6.867 Machine Learning

Problem Set 2

Due date: Wednesday October 6

Please address all questions and comments about this problem set to `6867-staff@csail.mit.edu`. You will need to use MATLAB for some of the problems but essentially all the code is provided. If you are not familiar with MATLAB, please consult

`http://www.ai.mit.edu/courses/6.867/matlab.html`

and the links therein.

Problem 1: active learning

Consider a simpler linear regression model from one dimensional bounded input $x \in [-1, 1]$ to $y \in \mathcal{R}$. In a vector form for n inputs $\{x_1, \ldots, x_n\}$ we can write the model as

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \mathbf{e}, \tag{1}$$

where $\mathbf{X} = [1 \;\; x_1; \cdots ; 1 \;\; x_N]$; $\mathbf{e} = [\epsilon_1, \ldots, \epsilon_n]^T$ is a random vector where each ϵ_i is independent zero-mean Gaussian noise with variance σ^2.

There are many criteria we can use for input selection in this context. For example, we can use the determinant of the induced covariance matrix of the ML parameter estimate $\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$ given by $\mathbf{C} = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}$. The determinant, product of variances in principal directions, can be interpreted as a measure of effective (squared) "volume" of variation. The trace of the covariance matrix, sum of the variances in the principal directions, is an alternative (equally reasonable) measure to optimize.

1. *(10 points)* Show that, similarly to the determinant criterion, the first two training inputs, x_1 and x_2, that should be selected to minimise the trace of the covariance matrix $\mathbf{C}$ are $x_1 = 1$ and $x_2 = -1$. Assume $\sigma^2 = 1$ from now on.

   Hint: The trace of a symmetric matrix is simply the sum of its diagonal elements. For a 2×2 symmetric matrix,

$$\mathbf{A} = \begin{bmatrix} a & b \\ b & c \end{bmatrix}, \qquad \mathbf{A}^{-1} = \frac{1}{ac - b^2}\begin{bmatrix} c & -b \\ -b & a \end{bmatrix} \tag{2}$$

2. *(5 points)* Based on the first two training inputs, plot the output variance as a function of the input x.

3. *(10 points)* We have seen in class that incorporating any additional input point is guaranteed to reduce (not increase) the output variance at all points. Moreover, the output variance is guaranteed to go down for the selected input. We can therefore always query the point with the highest variance to successively and uniformly reduce of the output variances. This is the sequential selection criterion.

   Select the third training point x_3 as the one which currently has highest output variance (e.g., by looking at your plot). Plot the output variance resulting from querying at your selected point. You will need to write a line or two of MATLAB code for this.

4. *(10 points)* Suppose we were interested only in knowing the output at a single point $x = 0$. Compare the output variance at $x = 0$ resulting from adding the third point according to the sequential selection criterion or by simply querying at $x = 0$. What conclusions can you draw from this?

5. *(10 points)* You have been provided with MATLAB code to test the sequential selection active learning criterion on real data obtained from the UCI machine learning repository. The regression problem considered here involves predicting the fuel efficiency of a car (in miles per gallon) from two features: horsepower and weight (scaled to suitable units). The regression model we use for this is a simple linear one. The script `activeReg.m` selects a few training points at random to start with. It then performs both active and passive learning (independent of each other), and keeps track of the test errors. For active learning, the next training example (car) to be selected is the one which currently has the highest output variance. The active learning method can query the same car multiple times, each time getting the same answer. For passive learning, the next example is selected simple at random with replacement. For each method the available training examples are the first 150 cars in the dataset, while the remaining cars are reserved for testing.

   Compare the performance of active and passive learners by looking at how test errors behave as a function of the number of training samples. For your convenience, our script will automatically generate these plots. The plots are averaged over 30 independent runs since the random selection of initial points and successive random selections by the passive method can create considerable variation in the test performance. Can you explain the plots?

Problem 2: linear discriminant, optimality

We would like to classify vectors $\mathbf{x}$ of dimension d into one of two classes, $y = 0$ or $y = 1$. Assume that we know in advance that data from each class is sampled according to Gaussian distributions of equal covariance:

$$\begin{aligned} p(\mathbf{x}|y=0;\boldsymbol{\mu}_0,\boldsymbol{\Sigma}) &= \frac{1}{\sqrt{2\pi|\boldsymbol{\Sigma}|}}\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_0)\right) \quad (3)\\ p(\mathbf{x}|y=1;\boldsymbol{\mu}_1,\boldsymbol{\Sigma}) &= \frac{1}{\sqrt{2\pi|\boldsymbol{\Sigma}|}}\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)\right) \quad (4)\end{aligned}$$

To classify each point $\mathbf{x}$ optimally (in the sense of minimizing the expected classification error) we must assign it to the class y that maximizes the posterior probability

$$P(y|\mathbf{x}) = \frac{p(\mathbf{x}|y)P(y)}{p(\mathbf{x})} \quad (5)$$

The resulting decision boundary, separating the two classes, is defined by the equation

$$\log\frac{P(y=1|\mathbf{x})}{P(y=0|\mathbf{x})} = \log\frac{p(\mathbf{x}|y=1)P(y=1)}{p(\mathbf{x}|y=0)P(y=0)} = 0 \quad (6)$$

1. *(5 points)* Assuming $\boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}, P(y)$ are known, show that the decision boundary is given by the following line

$$(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{x}-\frac{\boldsymbol{\mu_1}+\boldsymbol{\mu_0}}{2}\right)+\log\frac{P(y=1)}{P(y=0)} = 0 \quad (7)$$

Can a linear logistic regression model give rise to the same decision boundary?

(Hint: if $\mathbf{A}$ is symmetric then $\mathbf{v}^T\mathbf{A}\mathbf{u} = \mathbf{u}^T\mathbf{A}\mathbf{v}$.)

In practice the parameters of the two Gaussians are unknown but we are given n_0 samples from class $y = 0$ and n_1 samples from class $y = 1$. Let $\hat{\boldsymbol{\mu}}_0$ and $\hat{\boldsymbol{\Sigma}}_0$ be the mean and covariance of the samples from class $y = 0$; similarly, we define $\hat{\boldsymbol{\mu}}_1$ and $\hat{\boldsymbol{\Sigma}}_1$ based on the samples from class $y = 1$.

In Fisher linear discriminant analysis we find $\mathbf{w}$ such that when each point is projected onto the line $t \cdot \mathbf{w}$, $t \in \mathcal{R}$, the classes are "maximally" separable by a simple threshold. The criterion for finding $\mathbf{w}$ is to *maximize* the separation of the projected means over the projeced variances:

$$\frac{(\hat{\boldsymbol{\mu}}_1^T\mathbf{w}-\hat{\boldsymbol{\mu}}_0^T\mathbf{w})^2}{n_0\mathbf{w}^T\hat{\boldsymbol{\Sigma}}_0\mathbf{w}+n_1\mathbf{w}^T\hat{\boldsymbol{\Sigma}}_1\mathbf{w}} \quad (8)$$

2. *(5 points)* To simplify our calculationwe first write the above criterion as

$$\frac{(\mathbf{m}^T\mathbf{w})^2}{\mathbf{w}^T\mathbf{S}\mathbf{w}} \tag{9}$$

where $\mathbf{m}$ is a vector and $\mathbf{S}$ is a positive semi-definite and symmetric matrix. What are $\mathbf{m}$ and $\mathbf{S}$?

3. *(10 points)* When $\mathbf{S}$ is positive definite, we can write it as $\mathbf{S} = \mathbf{R}^T\mathbf{R}$, where $\mathbf{R}$ is invertible (the *square root* of the matrix). Since $\mathbf{R}$ is invertible we can always search for $\mathbf{v} = \mathbf{R}\mathbf{w}$ instead of $\mathbf{w}$ directly. Write the criterion in terms of $\mathbf{v}$ and show that the maximizing solution is given by

$$\hat{\mathbf{v}} = \mathbf{R}^{-T}\mathbf{m} \tag{10}$$

where $\mathbf{R}^{-T} = (\mathbf{R}^T)^{-1}$. Provide the resulting expression for $\hat{\mathbf{w}}$.

(*Hint:* maximum of $\mathbf{a}^T(\mathbf{v}/\|\mathbf{v}\|)$ over $\mathbf{v}$ is obtained by any $\mathbf{v}$ proportional to $\mathbf{a}$.

Problem 3: Fisher linear discriminat vs. logistic regression

For this problem we have provided a MATLAB data file `data.mat` that contains 4 datasets, all binary classification tasks. Each dataset consists of a training set (`train`*i*) and a test set (`test`*i*):

`train1, train1_2, test1` Data generated from two bivariate Gaussians with identical covariances

`train2, test2` Data generated from two bivariate Gaussians with different covariances

`train3, test3` A digit classification task. The vectors representing digits are projected to a plane defined by two dimensions that capture most of the variability.

`train4, test4` The same digit classificaction task as in the 3rd dataset, but now digit images are 64-dimensional vectors of 1's or 0's — pixels in a 8×8 bitmap. The 3rd dataset is derived from this representation by projecting onto a plane.

The `.X` field of each variable is the representation of the points (each row is one datapoint), while the `y` field contains the class labels (0 or 1).

We have provided the following MATLAB functions that implement both Fisher discriminant classification and logistic regression:

`plotdata(train`*i*`)` For 2D data, plots the data and associated labels.

`w = fisherdiscriminant(train`*i*`.X, train`*i*`.y)` Trains a Fisher discriminant linear classifiers and returns its parameters.

`w = logisticreg(train`*i*`.X, train`*i*`.y)` Trains a logistic regression classifier by maximizing likelihood with the Newton's method, and returns its parameters.

`boundary([w1 w2 ...], test`*i*`)` For 2D data, plots the data and the decisions boundaries of several logistic regression or Fisher discriminant sets of parameters in a single figure.

`errorrate(w, test`*i*`)` Computes the error rate of a Fisher discriminant or logistic regression model.

1. *(5 points)* Train logistic regression and Fisher discriminant classifiers on each of the first 3 training sets and report the test error on the corresponding test set (6 numbers). For each dataset plot on the same graph the test data and the decision boundaries corresponding to logistic regression and Fisher discriminant methods (3 plots).

2. *(5 points)* On the first data set (`test1.dat`) the performance of the two classifiers is similar. Would you expect the performances to be similar for all datasets sampled from class conditional distributions that are Gaussians with equal covariances?

3. *(5 points)* Training set `train1_2` is identical to `train1` except that it has an extra training point. Train the logistic regression and the Fisher discriminant on `train1_2` and compare the error rates on `test1` with those achieved by models trained on `train1`. For each method explain why the error rates change or not change with the addition of a single training point.

4. *(5 points)* Train logistic regression and the Fisher discriminant on the full 64 dimensional representation of the digits and report the error rates (`train4` and `test4`). Compare the error rates with those achieved on the reduced 2D representation (3rd dataset). Is the multivariate Gaussian assumption reasonable for the full representation of the digits? How sensitive logistic regression and Fisher discriminant classification are to the Gaussian assumption?