6.867 Machine learning

Mid-term exam

October 15, 2003

(2 points) Your name and MIT ID:

SOLUTIONS

Problem 1

Suppose we are trying to solve an active learning problem, where the possible inputs you can select form a discrete set. Specifically, we have a set of N unlabeled documents, $\Phi_1, \ldots, \Phi_N$, where each document is represented as a binary feature fector

$$\Phi = [\phi_1, \ldots, \phi_m]^T$$

and $\phi_i = 1$ if word i appears in the document and zero otherwise. Our goal is to quickly label these N documents with 0/1 labels. We can request a label for any of the N documents, preferably as few as possible. We also have a small number n of these documents already labeled to get us started.

We use a logistic regression model to solve the classification task:

$$P(y = 1|\Phi, \mathbf{w}) = g(\,\mathbf{w}^T\Phi\,)$$

where $g(\cdot)$ is the logistic function. Note that we do not include the bias term.

1. **(T/F – 2 points)** Any word that appears in all the N documents would effectively provide a bias term for the logistic regression model.

   T

2. **(T/F – 2 points)** Any word that appears only in the available n labeled documents used for initially training the logistic regression model, would serve equally well as a bias term.

   F

3. Having trained the logistic regression model on the basis of the n labeled documents, obtaining $\hat{\mathbf{w}}_n$, we'd like to request additional labeled documents. For this, we will use the following measure of uncertainty in our predictions:

$$E_{y\sim p_t}|y - p_t| = p_t|1 - p_t| + (1 - p_t)|0 - p_t| = 2p_t(1 - p_t)$$

where $p_t = P(y = 1|\Phi_t, \hat{\mathbf{w}}_n)$, our current prediction of the probability that $y = 1$ for the t^{th} unlabeled document Φ_t.

a) **(4 points)** We would request the label for the document/query point Φ_t that has

(   ) the smallest value of $2p_t(1 - p_t)$

(X) the largest value of $2p_t(1 - p_t)$

(   ) an intermediate value of $2p_t(1 - p_t)$

Briefly explain the rationale behind the selection criterion that you chose.

$2p_t(1-p_t)$ *is a measure of uncertainty about the label for document* t. *We expect to be able to reduce the uncertainty the most by requesting labels for those documents with the largest value of* $2p_t(1-p_t)$, *points that are closest to the decision boundary. Put another way, getting additional training examples that are close to the boundary ought to help the most in terms of figuring out exactly where the boundary should lie.*

b) **(2 points)** Sketch $\hat{\mathbf{w}}_n$ in Figure 1.1. Write down the equation, expressed solely in terms of Φ and $\hat{\mathbf{w}}_n$, that Φ has to satisfy for it to lie exactly on the decision boundary:

Points Φ *that lie on the decision boundary must satisfy* $\hat{\mathbf{w}}_n^T\Phi = 0$. *Hence,* $\hat{\mathbf{w}}_n^T$ *is orthogonal to the decision boundary. Moreover, since we decide* $y = 1$ *when* $\hat{\mathbf{w}}_n^T\Phi > 0$, $\hat{\mathbf{w}}_n^T$ *points into the '+' region.*

c) **(4 points)** In figure 1.2, circle the next point we would select according to the criterion. Draw two decision boundaries that would result from incorporating the new point in the training set, labeling the boundaries as $y = 1$ and $y = 0$, depending on the outcome of the query.

The uncertainty is largest when $P(y = 1|\Phi) = g(\hat{\mathbf{w}}_n^T \Phi)$ is near one-half, e.g. when $\hat{\mathbf{w}}_n^T \Phi$ is near zero so that the point Φ is close to the decision boundary. Hence, we circle the point nearest the boundary. If this point (initially in the $y = 0$ region) turns out to be a $y = 1$ document, then we "tilt" the boundary towards that point so as to tend to move that example to the other side of the decision boundary. Otherwise, if the document is in fact a $y = 0$, then we tilt the boundary away from that point so that it is "deeper" in the $y = 0$ region.

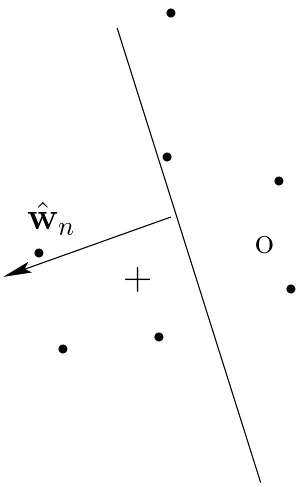

Figure 1.1. Two labeled points, unlabeled points, and the decision boundary. The point "+" corresponds to $y = 1$.

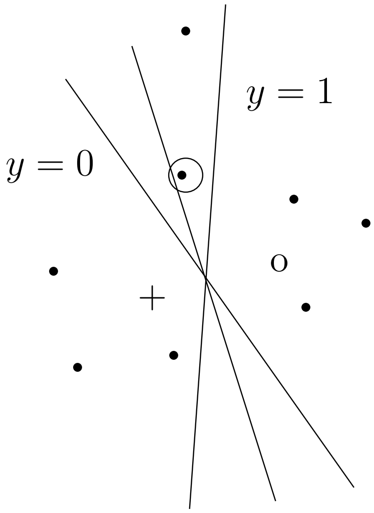

Figure 1.2. Two labeled points, unlabeled points, and the decision boundary. The point "+" corresponds to $y = 1$.

4. **(T/F – 2 points)** The criterion we have used here for active learning guarantees that the measure of uncertainty about the labels of the unlabeled points will decrease monotonically for each point after each query. $\boxed{\text{F}}$

Roughly speaking, when we "tilt" the boundary to incorporate each new labeled example some unlabeled points may end up closer to the boundary which tends to increase the uncertainty in those points.

Problem 2

Consider a regression problem where the two dimensional input points $\mathbf{x} = [x_1, x_2]^T$ are constrained to lie within the unit square: $x_i \in [-1, 1]$, $i = 1, 2$. The training and test input points $\mathbf{x}$ are sampled uniformly at random within the unit square. The target outputs y are governed by the following model

$$y \sim N(x_1^3 x_2^5 - 10x_1x_2 + 7x_1^2 + 5x_2 - 3,\ 1)$$

In other words, the outputs are normally distributed with mean given by

$$x_1^3 x_2^5 - 10x_1x_2 + 7x_1^2 + 5x_2 - 3$$

and variance 1.

We learn to predict y given $\mathbf{x}$ using linear regression models with 1st through 10th order polynomial features. The models are nested in the sense that the higher order models will include all the lower order features. The estimation criterion is the mean squared error.

We first train a 1st, 2nd, 8th, and 10th order model using $n = 20$ training points, and then test the predictions on a large number of independently sampled points.

1. **(6 points)** Select all the appropriate model(s) for each column. If you think the highest, or lowest, error would be shared among several models, be sure to list all models.

			Lowest test error
	Lowest training error	Highest training error	(typically)
1st order	( )	(X)	
2nd order	( )	( )	(X)
8th order	(X)	( )	
10th order	(X)	( )	( )

Briefly explain your selection in the last column, i.e., the model you would expect to have the lowest test error:

The 10th order regression model would seriously overfit when presented only with $n = 20$ training points. The second order model on the other hand might find some useful structure in the data based only on 20 points. The true model is also dominated by the second order terms. Since $|x_1| \leq 1$ and $|x_2| \leq 1$ any higher order terms without large coefficients are vanishingly small.

2. **(6 points)** We now train the polynomial regression models using $n = 10^6$ (one million) training points. Again select the appropriate model(s) for each column. If

you think the highest, or lowest, error would be shared among several models, be sure to list all models.

	Lowest structural error	Highest approx. error	Lowest test error
1st order	( )	( )	( )
2nd order	( )	( )	( )
8th order	(X)	( )	(X)
10th order	(X)	(X)	( )

3. **(T/F – 2 points)** The approximation error of a polynomial regression model depends on the number of training points. T

4. **(T/F – 2 points)** The structural error of a polynomial regression model depends on the number of training points. F

Problem 3

We consider here linear and non-linear support vector machines (SVM) of the form:

$$\begin{aligned} \min w_1^2/2 \quad &\text{subject to} \quad y_i(w_1 x_i + w_0) - 1 \geq 0, \quad i = 1, \ldots, n, \quad \text{or} \\ \min \mathbf{w}^T\mathbf{w}/2 \quad &\text{subject to} \quad y_i(\mathbf{w}^T\Phi_i + w_0) - 1 \geq 0, \quad i = 1, \ldots, n \end{aligned}$$

where Φ_i is a feature vector constructed from the corresponding real valued input x_i. We wish to compare the simple linear SVM classifier ($w_1 x + w_0$) and the non-linear classifier ($\mathbf{w}^T\Phi + w_0$), where $\Phi = [x, x^2]^T$.

1. **(3 points)** Provide three input points x_1, x_2, and x_3 and their associated ±1 labels such that they cannot be separated with the simple linear classifier, but are separable by the non-linear classifer with $\Phi = [x, x^2]^T$. You may find Figure 3.1. helpful in answering this question.

   $(x = 1, y = 1)$, $(x = 2, y = -1)$, $(x = 3, y = 1)$

2. **(3 points)** In the figure below (Figure 3.1), mark your three points x_1, x_2, and x_3 as points in the feature space with their associated labels. Draw the *decision boundary* of the non-linear SVM classifier with $\Phi = [x, x^2]^T$ that separates the points.

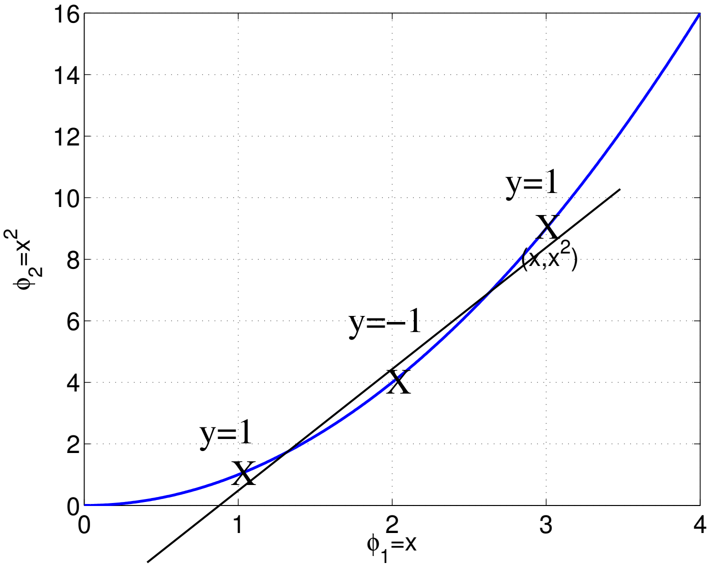

Figure 3.1. Feature space.

3. **(3 points)** Consider two labeled points ($x = 1, y = 1$) and ($x = 3, y = -1$). Is the margin we attain using feature vectors $\Phi = [x, x^2]^T$

(X) greater

(   ) equal

(   ) smaller

than the margin resulting from using the input x directly?

4. **(2 points)** In general, is the margin we would attain using scaled feature vectors $\Phi = [2x, 2x^2]^T$

(X) greater

(   ) equal

(   ) smaller

(   ) any of the above

in comparison to the margin resulting from using $\Phi = [x, x^2]^T$?

5. **(T/F – 2 points)** The values of the margins obtained by two different kernels $K(x, x')$ and $\tilde{K}(x, x')$ on the same training set do not tells us which classifier will perform better on the test set. $\boxed{\text{T}}$

We need to normalize the margin for it to be meaningful. For example, a simple scaling of the feature vectors would lead to a larger margin. Such a scaling does not change the decision boundary, however, and so the larger margin cannot directly inform us about generalization.

Problem 4

We consider here generative and discriminative approaches for solving the classification problem illustrated in Figure 4.1. Specifically, we will use a mixture of Gaussians model and regularized logistic regression models.

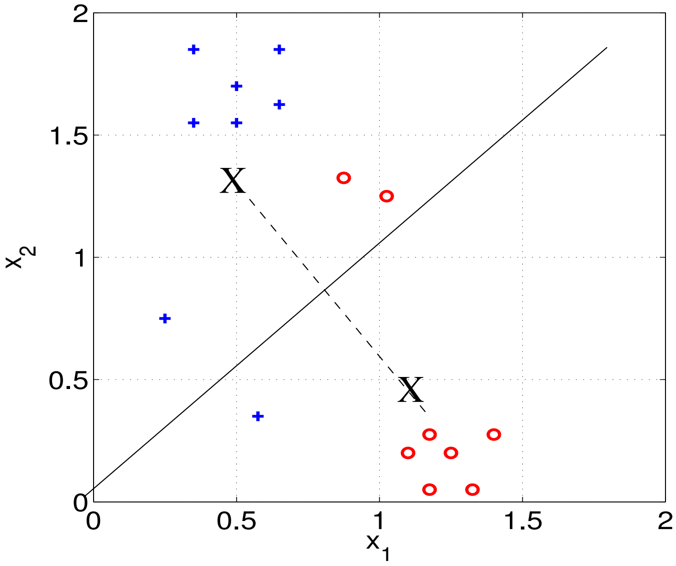

Figure 4.1. Labeled training set, where "+" corresponds to class $y = 1$.

1. We will first estimate a mixture of Gaussians model, one Gaussian per class, with the constraint that the covariance matrices are identity matrices. The mixing proportions (class frequencies) and the means of the two Gaussians are free parameters.

   a) **(3 points)** Plot the maximum likelihood estimates of the means of the two class conditional Gaussians in Figure 4.1. Mark the means as points "x" and label them "0" and "1" according to the class.

   The means should be close to the center of mass of the points.

   b) **(2 points)** Draw the decision boundary in the same figure.

   Since the two classes have the same number of points and the same covariance matrices, the decision boundary is a line and, moreover, should be drawn as the orthogonal bisector of the line segment connecting the class means.

2. We have also trained regularized linear logistic regression models

$$P(y = 1|\mathbf{x}, \mathbf{w}) = g(w_0 + w_1x_1 + w_2x_2)$$

for the same data. The regularization penalties, used in penalized conditional log-likelihood estimation, were $-Cw_i^2$, where $i = 0, 1, 2$. In other words, only one of the parameters were regularized in each case. Based on the data in Figure 4.1, we generated three plots, one for each regularized parameter, of the number of misclassified training points as a function of C (Figure 4.2). The three plots are not identified with the corresponding parameters, however. Please assign the "top", "middle", and "bottom" plots to the correct parameter, w_0, w_1, or w_2, the parameter that was regularized in the plot. Provide a brief justification for each assignment.

- **(3 points)** "top" = (w_1)

  By strongly regularizing w_1 we force the boundary to be horizontal in the figure. The logistic regression model tries to maximize the log-probability of classifying the data correctly. The highest penalty comes from the misclassified points and thus the boundary will tend to balance the (worst) errors. In the figure, this is roughly speaking $x_2 = 1$ line, resulting in 4 errors.

- **(3 points)** "middle" = (w_0)

  If we regularize w_0, then the boundary will eventually go through the origin (bias term set to zero). Based on the figure we can find a good linear boundary through the origin with only one error.

- **(3 points)** "bottom" = (w_2)

  The training error is unaffected if we regularize w_2 (constrain the boundary to be vertical); the value of w_2 would be small already without regularization.

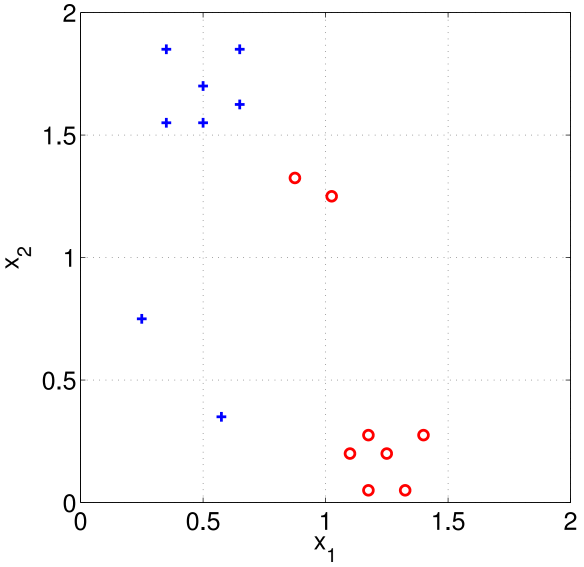

Figure 4.1 Labeled training set
(reproduced here for clarity)

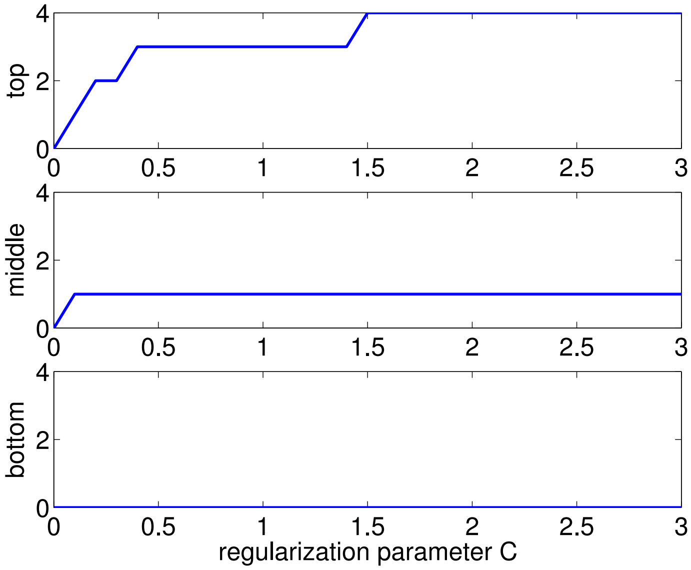

Figure 4.2. Training errors as a function
of regularization penalty