CS 195-5: Machine Learning
Problem Set 3

Douglas Lanman
dlanman@brown.edu
25 October 2006

1 Regularization

Problem 1

Show that the solution for the ridge regression problem

$$\hat{\mathbf{w}}_{ridge} = \underset{\mathbf{w}}{\operatorname{argmax}} -\sum_{i=1}^{N}(y_i - \mathbf{w}^T\mathbf{x}_i)^2 - \lambda\sum_{j=0}^{d} w_j^2 \tag{1}$$

is given by $\hat{\mathbf{w}}_{ridge} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$.

This proof is conceptually similar to the one presented in class on 9/8/06 for unregularized linear least-squares regression in d dimensions. Let us begin by defining the following quantities:

$$\mathbf{X} = \begin{pmatrix} 1 & x_1^{(1)} & \cdots & x_d^{(1)} \\ 1 & x_1^{(2)} & \cdots & x_d^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(N)} & \cdots & x_d^{(N)} \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}, \quad \mathbf{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{pmatrix},$$

where $\mathbf{X}$ is the design matrix and $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ and $\mathbf{y}$ are the observed points and their associated labels, respectively. By inspection, Equation 1 can be rewritten as

$$\hat{\mathbf{w}}_{ridge} = \underset{\mathbf{w}}{\operatorname{argmax}} -(\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w}) - \lambda\mathbf{w}^T\mathbf{w}.$$

Recall the following matrix identities: $(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$ and $(\mathbf{A}+\mathbf{B})^T = \mathbf{A}^T + \mathbf{B}^T$. Applying these expressions to the previous equation gives the following result.

$$\hat{\mathbf{w}}_{ridge} = \underset{\mathbf{w}}{\operatorname{argmax}} -\mathbf{y}^T\mathbf{y} + \mathbf{y}^T\mathbf{X}\mathbf{w} + \mathbf{w}^T\mathbf{X}^T\mathbf{y} - \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} - \lambda\mathbf{w}^T\mathbf{w}$$

Note that the maximum (or minimum) of this expression must occur where the derivative with respect to $\mathbf{w}$ equals zero. Equating the first partial derivative with zero, we find

$$\frac{\partial}{\partial\mathbf{w}}\left\{-\mathbf{y}^T\mathbf{y} + \mathbf{y}^T\mathbf{X}\mathbf{w} + \mathbf{w}^T\mathbf{X}^T\mathbf{y} - \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} - \lambda\mathbf{w}^T\mathbf{w}\right\} = 0$$

$$\Rightarrow -\frac{\partial(\mathbf{y}^T\mathbf{y})}{\partial\mathbf{w}} + \frac{\partial(\mathbf{y}^T\mathbf{X}\mathbf{w})}{\partial\mathbf{w}} + \frac{\partial(\mathbf{w}^T\mathbf{X}^T\mathbf{y})}{\partial\mathbf{w}} - \frac{\partial(\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w})}{\partial\mathbf{w}} - \lambda\frac{\partial(\mathbf{w}^T\mathbf{w})}{\partial\mathbf{w}} = 0.$$

Note that the first term is independent of $\mathbf{w}$ and can be eliminated to obtain the following equation.

$$\frac{\partial(\mathbf{y}^T\mathbf{X}\mathbf{w})}{\partial\mathbf{w}} + \frac{\partial(\mathbf{w}^T\mathbf{X}^T\mathbf{y})}{\partial\mathbf{w}} = \frac{\partial(\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w})}{\partial\mathbf{w}} + \lambda\frac{\partial(\mathbf{w}^T\mathbf{w})}{\partial\mathbf{w}}$$

Also note that $\mathbf{y}^T\mathbf{X}\mathbf{w}$ is a scalar quantity, so we must have $\mathbf{y}^T\mathbf{X}\mathbf{w} = (\mathbf{y}^T\mathbf{X}\mathbf{w})^T = \mathbf{w}^T\mathbf{X}^T\mathbf{y}$. Applying this identity to the previous equation leads to the following expression.

$$2\frac{\partial(\mathbf{w}^T\mathbf{X}^\mathbf{T})}{\partial\mathbf{w}}\mathbf{y} = \frac{\partial(\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w})}{\partial\mathbf{w}} + \lambda\frac{\partial(\mathbf{w}^T\mathbf{w})}{\partial\mathbf{w}} \tag{2}$$

Recall the following identities for the derivatives of scalar and matrix/vector forms [4].

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

From the first identity, we have

$$\frac{\partial(\mathbf{w}^T\mathbf{X}^\mathbf{T})}{\partial\mathbf{w}} = \mathbf{X}^T. \tag{3}$$

Similarly, the second identity implies

$$\frac{\partial(\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w})}{\partial\mathbf{w}} = \left(\mathbf{X}^T\mathbf{X} + (\mathbf{X}^T\mathbf{X})^T\right)\mathbf{w} = 2\mathbf{X}^T\mathbf{X}\mathbf{w}. \tag{4}$$

Finally, from the third identity and application of the chain rule, we have

$$\frac{\partial(\mathbf{w}^T\mathbf{w})}{\partial\mathbf{w}} = 2\mathbf{w}. \tag{5}$$

Substituting Equations 3, 4, and 5 into Equation 2 gives the following expression.

$$2\mathbf{X}^T\mathbf{y} = 2\mathbf{X}^T\mathbf{X}\mathbf{w} + 2\lambda\mathbf{w} = 2(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})\mathbf{w}$$

$$\Rightarrow \mathbf{X}^T\mathbf{y} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})\mathbf{w}.$$

Assuming the inverse of the square matrix $\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}$ exists, we can solve for $\mathbf{w}$ in the previous equation to yield the desired expression for the solution of the ridge regression problem.

$$\boxed{\hat{\mathbf{w}}_{ridge} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}}$$

(QED)

Problem 2

In this problem we will examine the effect of L_2 regularization on the bias-variance tradeoff in regression. Using `genData.m` draw samples from $h(x) = \sin(x) + \nu$ for $\nu \sim \mathcal{N}(\nu; 0, 0.1)$. Modify `regularizationCode.m` to implement ridge regression. Discuss the bias-variance tradeoff.

Let us begin by summarizing the modifications made to `regularizationCode.m`. Lines 12-18 define the simulation parameters. Lines 20-87 apply 10^{th}-order polynomial ridge regression to $L = 200$ test sets with $N = 20$ points in each. Finally, on lines 89-100, the bias, variance, and test error are plotted. Note that line 37 applies ridge regression, as defined in Problem 1, to test set l to obtain the predictions $y^{(l)}(x_n)$ at each point x_n. In addition, the following definitions are used on lines 45-53 to define the bias, variance, and test error [2].

$$\bar{y}(x) = \frac{1}{L}\sum_{l=1}^{L} y^{(l)}(x)$$

$$(\text{bias})^2 = \frac{1}{N}\sum_{n=1}^{N}\{\bar{y}(x_n) - h(x_n)\}^2 \tag{6}$$

$$\text{variance} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{L}\sum_{l=1}^{L}\{y^{(l)}(x_n) - \bar{y}(x_n)\}^2 \tag{7}$$

Note that the prediction errors are typically specified as $e^{(l)}(x) = y^{(l)}(x) - h(x)$, however for this problem I will considered the test error to be given by the squared loss function as follows.

$$\text{test error} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{L}\sum_{l=1}^{L}\{y^{(l)}(x_n) - h(x_n)\}^2 \tag{8}$$

The resulting plots of $(\text{bias})^2$, variance, $(\text{bias})^2 + \text{variance}$, and test error are shown in Figure 1.

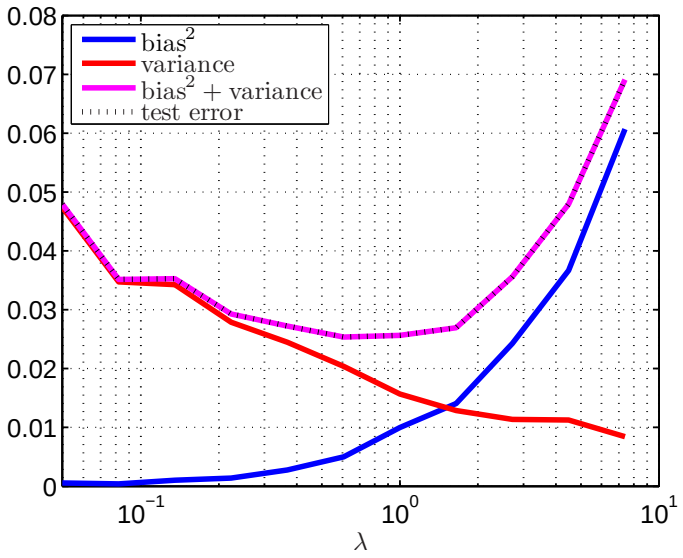
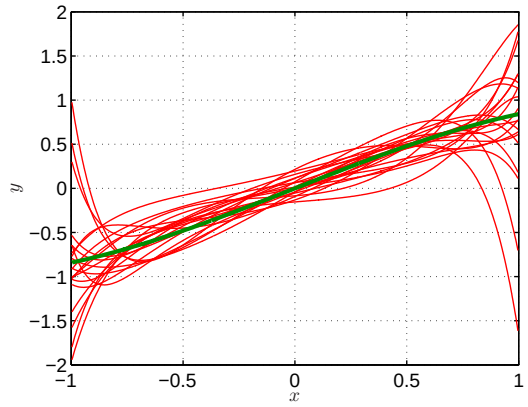
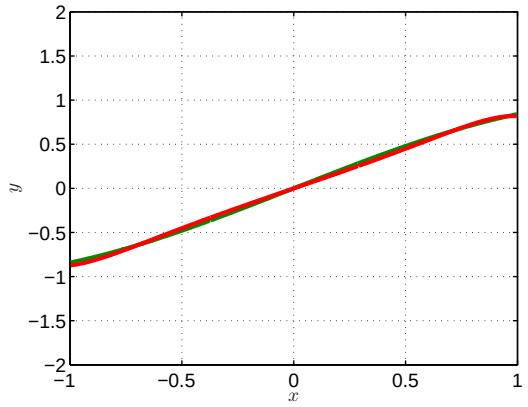


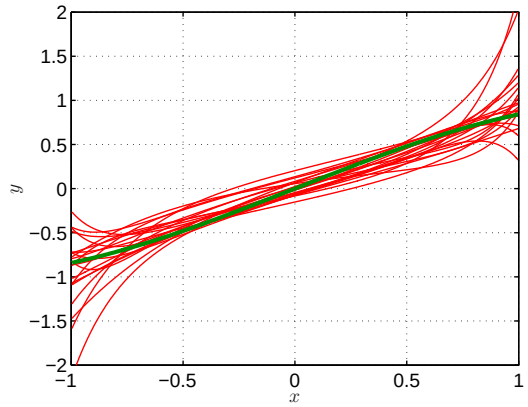
Figure 1: Comparison of squared bias, variance, their sum, and the test error.



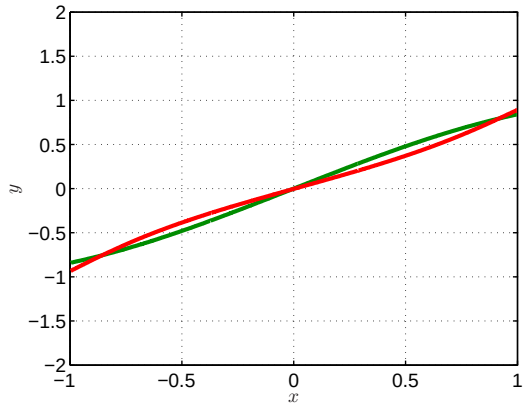
(a) $\lambda = 0.05$



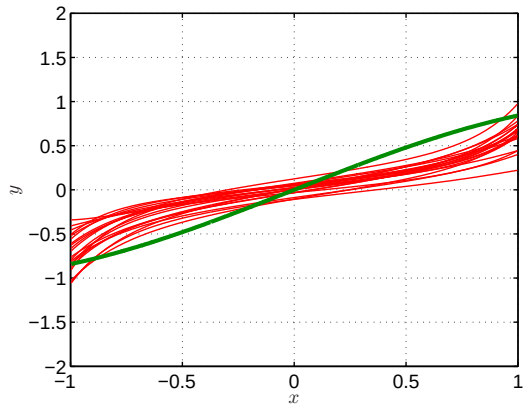
(b) $\lambda = 0.05$



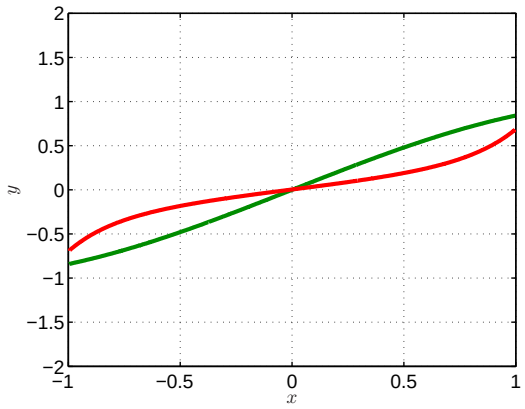
(c) $\lambda = 0.61$



(d) $\lambda = 0.61$



(e) $\lambda = 7.40$



(f) $\lambda = 7.40$

Figure 2: Illustration of the dependence of bias and variance on the ridge regression parameter λ. The left column shows the results of applying 10^{th}-order polynomial ridge regression to 20 random trials (with 20 points in each) drawn from the distribution defined in Problem 2. The right column compares the average over 200 random tests sets (in red) to the true underlying function (in green).

Several trends are apparent from an examination of Figure 1. First, we note that the regularization parameter λ effectively controls the bias-variance tradeoff. That is, for a small value of λ, the estimated model tends to have low bias, but high variance. For large values of λ, the estimated model tends to have low variance, but high bias. Note that these observations are consistent with the individual and average models shown in Figure 2.

From these observations it is apparent that we would like to select a value of the regularization parameter that leads to an effective bias-variance tradeoff and, ultimately, a low test error. In order to investigate this issue quantitatively, we can apply the decomposition of the expected loss derived in class on 9/25/06. Following the derivation in Chapter 3.2 of [2], we can express the test error $\{y(\mathbf{x};\mathcal{D}) - h(\mathbf{x})\}^2$ on a data set $\mathcal{D}$ as

$$\begin{aligned}\{y(\mathbf{x};\mathcal{D}) - h(\mathbf{x})\}^2 &= \{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] + \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2 \\ &= \{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2 + \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2 + \\ &\quad 2\{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\},\end{aligned}$$

where the expectation over the ensemble of data sets $\mathcal{D}$ is denoted by $\mathbb{E}_{\mathcal{D}}$. Taking the expectation of this expression with respect to $\mathcal{D}$ yields the following equation for the expected loss.

$$\underbrace{\mathbb{E}_{\mathcal{D}}[\{y(\mathbf{x};\mathcal{D}) - h(\mathbf{x})\}^2]}_{\text{expected loss}} = \underbrace{\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2}_{(\text{bias})^2} + \underbrace{\mathbb{E}_{\mathcal{D}}[\{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2]}_{\text{variance}} \tag{9}$$

We can approximate Equation 9 using Equations 6, 7, and 8 to obtain the following expression.

$$\underbrace{\frac{1}{N}\sum_{n=1}^{N}\frac{1}{L}\sum_{l=1}^{L}\{y^{(l)}(x_n) - h(x_n)\}^2}_{\text{test error}} = \underbrace{\frac{1}{N}\sum_{n=1}^{N}\{\bar{y}(x_n) - h(x_n)\}^2}_{(\text{bias})^2} + \underbrace{\frac{1}{N}\sum_{n=1}^{N}\frac{1}{L}\sum_{l=1}^{L}\{y^{(l)}(x_n) - \bar{y}(x_n)\}^2}_{\text{variance}}$$

Recall that, according to Equation 8, the test error is defined to be the squared loss; as a result, we obtain the familiar decomposition of the expected loss (i.e., test error) as the sum of the squared bias and variance. This result is confirmed by the plots in Figure 1. Specifically, we find that the plot of the test error is identical to the plot of $(\text{bias})^2 + \text{variance}$.

In conclusion, we find that the regularization parameter $\lambda = 0.61$ that achieves a minimum value of $(\text{bias})^2 + \text{variance}$ corresponds to the value that leads to a minimum error on the test set. This example illustrates model selection from a frequentist perspective; that is, the minimum error for a test set can be achieved with the optimal bias-variance tradeoff given by the value of the regularization parameter minimizing $(\text{bias})^2 + \text{variance}$. As discussed on page 152 in [2], this interpretation is instructive, however in practical circumstances a single data set will be available for training – preventing a direct determination of the optimal value of the regularization parameter. As a result, we are left with the heuristic: λ should neither be too small nor too large.

Problem 3

In this problem we will investigate the effect of regularization on classifiers. Specifically, we will consider the two-dimensional quadratic logistic regression problem, i.e.

$$p(1|\mathbf{x}) = 1/\left(1 + \exp(-w_0 - w_1 x_1 - w_2 x_2 - w_3 x_1^2 - w_4 x_2^2)\right).$$

Modify your logistic regression code to implement the following regularized objective.

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmax}} \sum_{i=1}^{N} \log p(y_i|\mathbf{x}_i; \mathbf{w}) - \lambda(w_3^2 + w_4^2). \tag{10}$$

Plot the decision boundaries obtained for the data in `lrDataApricot.mat` with $\lambda = \{0, 5, 20, 100, 500\}$. Discuss the effect of regularization on the geometry of the decision boundary.

Let's begin our discussion by reviewing the modifications made to `logisticRegression.m`. First, note that the modified objective function is implemented on line 46. In addition, the modified Hessian $\mathbf{H}$ is given by the expression on line 38. Finally, we note that the gradient expression on line 34 (and in `gradient.m`) was modified to add the correct penalty-related term. We can derive this term in a manner similar to that presented in class on 9/29/06. First, note that the log-likelihood (under the regularized logistic regression objective) can be expressed as follows.

$$\ell(X_N; \mathbf{w}) = \sum_{i=1}^{N} \log p(y_i|\mathbf{x}_i; \mathbf{w}) - \lambda(w_3^2 + w_4^2) \tag{11}$$

Recall that the update rule for the Newton-Raphson algorithm is given by

$$\mathbf{w}_{new} = \mathbf{w}_{old} + \mathbf{H}^{-1} \frac{\partial}{\partial \mathbf{w}} \ell(X_N; \mathbf{w}).$$

From Equation 11, the gradient of the log-likelihood $\ell(X_N; \mathbf{w})$ is given the following expression.

$$\frac{\partial}{\partial \mathbf{w}} \ell(X_N; \mathbf{w}) = \frac{\partial}{\partial \mathbf{w}} \sum_{i=1}^{N} \log p(y_i|\mathbf{x}_i; \mathbf{w}) - \lambda \frac{\partial}{\partial \mathbf{w}} (w_3^2 + w_4^2)$$

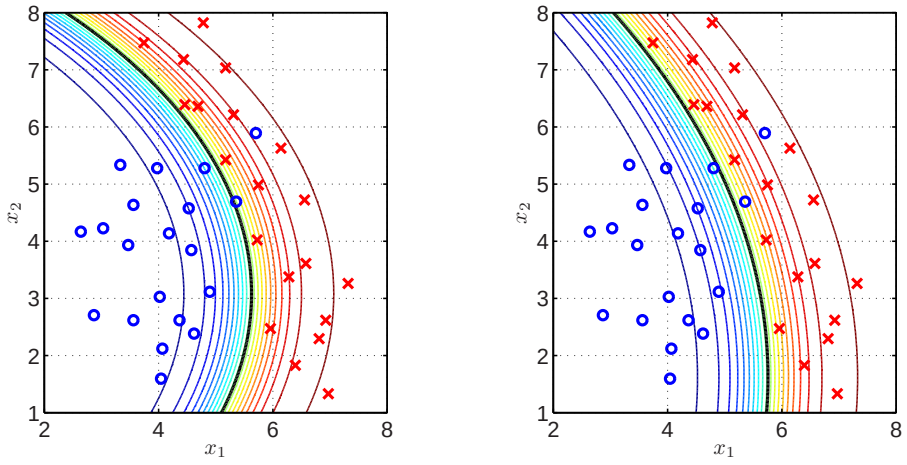
Recall from class on 10/2/06 that the gradient of the first term is given by $\mathbf{X}^T(\mathbf{y} - \sigma(\mathbf{X}\mathbf{w}))$, where $\mathbf{X}$ is the design matrix. The gradient of the regularization term can be computed directly as follows.

$$\frac{\partial}{\partial \mathbf{w}} (w_3^2 + w_4^2) = \left[0, 0, 0, \frac{\partial\, w_3^2}{\partial w_3}, \frac{\partial\, w_4^2}{\partial w_4}\right]^T = [0, 0, 0, 2w_3, 2w_4]^T$$

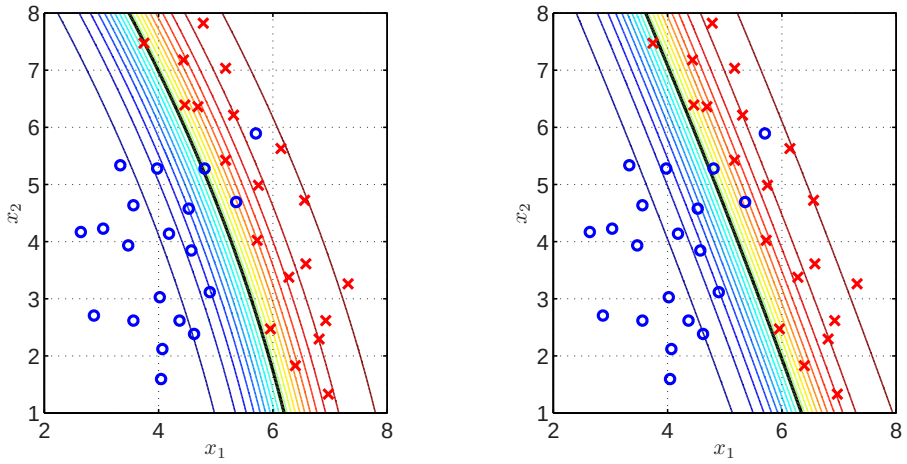
Substituting this result into the previous equation gives the desired expression for the gradient.

$$\boxed{\frac{\partial}{\partial \mathbf{w}} \ell(X_N; \mathbf{w}) = \mathbf{X}^T(\mathbf{y} - \sigma(\mathbf{X}\mathbf{w})) - 2\lambda[0, 0, 0, w_3, w_4]^T} \tag{12}$$

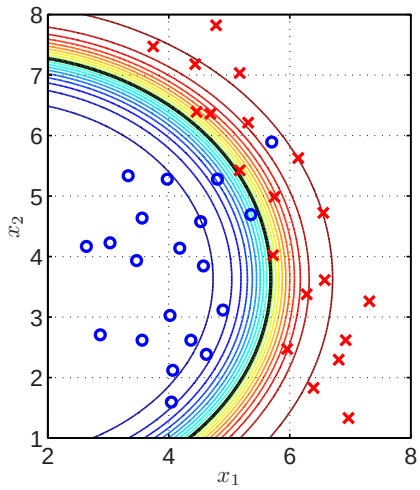
The decision boundaries shown in Figure 3 were plotted using `prob3.m`. From these plots it is apparent that **increasing λ (i.e., the degree of regularization) causes the decision boundary to become nearly linear**. That is, for $\lambda = 0$ there is no regularization and the decision boundary is a circular arc, as shown in Figure 3(e). As we increase the value of λ in Equation 12, we are effectively preventing the quadratic terms $\{w_3, w_4\}$ from being very large. As a result, the quadratic decision boundary is constrained to a set of linear coefficients. This effect is apparent in the near-linear decision boundary resulting from $\lambda = 500$ (i.e., a relatively large value of the regularization parameter), as shown in Figure 3(d).



(a) $\lambda = 5$ (b) $\lambda = 20$



(c) $\lambda = 100$ (d) $\lambda = 500$



(e) $\lambda = 0$

Figure 3: Decision boundaries for the regularized logistic classifier trained on `lrDataApricot.mat`. The decision boundary, for each value of the regularization parameter λ, is shown as a solid black line. The contour lines denote level sets of the logistic function evaluated using the optimal regression parameters $\hat{\mathbf{w}}$, with red and blue lines indicated values close to 1 and 0, respectively.

2 Support Vector Machines

Problem 4

Let $\{\mathbf{x}_i, y_i\}$, $i = 1, \ldots, N$ be the training samples and their labels, and let α_i, $i = 1, \ldots, N$ be the Lagrange multipliers found by solving the Support Vector Machine (SVM) optimization problem. Recall that the SVM classifier is given by

$$\hat{y}(\mathbf{x}) = \text{sign}\left(w_0 + \sum_{\alpha_i > 0} \alpha_i y_i \mathbf{x}_i^T \mathbf{x}\right). \tag{13}$$

Write down the expression for the optimal value of w_0.

In the following analysis we limit ourselves to the case where the training samples are linearly-separable. Recall from the lecture on 10/11/06 that, since the samples are linearly-separable, we can always find $\mathbf{w}$ such that

$$y_i(w_0 + \mathbf{w}^T \mathbf{x}_i) > 0, \forall i = 1, \ldots, N.$$

In fact, by adjusting the value of $\|\mathbf{w}\|$, we can further guarantee that

$$y_i(w_0 + \mathbf{w}^T \mathbf{x}_i) \geq 1, \forall i = 1, \ldots, N.$$

As was done in class, this gives us the freedom to select $y^*(w_0 + \mathbf{w}^T \mathbf{x}^*) = 1$ for the set of points $\{\mathbf{x}^*, y^*\}$ closest to the decision surface implicitly defined by Equation 13. Recall that the set of support vectors $\mathcal{S}$ (i.e., the training samples $\{\mathbf{x}_i, y_i\}$ for which $a_i > 0$) are the closest points equidistant to the decision surface. As a result, we have

$$y_s\left(w_0 + \mathbf{w}^T \mathbf{x}_s\right) = 1, \forall s \in \mathcal{S}$$

$$\Rightarrow y_s\left(w_0 + \sum_{i \in \mathcal{S}} \alpha_i y_i \mathbf{x}_i^T \mathbf{x}_s\right) = 1, \forall s \in \mathcal{S}$$

since, by the lecture on 10/11/06, we have $\mathbf{w} = \sum_{i \in \mathcal{S}} \alpha_i y_i \mathbf{x}_i$. Multiplying both sides of this expression by the class label y_s gives

$$y_s^2\left(w_0 + \sum_{i \in \mathcal{S}} \alpha_i y_i \mathbf{x}_i^T \mathbf{x}_s\right) = y_s \quad \Rightarrow \quad w_0 = y_s - \sum_{i \in \mathcal{S}} \alpha_i y_i \mathbf{x}_i^T \mathbf{x}_s, \tag{14}$$

since $y_s^2 = 1$. Note that Equation 14 gives a solution for w_0 for any support vector $s \in \mathcal{S}$. In order to provide a numerically robust solution for the bias term w_0, we propose averaging the estimates obtained with each support vector, such that

$$\boxed{w_0 = \frac{1}{N_{\mathcal{S}}} \sum_{s \in \mathcal{S}} \left(y_s - \sum_{i \in \mathcal{S}} \alpha_i y_i \mathbf{x}_i^T \mathbf{x}_s\right)}$$

where $N_{\mathcal{S}}$ is the number of support vectors. (Note that this derivation follows a similar approach as that presented in Chapter 7.1 in [2].)

Problem 5

If no test data is available we may estimate the risk of a classifier by computing the leave-one-out cross-validation error ϵ_N. That is,

$$\epsilon_N = \frac{1}{N}\sum_{i=1}^{N} L(\hat{y}_{-i}(\mathbf{x}_i), y_i), \tag{15}$$

where L is the zero-one loss function and $\hat{y}_{-i}(\mathbf{x})$ is the prediction of the SVM trained on all the data *except* the i^{th} sample. Suppose that, after training a SVM on a data set with N examples, we obtain m support vectors (i.e., m out of N samples have non-zero multipliers α_i). If ϵ_N is the leave-one-out estimate of SVM risk *on the same training data set*, show that

$$\epsilon_N \leq \frac{m}{N}.$$

Let's assume that the training samples are separable under the selected SVM kernel. As a result, if we train the SVM classifier on all input samples, then the classification error must satisfy

$$L(\hat{y}(\mathbf{x}_i), y_i) = 0, \forall i = 1, \ldots, N. \tag{16}$$

In other words, since the training samples are separable, the SVM classifier is error-free. At this point, we recall a key property for SVMs: the decision boundary is completely determined by the set of support vectors $\mathcal{S}$ [2]. As a result, if we remove any individual data point $i \notin \mathcal{S}$, then the resulting decision boundary will be identical to that found by training on the complete data set. From this observation and the result given in Equation 16, we can conclude that

$$L(\hat{y}_{-i}(\mathbf{x}_i), y_i) = 0, \forall i \notin \mathcal{S}. \tag{17}$$

Substituting Equation 17 into Equation 15 gives the following expression for the leave-one-out estimate of SVM risk.

$$\epsilon_N = \frac{1}{N}\sum_{i=1}^{N} L(\hat{y}_{-i}(\mathbf{x}_i), y_i) = \frac{1}{N}\sum_{s\in\mathcal{S}} L(\hat{y}_{-s}(\mathbf{x}_s), y_s) \tag{18}$$

Note that, if we remove a support vector, the SVM decision boundary may be altered. As a result, we could potentially misclassify the data point we removed. In general, we could misclassify any support vector $s \in \mathcal{S}$ such that

$$L(\hat{y}_{-s}(\mathbf{x}_s), y_s) \leq 1, \forall s \in \mathcal{S}. \tag{19}$$

Substituting Equation 19 into Equation 18 gives the desired bound for the leave-one-out estimate of SVM risk.

$$\epsilon_N = \frac{1}{N}\sum_{s\in\mathcal{S}} L(\hat{y}_{-s}(\mathbf{x}_s), y_s) \leq \frac{1}{N}\sum_{s\in\mathcal{S}} 1 = \frac{m}{N}$$

$$\therefore \boxed{\epsilon_N \leq \frac{m}{N}}$$

(QED)

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`.