ECE414 Project 1: Conjugate Priors
Junbum Kim, Andy Jeong

1 Posterior for Bernoulli Distribution

From Eq. 2.18 of the textbook [1], the update equation for Beta distribution as conjugate prior is:

$$P(\mu|m,l,a,b) = \frac{\Gamma(m+a+l+b)}{\Gamma(m+a)\Gamma(l+b)}\mu^{m+a-1}(1-\mu)^{l+b-1}$$

Expected value for this posterior density is then:

$$E[\mu|m,l,a,b] = \int \mu P(\mu|m,l,a,b)d\mu$$

$$= \frac{\Gamma(m+a+l+b)}{\Gamma(m+a)\Gamma(l+b)}\int \mu\mu^{m+a-1}(1-\mu)^{l+b-1}d\mu$$

$$= \frac{\Gamma(m+a+l+b)}{\Gamma(m+a)\Gamma(l+b)}\frac{\Gamma(m+a+1)\Gamma(l+b)}{\Gamma(m+a+l+b+1)}^{*}$$

$$= \frac{\Gamma(m+a+l+b)}{\Gamma(m+a)\Gamma(l+b)}\frac{(m+a)\Gamma(m+a)\Gamma(l+b)}{(m+a+l+b)\Gamma(m+a+l+b)}$$

$$= \frac{m+a}{m+a+l+b}$$

* using property of Beta distribution (Eq. 2.15):

$$E[\mu|a,b] = \int \mu K(a,b)\mu^{a-1}(1-\mu)^{b-1}d\mu$$

$$= \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\frac{\Gamma(a+1)\Gamma(b)}{\Gamma(a+1+b)} = \frac{a}{a+b} \qquad (\text{using } \Gamma(a+1) = a\Gamma(a))$$

2 Posterior for Normal Distribution

$$\text{Likelihood: } p(X|\mu) = \frac{1}{(2\pi\sigma^2)^{N/2}}e^{-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2} \qquad (Eq.2.137) \qquad [1]$$

$$\mu_{ML} = \frac{1}{N}\sum_{n=1}^{N}x_n \qquad (Eq.2.121) \qquad [1]$$

$$\Sigma_{ML} = \frac{1}{N}\sum_{n=1}^{N}(x_n-\mu_{ML})(x_n-\mu_{ML})^T \qquad (Eq.2.122) \qquad [1]$$

$$\text{Posterior: } p(\mu|X) = N(\mu|\mu_N, \sigma_N^2)$$

$$\mu_N = \frac{\sigma^2}{N\sigma_0^2+\sigma^2}\mu_0 + \frac{N\sigma_0^2}{N\sigma_0^2+\sigma^2}\mu_{ML} \qquad (Eq.2.141) \qquad [1]$$

$$\frac{1}{\sigma_N^2} = \frac{1}{\sigma_0^2} + \frac{N}{\sigma^2} \qquad (Eq.2.142) \qquad [1]$$

3 Posterior for Normal-Gamma Distribution

$$\text{Parameters: } \mu(real), \lambda > 0(real), \alpha > 0(real), \beta > 0(real) \qquad [3]$$

$$\text{PDF: } f(x,\tau|\mu,\lambda,\alpha,\beta) = \frac{\beta^\alpha\sqrt{\lambda}}{\Gamma(\alpha)\sqrt{2\pi}}\tau^{\alpha-\frac{1}{2}}e^{\alpha-\frac{1}{2}}e^{-\frac{\lambda\tau(x-\mu)^2}{2}}$$

$$\text{Likelihood: } p(D|\mu,\lambda) = \frac{1}{(2\pi)^{n/2}}\lambda^{n/2}e^{(-\frac{\lambda}{2}\sum_{i=1}^{n}(x_i-\mu)^2)} \qquad (Eq.61) \qquad [2]$$

$$\text{Posterior: } p(\mu,\tau|D) = NG(\mu,\lambda|\mu_n,\lambda_n,\alpha_n,\beta_n)$$

$$\mu_n = \frac{\lambda\mu_0 + n\bar{x}}{\lambda + n}, \qquad \lambda_n = \lambda_0 + n, \qquad \alpha_n = \alpha_0 + n/2 \qquad (Eq.86-89) \qquad [2]$$

$$\beta_n = \beta_0 + \frac{1}{2}\sum_{i=1}^{n}(x_i - \bar{x})^2 + \frac{\lambda_0 n(\bar{x}-\mu_0)^2}{2(\lambda_0+n)}$$

$$E[\tau] = \frac{\alpha}{\beta} \qquad (precision) \qquad [3]$$

Reference:

Pattern Recognition and Machine Learning, C.M. Bishop (2006)[1]

`https://www.cs.ubc.ca/~murphyk/Papers/bayesGauss.pdf` [2]

`https://en.wikipedia.org/wiki/Normal-gamma_distribution` [3]

Contents

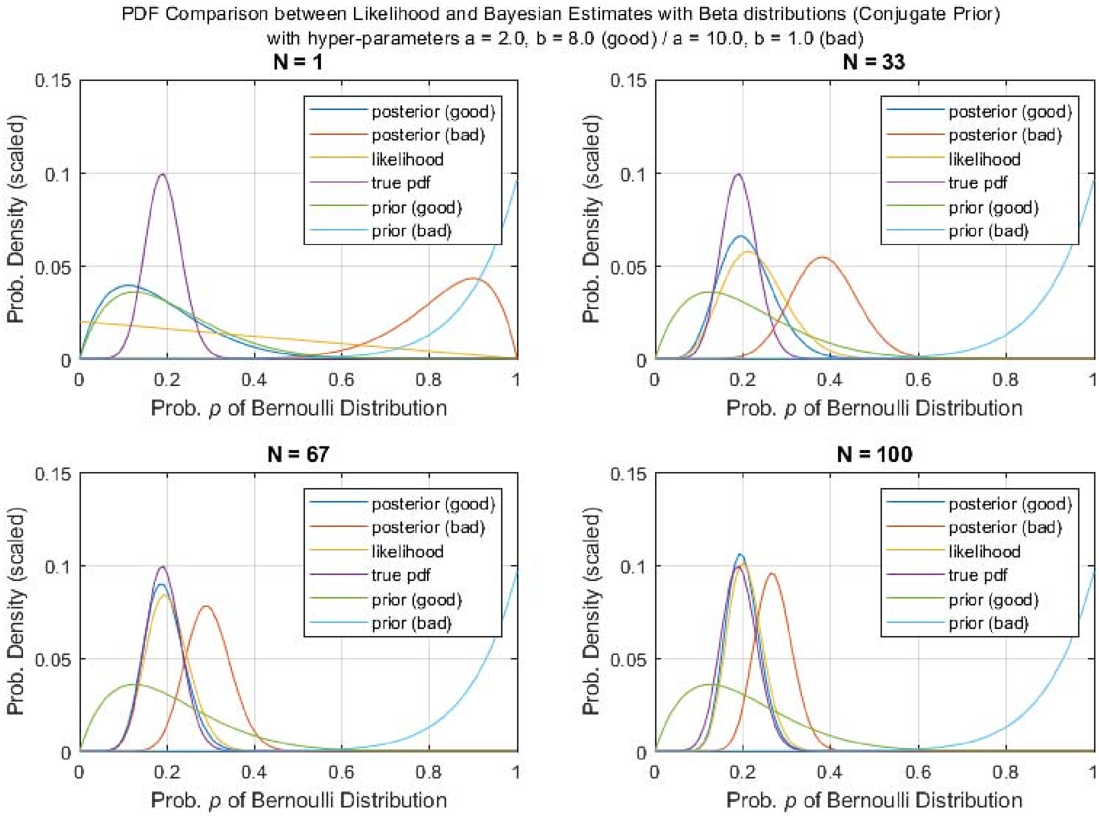

```
% Junbum Kim, Andy Jeong
% ECE414 - Bayesian Machine Learning
% Project 1: Bayesian Estimation with Conjugate Priors
% October 2, 2019
% Primary Reference: [1] Pattern Recognition and Machine Learning
%                                         by Chris. M. Bishop (2006)
% Assumption: all drawn random variables are i.i.d
clear all; close all; clc;
rng default; Extent = matlab.desktop.commandwindow.size;
```

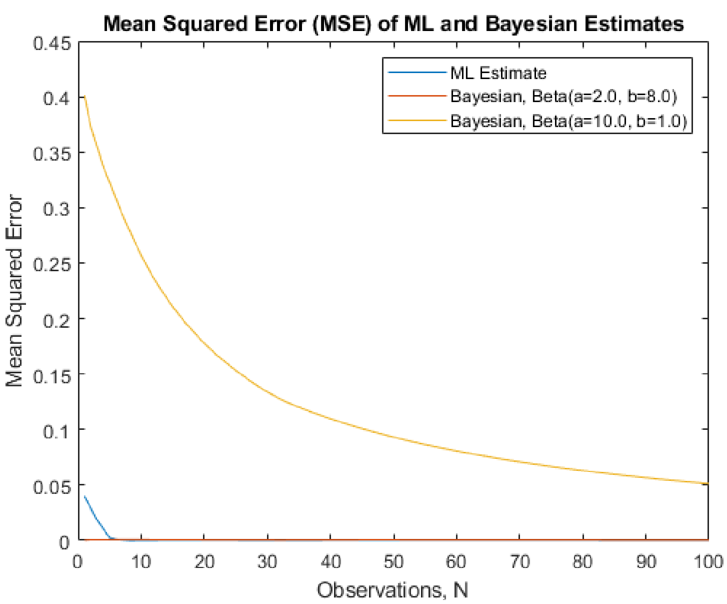
Simulation 1

Bernoulli Random Variables

```
% - Comparison between Maximum Likelihood (ML) estimate and estimation
% using conjugate priors (Beta distribution), with one "good" and one "bad"
% set of parameters for the conjugate prior

% ---- Relevant Equations ----
% (1) ML estimate of the mean (Equation 2.7 of [1])
% mu_{ML} = 1/N * sum(x_n) [from n=1 to N] (= sample mean)
binomial_likelihood = @(mu, m, l) (mu).^(m) .* (1-mu).^(l);

% (2) Conjugate Prior (Beta distribution) density (Equation 2.13 of [1])
% p(mu | a,b) = \frac{gamma(a+b)}{gamma(a}*gamma(b)} ...
%                      * mu^{a-1} * (1-mu)^{b-1}
beta_conj_prior = @(a, b, mu) gamma(a+b)/(gamma(a)*gamma(b)) ...
                    .* mu.^(a-1) .* (1-mu).^(b-1);

% (3) Conjugate prior (Beta distribution) update (Equation 2.18 of [1])
% p(mu | m,l,a,b) = \frac{gamma(m+a+l+b)}{gamma(m+a}*gamma(l+b)} ...
%                      * mu^{m+a+-1} * (1-mu)^{l+b-1}
beta_conj_prior_update = @(m, l, a, b, mu) ...
  gamma(m+b+l+b)/(gamma(m+a)*gamma(l+b)) .* mu.^(m+a-1) .* (1-mu).^(l+b-1);

% (4) Mean of the posterior beta distribution (Equation 2.20 of [1])
% expected value of (3) update equation (= E[mu | m,l,a,b])
% ** see attached page for equations
Bayes_mean = @(m,l,a,b) (m+a)/(m+a+l+b);

% (Other utility) Scaling to [0 1]
scale_pdf = @(x) x./sum(x);
% ---- end Relevant Equations ----

% Define the true mean and pdf, and the number of observations
% since mu is the parameter we want to predict, we aim to see
% the distributions over [0 1] range and determine its most likely value
true_mean = 0.2;            % arbitrary value set to be 'true' to compare
Nmin = 1; Nmax = 100;       % run up to 100 observations
mu = linspace(0,1,Nmax);    % sweep through the range
true_pdf = binopdf(Nmin:Nmax, Nmax, true_mean); % generate pdf with above
bernoulli_rvs = binornd(1, true_mean, [1, Nmax]); % generate random var's

% Hyper-parameters for Beta-distributions
good_a = 2; good_b = 8; % arbitrarily chosen to yield "good" results
bad_a = 10; bad_b = 1;  % arbitrarily chosen to yield "bad" results

% Initialization for memory space
max_likelihood = zeros(1, Nmax);
mse_max_likelihood = zeros(1, Nmax);
Bayes_estimates_good = zeros(1, Nmax);
Bayes_estimates_bad = zeros(1, Nmax);
Bayes_error_good = zeros(1, Nmax);
Bayes_error_bad = zeros(1, Nmax);

figure('Renderer', 'painters', 'Position', [100 100 900 600]);
figure(1);
i = 1;
% iterate from Nmin to Nmax observations
for N = Nmin:Nmax
    % take Bernoulli random variables of only up to size N
    bernoulli_rv = bernoulli_rvs(1, 1:N);
```

```
    % define m = # of observations of x = 1 for N observations
    %        l = # of observations of x = 0
    m = sum(bernoulli_rv,2);
    l = N - m;

    % compute the (conjugate) prior density
    % beta_conj_prior: anonymous function
    good_prior = beta_conj_prior(good_a, good_b, mu);
    bad_prior = beta_conj_prior(bad_a, bad_b, mu);

    % compute the likelihood density
    % binomial_likelihood: anonymous function
    likelihood = binomial_likelihood(mu, m, l);

    % compute the posterior density
    % beta_conj_prior_update: anonymous function
    good_posterior = beta_conj_prior_update(m, l, good_a, good_b, mu);
    bad_posterior = beta_conj_prior_update(m, l, bad_a, bad_b, mu);

    % normalize the distributions to [0 1] range
    good_prior = scale_pdf(good_prior);
    bad_prior = scale_pdf(bad_prior);
    likelihood = scale_pdf(likelihood);
    good_posterior = scale_pdf(good_posterior);
    bad_posterior = scale_pdf(bad_posterior);

    % compute max likelihood and bayes estimates
    % Bayes_mean: anonymous function
    max_likelihood(1,N) = mean(bernoulli_rv,2);     % = sample mean
    Bayes_estimates_good(1,N) = Bayes_mean(m, l, good_a, good_b);
    Bayes_estimates_bad(1,N) = Bayes_mean(m, l, bad_a, bad_b);

    % plots of PDF for ML and Bayesian Estimates
    %   - posteriors, priors, and true distribution densities
    %     at 4 points: N = Nmin, 1/3 of way, 2/3 of way, Nmax
    if N == 1 || N == round(Nmax/3) || N == round(Nmax/3*2) || N == Nmax
        figure(1);          % specify which figure to plot onto
        subplot(2,2,i);
        plot(mu, good_posterior, mu, bad_posterior, mu, likelihood, ...
            mu, true_pdf, mu, good_prior, mu, bad_prior);
        legend('posterior (good)', 'posterior (bad)', 'likelihood', ...
            'true pdf','prior (good)','prior (bad)', ...
            'Location', 'NorthEast');
        xlabel('Prob.{\it p} of Bernoulli Distribution');
        ylabel('Prob. Density (scaled)'); ylim([0 0.15]);
        title(sprintf('N = %d',N)); grid on;
        i = i + 1;
    end

    % compute the Mean Squared Error (MSE) for ML, Bayes estimators
    mse_max_likelihood(1,N) = mean(true_mean - max_likelihood(1,1:N)).^2;
    Bayes_error_good(1,N) = mean((true_mean - Bayes_estimates_good(1,1:N)).^2,2);
    Bayes_error_bad(1,N) = mean((true_mean - Bayes_estimates_bad(1,1:N)).^2,2);
end
% set title for figure 1
figure(1);
t = suptitle( ...
    sprintf('PDF Comparison between Likelihood and Bayesian Estimates with Beta distributions (Conjugate Prior) \n with hyper-parameters %s
/ %s', ...
    sprintf('a = %.1f, b = %.1f (good)', good_a, good_b), ...
    sprintf('a = %.1f, b = %.1f (bad)', bad_a, bad_b)));
set(t, 'FontSize', 10, 'Position', get(t,'Position')-[0 0.01 0], ...
    'FontWeight', 'normal');

% plot mean squared errors
figure(2);
plot(Nmin:Nmax, smooth(mse_max_likelihood), ...
                     Nmin:Nmax, smooth(Bayes_error_good), ...
                     Nmin:Nmax, smooth(Bayes_error_bad));
title('Mean Squared Error (MSE) of ML and Bayesian Estimates');
xlabel('Observations, N'); ylabel('Mean Squared Error');
legend('Location','Northeast', ...
        'ML Estimate', ...
        sprintf('Bayesian, Beta(a=%.1f, b=%.1f)', good_a, good_b), ...
        sprintf('Bayesian, Beta(a=%.1f, b=%.1f)', bad_a, bad_b));
```





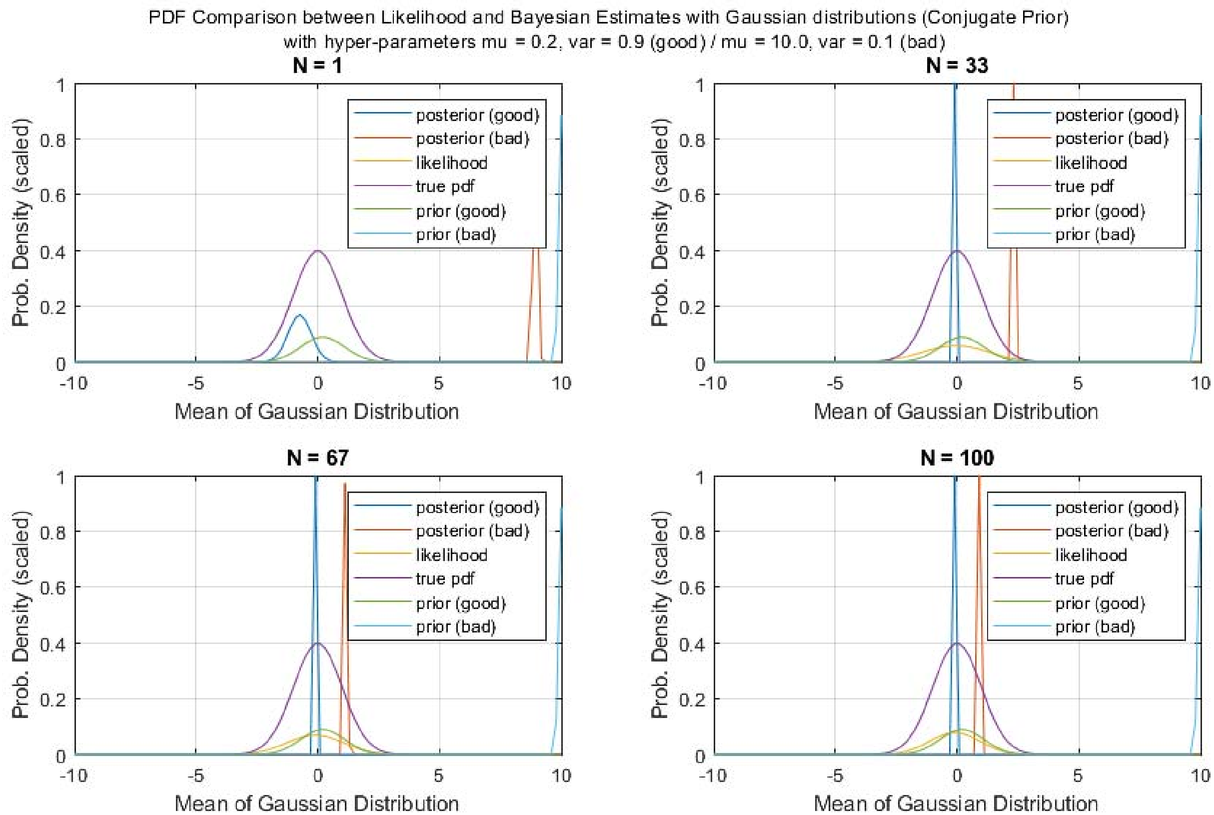
Simulation 2

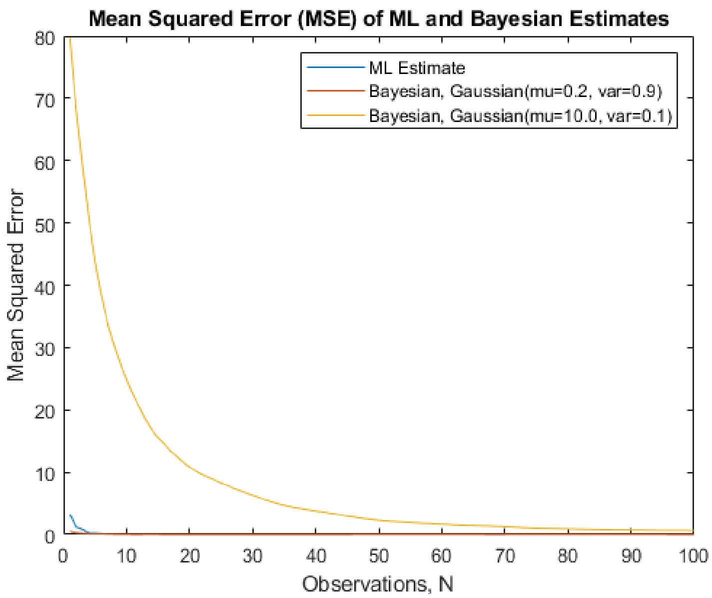
Gaussian Random Variables

```
% - Comparison between Maximum Likelihood (ML) estimate and estimation
% using conjugate priors (Gaussian distribution),
% with one "good" and one "bad" sets of parameters for the conjugate prior
% - Assume unknown mean (mu), known variance (sigma^2)

% ---- Relevant Equations ----
% (Other utility) Scaling to [0 1]
scale_pdf = @(x) x./sum(x);
% ---- end Relevant Equations ----

% Define the true mean and pdf, and the number of observations
% since mu is the parameter we want to predict, we aim to see
% the distributions over [-10 10] range and determine its most likely value
```

```matlab
true_mean = 0;          % initial guess to unknown mean
true_variance = 1;      % known variance
Nmin = 1; Nmax = 100;   % run up to 100 observations

mu = linspace(-10, 10, Nmax); % sweep through [-10 10]
norm_rvs = normrnd(true_mean, true_variance, [1, Nmax]);% generate r.v.'s
true_pdf = normpdf(mu, true_mean, true_variance);       % generate true pdf

% Hyper-parameters for Gaussian distributions
good_mu = 0.2; good_var = 0.9;
bad_mu = 10; bad_var = 0.1;

% Initialization for memory space
mse_max_likelihood = zeros(1, Nmax);
Bayes_error_good = zeros(1, Nmax);
Bayes_error_bad = zeros(1, Nmax);

figure('Renderer', 'painters', 'Position', [100 100 1000 600]);
figure(3);
i = 1;
% iterate from Nmin to Nmax observations
for N = Nmin:Nmax
    % take Gaussian random variables of size N
    norm_rv = norm_rvs(1, 1:N);

    % compute max-likelihood mean (Equation 2.121 or 2.143 of [1])
    % mu_{ML} = 1/N * sum(x_n) [from n=1 to N]
    ml_mean = mean(norm_rv,2);
    % compute variance (Equation 2.122 of [1])
    % var_{ML} = 1/N * sum((x_n - ml_mean)*(x_n - ml_mean)^T)[from n=1 to N]
    % this is equal to (N-1)/N * var(norm_rv), using built-in 'var()'
    ml_variance = mean((norm_rv - ml_mean).^2,2);

    % compute the (conjugate) prior density
    good_prior = normpdf(mu, good_mu, good_var);
    bad_prior = normpdf(mu, bad_mu, bad_var);

    % compute the likelihood density
    likelihood = normpdf(mu, ml_mean, ml_variance);

    % compute the posterior density (Equation 2.141, 2.142 of [1])
    % mu_N = [true_mean / (N*prior_var + true_var)] * prior_mu + ...
    %           [(N*prior_var)/(N*prior_var + true_var)] * ml_mu
    % 1/var_N = 1/prior_var + N/true_var
    % "good" set of mu, var
    good_mu_N = (true_variance/(N*good_var+true_variance))* good_mu ...
                + (N*good_var)/(N*good_var+true_variance)*ml_mean;
    good_var_N = ((1/good_var) + (N/true_variance))^(-1);
    good_posterior = normpdf(mu, good_mu_N, good_var_N);
    % "bad" set of mu, var
    bad_mu_N = (true_variance/(N*bad_var+true_variance))* bad_mu ...
                + (N*bad_var)/(N*bad_var+true_variance)*ml_mean;
    bad_var_N = ((1/bad_var) + (N/true_variance))^(-1);
    bad_posterior = normpdf(mu, bad_mu_N, bad_var_N);

    % normalize the distributions to [0 1] range
    good_prior = scale_pdf(good_prior);
    bad_prior = scale_pdf(bad_prior);
    likelihood = scale_pdf(likelihood);
    good_posterior = scale_pdf(good_posterior);
    bad_posterior = scale_pdf(bad_posterior);

    % plots of PDF for observations (N) for ML and Bayesian Estimates
    %   - posteriors, priors, and true distribution densities
    %   at 4 points: N = Nmin, 1/3 of way, 2/3 of way, Nmax
    if N == 1 || N == round(Nmax/3) || N == round(Nmax/3*2) || N == Nmax
        figure(3);
        subplot(2,2,i);
        plot(mu, good_posterior, mu, bad_posterior, mu, likelihood, ...
            mu, true_pdf, mu, good_prior, mu, bad_prior);
        title(sprintf('N = %d',N)); grid on;
        legend('posterior (good)', 'posterior (bad)', 'likelihood', ...
            'true pdf','prior (good)','prior (bad)', 'Location', 'NorthEast');
        xlabel('Mean of Gaussian Distribution'); ylabel('Prob. Density (scaled)');
        i = i + 1;
    end

    % compute Mean Squared Error (MSE)
    mse_max_likelihood(1,N) = mean(true_mean - ml_mean).^2;
    Bayes_error_good(1,N) = mean((true_mean - good_mu_N).^2,2);
    Bayes_error_bad(1,N) = mean((true_mean - bad_mu_N).^2,2);
end
% set title to figure 3
figure(3);
```

```
t = suptitle(sprintf('PDF Comparison between Likelihood and Bayesian Estimates with Gaussian distributions (Conjugate Prior) \n with hyper-p
arameters %s / %s', ...
    sprintf('mu = %.1f, var = %.1f (good)', good_mu, good_var),sprintf('mu = %.1f, var = %.1f (bad)', bad_mu, bad_var)));
set(t, 'FontSize', 10, 'Position', get(t,'Position')-[0 0.01 0], 'FontWeight', 'normal');

% plot MSE's for ML and Bayesian estimators
figure(4);
plot(Nmin:Nmax, smooth(mse_max_likelihood), ...
                     Nmin:Nmax, smooth(Bayes_error_good), ...
                     Nmin:Nmax, smooth(Bayes_error_bad));
title('Mean Squared Error (MSE) of ML and Bayesian Estimates');
xlabel('Observations, N'); ylabel('Mean Squared Error');
legend('Location','Northeast', ...
        'ML Estimate', ...
        sprintf('Bayesian, Gaussian(mu=%.1f, var=%.1f)', good_mu, good_var), ...
        sprintf('Bayesian, Gaussian(mu=%.1f, var=%.1f)', bad_mu, bad_var) ...
      );
```





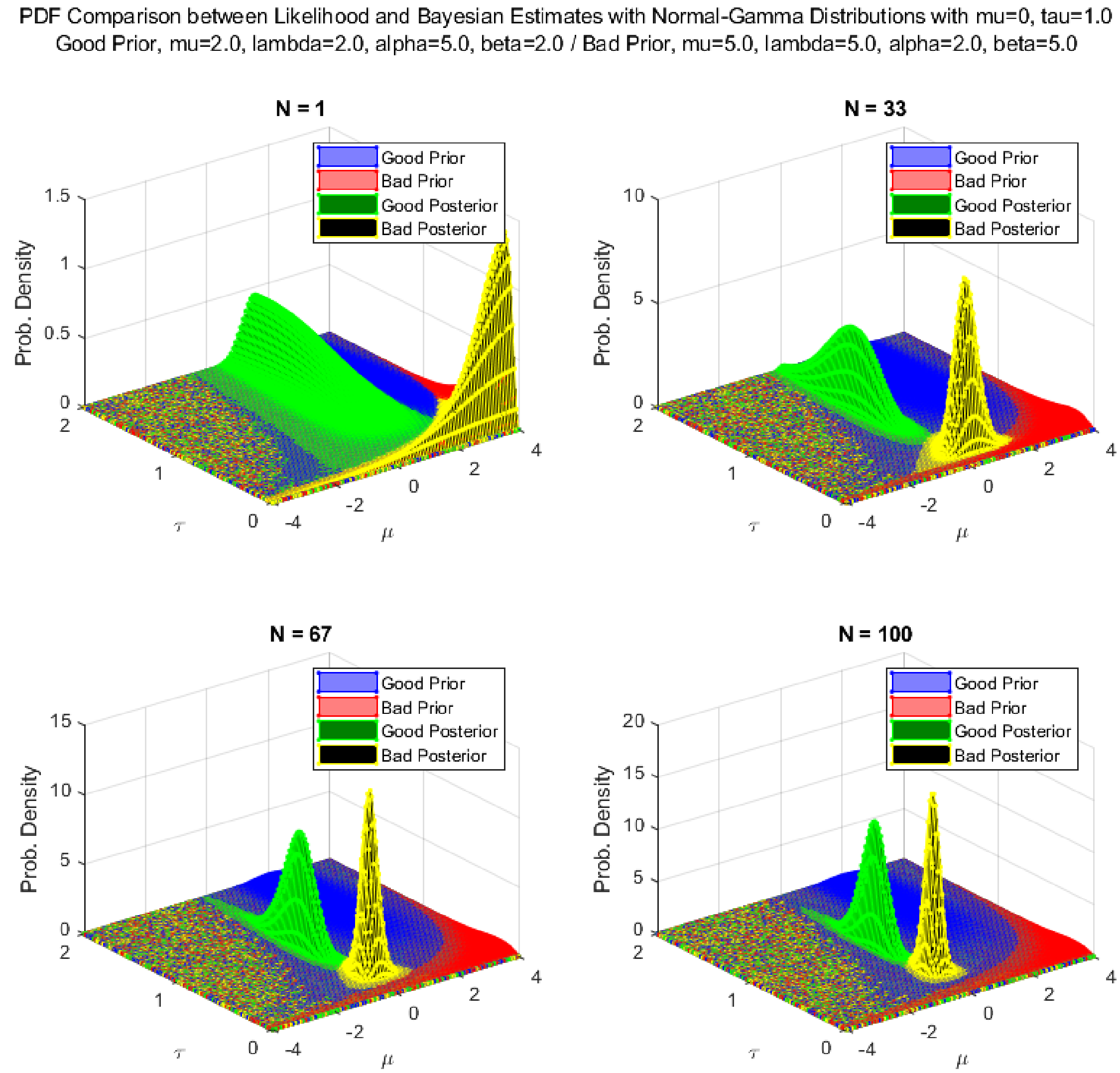
Simulation 3

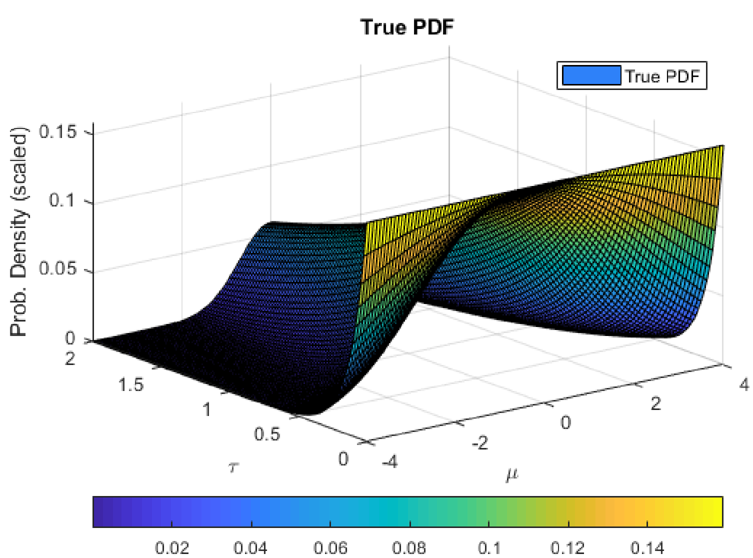
Gaussian Random Variables

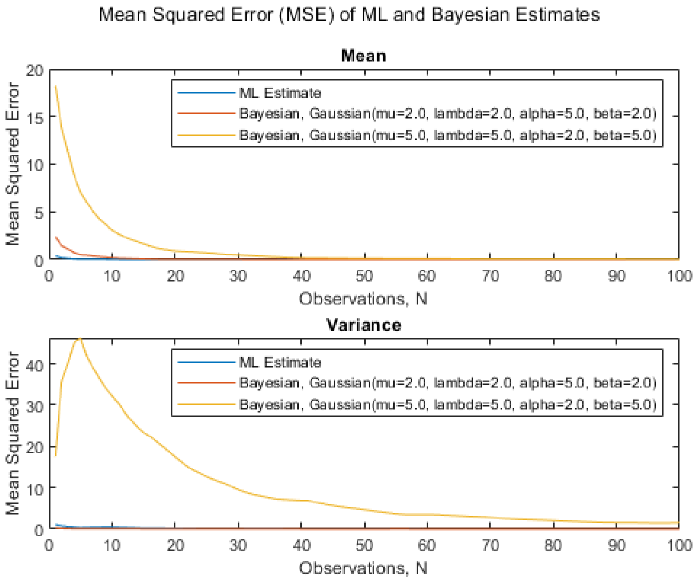
```
% - Comparison between Maximum Likelihood (ML) estimate and estimation
% using conjugate priors (Gaussian distribution), with one "good" and one "bad"
% set of parameters for the conjugate prior
% - Assume unknown mean, unknown variance

% approach: place a prior on mean and variance that is jointly conjugate.
% mean: Gaussian, variance: inverse gamma distribution

% ---- Relevant Equations ----
% (1) Conjugate Prior
% Equation for PDF for Normal-Gamma distribution
normal_gamma_pdf = @(x, tau, mu, lambda, alpha, beta) ...
        abs(((beta.^alpha).*sqrt(lambda))./(gamma(alpha)*sqrt(2*pi)) ...
        .* tau.^(alpha-1/2) .* exp(-beta.*tau) ...
        .* exp(-(lambda.*tau.*(x-mu).^2)/2));

% (2) Adjust hyperparameters for the posterior update
% Equations 86-89 of [2]
adjust_mean = @(lambda, mu, N, x) (lambda*mu + N*mean(x,2))/(lambda+N);
adjust_lambda = @(lambda, N) lambda + N;
adjust_alpha = @(alpha, N) alpha + N/2;
adjust_beta = @(beta, lambda, mu, N, x) ...
  beta + 1/2*sum((x-mean(x,2)).^2) + (lambda*N*(mean(x,2) - mu).^2)/(2*(lambda+N));

% (Other utility) Scaling to [0 1]
scale_pdf = @(x) x./sum(x);
% ---- end Relevant Equations ----

% Define the true mean and pdf, and the number of observations
true_mu = 0; % initial guess to unknown mean
true_lambda = 1; % initial guess to unknown variance
true_tau = 1/true_lambda; % set variance = 1
true_alpha = 0.5;
true_beta = 0.5;
Nmin = 1; Nmax = 100;

% range of mean and precision (= 1/var)
[mu, tau] = meshgrid(linspace(-4, 4, Nmax), linspace(0, 2, Nmax));

% generate pdf for Normal-Gamma distribution
% normal_gamma_pdf: anonymous function
true_pdf = normal_gamma_pdf(mu, tau, ...
        true_mu, true_lambda, true_alpha, true_beta);

% generate joint random variables (Normal-Gamma)
norm_rvs = normrnd(true_mu, 1/(true_lambda*true_tau), [1, Nmax]);% generate r.v.'s

% Hyper-parameters for Normal-Gamma distribution
good_mu = 2; good_lambda = 2; good_alpha = 5; good_beta = 2;
bad_mu = 5; bad_lambda = 5;  bad_alpha = 2; bad_beta = 5;

% Initialization for memory space
```

```
% (1) Max Likelihood
ml_mean = zeros(1, Nmax);
ml_precision = zeros(1, Nmax);
mse_max_likelihood_mean = zeros(1, Nmax);
mse_max_likelihood_var = zeros(1, Nmax);
% (2) Bayesian - good hyperparameter choices
Bayes_error_good_mean = zeros(1, Nmax);
Bayes_error_good_var = zeros(1, Nmax);
% (2) Bayesian - bad hyperparameter choices
Bayes_error_bad_mean = zeros(1, Nmax);
Bayes_error_bad_var = zeros(1, Nmax);

figure('Renderer', 'painters', 'Position', [50 50 900 800]);
figure(5);
i = 1;
for N = Nmin:Nmax
    % take Gaussian random variables of size N
    norm_rv = norm_rvs(1, 1:N);

    % compute max-likelihood mean (Equation 2.121 or 2.143 of [1])
    % mu_{ML} = 1/N * sum(x_n) [from n=1 to N]
    ml_mean = mean(norm_rv,2);
    % compute variance (Equation 2.124 of [1])
    % var_{ML} = (N-1)/N * (variance)
    ml_variance = mean((norm_rv - ml_mean).^2,2);

    % compute the (conjugate) prior density
    good_prior = normal_gamma_pdf(mu, tau, ...
            good_mu, good_lambda, good_alpha, good_beta);
    bad_prior = normal_gamma_pdf(mu, tau, ...
            bad_mu, bad_lambda, bad_alpha, bad_beta);

    % compute the posterior density
    good_mu_N = adjust_mean(good_lambda, good_mu, N, norm_rv);
    good_lambda_N = adjust_lambda(good_lambda, N);
    good_alpha_N = adjust_alpha(good_alpha, N);
    good_beta_N = adjust_beta(good_beta, good_lambda, good_mu, N, norm_rv);
    good_posterior = normal_gamma_pdf(mu, tau, ...
                      good_mu_N, good_lambda_N, good_alpha_N, good_beta_N);

    bad_mu_N = adjust_mean(bad_lambda, bad_mu, N, norm_rv);
    bad_lambda_N = adjust_lambda(bad_lambda, N);
    bad_alpha_N = adjust_alpha(bad_alpha, N);
    bad_beta_N = adjust_beta(bad_beta, bad_lambda, bad_mu, N, norm_rv);
    bad_posterior = normal_gamma_pdf(mu, tau, ...
                      bad_mu_N, bad_lambda_N, bad_alpha_N, bad_beta_N);

    % plots of PDF for observations (N) for ML and Bayesian Estimates
    %   at 4 points: N = Nmin, 1/3 of way, 2/3 of way, Nmax
    if N == 1 || N == round(Nmax/3) || N == round(Nmax/3*2) || N == Nmax
        figure(5);
        subplot(2,2,i);
        surf(mu, tau, good_prior,'EdgeColor','blue',...
                'FaceColor',[0,0,255]/255,'FaceAlpha',0.5,'Marker','.');
        hold on;
        surf(mu, tau, bad_prior,'EdgeColor','red',...
                'FaceColor',[255,0,0]/255,'FaceAlpha',0.5,'Marker','.');
        hold on;
        surf(mu, tau, good_posterior,'EdgeColor','green',...
                'FaceColor',[0,128,0]/255,'FaceAlpha',1,'Marker','.');
        hold on;
        surf(mu, tau, bad_posterior,'EdgeColor','yellow',...
                'FaceColor',[0,0,0]/255,'FaceAlpha',1,'Marker','.');
        title(sprintf('N = %d', N));
        xlabel('{\mu}'); ylabel('{\tau}'); zlabel('Prob. Density');
        legend('Good Prior', 'Bad Prior','Good Posterior','Bad Posterior','Location','Northeast');
        hold off; grid on;
        i = i + 1;
    end

    % compute Mean Squared Error (MSE)
    % - compare mean, variance (inverse of found precision) to the initially chosen 'true' values
    mse_max_likelihood_mean(1,N) = mean((true_mu - ml_mean).^2);
    mse_max_likelihood_var(1,N) = mean((1/(true_lambda*true_tau) - ml_variance).^2);
    Bayes_error_good_mean(1,N) = mean((true_mu - good_mu_N).^2);
    Bayes_error_good_var(1,N) = mean((1/(true_lambda*true_tau) - 1./(good_alpha_N/good_beta_N)).^2);
    Bayes_error_bad_mean(1,N) = mean((true_mu - bad_mu_N).^2);
    Bayes_error_bad_var(1,N) = mean((1/(true_lambda*true_tau) - 1./(bad_alpha_N/bad_beta_N)).^2);
end
% set title for PDF plots
figure(5);
t = suptitle( strcat({...
            sprintf('PDF Comparison between Likelihood and Bayesian Estimates with Normal-Gamma Distributions with mu=%1.f, tau=%.1f',true_m
u, true_tau), ...
```

```
            sprintf('Good Prior, mu=%.1f, lambda=%.1f, alpha=%.1f, beta=%.1f / Bad Prior, mu=%.1f, lambda=%.1f, alpha=%.1f, beta=%.1f\n', ..
.
            good_mu, good_lambda, good_alpha, good_beta,bad_mu, bad_lambda, bad_alpha, bad_beta)
            }));
set(t, 'FontSize', 11,'Position', get(t,'Position')+[0 0.01 0], 'FontWeight', 'normal');

% plot the true PDF
figure(6);
surf(mu, tau, true_pdf); colorbar('southoutside'); grid on;
title('True PDF'); legend('True PDF','Location','Northeast');
xlabel('{\mu}'); ylabel('{\tau}'); zlabel('Prob. Density (scaled)');

% plot mean squared errors for Max-likelihood and Bayesian estimators
figure(7);
subplot(211);
plot(Nmin:Nmax, smooth(mse_max_likelihood_mean), ...
                     Nmin:Nmax, smooth(Bayes_error_good_mean), ...
                     Nmin:Nmax, smooth(Bayes_error_bad_mean));
xlabel('Observations, N'); ylabel('Mean Squared Error');
title('Mean');
legend('Location','Northeast', ...
        'ML Estimate', ...
        sprintf('Bayesian, Gaussian(mu=%.1f, lambda=%.1f, alpha=%.1f, beta=%.1f)', good_mu, good_lambda, good_alpha, good_beta), ...
        sprintf('Bayesian, Gaussian(mu=%.1f, lambda=%.1f, alpha=%.1f, beta=%.1f)', bad_mu, bad_lambda, bad_alpha, bad_beta) ...
      );
subplot(212);
plot(Nmin:Nmax, smooth(mse_max_likelihood_var), ...
                     Nmin:Nmax, smooth(Bayes_error_good_var), ...
                     Nmin:Nmax, smooth(Bayes_error_bad_var));
xlabel('Observations, N'); ylabel('Mean Squared Error');
title('Variance');
legend('Location','Northeast', ...
        'ML Estimate', ...
        sprintf('Bayesian, Gaussian(mu=%.1f, lambda=%.1f, alpha=%.1f, beta=%.1f)', good_mu, good_lambda, good_alpha, good_beta), ...
        sprintf('Bayesian, Gaussian(mu=%.1f, lambda=%.1f, alpha=%.1f, beta=%.1f)', bad_mu, bad_lambda, bad_alpha, bad_beta) ...
      );
t = suptitle('Mean Squared Error (MSE) of ML and Bayesian Estimates');
set(t, 'FontSize', 10, 'FontWeight', 'normal');
```


PDF Comparison between Likelihood and Bayesian Estimates with Normal-Gamma Distributions with mu=0, tau=1.0
Good Prior, mu=2.0, lambda=2.0, alpha=5.0, beta=2.0 / Bad Prior, mu=5.0, lambda=5.0, alpha=2.0, beta=5.0
N = 1
N = 33
N = 67
N = 100
Good Prior
Bad Prior
Good Posterior
Bad Posterior
Prob. Density
τ
μ


True PDF
True PDF
Prob. Density (scaled)
τ
μ



Published with MATLAB® R2018a