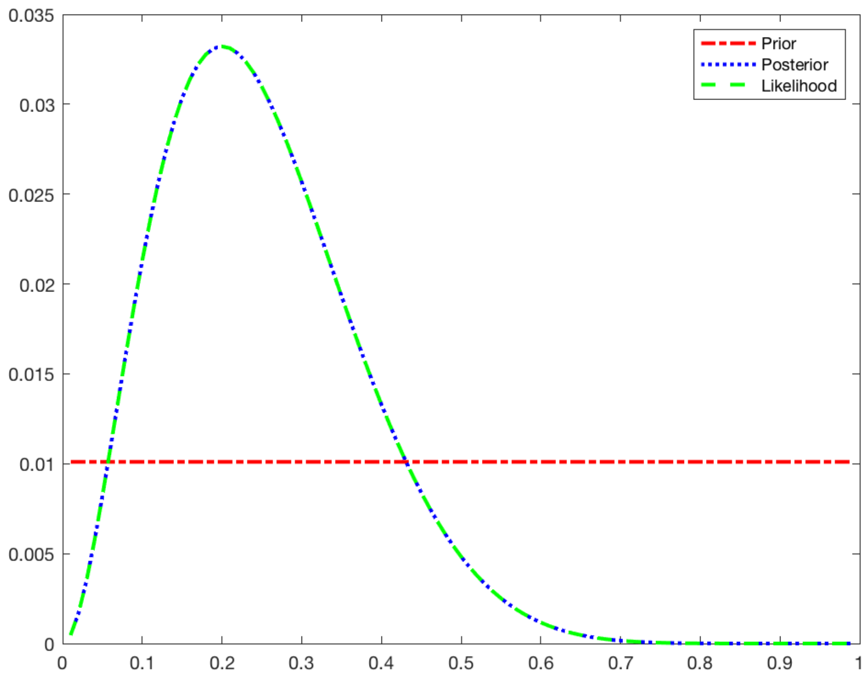

Figure 2. The prior, the posterior and the likelihood for $\alpha = 1$ and $\beta = 1$

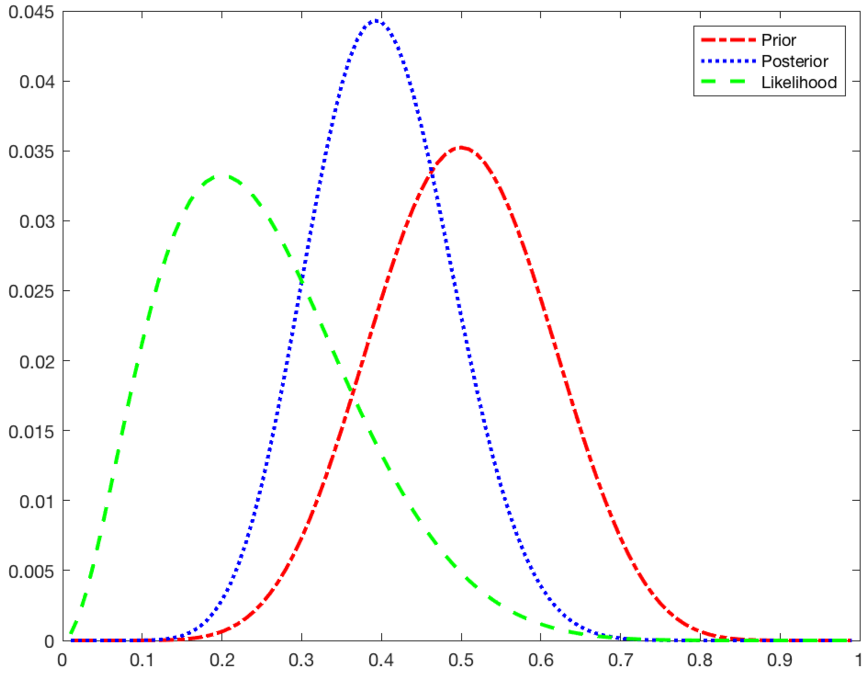

Figure 3. The prior, the posterior and the likelihood for $\alpha = 10$ and $\beta = 10$

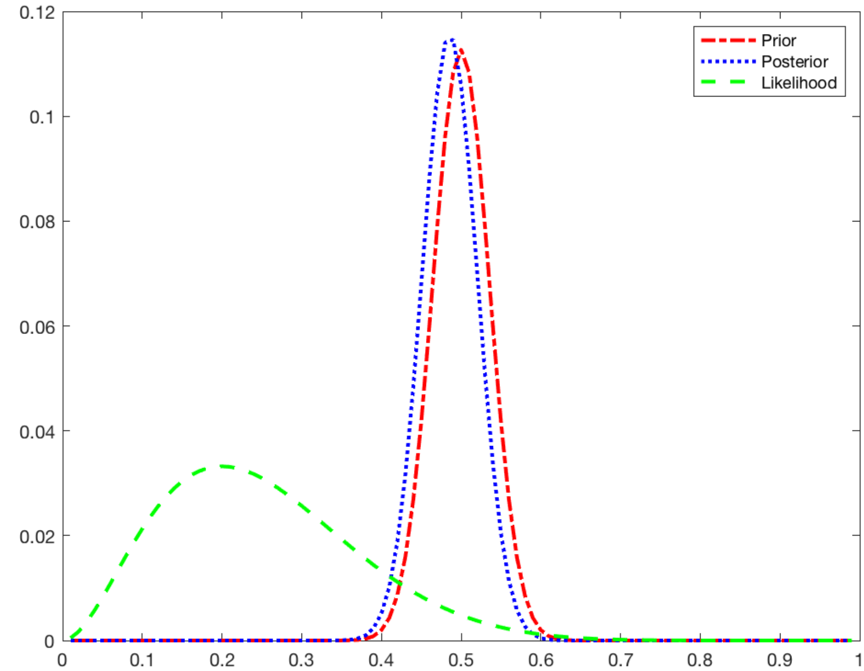

Figure 4. The prior, the posterior and the likelihood for $\alpha = 100$ and $\beta = 100$

2 Principal Component Analysis

Listing 2. mypca.m

```
close all;
clear all;

%load data
data=load('Dax_data.mat');
q = data.Quotes';

q_mean = mean(q,2);

[n,m] = size(q);

X = q - repmat(q_mean,1,m);

figure;
plot(X);

%Reduce the data to 3-dim space
covar = 1 / (m-1) * X * X';

[V,D] = eig(covar);
eigenvals = diag(D);
reduced_V = V(:,1:3);
reduced_X = X'*reduced_V;

figure;
plot(reduced_X);

%Reconstruct the data to original size
reconstructed_X = reduced_V*reduced_X';

figure;
plot(reconstructed_X);

%Compute reconstruction error using the eigenvalues of the covariance matrix
re = (sum(eigenvals(4:n))/sum(eigenvals)) * 100;

disp(['Reconstruction Error: ', num2str(re)])
```

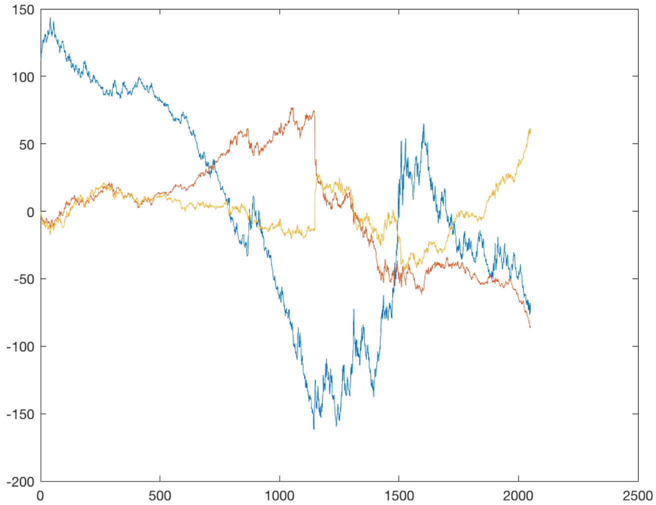

Figure 5. Data after PCA

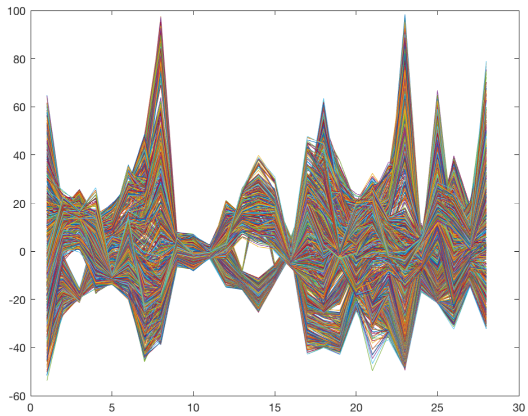

Figure 6. Data before Dimensionality Reduction

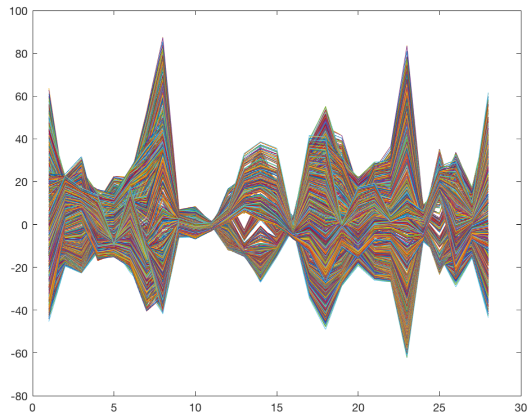

Figure 7. Data after reconstruction

The reconstruction error is **6.5609%**

3 Optimization with constraints

Listing 3. myopti.m

```
clear all;
close all;

x0=[-2,-1];
A=[];
b=[];
Aeq=[];
beq=[];
lb=[];
ub=[];
options = optimoptions('fmincon','Algorithm','interior-point','Display','iter');
fun1=@(x)x(1)+x(2);
x1 = fmincon(fun1,x0,A,b,Aeq,beq,lb,ub,@nonlcon1,options);

disp(['Optimal Solution for first constrained optimization problem: ', num2str(x1)])

x0=[-2,-1];
A=[];
b=[];
Aeq=[];
beq=[];
lb=[];
ub=[];
options = optimoptions('fmincon','Algorithm','interior-point','Display','iter');
fun2=@(x)2*(x(1)-5)^2+3*x(2)^2;
x2 = fmincon(fun2,x0,A,b,Aeq,beq,lb,ub,@nonlcon2,options);

disp(['Optimal Solution for second constrained optimization problem: ', num2str(x2)])

[X,Y] = meshgrid(-10:0.5:10,-10:20);
z1 = 2*((X)-5).^2 + 3*(Y).^2;
z2 = X+Y-10;
figure;
surf(X,Y,z1);
hold on
surf(X,Y,z2);
legend('Objective Function','Constraint');
plot(x1(1),x1(2),'DisplayName','Optimal Solution');
hold off

[X,Y] = meshgrid(-10:0.5:10,-10:20);
z1 = X+Y;
z2 = 4*X.^2+Y.^2-20;
figure;
surf(X,Y,z1);
hold on
surf(X,Y,z2);
legend('Objective Function','Constraint')
plot(x2(1),x2(2),'DisplayName','Optimal Solution')
hold off

function [const, ceq] = nonlcon1(x)
const = 4*x(1).^2+x(2).^2-20;
ceq= [];
end

function [const, ceq] = nonlcon2(x)
const = x(1)+x(2)-10;
```

```
ceq= [];
end
```

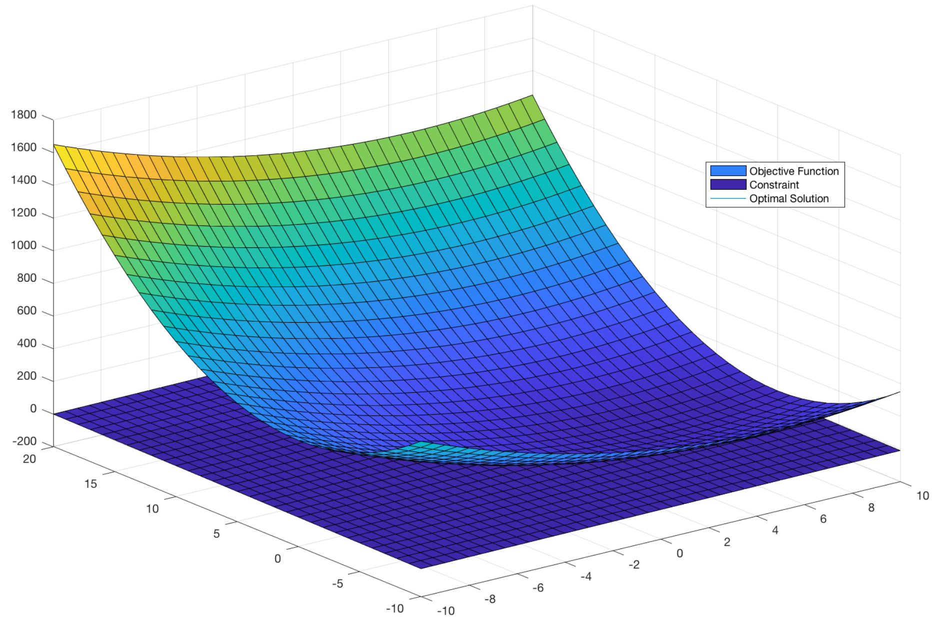

Figure 8. Surface plot of the objective function and the constraint

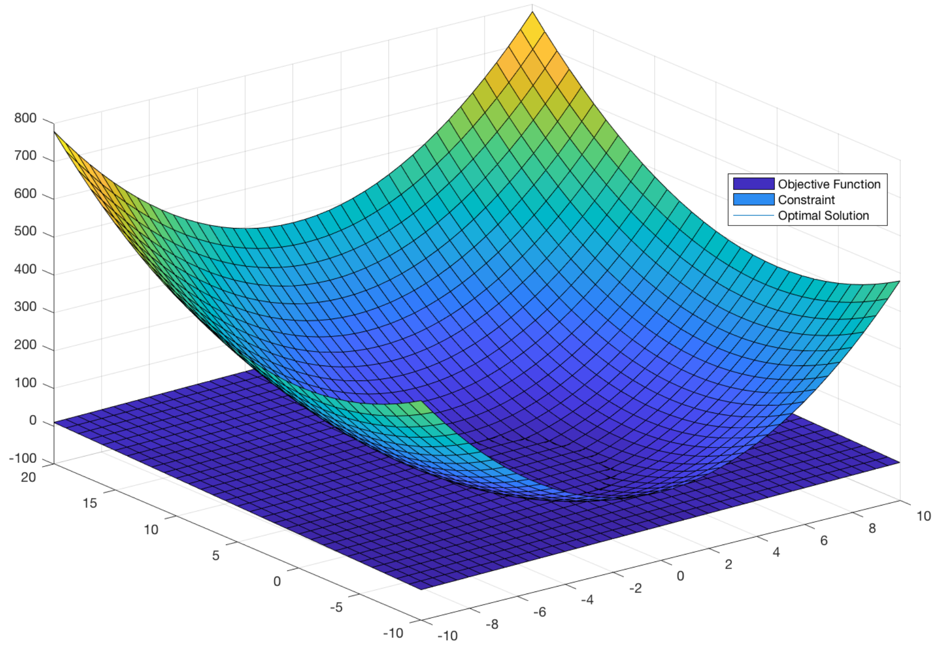

Figure 9. Surface plot of the objective function and the constraint

Optimal Solution for first constrained optimization problem: **-1, -4**
Optimal Solution for second constrained optimization problem: **5, -5.8965e-08**