# STA 601 - Homework 9

Kedar Prabhudesai

October 11, 2013

## Hierarchical Model: This model borrows information across experimental conditions.

Reaction times, $y_{ij}$, for subjects $j = 1, 2, \ldots, n_i$, in experimental conditions, $i = 1, 2, \ldots, n$.

$y_i \sim Exp(\lambda_i)$, - Within Condition Model
$\lambda_i \sim Gamma(a, b)$. - Between Condition Model.

Priors on $a$ and $b$.
$a \sim e^{-a\alpha_a}$
$b \sim Gamma(\alpha_b, \beta_b)$.

**Posterior Computation:**

$$
\begin{aligned}
p(\lambda_1, \lambda_2, \ldots, \lambda_n, a, b \mid y_1, y_2, \ldots, y_n) &\propto p(y_1, y_2, \ldots, y_n \mid \lambda_1, \lambda_2, \ldots, \lambda_n, a, b) \times p(\lambda_1, \lambda_2, \ldots, \lambda_n \mid a, b) \times p(a) \times p(b) \\
&\propto \prod_{i=1}^{n} \prod_{j=1}^{n_i} p(y_{ij} \mid \lambda_i, a, b) \prod_{i=1}^{n} p(\lambda_i \mid a, b) \times p(a) \times p(b)
\end{aligned}
$$

**Full Conditionals:**

- $p(\lambda_i \mid a, b, y_1, y_2, \ldots, y_n)$

$$\begin{aligned}
p(\lambda_i \mid a, b, y_1, y_2, \ldots, y_n) &\propto p(\lambda_i \mid a, b) \prod_{j=1}^{n_i} p(y_j \mid \lambda_i, a, b) \\
&\propto \lambda_i^{a-1} exp(-\lambda_i b) \prod_{j=1}^{n_i} \lambda_i exp(-\lambda_i y_j) \\
&\propto \lambda_i^{a-1} exp(-\lambda_i b) \lambda_i exp\left(-\lambda_i \sum_{j=1}^{n_i} y_j\right) \\
&\propto \lambda_i^{(a+n_i)-1} exp\left[-\lambda_i \left(b + \sum_{j=1}^{n_i} y_j\right)\right] \\
\therefore p(\lambda_i \mid a, b, y_1, y_2, \ldots, y_n) &\sim Gamma\left(a + n_i, b + \sum_{j=1}^{n_i} y_j\right).
\end{aligned}$$

- $p(a \mid \lambda_1, \lambda_2, \ldots, \lambda_n, b, y_1, y_2, \ldots, y_n)$

$$\begin{aligned}
p(a \mid \lambda_1, \lambda_2, \ldots, \lambda_n, b, y_1, y_2, \ldots, y_n) &\propto p(a) \prod_{i=1}^{n} p(\lambda_i \mid a, b) \\
&\propto exp(-a\alpha_a) \prod_{i=1}^{n} \lambda_i^{a-1} exp(-\lambda_i b) \\
&\propto \left(\prod_{i=1}^{n} \lambda_i\right)^{a-1} exp(-a\alpha_a)
\end{aligned}$$

  We can sample from this distribution in Matlab by using 'randsample' which is similar to 'sample' in R.

- $p(b \mid \lambda_1, \lambda_2, \ldots, \lambda_n, a, y_1, y_2, \ldots, y_n)$

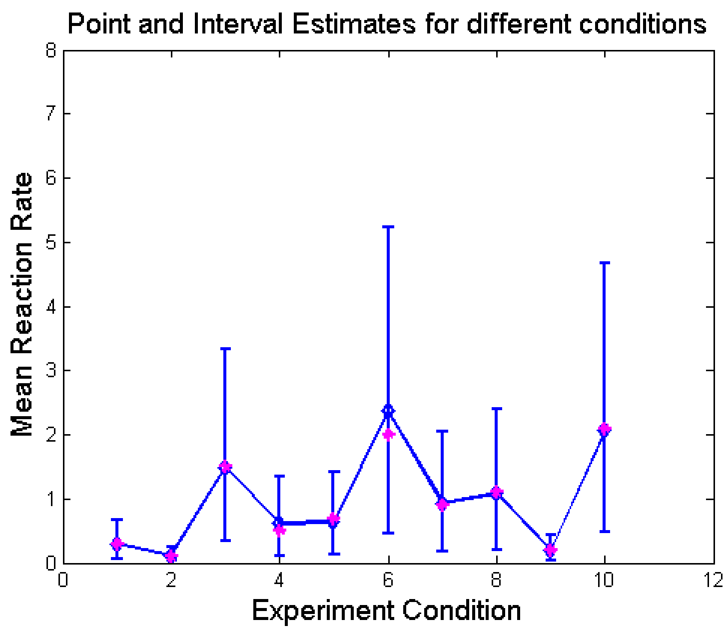
$$\begin{aligned}
p(b \mid \lambda_1, \lambda_2, \ldots, \lambda_n, a, y_1, y_2, \ldots, y_n) &\propto p(b) \prod_{i=1}^{n} p(\lambda_i \mid a, b) \\
&\propto b^{\alpha_b - 1} exp(-b\beta_b) \prod_{i=1}^{n} \lambda_i^{a-1} exp(-\lambda_i b) \\
&\propto b^{\alpha_b - 1} exp(-b\beta_b) exp\left(-\sum_{i=1}^{n} \lambda_i b\right) \\
&\propto b^{\alpha_b - 1} exp\left[-b\left(\beta_b + \sum_{i=1}^{n} \lambda_i\right)\right] \\
\therefore p(b \mid \lambda_1, \lambda_2, \ldots, \lambda_n, a, y_1, y_2, \ldots, y_n) &\sim Gamma\left(\alpha_b, \beta_b + \sum_{i=1}^{n} \lambda_i\right).
\end{aligned}$$

**Gibbs Sampling:**

Start with $\{a^{(0)}, b^{(0)}\}$.

- Draw, $\lambda_i^{(s)} \sim Gamma\left(a^{(s)} + n_i, b^{(s)} + \sum_{j=1}^{n_i} y_j\right)$, i=1,2,...,n.
- Draw, $a^{(s+1)} \sim \left(\prod_{i=1}^{n} \lambda_i^{(s)}\right)^{a_{grid}-1} exp(-a_{grid}\alpha_a)$, using Griddy Gibbs ('randsample' in Matlab).
- Draw, $b^{(s+1)} \sim Gamma\left(\alpha_b, \beta_b + \sum_{i=1}^{n} \lambda_i^{(s)}\right)$.
- Repeat.

Given below are results. The dots in Magenta denote the true values of $\lambda_i$ used to simulate data. Blue lines denote means and 95% Credible Intervals.



**Non-Hierarchical Model:** This model does NOT borrow information across experimental conditions. To achieve that, we group together data across experimental conditions and use a common model for the data.

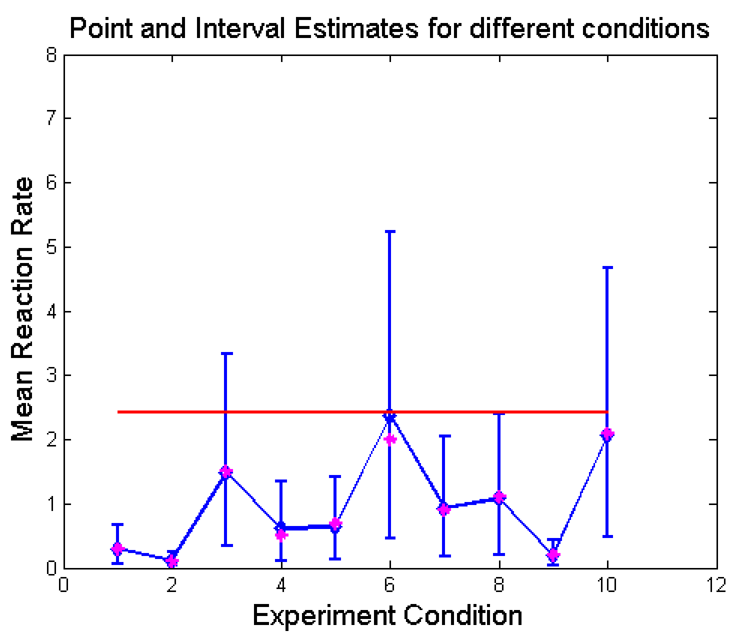
Reaction times, $y_k$, for subjects $k = 1, 2, \ldots, m$. Where, $m$ is the total number of data points pooled across all conditions.

$y \sim Exp(\lambda)$,
$\lambda \sim Gamma(a, b)$.

**Posterior:**

$$\begin{aligned}
p(\lambda \mid y) &\propto L(y; \lambda) p(\lambda) \\
&\propto \prod_{k=1}^{m} \lambda exp(-\lambda y_k) \lambda^{a-1} exp(-\lambda b) \\
&\propto \lambda^{a+m-1} exp\left[-\lambda \left(b + \sum_{k=1}^{m} y_k\right)\right] \\
p(\lambda \mid y) &\sim Gamma\left(a + m, b + \sum_{k=1}^{m} y_k\right)
\end{aligned}$$

Therefore, the posterior mean is given by, $E(\lambda \mid y) = \frac{a+m}{b+\sum_{k=1}^{m} y_k}$ Using, $a = 2$, $b = 3$, (same initial values used for Gibbs Sampler), $m = 835$ we get, $E(\lambda \mid y) = 2.4191$. If we plot this on the previous plot (red line), we can see that it is not a very good estimator for the condition specific reaction times.



# Appendix:

```
%% STA 601 — Homework 9
% Author: Kedar Prabhudesai
% Created on: 10/09/2013

close all;
clear all;

%% This is the hieracrchical Model
% Yij ¬ Exp(Li)
%  Li ¬ Gamma(a,b)
%   a ¬ Gamma(Aa,Ba)
%   b ¬ Gamma(Ab,Bb)

%% Simulate Data
% Dummmy Variables to simulate data
nConditions = 10;
nSubjectsPerCondition = [100 75 65 100 80 90 100 95 70 60];
LambdaI = [0.3 0.1 1.5 0.5 0.7 2 0.9 1.1 0.2 2.1];
Yij = cell(1,nConditions);

% Create Distribution Objects
for iCond = 1:nConditions
    DistObj = makedist('Gamma','a',1,'b',1/LambdaI(iCond));
    Yij{1,iCond} = DistObj.random(1,nSubjectsPerCondition(iCond));
end

%% Gibbs Sampler
nGibbs = 5000;
nBurnIn = 1000;
LiSamples = zeros(nConditions,nGibbs);
aSamples = zeros(1,nGibbs);
bSamples = zeros(1,nGibbs);
%   p(a) ¬ Exp(-a*Aa)
Aa = 5;
%   b ¬ Gamma(Ab,Bb)
Ab = 4;
Bb = 5;
% Initialize
aSamples(1) = 2;
bSamples(1) = 3;
LiGivenAll = makedist('Gamma','a',2,'b',1/3);
bGivenAll = makedist('Gamma','a',Ab,'b',1/Bb);
SumYj = zeros(1,nConditions);

for iCond = 1:nConditions
    SumYj(iCond) = sum(Yij{1,iCond});
end

aGrid = (1:0.1:60);
aGridWeights = -aGrid.*Aa;

for iGibbs = 2:nGibbs
    home;
    disp(iGibbs);
    % Update Li for all i
    for iCond = 1:nConditions
        LiGivenAll.a = aSamples(iGibbs-1) + nSubjectsPerCondition(iCond);
        LiGivenAll.b = 1/(bSamples(iGibbs-1) + SumYj(iCond));
        LiSamples(iCond,iGibbs) = LiGivenAll.random();
    end
```

```

    % Update a
%     aProbWeights = (aSamples(iGibbs-1)-1)*sum(log(LiSamples(:,iGibbs))) + aGridWeights;
    aProbWeights = (aGrid-1)*sum(log(LiSamples(:,iGibbs))) + aGridWeights;
    aProbWeights = exp(aProbWeights - max(aProbWeights));
    aSamples(iGibbs) = randsample(aGrid,1,true,aProbWeights);

    % Update b
    bGivenAll.b = 1/(Bb + sum(LiSamples(:,iGibbs)));
    bSamples(iGibbs) = bGivenAll.random();
end

% Burn-In
aSamples(1:nBurnIn) = [];
bSamples(1:nBurnIn) =  [];
LiSamples(:,1:nBurnIn) = [];

%% Manage Plotting
MeanLi = zeros(nConditions,1);
ConfInts = zeros(nConditions,2);
for iCond = 1:nConditions
    MeanLi(iCond) = mean(LiSamples(iCond,:));
    ConfInts(iCond,:) = quantile(LiSamples(iCond,:),[0.025 0.975]);
end

figure;
errorbar(1:nConditions,MeanLi,ConfInts(:,1),ConfInts(:,2),'Marker','diamond','Linewidth',2)...
    ;hold on;
title('Point and Interval Estimates for different conditions','FontSize',14);
xlabel('Experiment Condition','FontSize',14);
ylabel('Mean Reaction Rate','FontSize',14);
ylim([0 8]);

plot(1:nConditions,LambdaI,'m*','Linewidth',2);hold off;
keyboard
```