CS 195-5: Machine Learning
Problem Set 1

Douglas Lanman
dlanman@brown.edu
27 September 2006

1 Regression

Problem 1

Show that the prediction errors $y - f(\mathbf{x}; \hat{\mathbf{w}})$ are necessarily uncorrelated with any linear function of the training inputs. That is, show that for any $\mathbf{a} \in \mathbb{R}^{d+1}$

$$\hat{\sigma}(e, \mathbf{a}^T\mathbf{x}) = 0,$$

where $e_i = y_i - \hat{\mathbf{w}}^T\mathbf{x}_i$ is the prediction error for the i^{th} training example.

First, recall that the least squares estimate for the linear regression parameters is given by

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{n}(y_i - w_0 - \sum_{j=1}^{d} w_j x_j^{(i)})^2 = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{n}(y_i - \hat{\mathbf{w}}^T\mathbf{x}_i)^2 = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{n}(e_i)^2, \quad (1)$$

where we have augmented the inputs $\mathbf{x} \in \mathbb{R}^d$ by adding 1 as the "zeroth" dimension such that $x_0 = 1$. Recall that the minimum (or maximum) of the argument of Equation 1 will be achieved where $\partial/\partial w_i$ equals zero for every regression parameter w_i. Differentiating with respect to w_0 and equating with zero gives

$$\frac{\partial}{\partial w_0} \sum_{i=1}^{n}(y_i - w_0 - \sum_{j=1}^{d} w_j x_j^{(i)})^2 = \sum_{i=1}^{n} 2(y_i - w_0 - \sum_{j=1}^{d} w_j x_j^{(i)}) \cdot (-1) = 0$$

$$\Rightarrow \frac{1}{n}\sum_{i=1}^{n}(y_i - w_0 - \sum_{j=1}^{d} w_j x_j^{(i)}) = \frac{1}{n}\sum_{i=1}^{n} e_i = \bar{e} = 0. \quad (2)$$

In other words, a necessary condition for $\hat{\omega}$ is that the prediction errors have zero mean. Similarly, differentiating with respect to any w_i (for $i \in \{0, \ldots, n\}$) and equating with zero gives

$$\frac{\partial}{\partial w_i} \sum_{i=1}^{n}(y_i - w_0 - \sum_{j=1}^{d} w_j x_j^{(i)})^2 = \sum_{i=1}^{n} 2(y_i - w_0 - \sum_{j=1}^{d} w_j x_j^{(i)}) \cdot (-w_j x_j^{(i)}) = 0$$

$$\Rightarrow \sum_{i=1}^{n}(y_i - w_0 - \sum_{j=1}^{d} w_j x_j^{(i)})x^{(i)} = \sum_{i=1}^{n} e_i x_i = 0. \quad (3)$$

As in class, this result implies that the prediction errors are uncorrelated with the training data.

We now turn our attention to proving the claim that the prediction errors are necessarily uncorrelated with any linear function of the training inputs. Recall that the correlation is written

$$\hat{\sigma}(e, \mathbf{a}^T\mathbf{x}) = \sum_{i=1}^{n}(e_i - \bar{e})(\mathbf{a}^T\mathbf{x}_i - \overline{\mathbf{a}^T\mathbf{x}_i}) = \sum_{i=1}^{n} e_i(\mathbf{a}^T\mathbf{x}_i - \overline{\mathbf{a}^T\mathbf{x}_i}), \quad (4)$$

where we have applied Equation 2 to conclude $\bar{e} = 0$. Now let us examine the term $\overline{\mathbf{a}^T\mathbf{x}}$.

$$\overline{\mathbf{a}^T\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{a}^T\mathbf{x}_i = \mathbf{a}^T\left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\right) = \mathbf{a}^T\bar{\mathbf{x}}$$

Note that we are allowed to bring $\mathbf{a}^T$ out of the summation by linearity. Also note that $\mathbf{a}^T\bar{\mathbf{x}}$ is a scalar quantity and, as a result, we can write Equation 4 as

$$\hat{\sigma}(e, \mathbf{a}^T\mathbf{x}) = \left(\mathbf{a}^T\sum_{i=1}^{n}e_i\mathbf{x}_i\right) - \left(\mathbf{a}^T\bar{\mathbf{x}}\sum_{i=1}^{n}e_i\right) = 0,$$

where, by Equations 2 and 3, we know that both summations are equal to zero. As a result, we have proven desired result.

$$\boxed{\hat{\sigma}(e, \mathbf{a}^T\mathbf{x}) = 0}$$

(QED)

Problem 2

Suppose that the data in a regression problem are scaled by multiplying the j^{th} dimension of the input by a non-zero number c_j. Let $\tilde{\mathbf{x}} \triangleq [1, c_1 x_1, \ldots, c_d x_d]^T$ denote a single scaled data point and let $\tilde{\mathbf{X}}$ represent the corresponding design matrix. Similarly, let $\hat{\mathbf{w}}$ be the maximum likelihood (ML) estimate of the regression parameters from the unscaled $\mathbf{X}$, and let $\hat{\tilde{\mathbf{w}}}$ be the solution obtained from the scaled $\tilde{\mathbf{X}}$. Show that scaling does not change optimality, in the sense that $\hat{\mathbf{w}}^T\mathbf{x} = \hat{\tilde{\mathbf{w}}}^T\tilde{\mathbf{x}}$.

First, note that scaling can be represented as a linear operator $\mathbf{C}$ composed of the scaling factors along its main diagonal.

$$\mathbf{C} = \begin{pmatrix} 1 & & & \\ & c_1 & & \\ & & \ddots & \\ & & & c_d \end{pmatrix}$$

Using the scaling operator $\mathbf{C}$, we can express the scaled inputs $\tilde{\mathbf{x}}$ and design matrix $\tilde{\mathbf{X}}$ as functions of $\mathbf{x}$ and $\mathbf{X}$, respectively.

$$\tilde{\mathbf{x}} = \mathbf{C}\mathbf{x} \qquad \tilde{\mathbf{X}} = \mathbf{X}\mathbf{C} \tag{5}$$

As was demonstrated in class, under the Gaussian noise model, the ML estimate of the regression parameters is given by

$$\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}.$$

Using the expressions in Equation 5, we find

$$\begin{aligned} \hat{\tilde{\mathbf{w}}} &= (\tilde{\mathbf{X}}^T\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}^T\mathbf{y} = \left((\mathbf{X}\mathbf{C})^T\mathbf{X}\mathbf{C}\right)^{-1}(\mathbf{X}\mathbf{C})^T\mathbf{y} \\ &= \left(\mathbf{C}^T\mathbf{X}^T\mathbf{X}\mathbf{C}\right)^{-1}\mathbf{C}^T\mathbf{X}^T\mathbf{y}. \end{aligned}$$

At this point, we can apply the following matrix identity: if the individual inverses $\mathbf{A}^{-1}$ and $\mathbf{B}^{-1}$ exist, then $(\mathbf{A}\mathbf{B})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$. Since $\mathbf{C}$ is a real, symmetric square matrix, $\mathbf{C}^{-1}$ must exist. Similarly, $\mathbf{X}^T\mathbf{X}$ is a real, symmetric square matrix so $(\mathbf{X}^T\mathbf{X}\mathbf{C})^{-1}$ must also exist. As a result, we can apply the matrix identity to the previous expression.

$$\begin{aligned} \hat{\tilde{\mathbf{w}}} &= \left((\mathbf{C}^T)(\mathbf{X}^T\mathbf{X}\mathbf{C})\right)^{-1}\mathbf{C}^T\mathbf{X}^T\mathbf{y} \\ &= (\mathbf{X}^T\mathbf{X}\mathbf{C})^{-1}(\mathbf{C}^T)^{-1}\mathbf{C}^T\mathbf{X}^T\mathbf{y} \\ &= \mathbf{C}^{-1}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \\ &= \mathbf{C}^{-1}\hat{\mathbf{w}} \end{aligned} \tag{6}$$

To prove that the scaled solution is optimal, we apply Equations 5 and 6 as follows.

$$\hat{\tilde{\mathbf{w}}}^T\tilde{\mathbf{x}} = (\mathbf{C}^{-1}\hat{\mathbf{w}})^T\mathbf{C}\mathbf{x} = \hat{\mathbf{w}}^T(\mathbf{C}^{-1})^T\mathbf{C}\mathbf{x}$$

Recall from [4] that $(\mathbf{A}^{-1})^T = (\mathbf{A}^T)^{-1}$. As a result, $(\mathbf{C}^{-1})^T\mathbf{C} = (\mathbf{C}^T)^{-1}\mathbf{C}$. Since $\mathbf{C}$ is a real, symmetric matrix, $\mathbf{C} = \mathbf{C}^T$ and, as a result, $(\mathbf{C}^{-1})^T\mathbf{C} = \mathbf{I}$. In conclusion we have proven the desired result.

$$\boxed{\hat{\mathbf{w}}^T\mathbf{x} = \hat{\tilde{\mathbf{w}}}^T\tilde{\mathbf{x}}}$$

(QED)

Problem 3

Derive the maximum likelihood (ML) estimate of σ^2 under the Gaussian noise model.

Recall that the likelihood $\mathcal{P}$ of the noise variance σ^2, given the observations $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T$ and $\mathbf{Y} = [y_1, \ldots, y_N]^T$, is defined to be

$$\mathcal{P}(\mathbf{Y}; \mathbf{w}, \sigma) \triangleq p(\mathbf{Y}|\mathbf{X}, \mathbf{w}, \sigma).$$

Also recall that, under the Gaussian noise model, the label y is a random variable

$$y = f(\mathbf{x}; \mathbf{w}) + \nu, \;\; \nu \sim \mathcal{N}(\nu, 0, \sigma),$$

with the following distribution

$$p(y|\mathbf{x}, \mathbf{w}, \sigma) = \mathcal{N}(y; f(\mathbf{x}; \mathbf{w}), \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y - f(\mathbf{x}; \mathbf{w}))^2}{2\sigma^2}\right).$$

Assuming that the observations are independent and identically distributed (i.i.d.), then the likelihood can be expressed as the following product.

$$\mathcal{P}(\mathbf{Y}; \mathbf{w}, \sigma) = \prod_{i=1}^{N} p(y_i|\mathbf{x}_i, \mathbf{w}, \sigma) = \prod_{i=1}^{N} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y_i - f(\mathbf{x}_i; \mathbf{w}))^2}{2\sigma^2}\right),$$

with the corresponding ML estimator for σ^2 given by

$$\hat{\sigma}^2_{ML} = \underset{\sigma^2}{\operatorname{argmax}} \prod_{i=1}^{N} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y_i - f(\mathbf{x}_i; \mathbf{w}))^2}{2\sigma^2}\right).$$

As was done in class, we consider the log-likelihood ℓ which converts this product into a summation as follows.

$$\ell(\mathbf{Y}; \mathbf{w}, \sigma) \triangleq \log \mathcal{P}(\mathbf{Y}; \mathbf{w}, \sigma) = -\frac{1}{2\sigma^2} \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i; \mathbf{w}))^2 - N \log(\sigma\sqrt{2\pi}) \tag{7}$$

Since the logarithm is a monotonically-increasing function, the ML estimator becomes

$$\hat{\sigma}^2_{ML} = \underset{\sigma^2}{\operatorname{argmax}} \left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i; \mathbf{w}))^2 - N \log(\sigma\sqrt{2\pi}) \right\}.$$

The maximum (or minimum) of this function will necessarily be obtained where the derivative with respect to σ^2 equals zero.

$$\frac{\partial}{\partial \sigma^2} \left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i; \mathbf{w}))^2 - N \log(\sigma\sqrt{2\pi}) \right\} = 0$$

$$\Rightarrow \frac{1}{2\sigma^4} \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i; \mathbf{w}))^2 - \frac{N}{2\sigma^2} = 0$$

In conclusion, the ML estimate of σ^2 is given by the following expression.

$$\boxed{\hat{\sigma}^2_{ML} = \frac{1}{N}\sum_{i=1}^{N}(y_i - f(\mathbf{x}_i; \mathbf{w}))^2} \tag{8}$$

Note that this expression maximizes the likelihood of the observations, assuming $\mathbf{w}$ is known. If this is not the case, then the ML estimate $\hat{\mathbf{w}}_{ML}$ should be substituted for $\mathbf{w}$ in Equation 8. As was shown in class, the ML estimate $\hat{\mathbf{w}}_{ML}$ is independent of σ^2 and, as a result, can be estimated independently. In other words, if the ML estimates of *both* $\mathbf{w}$ and σ^2 are required, then the following expressions should be used.

$$\hat{\mathbf{w}}_{ML} = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{N}(y_i - f(\mathbf{x}_i; \mathbf{w}))^2$$

$$\hat{\sigma}^2_{ML} = \frac{1}{N}\sum_{i=1}^{N}(y_i - f(\mathbf{x}_i; \hat{\mathbf{w}}_{ML}))^2$$

Problem 4

Part 1: Consider the noise model $y = f(\mathbf{x}; \mathbf{w}) + \nu$ in which ν is drawn from the following distribution.

$$p(\nu) = C \exp(-\nu^4), \text{ for } C = \left(\int_{-\infty}^{\infty} \exp(-x^4)\, dx \right)^{-1}$$

Derive the conditions on the maximum likelihood estimate $\hat{\mathbf{w}}_{ML}$. What is the corresponding loss function? Compare this loss function with squared loss on the interval $[-3, 3]$. How do you expect these differences to affect regression?

As in Problem 3, let's begin by defining the distribution of the label y, given the input $\mathbf{x}$.

$$p(y|\mathbf{x}, \mathbf{w}) = C \exp\left(-(y - f(\mathbf{x}; \mathbf{w}))^4\right)$$

Once again, we assume that the observations are i.i.d. such that the likelihood $\mathcal{P}$ is given by

$$\mathcal{P}(\mathbf{Y}; \mathbf{w}) = \prod_{i=1}^{N} p(y_i|\mathbf{x}_i, \mathbf{w}) = \prod_{i=1}^{N} C \exp\left(-(y_i - f(\mathbf{x}_i; \mathbf{w}))^4\right).$$

The log-likelihood ℓ is then given by

$$\ell(\mathbf{Y}; \mathbf{w}) = \log \mathcal{P}(\mathbf{Y}; \mathbf{w}) = NC - \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i; \mathbf{w}))^4,$$

with the corresponding ML estimate for $\mathbf{w}$ given by

$$\boxed{\hat{\mathbf{w}}_{ML} = \underset{\mathbf{w}}{\operatorname{argmax}}\, \ell(\mathbf{Y}; \mathbf{w}) = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i; \mathbf{w}))^4 = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{N} L'(y_i, f(\mathbf{x}_i; \mathbf{w}))}$$

where L' is the quadrupled loss function defined as follows.

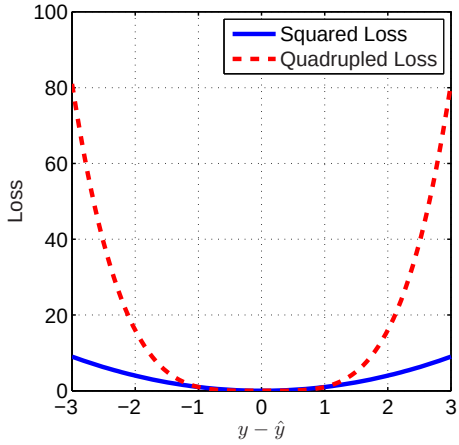
$$\boxed{L'(y, \hat{y}) = (y - \hat{y})^4}$$

The squared loss $L(y, \hat{y}) = (y - \hat{y})^2$ and quadrupled loss were plotted (as a function of $y - \hat{y}$) in Figure 1(a) using the Matlab script `prob4.m`. From this plot it is apparent that L' creates a greater penalty for outliers than L. That is, as the prediction error $|y - \hat{y}|$ increases, the regression with quadrupled loss will bias towards outliers more than with squared loss.

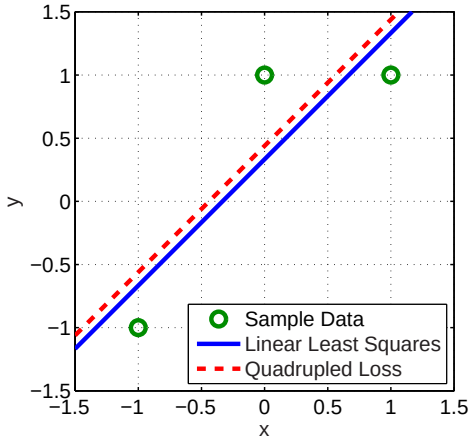
Part 2: Now consider the following data set $\mathbf{X} = [-1, 0, 1]^T$ and $\mathbf{Y} = [-1, 1, 1]^T$. Find the numerical solution for $\hat{\mathbf{w}}_{ML}$ using both linear least squares (i.e., squared loss) and quadrupled loss and report the empirical losses. Plot the corresponding functions and explain how what is seen results from the differences in loss functions.

Using `prob4.m`, the optimal values of (w_0, w_1) were determined using the provided `testWs.m` function for both the squared and quadrupled losses. The results are tabulated below.

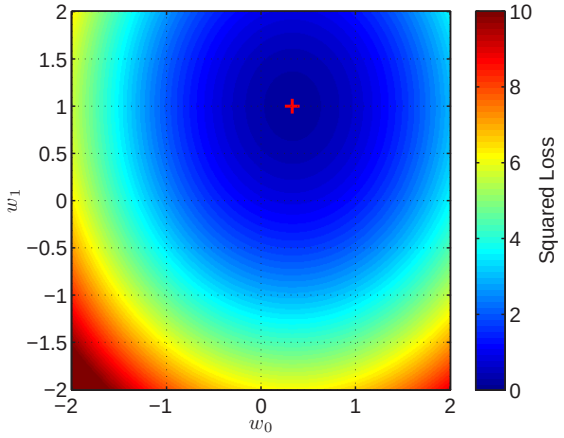
Loss Function	$\hat{w}_0$	$\hat{w}_1$	Empirical Loss
Squared Loss (LSQ): $L(y, \hat{y})$	$\approx$0.33	1.00	$\approx$0.222
Quadrupled Loss: $L'(y, \hat{y})$	0.44	1.00	$\approx$0.234



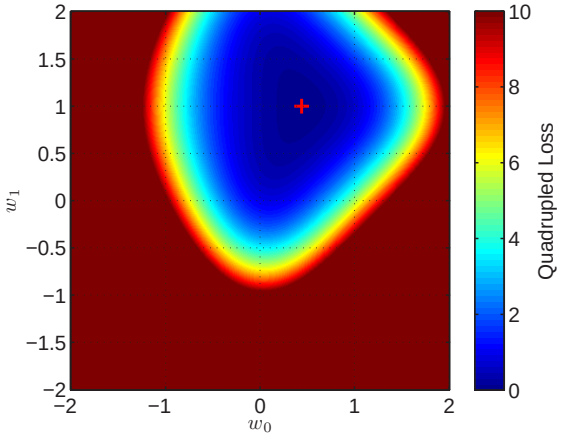
(a) Comparison of loss functions



(b) Comparison of ML-estimated models



(c) Squared loss surface



(d) Quadrupled loss surface

Figure 1: ML estimation using exhaustive search under varying loss functions.

Note that the empirical losses L_N or L'_N are given by the average sum of squares as follows.

$$L_N(\mathbf{w}) = L'_N(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - f(\mathbf{x}_i;\mathbf{w}))^2 \tag{9}$$

Using the values in the table, the corresponding regression models were plotted in Figure 1(b), with the associated loss surfaces as shown in Figures 1(c) and 1(d). Note that the two noise models resulted in lines with identical slopes (i.e., equal values of w_1), however the y-intercepts (given by w_0) differed. This is consistent with our previous observation that the quadrupled loss model biases towards outliers. For this example, the estimated line moves towards the second data point at $(x = 0, y = 1)$ since this point can be viewed as an outlier.

Problem 5

Apply linear and quadratic polynomial regression to the data in `meteodata.mat` using linear least squares. Use 10-fold cross-validation to select the best model and plot the results. Report the empirical loss and log-likelihood. Based on the plot, comment on the Gaussian noise model.

Recall from class on 9/13/06 that we can solve for the least squares polynomial regression coefficients $\hat{\mathbf{w}}$ by using the extended *design matrix* $\tilde{\mathbf{X}}$ such that

$$\hat{\mathbf{w}} = (\tilde{\mathbf{X}}^T\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}^T\mathbf{y}, \text{ with } \tilde{\mathbf{X}} = \begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^d \\ 1 & x_2 & x_2^2 & \dots & x_2^d \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & x_N^2 & \dots & x_N^d \end{pmatrix},$$

where d is the degree of the polynomial and $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ and $\mathbf{y}$ are the observed points and their associated labels, respectively. To prevent numerical round-off errors, this method was applied to the column-normalized design matrix $\tilde{\mathbf{X}}$ using `degexpand.m` within the main script `prob5.m` (as discussed in Problem 2). The resulting linear and quadratic polynomials are shown in Figure 2(a). The fitting parameters obtained using all data points and up to a fourth-order polynomial are tabulated below.

Polynomial Degree	Empirical Loss	Log-likelihood	10-fold Cross Validation Score
Linear ($d = 1$)	1.057	-529.5	1.063
Quadratic ($d = 2$)	0.930	-506.1	0.943
Cubic ($d = 3$)	0.930	-506.1	0.947
Quartic ($d = 4$)	0.925	-505.1	0.950

Note that the empirical loss L_N is defined to be the average sum of squared errors as given by Equation 9. The log-likelihood of the data (under a Gaussian model) was derived in class on 9/11/06 and is given by Equation 7 as

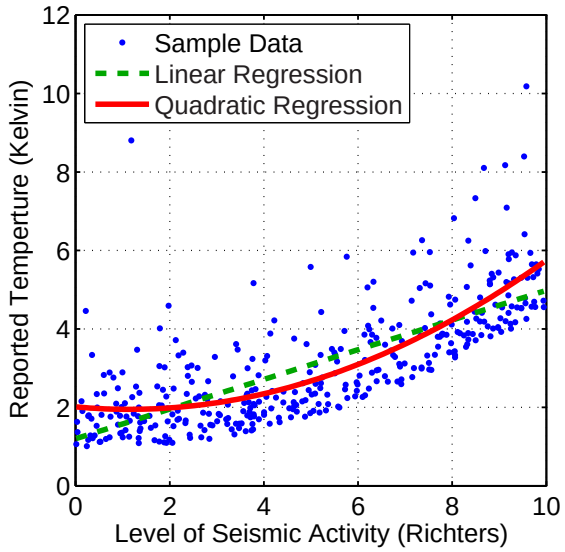
$$\ell(\mathbf{Y}; \mathbf{w}, \sigma) = -\frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - f(\mathbf{x}_i; \mathbf{w}))^2 - N\log(\sigma\sqrt{2\pi}),$$

where $\sigma \to \hat{\sigma}_{ML}$ is the maximum likelihood estimate of σ under a Gaussian noise model (as given by Equation 8 in Problem 3).

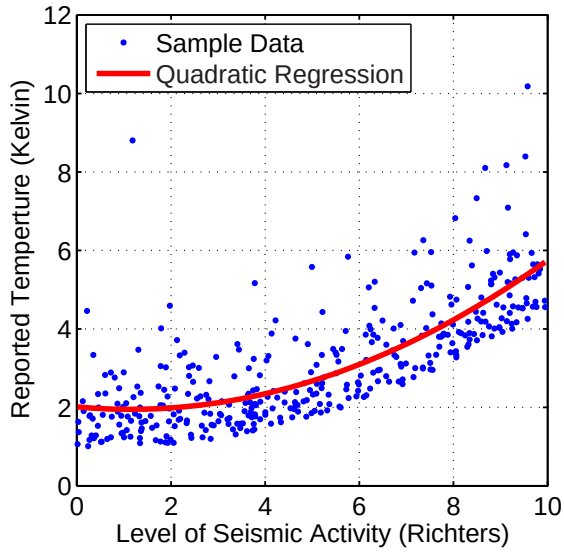
At this point, we turn our attention to the model-order selection task (i.e., deciding what degree of polynomial best-represents the data). As discussed in class of 9/13/06, we will use 10-fold cross validation to select the best model. First, we partition the data into 10 roughly equal parts (see lines 65-70 of `prob5.m`). Next, we perform 10 sequential trials where we train on all but the i^{th} fold of the data and then measure the empirical error on the remaining samples. In general, we formulate the k-fold cross validation score as

$$\hat{L}_k = \frac{1}{N}\sum_{i=1}^{k}\sum_{j \in \text{fold } i}(y_j - f(\mathbf{x}_j; \hat{\mathbf{w}}_i)^2,$$

where $\hat{\mathbf{w}}_i$ is fit to all samples except those in the i^{th} fold. The resulting 10-fold cross-validation scores are tabulated above (for up to a fourth-order polynomial). Since the lowest cross-validation score is achieved for the **quadratic polynomial** we select this as the best model.



(a) Least squares polynomial regression results



(b) Quadratic polynomial (selected by cross-validation)

Figure 2: Comparison of linear and quadratic polynomials estimated using linear least squares.

In conclusion, we find that the quadratic polynomial has the lowest cross-validation score. However, as shown in Figure 2(b), it is immediately apparent that the Gaussian noise model does not accurately represent the underlying distribution; more specifically, if the underlying distribution was Gaussian, we'd expect half of the data points to be above the model prediction (and the other half below). This is clearly not the case for this example – motivating the alternate noise model we'll analyze in Problem 6.

Problem 6

Part 1: Consider the noise model $y = f(\mathbf{x}; \mathbf{w}) + \nu$ in which ν is drawn from $p(\nu)$ as follows.

$$p(\nu) = \begin{cases} e^{-\nu} & \text{if } \nu > 0, \\ 0 & \text{otherwise} \end{cases}$$

Perform 10-fold cross-validation for linear and quadratic regression under this noise model, using exhaustive numerical search similar to that used in Problem 4. Plot the selected model and report the empirical loss and the log-likelihood under the estimated exponential noise model.

As in Problems 3 and 4, let's begin by defining the distribution of the label y, given the input $\mathbf{x}$.

$$p(y|\mathbf{x}, \mathbf{w}) = \begin{cases} \exp\left(-(y - f(\mathbf{x}; \mathbf{w}))\right) & \text{if } y > f(\mathbf{x}; \mathbf{w}), \\ 0 & \text{otherwise} \end{cases}$$

Once again, we assume that the observations are i.i.d. such that the likelihood $\mathcal{P}$ is given as follows.

$$\mathcal{P}(\mathbf{Y}; \mathbf{w}) = \prod_{i=1}^{N} p(y_i|\mathbf{x}_i, \mathbf{w}) = \prod_{i=1}^{N} \begin{cases} \exp\left(f(\mathbf{x}_i; \mathbf{w}) - y_i\right) & \text{if } y_i > f(\mathbf{x}_i; \mathbf{w}), \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

The log-likelihood ℓ is then given by

$$\ell(\mathbf{Y}; \mathbf{w}) = \log \mathcal{P}(\mathbf{Y}; \mathbf{w}) = \sum_{i=1}^{N} \begin{cases} f(\mathbf{x}_i; \mathbf{w}) - y_i & \text{if } y_i > f(\mathbf{x}_i; \mathbf{w}), \\ -\infty & \text{otherwise} \end{cases} \tag{11}$$

since the logarithm is a monotonic function and $lim_{x \to 0} \log x = -\infty$. The corresponding ML estimate for $\mathbf{w}$ is given by the following expression.

$$\hat{\mathbf{w}}_{ML} = \underset{\mathbf{w}}{\operatorname{argmax}}\, \ell(\mathbf{Y}; \mathbf{w}) = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{N} \begin{cases} y_i - f(\mathbf{x}_i; \mathbf{w}) & \text{if } y_i > f(\mathbf{x}_i; \mathbf{w}), \\ \infty & \text{otherwise} \end{cases} \tag{12}$$

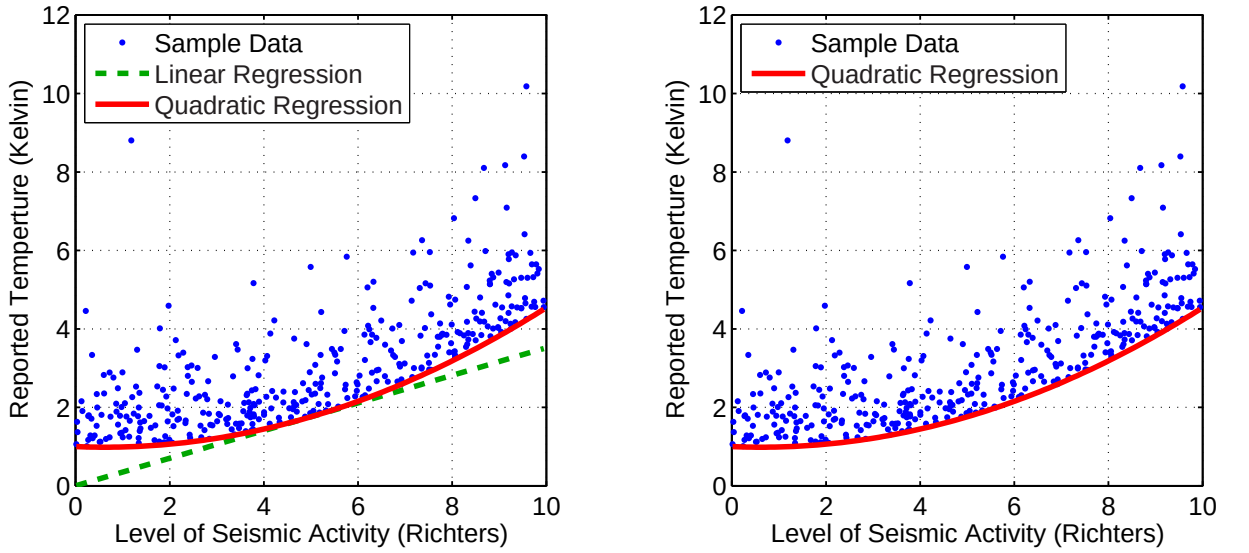
Note that Equations 10 and 11 prevent *any* prediction $f(\mathbf{x}_i; \mathbf{w})$ from being above the corresponding label y_i. As a result, the exponential noise distribution will effectively lead to a model corresponding to the lower-envelope of the training data.

Using the maximum likelihood formulation in Equation 12, we can solve for the optimal regression parameters using an exhaustive search (similar to what was done in Problem 4). This approach was applied to all the data points on lines 19-72 of `prob6.m`. The resulting best-fit linear and quadratic polynomial models are shown in Figure 3(a). The fitting parameters obtained using all the data points and up to a second-order polynomial are tabulated below.

Polynomial Degree	Empirical Loss	Log-likelihood	10-fold Cross Validation Score
Linear ($d = 1$)	2.837	-487.5	2.837
Quadratic ($d = 2$)	1.901	-359.2	1.900

Note that the empirical loss L_N was calculated using Equation 9. The log-likelihood of the data (under the exponential noise model) was determined using Equation 11.

Model selection was performed using 10-fold cross-validation as in Problem 5. The resulting scores are tabulated above. Since the lowest cross-validation score was achieved for the **quadratic polynomial** we select this as the best model and plot the result in Figure 3(b). Comparing the



(a) Polynomial regression results

(b) Quadratic polynomial (selected by cross-validation)

Figure 3: ML estimation using exhaustive search under an exponential noise model.

model in Figure 3(b) with that in Figure 2(b), we conclude that the quadratic polynomial under the exponential noise model better approximates the underlying distribution; more specifically, the quadratic polynomial (under an exponential noise model) achieves a log-likelihood of -359.2, whereas it only achieves a log-likelihood of -506.1 under the Gaussian noise model. It is also important to note that the Gaussian noise model leads to a lower squared loss (i.e., empirical loss), however this is by the construction of the least squares estimator used in Problem 5. This highlights the important observation that a lower empirical loss does not necessarily indicate a better model – this only applies when the choice of noise model appropriately models the actual distribution.

Part 2: Now evaluate the polynomials selected under the Gaussian and exponential noise models for the data in 2005. Report which performs better in terms of likelihood and empirical loss.

In both Problems 5 and 6.1, the quadratic polynomial was selected as the best model by 10-fold cross validation. In order to gauge the generalization capabilities of these models, the 2004 model parameters we used to predict the 2005 samples. The results for each noise model are tabulated below.

Noise Model	Empirical Loss	Log-likelihood
Gaussian ($d = 2$)	1.136	-541.2
Exponential ($d = 2$)	2.081	-358.5

In conclusion, we find that the Gaussian model achieves a lower empirical loss on the 2005 samples. This is expected, since the Gaussian model is equivalent to the least squares estimator which minimizes empirical loss. The exponential model, however, achieves a significantly higher log-likelihood – indicating that it models the underlying data more effectively than the Gaussian noise model. As a result, we reiterate the point made previously: low empirical loss can be achieved even if the noise model does not accurately represent the underlying noise distribution.

2 Multivariate Gaussian Distributions

Problem 7

Recall that the probability density function (pdf) of a Gaussian distribution in $\mathbb{R}^d$ is given by

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\right).$$

Show that a contour corresponding to a fixed value of the pdf is an ellipse in the 2D x_1, x_2 space.

An iso-contour of the pdf along $p(\mathbf{x}) = p_0$ is given by

$$\frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\right) = p_0.$$

Taking the logarithm of each side and rearranging terms gives

$$(\mathbf{x}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu) = -2\log\left(p_0(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}\right) = C,$$

where $C \in \mathbb{R}$ is a constant. As this point it is most convenient to write this expression explicitly as a function of x_1 and x_2 such that $\mathbf{x} = [x_1, x_2]^T$ and $\mu = [\bar{x}_1, \bar{x}_2]^T$.

$$\Rightarrow \begin{pmatrix} x_1-\bar{x}_1 & x_2-\bar{x}_2 \end{pmatrix} \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{pmatrix}^{-1} \begin{pmatrix} x_1-\bar{x}_1 \\ x_2-\bar{x}_2 \end{pmatrix} = C$$

$$\begin{pmatrix} x_1-\bar{x}_1 & x_2-\bar{x}_2 \end{pmatrix} \begin{pmatrix} \sigma_2^2 & -\sigma_{12} \\ -\sigma_{12} & \sigma_1^2 \end{pmatrix} \begin{pmatrix} x_1-\bar{x}_1 \\ x_2-\bar{x}_2 \end{pmatrix} = (\sigma_1^2\sigma_2^2 - \sigma_{12}^2)C = C'$$

Multiplying the terms in this expression, we obtain

$$\sigma_2^2(x_1-\bar{x}_1)^2 - 2\sigma_{12}(x_1-\bar{x}_1)(x_2-\bar{x}_2) + \sigma_1^2(x_2-\bar{x}_2)^2 = C'.$$

Simplifying, we obtain a solution for the 2D iso-contours given by

$$\begin{gathered} \tfrac{1}{\sigma_1^2}(x_1-\bar{x}_1)^2 - 2\tfrac{\sigma_{12}}{\sigma_1^2\sigma_2^2}(x_1-\bar{x}_1)(x_2-\bar{x}_2) + \tfrac{1}{\sigma_2^2}\sigma_1^2(x_2-\bar{x}_2)^2 = C_0, \\ C_0 \triangleq 2\left(\tfrac{\sigma_{12}^2-\sigma_1^2\sigma_2^2}{\sigma_1^2\sigma_2^2}\right)\log\left(p_0(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}\right). \end{gathered} \tag{13}$$

Recall from [6] that a general quadratic curve can be written as

$$ax_1^2 + bx_1x_2 + cx_2^2 + dx_1 + fx_2 + g = 0,$$

with an associated *discriminant* given by $b^2 - 4ac$. Also recall that if the discriminant $b^2 - 4ac < 0$, then the quadratic curve represents either an ellipse, a circle, or a point (with the later two representing certain degenerate configurations of an ellipse) [6]. Since Equation 13 has the form of a quadratic curve, it has a discriminant given by

$$b^2 - 4ac = 4\left(\frac{\sigma_{12}^2}{\sigma_1^4\sigma_2^4} - \frac{1}{\sigma_1^2\sigma_2^2}\right) = \frac{4}{\sigma_1^4\sigma_2^4}\left(\sigma_{12}^2 - \sigma_1^2\sigma_2^2\right) \overset{?}{<} 0.$$

Recall that the covariance must be non-negative, therefore $\{\sigma_1, \sigma_2, \sigma_{12}\} \geq 0$. As a result, $4/(\sigma_1^4\sigma_2^4)$ is positive and we only need to prove that $\sigma_{12}^2 < \sigma_1^2\sigma_2^2$ in order to show that Equation 13 represents an ellipse. Recall from class on 9/15/06 that the definition of the *cross-correlation coefficient* ρ_{12} is

$$\rho_{12} = \frac{\sigma_{12}}{\sigma_1\sigma_2} \;\Rightarrow\; \sigma_{12}^2 = \rho_{12}^2\sigma_1^2\sigma_2^2,$$

where $-1 \leq \rho_{12} \leq 1$. As a result, $0 \leq \rho_{12}^2 \leq 1$ which implies $\sigma_{12}^2 < \sigma_1^2\sigma_2^2$ and the discriminant has the form of an ellipse. In conclusion, Equation 13 has the associated discriminant

$$b^2 - 4ac = \frac{4}{\sigma_1^4\sigma_2^4}\left(\sigma_{12}^2 - \sigma_1^2\sigma_2^2\right) < 0,$$

which by [6] defines an ellipse in the 2D x_1, x_2 space. (QED)

Problem 8

Suppose we have a sample $\mathbf{x}_1, \ldots, \mathbf{x}_n$ drawn from $\mathcal{N}(\mathbf{x}; \mu, \mathbf{\Sigma})$. Show that the ML estimate of the mean vector μ is

$$\hat{\mu}_{ML} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$$

and the ML estimate of the covariance matrix $\mathbf{\Sigma}$ is

$$\hat{\mathbf{\Sigma}}_{ML} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \hat{\mu}_{ML})(\mathbf{x}_i - \hat{\mu}_{ML})^T.$$

Qualitatively, we will repeat the derivation used for the univariate Gaussian in Problem 3. Let's begin by defining the multivariate Gaussian distribution in $\mathbb{R}^d$.

$$p(\mathbf{x}; \mu, \mathbf{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}}\exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\right)$$

Once again, we assume that the samples are i.i.d. such that the likelihood $\mathcal{P}$ is given by

$$\mathcal{P}(\mathbf{X}; \mu, \mathbf{\Sigma}) = \prod_{i=1}^{n} p(\mathbf{x}_i|\mu, \mathbf{\Sigma}).$$

The log-likelihood ℓ is then given by

$$\begin{aligned}\ell(\mathbf{X}; \mu, \mathbf{\Sigma}) &= \log \mathcal{P}(\mathbf{X}; \mu, \mathbf{\Sigma}) = \sum_{i=1}^{n}\log\left(p(\mathbf{x}_i|\mu, \mathbf{\Sigma})\right) \\ &= -n\log\left((2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}\right) - \frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}_i-\mu), \qquad (14)\end{aligned}$$

with the corresponding ML estimate for μ given by

$$\hat{\mu}_{ML} = \underset{\mu}{\operatorname{argmax}}\,\ell(\mathbf{X}; \mu, \mathbf{\Sigma}) = \underset{\mu}{\operatorname{argmin}}\sum_{i=1}^{n}(\mathbf{x}_i-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}_i-\mu)$$

since the first term in Equation 14 is independent of μ. The minimum (or maximum) of this function will occur where the derivative with respect to μ equals zero. Let's proceed by making the substitution $\mathbf{\Sigma}^{-1} \rightarrow \mathbf{A}$. Equating the first partial derivative with zero, we find

$$\frac{\partial}{\partial\mu}\left\{\sum_{i=1}^{n}(\mathbf{x}_i-\mu)^T\mathbf{A}(\mathbf{x}_i-\mu)\right\} = \sum_{i=1}^{n}\frac{\partial}{\partial\mu}\left[(\mathbf{x}_i-\mu)^T\mathbf{A}(\mathbf{x}_i-\mu)\right] = 0$$

$$\Rightarrow \sum_{i=1}^{n}\frac{\partial}{\partial\mu}\left[\mathbf{x}_i^T\mathbf{A}\mathbf{x}_i - \mathbf{x}_i^T\mathbf{A}\mu - \mu^T\mathbf{A}\mathbf{x}_i + \mu^T\mathbf{A}\mu\right] = 0.$$

Note that the first term is independent of μ and can be eliminated.

$$\Rightarrow \sum_{i=1}^{n}\left\{-\frac{\partial(\mathbf{x}_i^T\mathbf{A}\mu)}{\partial\mu} - \frac{\partial(\mu^T\mathbf{A}\mathbf{x}_i)}{\partial\mu} + \frac{\partial(\mu^T\mathbf{A}\mu)}{\partial\mu}\right\} = 0$$

We can now apply the following identities for the derivatives of scalar and matrix/vector forms [4].

$$\frac{\partial(\mathbf{A}\mathbf{x})}{\partial\mathbf{x}} = \frac{\partial(\mathbf{x}^T\mathbf{A})}{\partial\mathbf{x}} = \mathbf{A} \qquad \frac{\partial(\mathbf{x}^T\mathbf{A}\mathbf{x})}{\partial\mathbf{x}} = (\mathbf{A}+\mathbf{A}^T)\mathbf{x}$$

Applying these expressions to the previous equation gives

$$\sum_{i=1}^{n}\left\{-\mathbf{x}_i^T\mathbf{A} - \mathbf{A}\mathbf{x}_i + \mathbf{A}\mu + \mathbf{A}^T\mu\right\} = 0.$$

Recall that $\mathbf{\Sigma}$ is a real, symmetric matrix such that $\mathbf{\Sigma}^T = \mathbf{\Sigma}$ [1]. In addition, we have $\mathbf{A}^T = \mathbf{A}$ since $\mathbf{A}$ is also a real, symmetric matrix. Finally, since $\mathbf{A}$ is symmetric, we have $\mathbf{x}_i^T\mathbf{A} = \mathbf{A}\mathbf{x}_i$. Applying these identities to the previous equation, we find

$$2\mathbf{A}\sum_{i=1}^{n}(\mathbf{x}_i - \mu) = 0 \quad \Rightarrow \quad \sum_{i=1}^{n}\mu = n\mu = \sum_{i=1}^{n}\mathbf{x}_i.$$

Diving by n, we obtain the desired result.

$$\boxed{\hat{\mu}_{ML} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i}$$

Similar to the derivation of $\hat{\mu}_{ML}$, the ML estimate for $\hat{\mathbf{\Sigma}}_{ML}$ is given by

$$\begin{aligned}\hat{\mathbf{\Sigma}}_{ML} = \underset{\mathbf{\Sigma}}{\operatorname{argmax}}\,\ell(\mathbf{X};\mu,\mathbf{\Sigma}) &= \underset{\mathbf{\Sigma}}{\operatorname{argmin}}\left\{\frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}_i-\mu) + n\log\left((2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}\right)\right\} \\ &= \underset{\mathbf{\Sigma}}{\operatorname{argmin}}\left\{\sum_{i=1}^{n}(\mathbf{x}_i-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}_i-\mu) + n\log\left(|\mathbf{\Sigma}|\right)\right\} \qquad (15)\end{aligned}$$

Once again, we make the substitution $\mathbf{\Sigma}^{-1} \rightarrow \mathbf{A}$. As a result, the minimum (or maximum) of Equation 15 will occur where the derivative with respect to $\mathbf{A}$ equals zero.

$$\frac{\partial}{\partial\mathbf{A}}\left\{\sum_{i=1}^{n}(\mathbf{x}_i-\mu)^T\mathbf{A}(\mathbf{x}_i-\mu) + n\log\left(|\mathbf{A}^{-1}|\right)\right\} = 0$$

Note that we can substitute $|\mathbf{A}^{-1}| = |\mathbf{A}|^{-1}$ to obtain the following result [4].

$$\Rightarrow \frac{\partial}{\partial\mathbf{A}}\log\left(|\mathbf{A}|\right) = \frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial\mathbf{A}}\left[(\mathbf{x}_i-\mu)^T\mathbf{A}(\mathbf{x}_i-\mu)\right]$$

We can now apply the following identities for the derivatives of scalar and matrix/vector forms [4].

$$\frac{\partial}{\partial\mathbf{A}}\left[(\mathbf{x}-\mu)^T\mathbf{A}(\mathbf{x}-\mu)\right] = (\mathbf{x}-\mu)(\mathbf{x}-\mu)^T \qquad \frac{\partial}{\partial\mathbf{A}}\log\left(|\mathbf{A}|\right) = \mathbf{A}^{-1} = \mathbf{\Sigma}$$

Applying these identities to the previous equation, we find the derived result.

$$\boxed{\hat{\mathbf{\Sigma}}_{ML} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i-\hat{\mu}_{ML})(\mathbf{x}_i-\hat{\mu}_{ML})^T}$$

(QED)

3 Linear Discriminant Analysis

Problem 9

This problem will examine an extension of linear discriminant analysis (LDA) to multiple classes (assuming equal covariance matrices such that $\mathbf{\Sigma}_c = \mathbf{\Sigma}$ for every class c). Show that the optimal decision rule is based on calculating a set of C linear discriminant functions

$$\delta_c(\mathbf{x}) = \mathbf{x}^T\mathbf{\Sigma}^{-1}\mu_c - \frac{1}{2}\mu_c^T\mathbf{\Sigma}^{-1}\mu_c$$

and selecting $C^* = \text{argmax}_c\, \delta_c(\mathbf{x})$. Assume that the prior probability is uniform for each class such that $P_c = p(y = c) = 1/C$. Apply this method to the data in `apple_lda.mat` and plot the resulting linear decision boundaries or, alternatively, the decision regions. Report the classification error.

Recall from class on 9/20/06 and [3] that the *Bayes Classifier* minimizes the conditional risk, for a given class-conditional density $p_c(\mathbf{x}) = p(\mathbf{x}|y = c)$ and prior probability $P_c = p(y = c)$, such that

$$C^* = \underset{c}{\text{argmax}}\, \delta_c(\mathbf{x}), \text{ for } \delta_c(\mathbf{x}) \triangleq \log p_c(\mathbf{x}) + \log P_c.$$

For this problem we can eliminate the class prior term to obtain $\delta_c(\mathbf{x}) = \log p_c(\mathbf{x})$, since the class prior is identical for all classes. In addition, we'll assume that the class-conditionals are multivariate Gaussians with identical covariance matrices such that

$$p_c(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_c)^T\mathbf{\Sigma}^{-1}(\mathbf{x} - \mu_c)\right).$$

Substituting into the previous expression we find

$$\begin{aligned}
\delta_c(\mathbf{x}) = \log p_c(\mathbf{x}) &= -\frac{1}{2}(\mathbf{x} - \mu_c)^T\mathbf{\Sigma}^{-1}(\mathbf{x} - \mu_c) - \frac{d}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{\Sigma}| \\
&= -\frac{1}{2}(\mathbf{x} - \mu_c)^T\mathbf{\Sigma}^{-1}(\mathbf{x} - \mu_c) \\
&= -\frac{1}{2}\mathbf{x}^T\mathbf{\Sigma}^{-1}\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{\Sigma}^{-1}\mu_c + \frac{1}{2}\mu_c^T\mathbf{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\mu_c^T\mathbf{\Sigma}^{-1}\mu_c \\
&= \frac{1}{2}\mathbf{x}^T\mathbf{\Sigma}^{-1}\mu_c + \frac{1}{2}\mu_c^T\mathbf{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\mu_c^T\mathbf{\Sigma}^{-1}\mu_c \qquad (16)
\end{aligned}$$

where terms that are independent of c have been eliminated. To complete our proof, note that $\mu_c^T\mathbf{\Sigma}^{-1}\mathbf{x} = (\mu_c^T\mathbf{\Sigma}^{-1}\mathbf{x})^T = \mathbf{x}^T(\mathbf{\Sigma}^{-1})^T\mu_c = \mathbf{x}^T\mathbf{\Sigma}^{-1}\mu_c$, since $\mathbf{\Sigma}$ and $\mathbf{\Sigma}^{-1}$ are symmetric matrices and, as a result, $\mathbf{\Sigma}^{-1} = (\mathbf{\Sigma}^{-1})^T$. Applying this observation to Equation 16, we prove the desired result.

$$\boxed{\delta_c(\mathbf{x}) = \mathbf{x}^T\mathbf{\Sigma}^{-1}\mu_c - \frac{1}{2}\mu_c^T\mathbf{\Sigma}^{-1}\mu_c}$$

Using `prob9.m`, this method was applied to the data in `apple_lda.mat`. The means and covariance were obtained using the ML estimators derived in Problem 8; for a given class c, the mean was calculated as

$$\hat{\mu}_c = \frac{1}{n_c}\sum_{i=1}^{n_c}\mathbf{x}_{ci}.$$

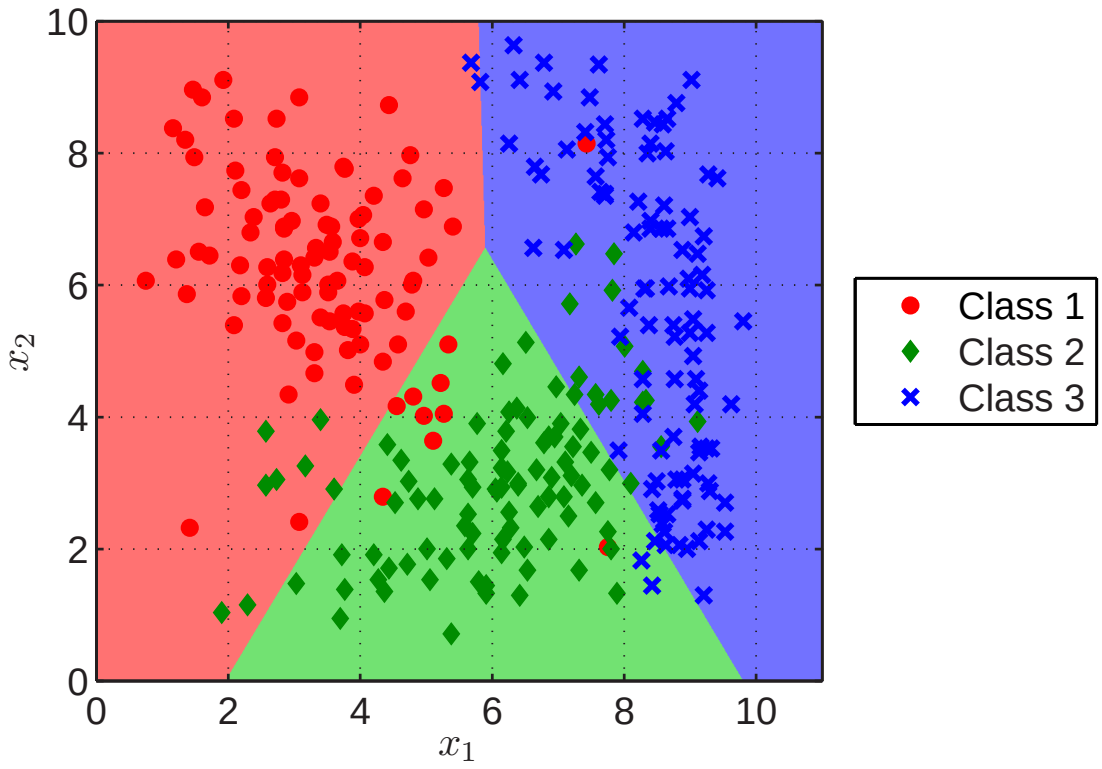


Figure 4: Optimal linear decision boundaries/regions for `apple_lda.mat`.

Similarly, the the covariance matrix $\mathbf{\Sigma}$ was calculated as

$$\hat{\mathbf{\Sigma}} = \frac{1}{n}\sum_{c}\sum_{i=1}^{n_c}(\mathbf{x}_{ci} - \hat{\mu}_c)(\mathbf{x}_{ci} - \hat{\mu}_c)^T.$$

The resulting decision boundaries are shown in Figure 4. Note that $\delta_c(\mathbf{x})$ is linear in $\mathbf{x}$, so we only obtain linear decision boundaries. The classification error (measured as the percent of incorrect classifications on the training data) was $\approx$**12.7%**.

Problem 10

Assume that the covariances $\mathbf{\Sigma}_c$ are no longer required to be equal. Derive the discriminant function and apply the resulting decision rule to `apple_lda.mat`. Plot the decision boundaries/regions and report the classification error. Compare the performance of the two classifiers.

As in Problem 9, we begin by writing the Bayes Classifier

$$C^* = \underset{c}{\operatorname{argmax}}\, \delta_c(\mathbf{x}), \text{ for } \delta_c(\mathbf{x}) = \log p_c(\mathbf{x}) + \log P_c = \log p_c(\mathbf{x}),$$

since $P_c = 1/C$ is independent of c. The class-conditionals are multivariate Gaussians given by

$$p_c(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}_c|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_c)^T\mathbf{\Sigma}_c^{-1}(\mathbf{x}-\mu_c)\right).$$

Substituting into the previous expression we find

$$\begin{aligned}\delta_c(\mathbf{x}) = \log p_c(\mathbf{x}) &= -\frac{1}{2}(\mathbf{x}-\mu_c)^T\mathbf{\Sigma}_c^{-1}(\mathbf{x}-\mu_c) - \frac{d}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{\Sigma}_c| \\ &= -\frac{1}{2}(\mathbf{x}-\mu_c)^T\mathbf{\Sigma}_c^{-1}(\mathbf{x}-\mu_c) - \frac{1}{2}\log|\mathbf{\Sigma}_c| \\ &= -\frac{1}{2}\mathbf{x}^T\mathbf{\Sigma}_c^{-1}\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{\Sigma}_c^{-1}\mu_c + \frac{1}{2}\mu_c^T\mathbf{\Sigma}_c^{-1}\mathbf{x} - \frac{1}{2}\mu_c^T\mathbf{\Sigma}_c^{-1}\mu_c - \frac{1}{2}\log|\mathbf{\Sigma}_c|\end{aligned}$$

where terms that are independent of c have been eliminated. Once again, we can simplify this

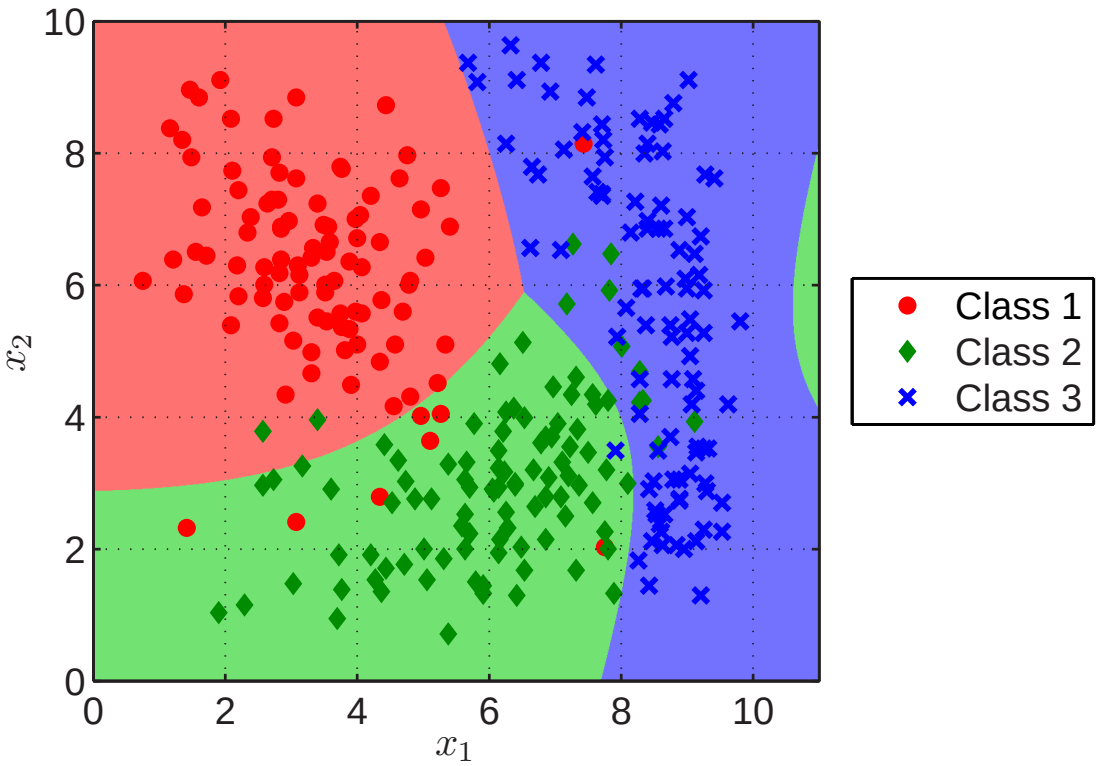


Figure 5: Non-linear decision boundaries/regions for `apple_lda.mat`.

expression since $\mu_c^T \mathbf{\Sigma}_c^{-1}\mathbf{x} = \mathbf{x}^T \mathbf{\Sigma}_c^{-1}\mu_c$. In conclusion, the discriminant function is given as follows.

$$\boxed{\delta_c(\mathbf{x}) = -\frac{1}{2}\mathbf{x}^T \mathbf{\Sigma}_c^{-1}\mathbf{x} + \mathbf{x}^T \mathbf{\Sigma}_c^{-1}\mu_c - \frac{1}{2}\mu_c^T \mathbf{\Sigma}_c^{-1}\mu_c - \frac{1}{2}\log|\mathbf{\Sigma}_c|}$$

Using `prob10.m`, this discriminant function was applied to the data in `apple_lda.mat`. The means and covariances were obtained using the ML estimators derived in Problem 8. The resulting decision boundaries are shown in Figure 5. The classification error (measured as the percent of incorrect classifications on the training data) was **7%**.

Note that the decision functions $\delta_c(\mathbf{x})$ are no longer linear in $\mathbf{x}$, so now we can obtain non-linear decision boundaries. In fact, the leading-order term $-\frac{1}{2}\mathbf{x}^T \mathbf{\Sigma}_c^{-1}\mathbf{x}$ is a quadratic form which, together with the lower-order terms, can be used to express any second-degree curve (i.e., conic sections in two dimensions) of the form

$$ax_1^2 + bx_1x_2 + cx_2^2 + dx_1 + fx_2 + g = 0$$

as the decision boundary separating classes. In contrast to the linear decision boundaries in Problem 9, by allowing different covariance matrices for each class, we can now achieve a variety of decision boundaries such as lines, parabolas, hyperbolas, and ellipses [5]. As shown in Figure 5, the resulting decision boundaries are curvilinear and, as a result, can achieve a lower classification error due to the increase in their degrees of freedom.

Problem 11

An alternative approach to learning quadratic decision boundaries is to map the inputs into an extended polynomial feature space. Implement this method and compare the resulting decision boundaries to those in Problem 10.

As discussed in class on 9/15/06, we can utilize the linear regression framework to implement a naïve classifier as follows. First, we form an *indicator matrix* $\mathbf{Y}$ such that

$$\mathbf{Y}_{ij} = \begin{cases} 1 & \text{if } y_i = c, \\ 0 & \text{otherwise} \end{cases}$$

where the possible classes are labeled as $1, \ldots, C$. Recall from Problem 10 that any quadratic curve can be written as

$$ax_1^2 + bx_1x_2 + cx_2^2 + dx_1 + fx_2 + g = 0,$$

where $\{a, b, c, d, f, g\}$ are unknown regression coefficients. As a result, we can define a design matrix with extended polynomial features as follows.

$$\mathbf{X} = \begin{pmatrix} 1 & x_{2(1)} & x_{1(1)} & x_{2(1)}^2 & x_{1(1)}x_{2(1)} & x_{1(1)}^2 \\ 1 & x_{2(2)} & x_{1(2)} & x_{2(2)}^2 & x_{1(2)}x_{2(2)} & x_{1(2)}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{2(N)} & x_{1(N)} & x_{2(N)}^2 & x_{1(N)}x_{2(N)} & x_{1(N)}^2 \end{pmatrix}$$

where $x_{i(j)}$ denotes the i^{th} coordinate of the j^{th} training sample. (Note that inclusion of the cross

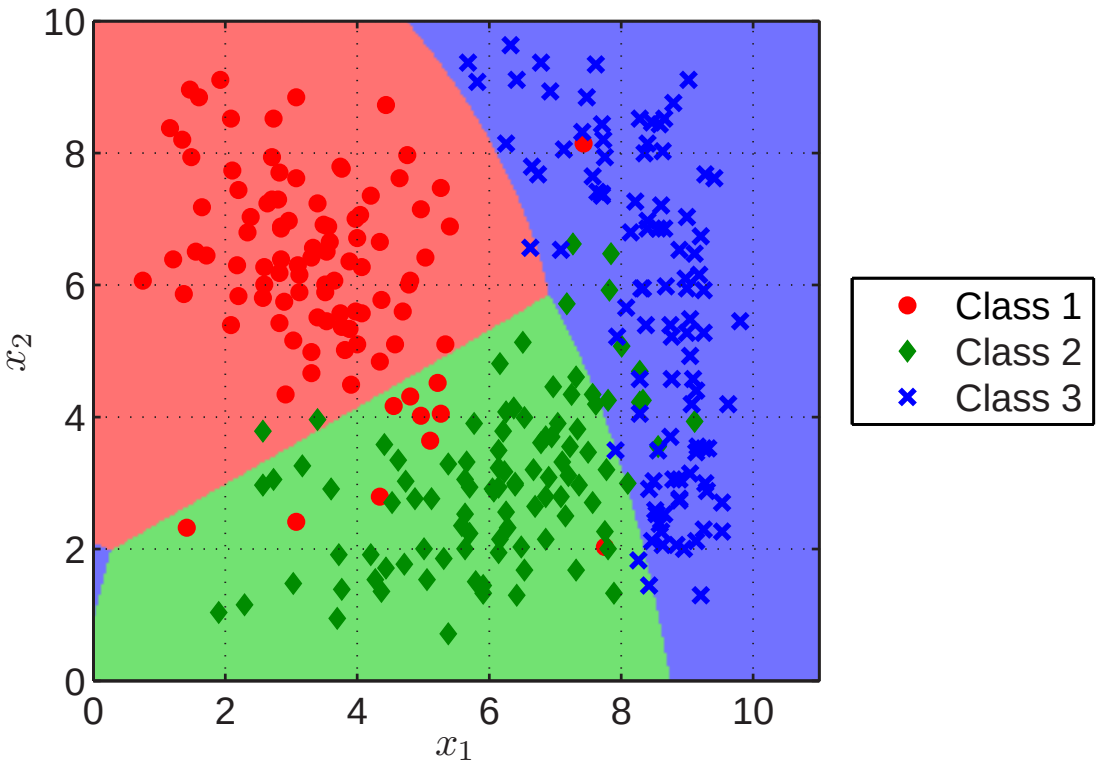
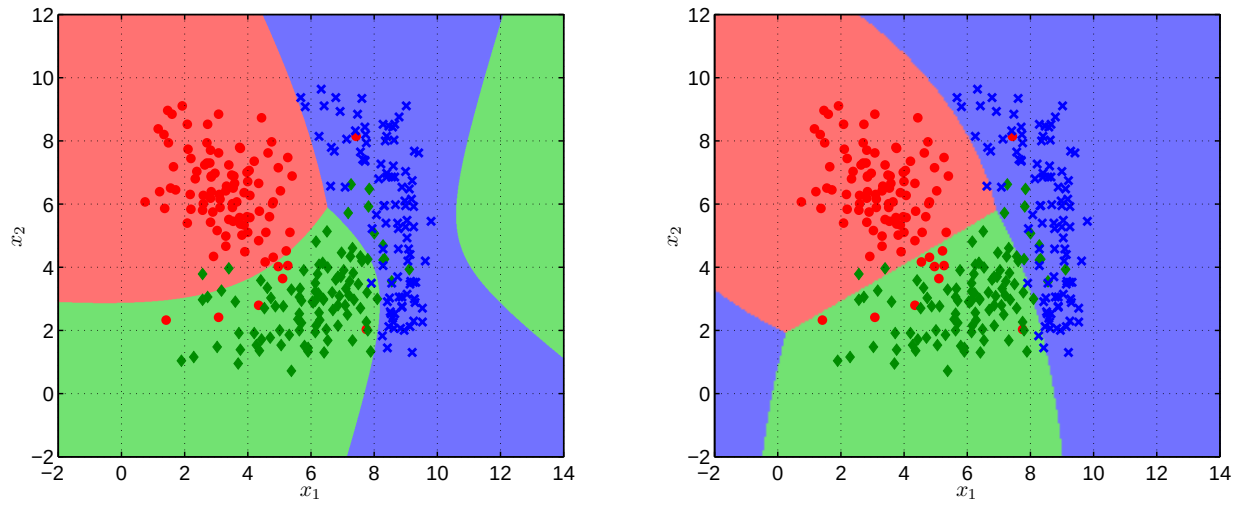


Figure 6: Quadratic decision boundaries/regions using extended polynomial features.



(a) Decision boundaries using extended LDA

(b) Decision boundaries using extended features

Figure 7: Comparison of classification results in Problems 10 and 11.

term x_1x_2 had little effect on the decision regions.) Together, $\mathbf{X}$ and $\mathbf{Y}$ define C independent linear regression problems with least squares solutions given by

$$\hat{\mathbf{W}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y},$$

where the regression coefficients for each class correspond to the columns of $\hat{\mathbf{W}}$. For a general set of samples $\mathbf{x}$, we form the extended design matrix $\mathbf{X}_0$ with the corresponding linear predictions given by

$$\hat{\mathbf{Y}}_0 = \mathbf{X}_0(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} = \mathbf{X}_0\hat{\mathbf{W}},$$

as shown in class. Finally, we assign a class label $\hat{\mathbf{C}}$ by choosing the column of $\hat{\mathbf{Y}}_0$ with the largest value (independently for each row/sample).

Using `prob11.m`, this classification procedure was applied to the data in `apple_lda.mat`. The resulting decision boundaries are shown in Figure 6. The classification error (measured as the percent of incorrect classifications on the training data) was $\approx$**9.3%**. Since the regression features were extended to allow quadratic decision boundaries, we find that the resulting regions are delineated by curvilinear boarders. While the classification error is lower than in Problem 9, we find that the resulting decision regions possess some odd/undesirable properties.

Compare the decision boundaries/regions in Figures 7(a) and 7(b). Using extended linear discriminant analysis (LDA) as in Problem 10, we find that the decision boundaries are quadratic and, assuming underlying Gaussian distributions, seem to make reasonable generalizations. More specifically, examine the decision region for the second class (green) in Figure 7(a). Not surprisingly, it continues on the other side of Class 3. This is reasonable since Class 3 varies in the opposite direction of Class 2.

Unfortunately, this generalization behavior is not reflected in the decision regions found in this problem. Surprisingly, we find that Class 3 becomes the most likely class label in the bottom left corner of Figure 7(b). This is inconsistent with the covariance matrix of the underlying distribution (assuming it is truely Gaussian). In conclusion, we find that the results in Problems 10 and 11 illustrate the benefits of generative models over naïve classification schemes (when there is some knowledge of the underlying distributions).

References

[1] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1996.

[2] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006.

[3] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (Second Edition)*. Wiley-Interscience, 2000.

[4] Sam Roweis. Matrix identities. `http://www.cs.toronto.edu/~roweis/notes/matrixid.pdf`.

[5] Eric W. Weisstein. Quadratic curve. `http://mathworld.wolfram.com/QuadraticCurve.html`.

[6] Eric W. Weisstein. Quadratic curve discriminant. `http://mathworld.wolfram.com/QuadraticCurveDiscriminant.html`.