

Due data: 2/25/2020, end of the day.

For question 1-3, please submit a **PDF file** via Canvas.

For question 4 (programming question), please submit an **.ipynb file** via Canvas.

Please answer the following questions:

- [6 points] Prove Bayes' Theorem. Briefly explain why it is useful for machine learning problems, i.e., by converting posterior probability to likelihood and prior probability.
- [10 points] In Lecture 3-1, we gave the normal equation (i.e., closed-form solution) for linear regression using MSE as the cost function. **Prove that the closed-form solution for Ridge Regression** is $\mathbf{w} = (\lambda I + X^T \cdot X)^{-1} \cdot X^T \cdot \mathbf{y}$, where I is the identity matrix, $X = (x^{(1)}, x^{(2)}, \dots, x^{(m)})^T$ is the input data matrix, $x^{(i)} = (1, x_1, x_2, \dots, x_n)$ is the i th data sample, and $\mathbf{y} = (y^{(1)}, y^{(2)}, \dots, y^{(m)})$. Assume the hypothesis function $h_{\mathbf{w}}(x) = w_0 + w_1 x_1 + w_2 x_2 + \dots + w_n x_n$, and $y^{(j)}$ is the measurement of $h_{\mathbf{w}}(x)$ for the j th training sample. The cost function of the Ridge Regression is $E(\mathbf{w}) = \text{MSE}(\mathbf{w}) + \frac{\lambda}{2} \sum_{i=1}^m w_i^2$. [Hint: please refer to the proof of the normal equation of linear regression. [Note: Please use the following rectified definition of MSE when you prove: $\text{MSE}(\mathbf{w}) = \sum_{i=1}^m (\mathbf{w}^T \cdot \mathbf{x}^{(i)} - y^{(i)})^2$.] .
- [10 points] Recall the multi-class Softmax Regression model on page 16 of Lecture 3-3. Assume we have K different classes. The posterior probability is $\hat{p}_k = \delta(s_k(x))_k = \frac{\exp(s_k(x))}{\sum_{j=1}^K \exp(s_j(x))}$ for $k = 1, 2, \dots, K$, where $s_k(x) = \theta_k^T \cdot x$, and input x is an n -dimension vector.
 - To learn this Softmax Regression model, how many parameters we need to estimate? What are these parameters?
 - Consider the cross-entropy cost function $J(\theta)$ (see page 16 of Lecture 3-3) of m training samples $\{(x_i, y_i)\}_{i=1,2,\dots,m}$. Derive the gradient of $J(\theta)$ regarding to θ_k as shown in page 17 of Lecture 3-3

Programming Problem:

- [44 points] In this problem, we write a program to find the coefficients for a linear regression model for the dataset provided (data2.txt). Assume a linear model: $y = w_0 + w_1 \cdot x$. You need to
 - Plot the data (i.e., x-axis for 1st column, y-axis for 2nd column), and use Python to implement the following methods to find the coefficients:
 - Normal equation, and
 - Gradient Descent using **batch** AND **stochastic** modes respectively:
 - Determine an appropriate termination condition (e.g., when cost function is less than a threshold, and/or after a given number of iterations).
 - Print the cost function vs. iterations for each mode; compare and discuss batch and stochastic modes in terms of the accuracy and the speed of convergence.
 - Choose a best learning rate. For example, you can plot cost function vs. learning rate to determine the best learning rate.

Please implement the algorithms by yourself and **do NOT use the fit() function** of the library.