New Mexico State University
Klipsch School of Electrical Engineering

EE565 Pattern Recognition and Machine Learning
Fall 2016 – Project #4
Due: 5:00pm Thu. Oct. 13

Name: _____

Grade: _____

Project

The goal of this project is to gain experience with

- linear models for regression
- linear models for classification

Companion files, which include training points for the linear regression problems, may be found at

`http://www.ece.nmsu.edu/~pdeleon/Teaching/EE565/Projects/CompanionFiles4.zip`

For Problem 3, you may use the Statistical Pattern Recognition (STPR) toolbox for MATLAB

`http://cmp.felk.cvut.cz/cmp/software/stprtool/`

and scikit-learn for Python implementations

`http://http://scikit-learn.org/`

Report

Please submit a hardcopy report with your results including commentary and plots. Please `mailto:pdeleon@nmsu.edu` a zip file containing all MATLAB code to recreate your results.

Notes

Students are encouraged to discuss detailed, technical aspects with each other and Prof. De Leon. Students are encouraged to utilize existing MATLAB functions in this project. However, students must write all other required codes on an *individual* basis.

1. Linear Models for Regression–Part 1

Create four plots similar to Fig. 1.4 in the textbook using the author's `curvefitting.txt` containing the ten training data points for each of the following basis functions.

a Identity or linear, $\phi_j(x) = x$

b Polynomial, $\phi_j(x) = x^j$ (repeated from Project #2)

c Gaussian, $\phi_j(x) = \exp\left\{-\dfrac{(x-\mu_j)^2}{2s^2}\right\}$

d Sigmoidal, $\phi_j(x) = \sigma\left(\dfrac{x-\mu_j}{s}\right)$ where $\sigma(a) = \dfrac{1}{1+\exp(-a)}$

Note that in Chapter 1, M is the order of the polynomial while in Chapter 3, M is the number of model parameters (equal to polynomial order + 1). Denote your plots according to Chapter 3 convention and title them with the basis function and model order. For Gaussian and Sigmoidal basis functions, linearly space $\{\mu_j\}$ over the interval $[0, 1]$ and let $s = 1$.

You must implement (3.15) as part of your solution and NOT use MATLAB's polyfit, glmfit, or other equivalent functions.

2. Linear Models for Regression–Part 2

Create a plot similar to Fig. 1.5 in the textbook using the author's curvefitting.txt file containing the ten training data points and 100 of your own test data points (generated according to Appendix A Synthetic Data) for each of the following basis functions.

a Identity or linear, $\phi_j(x) = x$

b Polynomial, $\phi_j(x) = x^j$ (repeated from Project #2)

c Gaussian, $\phi_j(x) = \exp\left\{-\dfrac{(x-\mu_j)^2}{2s^2}\right\}$

d Sigmoidal, $\phi_j(x) = \sigma\left(\dfrac{x-\mu_j}{s}\right)$ where $\sigma(a) = \dfrac{1}{1+\exp(-a)}$

3. Linear Models for Two-Class Classification

a Recreate Fig. 4.4 [least-squares (LS) classifier (magenta curve) and not the logistic model (green curve)] in the textbook using the supplied code to generate the training data points. You must implement (4.16) as part of your solution.

b Produce the same plot as in Fig. 4.4 but using the Fisher Linear Discriminant (FLD) (see STPR fld or SciPy).

c Produce the same plot as in Fig. 4.4 but using the Perceptron algorithm (see STPR perceptron or scikit-learn).

d For all three classifiers in (a)–(c), evaluate the system with 100 test data points (50 from each class) and measure system accuracy.

Note 1: For the LS two-class classifier, use 1-of-K coding for the target vectors of the form $[1\ \ 0]^T$ or $[0\ \ 1]^T$. For FLD and perceptron classifiers, see STPR and SciPy/scikit-learn functions for proper input.

Note 2: The 2-D model is of the form $y(\mathbf{x}) = \tilde{\mathbf{w}}^T\tilde{\mathbf{x}}$ where $\tilde{\mathbf{w}} = [w_0\ w_1\ w_2]^T$ and $\tilde{\mathbf{x}} = [1\ x_1\ x_2]^T$. If $y(\mathbf{x}) \geq 0.5$ we decide $\mathcal{C}_1$ otherwise we decide $\mathcal{C}_2$. Thus $y(\mathbf{x}) = 0.5 = \tilde{\mathbf{w}}^T\tilde{\mathbf{x}}$ represents the decision boundary. Graph the 1-D decision surface (magenta curve) with $x_2 = (-w_1/w_2)x_1 + (0.5 - w_0)/w_2$ where $x_2,\ x_1$ is the vertical, horizontal axis variable respectively.

4. Linear Models for Multi-Class Classification

a Recreate Fig. 4.5 left plot (LS classifier) in the textbook using the supplied code to generate the training data points. Kudos, if you can shade the decision regions as red, green, or blue. You must implement (4.16) as part of your solution.

b Evaluate the system with 150 test data points (50 from each class) and measure system accuracy.

Note: For the LS multi-class classifier, use 1-of-K coding for the target vectors of the form $[1 \;\; 0 \;\; 0]^T$, $[0 \;\; 1 \;\; 0]^T$, or $[0 \;\; 0 \;\; 1]^T$.

5. Probabilistic Generative Models

Produce the same plots as in Fig. 4.4 but using a probabilistic generative model. To do this, first estimate the distributional parameters of the training data using Section 4.2.2. Then use the distributional parameters in (4.66) and (4.67) to obtain the linear model and the plot. If you wish (not required), you can use the distributional parameters to recreate Fig. 4.10(a).