By

Rasesh Garg
Mohammed Nifaullah Sailappai

In [2]:
```
using Distributions;
using Gadfly;
```

In [3]:
```
d = Beta(2,2);
sample = rand(d,1000)
```

Out[3]:
```
1000-element Array{Float64,1}:
 0.143184
 0.29398
 0.712605
 0.859599
 0.592666
 0.440984
 0.601532
 0.138286
 0.498207
 0.554796
 0.492748
 0.881315
 0.882574
 ⋮
 0.527696
 0.295419
 0.578861
 0.471314
 0.349953
 0.4733
 0.565724
 0.623227
 0.260835
 0.475413
 0.583345
 0.500478
```

In [16]:
```
function dl_by_da(sample,a,b)
    n = length(sample)
    result = -n*digamma(a+b) + n*digamma(a) - sum(log.(sample));
    return result;
end
```

Out[16]:
```
dl_by_da (generic function with 1 method)
```

In [17]:
```
function dl_by_db(sample,a,b)
    n = length(sample)
    result = -n*digamma(a+b) + n*digamma(b) - sum(log.(1-sample));
    return result;
end
```

Out[17]: dl_by_db (generic function with 1 method)

In [20]:
```
function gradient_decent_beta(sample)
    n = length(sample);
    max_itr = 1000;
    lr = 0.001;
    a = rand()*10;
    b = rand()*10;

    for i=1:max_itr
        a_new = a -lr*dl_by_da(sample,a,b);
        b_new = b - lr*dl_by_db(sample,a,b);
        if(a_new < 0) a_new = rand()*10; end;
        if(b_new < 0) b_new = rand()*10; end;
        if (abs(a_new - a) <0.0001 && abs(b_new-b)< 0.0001) break; end;
        a = a_new; b = b_new;
    end
    return a,b;
end
```

Out[20]: gradient_decent_beta (generic function with 1 method)

In [25]:
```
gradient_decent_beta(sample)
```

Out[25]: (1.9913310267352218, 2.058715221582262)

Checking consistency for larger sample size

In [22]:
```
gradient_decent_beta(rand(d,10000))
```

Out[22]: (20.839491385525896, 20.60188461215319)

In [24]:
```
gradient_decent_beta(rand(d,100000))
```

Out[24]: (942.424536201785, 946.367852908452)

The function is not consisetent, actually the values are closer to the true values for the smaller sample size of 1000 and not for larger parameters.\ Its unexpecyed can be probably due to initialization or a local minima and hence needs to be explored further.

Lec - 8 (In-class)

i.
$$\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\,\Gamma(\beta)}\, x^{\alpha-1}\,(1-x)^{\beta-1}$$

each x_i is i.i.d variable with Beta distribution.

(1) Likelihood $(\alpha, \beta \mid x_n)$

a.
$$= \left(\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\,\Gamma(\beta)}\right)^{n} \left(\prod x_i\right)^{(\alpha-1)} \left(\prod (1-x_i)^{\beta-1}\right)$$

b. Log Likelihood

$$\log\left(L(\alpha,\beta \mid x_i)\right) = n \log \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\,\Gamma(\beta)} + (\alpha-1)\left[\sum \log x_i\right] + (\beta-1)\left[\sum \log(1-x_i)\right]$$

Negative log-likelihood $= -\log L(\alpha, \beta \mid x_i)$

c. $\alpha_{new} =$

$$= -n \log(\Gamma(\alpha+\beta)) + n\log \Gamma(\alpha) + n\log \Gamma(\beta)$$
$$- (\alpha-1)\left(\sum \log x_i\right) - (\beta-1)\left(\sum \log(1-x_i)\right)$$

c. Randomly initiate the parameter.
Update equations are given by:

$$\alpha_{new} = \alpha_{old} - \lambda \frac{\partial L(\alpha, \beta | \cancel{x})}{\partial \alpha}$$

$$\beta_{new} = \beta_{old} - \lambda \frac{\partial L(\alpha, \beta | x)}{\partial \beta}$$

Algorithm:

d. (i) Initiate with random values for α and β

Iteration step: 1 to m

(ii) Updated parameters based on gradient descent

$$\alpha_{new} = \alpha_{old} - \lambda \frac{\partial L(\alpha, \beta | x)}{\partial \alpha}$$

$$\beta_{new} = \beta_{old} - \lambda \frac{\partial L(\alpha, \beta | \cancel{x})}{\partial \beta}$$

(iii) Stopping criteria.

max. no. of iterations $\Rightarrow m$

or convergence of the ~~function~~ parameters.

i.e ~~$y_{new} - y_{old}$~~ $|w_i - w_{i-1}| < \varepsilon$

where w_i is the parameter in i^{th} iteration.

approximation to $\frac{\partial \Gamma(\alpha+\beta)}{\partial \alpha}$

c' $\frac{\partial}{\partial \alpha} = -n \text{ digamma}(\alpha+\beta) + n \text{digamma}(\alpha) - \left(\sum \log x_i\right)$

$\frac{\partial}{\partial \beta} = -n \text{ digamma}(\alpha+\beta) + n \text{digamma}(\beta) - \left(\sum \log(1-x_i)\right)$