Contents


```
% ECE414 - Bayesian Machine Learning
% Authors   : Junbum Kim, Andy Jeong
% Project 2 : Bayesian Linear Regression Models
% Date      : October 16, 2019
% Reference : Pattern Recognition and Machine Learning
%                           by Chris. M. Bishop (2006)
% Assumption: all drawn random variables are i.i.d
clear all; close all; clc; warning('off','all');
```

Equations

1 Figure 3.7

$$\text{Posterior: } p(w|\mathbf{t}) = N(w|m_N, S_N) \tag{1}$$

$$m_N = S_N(S_0^{-1}m_0 + \beta\Phi^T t) \tag{2}$$

$$S_N^{-1} = S_0^{-1} + \beta\phi^T\phi \tag{3}$$

2 Figure 3.8

$$\text{Posterior: } p(w|\alpha) = N(w|0, \alpha^{-1}I) \tag{4}$$

$$m_N = \beta S_N \phi^T t \tag{5}$$

$$S_N^{-1} = \alpha I + \beta\phi^T\phi \tag{6}$$

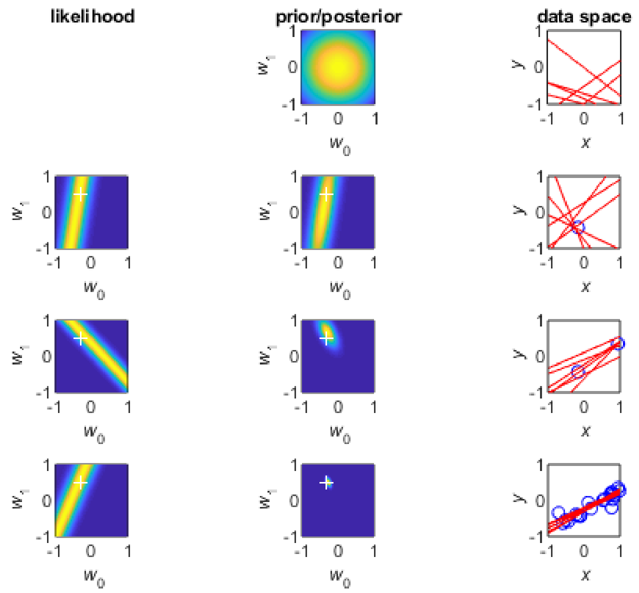
Visualization 1

Figure 3.7: Bayesian Linear Regression Kernel (Basis): Linear

```
% number of observations
N = 100;

% parameters {a0, a1} for generating synthetic data
% -- equation: f(x, a0, a1) = a0 + a1 * x
a0 = -0.3; a1 = 0.5;

% noise parameters (assume variance is known)
noise_mean = 0; noise_std = 0.2;

% generate random variables
x = rand(N,1)*2-1;      % uniform disribution U(-1, 1)
noisy_obs = @(x) a0 + a1*x + normrnd(noise_mean, noise_std,[length(x),1]);
t = noisy_obs(x);       % generate noisy observations

% hyperparameters
alpha = 2.0;
beta = (1/noise_std)^2; % precision

% prior distribution
m0 = [0; 0];
s0 = alpha * eye(2);    % 2 weights

% create gridspace for creating multivariate guassian distribution and for plotting
sampling_rate = 1001;
[w0, w1] = meshgrid(linspace(-1, 1, sampling_rate));
grid_space = [w0(:) w1(:)];
X = [ones(length(x),1) x]; % form: y = x0 + x1 * x

covars = zeros(2,2,N);
means = zeros(2,N);
w_ml = zeros(2,1,N);

for i = 1:N
    T = t(1:i);
```

```
    PHI = X(1:i,:); % for linear model, take PHI = X
    % true parameters w
    w_ml(:,:,i) = inv(PHI' * PHI) * PHI' * T;
    % update posterior distributions
    covars(:,:,i) = inv(inv(s0) + beta * PHI' * PHI);
    means(:,i) = covars(:,:,i)*(inv(s0) * m0 + beta * PHI' * T);
end

% plot likelihood, prior/posterior distribution, data space in order by rows
% pcolor: adjust color weights at each point (w0,w1) by density
figure(1);
subplot(4,3,1); title('likelihood'); axis off; pbaspect([1 1 1]);

subplot(4,3,2);
prior = mvnpdf(grid_space, m0', s0);
prior_grid = reshape(prior, [sampling_rate, sampling_rate]);
pcolor(w0, w1, prior_grid); title('prior/posterior');
shading interp; pbaspect([1 1 1])
xlabel('{\it w}_0'); ylabel('{\it w}_1');

subplot(4,3,3);
for i=1:6
    w_samples_idx = randperm(length(prior(:,1)),2);
    Y = [grid_space(w_samples_idx(1),1), grid_space(w_samples_idx(2),2)] * X';
    plot(X(:,2), Y,'r'); hold on;
end
axis([-1,1,-1,1]); pbaspect([1 1 1]); title('data space');
xlabel('{\it x}'); ylabel('{\it y}');

% N = 1 (observation)
N = 1;
subplot(4,3,4);
y = grid_space * X(N,:)';
t_tmp = t(N);
likelihood = reshape(mvnpdf(t_tmp, y, 1/beta), [sampling_rate, sampling_rate]);
pcolor(w0,w1,likelihood); hold on;
scatter(w_ml(1,:,end),w_ml(2,:,end),'w+');
shading interp; pbaspect([1 1 1]); axis([-1,1,-1,1]);
xlabel('{\it w}_0'); ylabel('{\it w}_1');

subplot(4,3,5);
posterior = mvnpdf(grid_space, means(:,N)', covars(:,:,N));
posterior_grid = reshape(posterior, [sampling_rate, sampling_rate]);
pcolor(w0, w1, posterior_grid); hold on;
scatter(w_ml(1,:,end),w_ml(2,:,end),'w+');
shading interp; pbaspect([1 1 1]); axis([-1,1,-1,1]);
xlabel('{\it w}_0'); ylabel('{\it w}_1');

subplot(4,3,6);
plot(x(1:N),t(1:N),'bo'); hold on;
rand_samples = mvnrnd(means(:,N)', covars(:,:,N), 6);
Y = rand_samples * X';
plot(X(:,2), Y,'r'); hold on;
axis([-1,1,-1,1]); pbaspect([1 1 1]);
xlabel('{\it x}'); ylabel('{\it y}');

% N = 2 (observations)
N = 2;
subplot(4,3,7);
y = grid_space * X(N,:)';
t_tmp = t(N);
likelihood = reshape(mvnpdf(t_tmp, y, 1/beta), [sampling_rate, sampling_rate]);
pcolor(w0,w1,likelihood); hold on;
scatter(w_ml(1,:,end),w_ml(2,:,end),'w+');
shading interp; pbaspect([1 1 1]); axis([-1,1,-1,1]);
xlabel('{\it w}_0'); ylabel('{\it w}_1');

subplot(4,3,8);
posterior = mvnpdf(grid_space ,means(:,N)', covars(:,:,N));
posterior_grid = reshape(posterior, [sampling_rate, sampling_rate]);
pcolor(w0,w1,posterior_grid); hold on;
scatter(w_ml(1,:,end),w_ml(2,:,end),'w+');
shading interp; pbaspect([1 1 1]); axis([-1,1,-1,1]);
xlabel('{\it w}_0'); ylabel('{\it w}_1');

subplot(4,3,9);
plot(x(1:N),t(1:N),'bo'); hold on;
rand_samples = mvnrnd(means(:,N)', covars(:,:,N), 6);
Y = rand_samples * X';
plot(X(:,2), Y,'r'); hold on;
axis([-1,1,-1,1]); pbaspect([1 1 1]);
xlabel('{\it x}'); ylabel('{\it y}');
```

```
% N = 20 (observations)
N = 20;
subplot(4,3,10);
y = grid_space * X(N,:)';
t_tmp = t(N);
likelihood = reshape(mvnpdf(t_tmp, y, 1/beta), [sampling_rate, sampling_rate]);
pcolor(w0,w1,likelihood); hold on;
scatter(w_ml(1,:,end),w_ml(2,:,end),'w+');
shading interp; pbaspect([1 1 1]); axis([-1,1,-1,1]);
xlabel('{\it w}_0'); ylabel('{\it w}_1');

subplot(4,3,11);
posterior = mvnpdf(grid_space ,means(:,N)', covars(:,:,N));
posterior_grid = reshape(posterior, [sampling_rate, sampling_rate]);
pcolor(w0,w1,posterior_grid); hold on;
scatter(w_ml(1,:,end),w_ml(2,:,end),'w+');
shading interp; pbaspect([1 1 1]); axis([-1,1,-1,1]);
xlabel('{\it w}_0'); ylabel('{\it w}_1');

subplot(4,3,12);
plot(x(1:N),t(1:N),'bo'); hold on;
axis([-1,1,-1,1]); pbaspect([1 1 1])
rand_samples = mvnrnd(means(:,N)', covars(:,:,N), 6);
Y = rand_samples * X';
plot(X(:,2), Y,'r'); hold on;
axis([-1,1,-1,1]); pbaspect([1 1 1]);
xlabel('{\it x}'); ylabel('{\it y}');
```

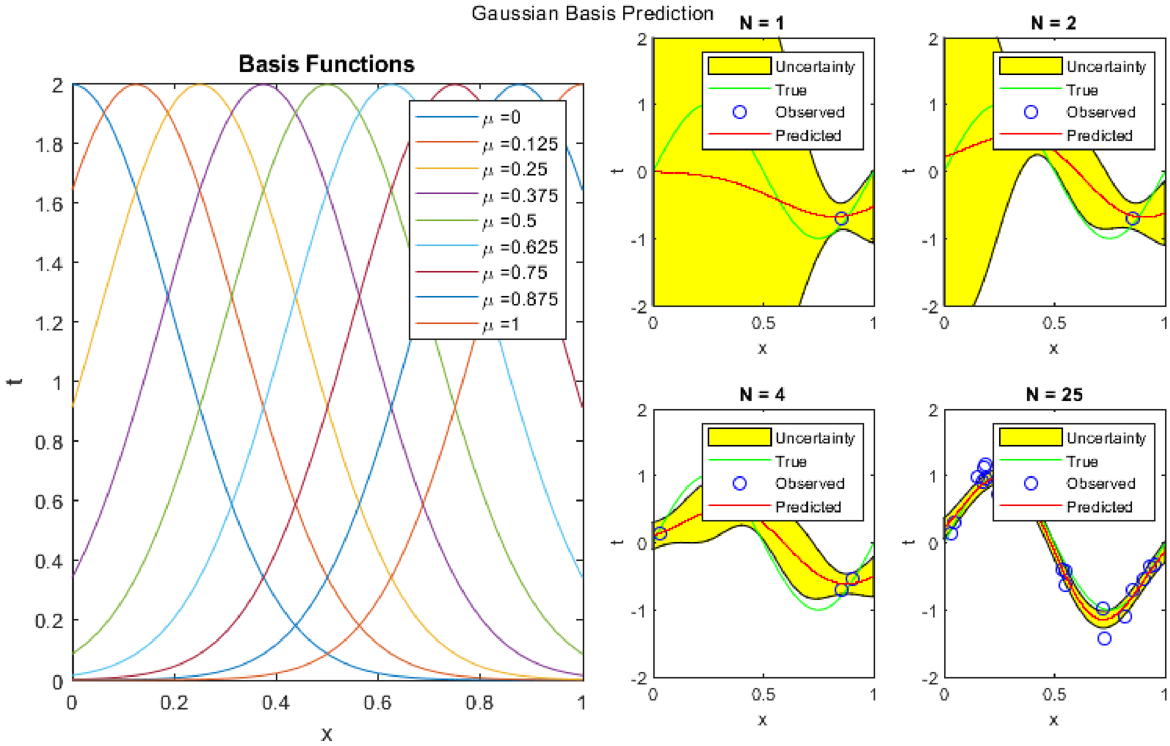


Visualization 2

Figure 3.8: Bayesian Linear Regression (predictive distribution) Kernel (Basis): Gaussian

```
N = 25;                              % number of observations
N_basis = 9;                         % number of basis functions
noise_mean = 0;                      % known noise parameter
noise_std = 0.2;
basis_mean = linspace(0,1,N_basis); % basis parameters
basis_std = 0.2;
alpha = 3; beta = 10;               % hyperparameters
sampling_rate = 1000;

X = linspace(0, 1, sampling_rate);  % true sinusoidal
y = sin(2 * pi * X);

% plot Gaussian basis functions
basis = normpdf(X,basis_mean',basis_std);
figure('Renderer', 'painters', 'Position', [100 100 900 500]);
figure(2); subplot(2,4,[1 6]);
t = suptitle('Gaussian Basis Prediction');
set(t, 'FontSize', 10, 'Position', get(t,'Position')+[0 0.01 0], ...
    'FontWeight', 'normal');
plot(X, basis); title('Basis Functions'); xlabel('x'); ylabel('t');
legend(strcat('{\mu} = ',string(num2cell(basis_mean))));
```

```
% drawn random variables and targets
x = rand(1,N);                   % [1xN] dimensional uniform RV (~U(0,1))
t = sin(2 * pi * x) + normrnd(0, noise_std, [1, N]);

% find distribution fitting the observation(s)
x_distributions = normpdf(x', basis_mean, basis_std);
predictions = zeros(N, sampling_rate);
variances = zeros(length(X),N);

for i = 1:N
    PHI = x_distributions(1:i,:);
    Sn = inv(alpha * eye(N_basis) + beta * (PHI)' * PHI);
    mn = beta * Sn * PHI' * t(1:i)';
    predictions(i,:) = mn' * basis;
    for j=1:length(X)
        phi = normpdf(X(j), basis_mean', basis_std);
        variances(j,i) = 1/beta + phi' * Sn * phi;
    end
end

% N = 1 (observation)
subplot(2,4,3);
N = 1;
std_dev = variances(:,N)';
upper = predictions(N,:) + std_dev;
lower = predictions(N,:) - std_dev;
X2 = [X, fliplr(X)];
inbetween = [upper, fliplr(lower)];
fill(X2, inbetween, 'y'); hold on;
xlabel('x'); ylabel('t');
plot(X, y, 'g', x(1:N), t(1:N), 'bo', X, predictions(N,:), 'r');
legend('Uncertainty','True', 'Observed', 'Predicted');
title('N = 1'); xlim([0 1]); ylim([-2 2]);

% N = 2 (observations)
subplot(2,4,4);
N = 2;
std_dev = variances(:,N)';
upper = predictions(N,:) + std_dev;
lower = predictions(N,:) - std_dev;
X2 = [X, fliplr(X)];
inbetween = [upper, fliplr(lower)];
fill(X2, inbetween, 'y'); hold on;
xlabel('x'); ylabel('t');
plot(X, y, 'g', x(1:N), t(1:N), 'bo', X, predictions(N,:), 'r');
legend('Uncertainty','True', 'Observed', 'Predicted');
title('N = 2'); xlim([0 1]); ylim([-2 2]);

% N = 4 (observations)
subplot(2,4,7);
N = 4;
std_dev = variances(:,N)';
upper = predictions(N,:) + std_dev;
lower = predictions(N,:) - std_dev;
X2 = [X, fliplr(X)];
inbetween = [upper, fliplr(lower)];
fill(X2, inbetween, 'y'); hold on;
xlabel('x'); ylabel('t');
plot(X, y, 'g', x(1:N), t(1:N), 'bo', X, predictions(N,:), 'r');
legend('Uncertainty','True', 'Observed', 'Predicted');
title('N = 4'); xlim([0 1]); ylim([-2 2]);

% N = 25 (observations)
subplot(2,4,8);
N = 25;
std_dev = variances(:,N)';
upper = predictions(N,:) + std_dev;
lower = predictions(N,:) - std_dev;
X2 = [X, fliplr(X)];
inbetween = [upper, fliplr(lower)];
fill(X2, inbetween, 'y'); hold on;
xlabel('x'); ylabel('t');
plot(X, y, 'g', x(1:N), t(1:N), 'bo', X, predictions(N,:), 'r');
legend('Uncertainty','True', 'Observed', 'Predicted');
title('N = 25'); xlim([0 1]); ylim([-2 2]);
```



Published with MATLAB® R2018a