STA 601 - Homework 8

Kedar Prabhudesai

September 27, 2013

Missing Data:

To simulate rejecting $p\%$ of data, I created a matrix of random values from $U[0,1]$ and ran a threshold of $p\%$ to get a random indicator matrix O_{ij}. I also made sure that for a given sample we have at least one value present. Using this approach I got a rejection percentage close to $p\%$.

This can be summarized as follows:

- $O_{ij} \in [0,1]$, $i = 1, 2, \ldots, 100$, $j = 1, 2$.
- If $O_{ij} \leq p/100 = 1$, else 0.
- 1: Reject Data, 0: Keep Data.
- Make sure that, $\forall i$ at least one j is present.

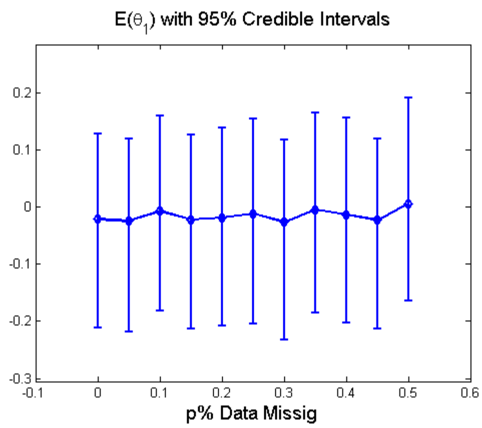
Gibbs Sampling:

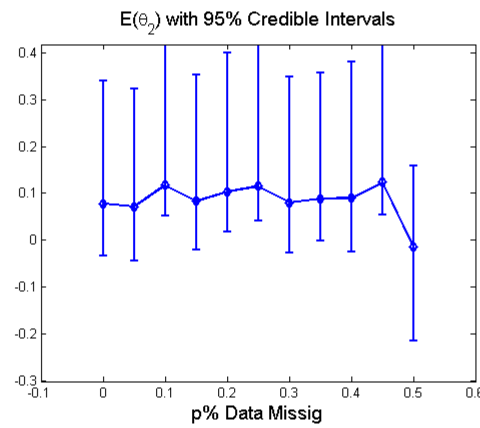
- Priors for θ and Σ :
  $\mu \sim \mathcal{N}_2\left(\mu_0, \Lambda_0\right)$. $\mu_0 = \begin{pmatrix} 0.2 \\ 0.2 \end{pmatrix}$, $\Lambda_0 = \begin{pmatrix} 1.25 & 0.6 \\ 0.6 & 1.25 \end{pmatrix}$
  $\Sigma \sim$ Inverse-Wishart$\left(\nu_0, S_0^{-1}\right)$. $\nu_0 = 4$, $S_0 = \begin{pmatrix} 1.2 & 0.4 \\ 0.4 & 1.2 \end{pmatrix}$.
- Missing Data: We know that our data is drawn from a Bivariate Normal. We can partition each sample as $\{y_{obs}, y_{miss}\}$
- Gibbs Sampler: Given the starting values of $\{\Sigma^{(0)}, y_{miss}^{(0)} = 2\}$, we generate $\{\theta^{(s+1)}, \Sigma^{(s+1)}, y_{miss}^{(s+1)}\}$, as follows:
  - $\theta^{(s+1)} \sim p(\theta | y_{obs}, y_{miss}^{(s)}, \Sigma^{(s)})$
  - $\Sigma^{(s+1)} \sim p(\Sigma | y_{obs}, y_{miss}^{(s)}, \theta^{(s+1)})$
  - $y_{miss}^{(s+1)} \sim p(y_{miss} | y_{obs}, \theta^{(s+1)}, \Sigma^{(s+1)})$

- Full Conditionals:
  - Update $\bar{y}$
  - $\theta^{(s+1)} \mid y_{obs}, y_{miss}^{(s)}, \Sigma^{(s)} \sim \mathcal{N}_2 \left[\left(\frac{n\Sigma^{(s)-1}\bar{y} + \Lambda_0^{-1}\mu_0}{n\Sigma^{(s)-1} + \Lambda_0^{-1}} \right), \left(n\Sigma^{(s)-1} + \Lambda_0^{-1} \right)^{-1} \right]$
  - $\Sigma^{(s+1)} \mid y_{obs}, y_{miss}^{(s)}, \theta^{(s+1)} \sim IW \left[\nu_0 + n, (S_0 + S_\theta)^{-1} \right], S_\theta = \sum_{i=1}^{n} (y_i - \theta^{(s+1)})(y_i - \theta^{(s+1)})'$
  - $y_{miss}^{(s+1)} \mid y_{obs}, \theta^{(s+1)}, \Sigma^{(s+1)} \sim \mathcal{N} \left(\theta_{b|a}^{(s+1)}, \Sigma_{b|a}^{(s+1)} \right)$
    $\theta_{b|a}^{(s+1)} = \theta_b^{(s+1)} + \Sigma_{b,a}^{(s+1)} \left(\Sigma_{a,a}^{(s+1)} \right)^{-1} \left(y_a - \theta^{(s+1)_a} \right)$
    $\Sigma_{b|a}^{(s+1)} = \Sigma_{b,b}^{(s+1)} - \Sigma_{b,a}^{(s+1)} \left(\Sigma_{a,a}^{(s+1)} \right)^{-1} \Sigma_{a,b}^{(s+1)}$,
    where, b is the index of 'Data Missing' and a is index of 'Data Present'.

Results:

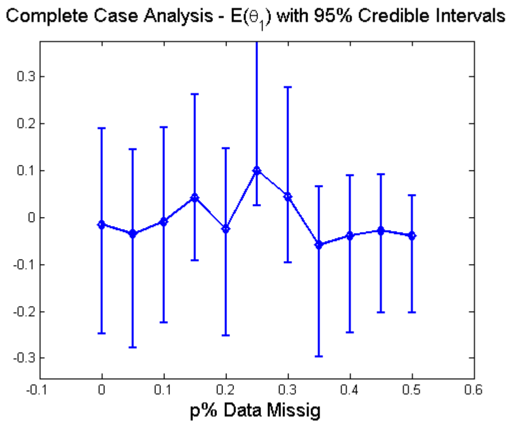
Gibbs Samples = 10000, Burn-In = 1000. Given here are estimates of $E(\theta_1)$ and $E(\theta_2)$ with their 95% Credible Intervals. We see that as the % of rejected data increases, our estimate of mean is less accurate and has a larger credible interval. Note that 0% point represents no data was rejected.

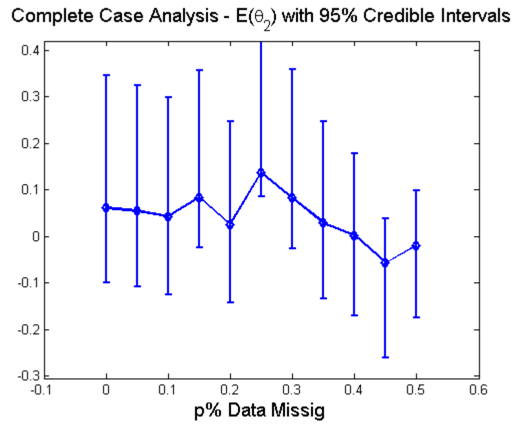




Complete Case Analysis:

For this we reject all samples which contain missing data. By doing this our estimate wanders even more about the true estimate and the intervals are much bigger than when we impute the missing data.





Appendix:

```
%% STA 601 — Homework 8
% Author: Kedar Prabhudesai
% Created on: 9/27/2013

close all;
clear all;

%% Setup Data and Distributions
% Create Bivariate Distribution
nSamples = 100;
rho = 0.8;
mu = [0 0];
SIGMA = [1 rho; rho 1];

% Get Data
rng('shuffle');
yTruth = mvnrnd(mu,SIGMA,nSamples);

pctMissing = 0:0.05:0.5;

y1Bayes = [];
y1BayesConf = [];
y2Bayes = [];
y2BayesConf = [];

%% Missing Data
for iPct = 1:numel(pctMissing)
    % Simulate Missing Data
    yTruthMissing = yTruth;
    yTruthMissing = yTruthMissing';

    % Get Random indices to reject data
    OInds = rand(size(yTruthMissing));
    % Indicator which tells us what is missing: 1—Miss, 0—Present
    Oij = (OInds ≤ pctMissing(iPct));
    disp(['Processing ',num2str(pctMissing(iPct)),' Actual Percent ',num2str(sum(Oij(:))/...
        numel(OInds))]);

    % Reject Data
    yTruthMissing(Oij) = NaN;
    % Make sure we do not have both y1 and y2 missing
    y1AndY2Missing = isnan(yTruthMissing(1,:)) & isnan(yTruthMissing(2,:));

    for iSample = 1:nSamples
        idxRnd = randperm(2,1);
        yTruthMissing(idxRnd,iSample) = yTruth(iSample,idxRnd);
        Oij(idxRnd,iSample) = 0;
    end
    idxMissing = find(sum(Oij,1)==1);

    %% Gibbs Sampler
    nGibbs = 10000;
    nBurnIn = 1000;

    mu0 = [0.2 0.2]';
    L0 = [1.25 0.6;0.6 1.25];

    nu0 = 4;
    S0 = [1.2 0.4;0.4 1.3];

```

```matlab
    thetaSamples = zeros(2,nGibbs);
    sigmaSamples = zeros(2,2,nGibbs);

    % Initialize first value of Sigma and missing data
    sigmaSamples(:,:,1) = S0;
    for iMiss = 1:numel(idxMissing)
        % b: Missing Index
        b = find(Oij(:,idxMissing(iMiss)));
        yTruthMissing(b,idxMissing(iMiss)) = 2;
    end

    for iGibbs = 2:nGibbs
        % Update ybar
        ybar = mean(yTruthMissing,2);

        % Update theta
        Ln = inv(inv(L0) + nSamples.*inv(sigmaSamples(:,:,iGibbs-1)));
        mun = Ln*(inv(L0)*mu0 + nSamples.*inv(sigmaSamples(:,:,iGibbs-1))*ybar);
        thetaHere = mvnrnd(mun,Ln);
        thetaSamples(:,iGibbs) = thetaHere;
        % Update Sigma
        Sn = S0 + (bsxfun(@minus,yTruthMissing,thetaSamples(:,iGibbs)))*(bsxfun(@minus,...
            yTruthMissing,thetaSamples(:,iGibbs)))';
        Z = mvnrnd([0 0],inv(Sn),nu0+nSamples);
        sigmaHere = inv(Z'*Z);
        sigmaSamples(:,:,iGibbs) = sigmaHere;

        % Update Missing Data
        for iMiss = 1:numel(idxMissing)
            % a: Present Index
            % b: Missing Index
            b = find(Oij(:,idxMissing(iMiss)));
            if b == 1
                a = 2;
            else
                a = 1;
            end
            ThetaMissGivenPres = thetaHere(b) + (sigmaHere(b,a)*(yTruthMissing(a,idxMissing...
                (iMiss))-thetaHere(a)))/sigmaHere(a,a);
            SigmaMissGivenPres = sqrt(sigmaHere(b,b) - (sigmaHere(b,a)*sigmaHere(a,b))/...
                sigmaHere(a,a));
            yTruthMissing(b,idxMissing(iMiss)) = ThetaMissGivenPres + SigmaMissGivenPres*...
                randn;
        end
    end

    % Burn-In
    thetaSamples(:,1:nBurnIn) = [];
    sigmaSamples(:,:,1:nBurnIn) = [];

    % Estimate mu and Sigma
    muBayes = mean(thetaSamples,2)';
    sigmaBayes = mean(sigmaSamples,3);

    % Organize to plot
    y1Bayes = [y1Bayes;muBayes(1)];
    y1BayesConf = [y1BayesConf;quantile(thetaSamples(1,:),[0.025 0.975])];
    y2Bayes = [y2Bayes;muBayes(2)];
    y2BayesConf = [y2BayesConf;quantile(thetaSamples(2,:),[0.025 0.975])];
end
figure;
errorbar(pctMissing,y1Bayes,y1BayesConf(:,1),y1BayesConf(:,2),'Marker','diamond','Linewidth...
    ',2);
title('E(\theta_1) with 95% Credible Intervals','FontSize',14);
```

```
xlabel('p% Data Missig','FontSize',14);
ylim([min(y1BayesConf(:,1))-0.1 max(y1BayesConf(:,2))+0.1]);
xlim([-0.1 0.6]);

figure;
errorbar(pctMissing,y2Bayes,y2BayesConf(:,1),y2BayesConf(:,2),'Marker','diamond','Linewidth...
    ',2);
title('E(\theta_2) with 95% Credible Intervals','FontSize',14);
xlabel('p% Data Missig','FontSize',14);
ylim([min(y2BayesConf(:,1))-0.1 max(y2BayesConf(:,2))+0.1]);
xlim([-0.1 0.6]);

%% Complete Case Analysis
y1Bayes = [];
y1BayesConf = [];
y2Bayes = [];
y2BayesConf = [];

for iPct = 1:numel(pctMissing)
    % Simulate Missing Data
    yTruthMissing = yTruth;
    yTruthMissing = yTruthMissing';

    % Get Random indices to reject data
    OInds = rand(size(yTruthMissing));
    % Indicator which tells us what is missing: 1-Miss, 0-Present
    Oij = (OInds <= pctMissing(iPct));
    disp(['Processing ',num2str(pctMissing(iPct)),' Actual Percent ',num2str(sum(Oij(:))/...
        numel(OInds))]);

    % Reject Data
    yTruthMissing(Oij) = NaN;
    % Make sure we do not have both y1 and y2 missing
    y1AndY2Missing = isnan(yTruthMissing(1,:)) & isnan(yTruthMissing(2,:));

    for iSample = 1:nSamples
        idxRnd = randperm(2,1);
        yTruthMissing(idxRnd,iSample) = yTruth(iSample,idxRnd);
        Oij(idxRnd,iSample) = 0;
    end
    idxMissing = find(sum(Oij,1)==1);

    %% Get rid of samples where data is missing
    for iMiss = 1:numel(idxMissing)
        yTruthMissing(:,idxMissing(iMiss)) = [0;0];
    end

    %% Gibbs Sampler
    nGibbs = 10000;
    nBurnIn = 1000;

    mu0 = [0.2 0.2]';
    L0 = [1.25 0.6;0.6 1.25];

    nu0 = 4;
    S0 = [1.2 0.4;0.4 1.3];

    thetaSamples = zeros(2,nGibbs);
    sigmaSamples = zeros(2,2,nGibbs);

    % Initialize first value of Sigma
    sigmaSamples(:,:,1) = S0;
    ybar = mean(yTruthMissing,2);

```

```
    for iGibbs = 2:nGibbs
        % Update theta
        Ln = inv(inv(L0) + nSamples.*inv(sigmaSamples(:,:,iGibbs-1)));
        mun = Ln*(inv(L0)*mu0 + nSamples.*inv(sigmaSamples(:,:,iGibbs-1))*ybar);
        thetaHere = mvnrnd(mun,Ln);
        thetaSamples(:,iGibbs) = thetaHere;

        % Update Sigma
        Sn = S0 + (bsxfun(@minus,yTruthMissing,thetaSamples(:,iGibbs)))*(bsxfun(@minus,...
            yTruthMissing,thetaSamples(:,iGibbs)))';
        Z = mvnrnd([0 0],inv(Sn),nu0+nSamples);
        sigmaHere = inv(Z'*Z);
        sigmaSamples(:,:,iGibbs) = sigmaHere;
    end

    % Burn-In
    thetaSamples(:,1:nBurnIn) = [];
    sigmaSamples(:,:,1:nBurnIn) = [];

    % Estimate mu and Sigma
    muBayes = mean(thetaSamples,2)';
    sigmaBayes = mean(sigmaSamples,3);

    % Organize to plot
    y1Bayes = [y1Bayes;muBayes(1)];
    y1BayesConf = [y1BayesConf;quantile(thetaSamples(1,:),[0.025 0.975])];
    y2Bayes = [y2Bayes;muBayes(2)];
    y2BayesConf = [y2BayesConf;quantile(thetaSamples(2,:),[0.025 0.975])];
end
figure;
errorbar(pctMissing,y1Bayes,y1BayesConf(:,1),y1BayesConf(:,2),'Marker','diamond','Linewidth',2);
title('Complete Case Analysis - E(\theta_1) with 95% Credible Intervals','FontSize',14);
xlabel('p% Data Missig','FontSize',14);
ylim([min(y1BayesConf(:,1))-0.1 max(y1BayesConf(:,2))+0.1]);
xlim([-0.1 0.6]);

figure;
errorbar(pctMissing,y2Bayes,y2BayesConf(:,1),y2BayesConf(:,2),'Marker','diamond','Linewidth',2);
title('Complete Case Analysis - E(\theta_2) with 95% Credible Intervals','FontSize',14);
xlabel('p% Data Missig','FontSize',14);
ylim([min(y2BayesConf(:,1))-0.1 max(y2BayesConf(:,2))+0.1]);
xlim([-0.1 0.6]);
```