# Machine Learning (Spring 2018)
# Homework 1

## Due: 11:59pm April 19, 2018

Use Python to do the programming. Do not use Python packages for machine learning. Submit your source code and the results through e3. Explain what you did as detailed as possible.

1) **(Polynomial regression)**
   In Chapter 1 and Chapter 3 of the textbook, we have learned polynomial regression. In this exercise, you will implement polynomial regression.

   First, train the coefficients $w_0, \ldots, w_M$ of the polynomial function (Eq. (1.1) in the textbook)

   $$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j \tag{1}$$

   using the training data (the first 70% data points of the dataset **data_1.csv**) and the error function (Eq. (1.2) in the textbook)

   $$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w} - t_n\}^2, \tag{2}$$

   where $N$ is the number of data points in the dataset, $x_n$ is the $n$th data point of the first array in **data_1.csv**, and $t_n$ is the corresponding target value of $x_n$ (stored in the second array in **data_1.csv**). Write your own code. Do not directly use Python functions.

   Then, test the performance of your implementation using the test data (the remaining 30% data points of the dataset **data_1.csv**).

   a) Calculate and plot the figure of root-mean-square (RMS) error, defined by $E_{\text{RMS}} = \sqrt{2E(\mathbf{w}^*)/N}$ (Eq. (1.3) in the textbook), versus $M$, similar to the one shown in Figure 1.5 of the textbook. Now, with the same training and test dataset, find the coefficients $w_0, \ldots, w_M$ of the $M = 9$ polynomial function by considering the modified error function with regularization: (Eq. (1.4) in the textbook)

   $$\tilde{E}(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w} - t_n\}^2 + \frac{\lambda}{2} ||\mathbf{w}||^2. \tag{3}$$

   b) Try various regularization coefficient $\lambda$ and plot $E_{\text{RMS}}$ versus $\ln \lambda$. Adjust the range of $\ln \lambda$ to see whether you can get the result similar to the one shown in Figure 1.8 of the textbook.

2) **($D$-dimensional polynomial regression)**

   a) In Exercise 1, the data is 1-dimensional. In practical applications, we have to deal with $D$-dimensional data ($D > 1$). In this exercise, train the coefficients $w_0, w_i, w_{ij}, w_{ijk},\ i, j, k = 1, \ldots, D$, of the following polynomial function (Eq. (1.74) in the textbook)

   $$y(x, \mathbf{w}) = w_0 + \sum_{i=1}^{D} w_i x_i + \sum_{i=1}^{D} \sum_{j=1}^{D} w_{ij} x_i x_j + \sum_{i=1}^{D} \sum_{j=1}^{D} \sum_{k=1}^{D} w_{ijk} x_i x_j x_k \tag{4}$$

and then calculate the RMS error of the training dataset and the test dataset.

Use the Iris dataset. Note that the Iris dataset (**Iris_X.mat**) has 4-dimensional data, representing respectively the sepal length in cm, the sepal width in cm, the petal length in cm, and the petal width in cm. The file **Iris_T.mat** contains 1-dimensional target, representing three classes, including Iris Setosa (class 1), Iris Versicolour (class 2), and Iris Virginica (class 3), with 50 samples in each class. Use the first 40 samples of each class for training and the last 10 samples of each class for testing.

b) Repeat (a) using the Wine dataset https://archive.ics.uci.edu/ml/datasets/wine (read the description and figure out how to use the dataset by yourself).

3) **(Bayesian linear regression)**
In this exercise, you will implement Bayesian linear regression.

a) Calculate and plot 5 sample curves of the function $y(x, \mathbf{w}) = \mathbf{w}^T \phi(x)$ for data size $N = 1, 2, 4, 25, 50, 80, 100$, similar to Figure 3.9 of the textbook, using the dataset **data_3.csv**. To calculate $y(x, \mathbf{w}) = \mathbf{w}^T \phi(x)$, choose Gaussian basis functions $\phi = [\phi_0, \ldots, \phi_{M-1}]^T$ of the form $\phi_j(x) = \exp -\frac{(x-\mu_j)^2}{2s^2}$, $j = 0, \ldots, M-1$, with $\mu_j = \frac{j-(M-1)/2}{(M-1)/2}$, $M = 9$, and $s = 1$. The posterior distribution is given by $p(\mathbf{w}|\mathbf{t}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N)$, where $\mathbf{m}_N$ is the mean vector and $\mathbf{S}_N$ is the covariance matrix. Let us consider a zero-mean isotropic Gaussian governed by a single precision parameter $\alpha$ as the prior distribution of $\mathbf{w}$, i.e., $p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|0, \alpha^{-1}\mathbf{I})$. Then after deriving the mean vector and the covariance matrix of the posterior distribution $p(\mathbf{w}|\mathbf{t})$ (find the formula by yourself in the textbook), you can obtain $\mathbf{w}$ and then the function $y(x, \mathbf{w})$. Assume that $\alpha = 10^{-6}$ and $\beta = 10$.

b) Calculate and plot, similar to Figure 3.8 of the textbook, the predictive distribution of the target value $t$ and show the mean curve and the region of variance with one standard deviation on either side of the mean curve for data size $N = 1, 2, 4, 25, 50, 80, 100$ of the dataset **data_3.csv**. Note that the predictive distribution takes the form

$$p(t|\mathbf{x}, \mathbf{t}, \alpha, \beta) = \mathcal{N}(t|\mathbf{m}_N^T \phi(\mathbf{x}), \sigma_N^2(\mathbf{x})), \tag{5}$$

where $\sigma_N^2(\mathbf{x}) = \frac{1}{\beta} + \phi(\mathbf{x})^T \mathbf{S}_N \phi(\mathbf{x})$.

c) Repeat (a) and (b) using a different set of basis functions

$$\phi_j(x) = \sigma\left(\frac{x - \mu_j}{s}\right), \tag{6}$$

where $\mu_j = \frac{2j}{M}$ and $\sigma(a)$ is the logistic sigmoid function defined by

$$\sigma(a) = \frac{1}{1 + \exp(-a)}. \tag{7}$$

Let $\alpha = 10^{-6}$ and $\beta = 1$.