New Mexico State University
Klipsch School of Electrical Engineering

EE565 Pattern Recognition and Machine Learning
Fall 2016 – Project #1
Due: 5:00pm Thu. Sep. 1

Name: _____

Grade: _____

Project

The goal of this project is to gain familiarity with

- polynomial curve fitting (linear regression) presented in Section 1.1
- regularized linear regression presented in Section 1.1
- maximum likelihood (ML) estimation of distributional parameters in Section 1.2
- decision theory in Section 1.5

Companion files for this project may be found at

`http://www.ece.nmsu.edu/~pdeleon/Teaching/EE565/Projects/CompanionFiles1.zip`

1 Polynomial Curve Fitting–Linear Model for Regression

a Solve Problem 1.1 in the text. Note that general solution is presented in Chapter 3 equation (3.15). To specialize the solution for linear regression, use $\phi_j(x) = x^j$.

b Compute and list $\mathbf{w}^*$ using data from the curvefitting.txt file and recreate the plots in Figure 1.4. You can check your solutions against Table 1.1. You may not use MATLAB's polyfit or other equivalent function.

c Use $\mathbf{w}^*$ and the data from curvefitting.txt file to recreate the plot in Figure 1.5 (you will need to generate your own test data points according to Appendix A Synthetic Data). You may need to omit the factor of 2 in the E_{RMS} formula in (1.3) to match the plot.

d Rereate the plots in Figure 1.6. Since the noise values used to create the plots in Figure 1.4 are unknown, your plots will be slightly different.

2 Polynomial Curve Fitting–Regularization

a Solve Problem 1.2 in the text. Note that general solution is presented in Chapter 3 equation (3.28). To specialize the solution to linear regression, use $\phi_j(x) = x^j$.

b Compute and list $\mathbf{w}^*$ using data from the curvefitting.txt file and recreate the plots in Figure 1.7. You can check your solutions against Table 1.2[1]. You may not use MATLAB's polyfit or other equivalent function.

c Use $\mathbf{w}^*$ and the curvefitting.txt file to recreate the plot in Figure 1.8 (you will need to generate your own test data points according to Appendix A Synthetic Data).

[1] Prof. DeLeon's values are close but not exact to these values

3 Maximum Likelihood Estimation of Normal Parameters μ, σ^2

a Solve Problem 1.11 in the text.

b Generate 1000 data points from $\mathcal{N}(0, 1)$ and estimate the distributional parameters by computing the sample mean and sample variance using (1.55) and (1.56). Repeat this experiment 1000 times and plot histograms of the sample means and sample variances. Compute the average of the sample means, $\bar{\mu}_{\mathrm{ML}}$ and average of the sample variances, $\bar{\sigma}^2_{\mathrm{ML}}$. Compare your averages against the theoretical values in (1.57) and (1.58). Comment.

4 Minimum Misclassification Rate Rule

In the CompanionFiles1.zip are files for simulating a classifier. Each row of `training.txt` has a label or class in the first column and a data point in the second column. Each row of `test.txt` has a data point. Each row of `key.txt` has the actual class of the corresponding data point in `test.txt`.

a Assuming the training data points are generated from normal processes, estimate the distributional parameters μ and σ^2 for each class.

b Determine the optimal decision boundary, i.e. where the distributions cross. This can be determined analytically by solving the quadratic based on your distributions in (a). Plot the joint probability density functions (pdfs) $p(x, \mathcal{C}_1)$ and $p(x, \mathcal{C}_2)$ using results from (a) and indicate the decision boundary on the plot.

c Classify each test point as either $\mathcal{C}_1$ or $\mathcal{C}_2$ based on whether it is less than or greater than the decision point. Compute a confusion matrix for your simulation using test points.

d Pick a different or non-optimal decision point and repeat (c). Compute a confusion matrix for your simulation and show the results are worse than those in (c).

e Assume equal class priors, i.e. $p(\mathcal{C}_1) = p(\mathcal{C}_2) = 1/2$. Classify each test point as either $\mathcal{C}_1$ or $\mathcal{C}_2$ based on maximum *a posterior* (MAP). As described in the text at the bottom of p. 39, this is equivalent to computing $p(x, \mathcal{C}_1)$ and $p(x, \mathcal{C}_2)$ for the test point and basing the classification decision on which is maximum. Show the results are identical to (c).

Report

Please submit a hardcopy report with plots, tables, values, and comments (do not list code). Please `mailto:pdeleon@nmsu.edu` a zip file containing all source code, e.g. if programming in MATLAB submit prob1.m, prob2.m, prob3.m, and prob4.m for each problem. It goes without saying that all code should run "out-of-the-box" (no hard-coded paths) and complete in a few seconds.

Notes

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