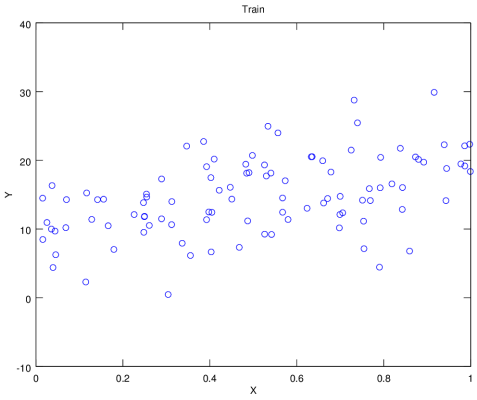
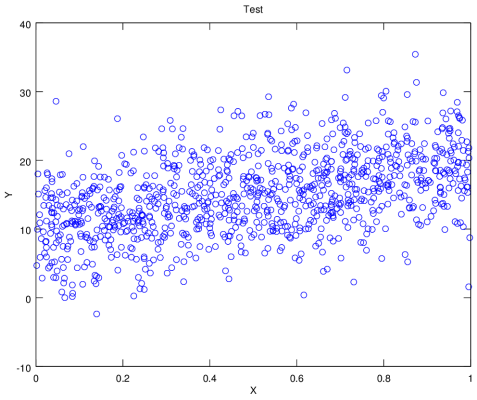
CS691GM: Graphical Models - Spring 2013

Assignment 5

Introduction: In this assignment, you will experiment with Bayesian linear regression. Linear regression is one of the most common models in machine learning and statistics. We will compare different methods for parameter estimation, inference and prediction.

Data Set and Task: To enable visualization of the posterior distribution over the model parameters, we will use a simple one dimensional linear regression data set. Training and test data files are included with the assignment package. The training data files are *Xtrain.txt* and *Ytrain.txt*. The test files are *Xtest.txt* and *Ytest.txt*. The two data sets are visualized below.





Model: A standard linear regression model can represent a linear trend $y = z\beta_1 + \beta_0$. β_1 represents the slope of the linear relationship while β_0 represents the offset. z is the independent variable and y is the dependent variable. The parameters of the model are $\beta = [\beta_0, \beta_1]^T$. We can express the linear regression equation more compactly by letting $\mathbf{x} = [1, z]$ and then writing $y = \mathbf{x}\beta$ ($\mathbf{x}$ is a row vector and β is a column vector). If the linear relationship is subject to random additive noise, we can express the relationship between $\mathbf{x}$ and y using a conditional probability density as shown below. We use a normal distribution for y where the mean is $\mathbf{x}\beta$ and the variance σ^2.

$$P(y|\mathbf{x}, \beta) = \mathcal{N}(y|\mathbf{x}\beta, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{1}{2\sigma^2}(y - \mathbf{x}\beta)^2) \tag{1}$$

Under the Bayesian inference framework, the model parameters β are viewed as random variables. We give them an independent zero-mean prior distribution with variance λ. We write $\Lambda = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}$. We can then express the prior distribution over the parameters as a multi-variate normal as shown below. Note that $|2\pi\Lambda|$ indicates the determinant of the matrix $2\pi\Lambda$.

$$P(\beta|\Lambda) = \mathcal{N}(\beta|0, \Lambda) = \frac{1}{|2\pi\Lambda|^{1/2}} \exp(-\frac{1}{2}\beta^T\Lambda^{-1}\beta) \tag{2}$$

Estimation and Inference: Suppose we have a training set $\mathcal{D} = \{(y_n, \mathbf{x}_n)\}_{1:N}$ consisting of N different observations $(y_n, \mathbf{x}_n)$. Each $\mathbf{x}_n$ is a row vector of length 2 and each y_n is a scalar. To simplify the notation, let X be a data matrix whose n^{th} row is $\mathbf{x}_n$ and Y be a column vector whose n^{th} entry is y_n. We'll assume for simplicity that the values of σ^2 and λ are both known and fixed. Our goal is to summarize the information about β given the data and the prior distribution.

The maximum likelihood estimate for β is obtained by ignoring the prior distribution on the parameters completely and selecting the value of β that makes the observed data the most likely. The maximum likelihood estimator is given below:

$$\beta_{ML} = (X^T X)^{-1} X^T Y \tag{3}$$

Instead of maximizing the likelihood of the data, we can maximize the posterior probability of the model parameters given the observed data and the prior. This is no more difficult than maximum likelihood estimation, but it also takes the prior distribution into account. Selecting the parameters that maximize the posterior is known as *maximum a posteriori* or MAP estimation. The MAP estimator is given below.

$$\beta_{MAP} = (X^T X + \sigma^2 \Lambda^{-1})^{-1} X^T Y \tag{4}$$

Bayesian inference is concerned not only with the maximum of the posterior distribution, but the full posterior distribution. The full posterior distribution for Bayesian linear regression is available in closed form. The posterior on β is multivariate normal as shown below.

$$\mu = (X^T X + \sigma^2 \Lambda^{-1})^{-1} X^T Y \tag{5}$$

$$\Sigma = (X^T X + \sigma^2 \Lambda^{-1})^{-1} \tag{6}$$

$$P(\beta|\mathcal{D}, \sigma^2, \Lambda) = \mathcal{N}(\beta|\mu, \Sigma) = \frac{1}{|2\pi\Sigma|^{1/2}} \exp(-\frac{1}{2}(\beta - \mu)^T \Sigma^{-1} (\beta - \mu)) \tag{7}$$

One issue with Bayesin linear regression in high dimensions is that the full covariance matrix has quadratic space complexity. To maintain some of the benefits of Bayesian inference while reducing both computational and space complexity, we can apply variational inference under the assumption that the approximate posterior covariance matrix is diagonal (equivalent to assuming the parameters are independent in the posterior). We obtain the following approximate posterior distribution.

$$m = (X^T X + \sigma^2 \Lambda^{-1})^{-1} X^T Y \tag{8}$$

$$S_{ij} = \begin{cases} (\sum_{n=1}^{N} x_{ni}^2 + \frac{\sigma^2}{\lambda})^{-1} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

$$Q(\beta|m, S) = \mathcal{N}(\beta|m, S) = \frac{1}{|2\pi S|^{1/2}} \exp(-\frac{1}{2}(\beta - m)^T S^{-1} (\beta - m)) \tag{10}$$

1. (*30 points*) **Derivations:** Perform the following derivations. Show your work.

(a) [5] Write down the log likelihood function for the linear regression model.

(b) [10] Derive the maximum likelihood estimate for the model parameters, β_{ML} as shown above.

(c) [5] Write down the log posterior function for the linear regression model.

(b) [10] Derive the maximum a posteriori estimate for the model parameters, β_{MAP} as shown above. Assume that σ^2 and Λ are known and fixed.

2. (*25 points*) **Estimation:** Use the supplied data to perform the following model estimation tasks. In all cases, use $\sigma = 5$ and $\lambda = 20$.

(a) [5] Compute β_{ML} using the first $2, 10, 100$ training cases. Produce a scatter plot showing the training data used to compute each of the four parameter sets. Overlay the estimated regression line $y = \mathbf{x}\beta_{ML}$ in each plot.

(b) [5] For each of the four parameter sets found in part (a), compute the mean squared error between the test cases and the predictions: $\frac{1}{N}\sum_{n=1}^{N}(y_n - \mathbf{x}_n\beta_{ML})^2$. Include a plot of the error versus the number of training cases used to estimate the model parameters.

(c) [5] Compute β_{MAP} using the first $2, 10, 100$ training cases. Produce a scatter plot showing the training data used to compute each of the four parameter sets. Overlay the estimated regression line $y = \mathbf{x}\beta_{MAP}$ in each plot.

(d) [5] For each of the four parameter sets found in part (c), compute the mean squared error between the test cases and the predictions: $\frac{1}{N}\sum_{n=1}^{N}(y_n - \mathbf{x}_n\beta_{MAP})^2$. Include a plot of the error versus the number of training cases used to estimate the model parameters.

(e) [5] How do the ML and MAP estimates differ? How does the difference depend on the the number of data cases available for estimation? Is ML estimation reasonable when the amount of data is low? Is MAP estimation always better than ML estimation in terms of error? Why or why not?

3. (*45 points*) **Posterior Inference:** Use the supplied data to perform the following posterior inference tasks. In all cases, use $\sigma = 5$ and $\lambda = 20$.

(a) [15] Use the first $2, 10, 100$ data cases to compute the true posterior mean μ and the posterior covariance matrix Σ. For each posterior mean and covariance, produce a contour plot of of the log posterior over the range $0 \leq \beta_0 \leq 20$, $0 \leq \beta_1 \leq 20$. Plot the location of the posterior mean in each plot (example matlab plotting code is included). Also draw 10 samples of β from each posterior distribution (use mvnrnd in matlab or numpy.random.multivariate_normal in python). Display the regression lines corresponding to each sample of β.

(b) [15] Use the first $2, 10, 100$ data cases to compute the variational posterior mean m and the variatinal posterior covariance matrix S. For each variational posterior mean and covariance, produce a contour plot of of the variational log posterior over the range $0 \leq \beta_0 \leq 20$, $0 \leq \beta_1 \leq 20$. Plot the location of the variational posterior mean in each plot. Also draw 10 samples of β from each variational posterior distribution (use mvnrnd in matlab or numpy.random.multivariate_normal in python). Display the regression lines corresponding to each sample of β.

(d) [8] Describe how the the true posterior and the variational approximation change as the amount of data increases. How do these changes relate to the variation in the collection of sampled regression lines?

(e) [7] Describe any differences between the log posterior and the variational log posterior. Are the differences apparent in the collection of sampled regression lines?