6.867 Machine Learning

Problem Set 2 Solutions

Due date: Wednesday October 6

Problem 1: Active Learning

1. **Solution:** The covariance matrix of the parameter vector, after selecting the first two training examples, x_1 and x_2, is given by

$$\begin{aligned}\mathbf{C} &= \sigma^2(\mathbf{X}^T\mathbf{X})^{-1} = \left(\begin{bmatrix} 1 & 1 \\ x_1 & x_2 \end{bmatrix}\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \end{bmatrix}\right)^{-1} = \begin{bmatrix} 2 & x_1+x_2 \\ x_1+x_2 & x_1^2+x_2^2 \end{bmatrix}^{-1} \\ &= \frac{1}{2(x_1^2+x_2^2)-(x_1+x_2)^2}\begin{bmatrix} x_1^2+x_2^2 & -(x_1+x_2) \\ -(x_1+x_2) & 2 \end{bmatrix}\end{aligned}$$

$$\mathrm{Tr}(\mathbf{C}) = \frac{1}{(x_1-x_2)^2}(x_1^2+x_2^2+2) = 1+\frac{2(1+x_1x_2)}{(x_1-x_2)^2}$$

To minimise the trace, the second term should be as small as possible. Since x_1x_2 lies between 1 and -1 in the input region of interest, the minimum occurs when $x_1x_2 = -1$. Thus, x_1 and x_2 take the extreme values 1 and -1. ■

2. **Solution:** After using two training examples, output variance at a test point x_0 is given by

$$\begin{aligned}\text{output variance}(x_0) &= \begin{bmatrix} 1 & x_0 \end{bmatrix}\mathbf{C}\begin{bmatrix} 1 \\ x_0 \end{bmatrix} \\ &= \frac{1}{4}\begin{bmatrix} 1 & x_0 \end{bmatrix}\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} 1 \\ x_0 \end{bmatrix} \\ &= \frac{1}{2}(1+x_0^2).\end{aligned}$$

See attached plot (Figure 1). ■

3. **Solution:** We should either choose $x_3 = 1$ or $x_3 = -1$ as these are the points where output variance is maximum for $x \in [-1, 1]$.

   After adding $x_3 = -1$, output variance at a test point x_0 is given by

$$\begin{aligned}
\text{output variance}(x_0) &= \begin{bmatrix} 1 & x_0 \end{bmatrix} \left(\begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 1 & -1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 \\ x_0 \end{bmatrix} \\
&= \frac{1}{8} \begin{bmatrix} 1 & x_0 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ x_0 \end{bmatrix} \\
&= \frac{1}{8}(3 + 2x_0 + 3x_0^2).
\end{aligned}$$

   (Note: had we picked $x_3 = 1$ instead, we would have got $\frac{1}{8}(3 - 2x_0 + 3x_0^2)$.)

   See attached plot (Figure 1). ■

4. **Solution:** Output variance at $x = 0$ after adding x_3 according to the sequential selection criterion is 0.375 (for both $x_3 = 1$ and $x_3 = -1$).

   Output variance at $x = 0$ after adding $x_3 = 0$ is given by

$$\begin{aligned}
\text{output variance}(x_0) &= \begin{bmatrix} 1 & 0 \end{bmatrix} \left(\begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 0 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\
&= \frac{1}{6} \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\
&= 0.333
\end{aligned}$$

   This shows that while the sequential selection criterion is good for reducing the uncertainty at all test points, it does not guarantee that the uncertainty at any specific point (other than the point queried) will be minimised. We can increase our confidence about the output at a specific point by querying that point itself instead. ■

5. **Solution:** See attached plot (Figure 2).

   We make two observations from this plot:

   1. For very few training examples, active learning does much better than passive learning in terms of test error on an independent data set.
   2. When there are a large number of training examples, active learning does somewhat worse than passive learning.

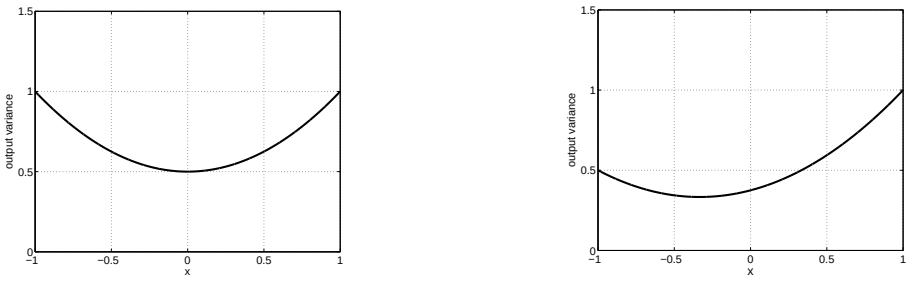

Figure 1: Plots for Problems 1.2 (left) and 1.3 (right).

The explanation for this is as follows. We have assumed a linear regression model for active learning. However, for the real data that we are working with, we have no guarantee of how good accurate a linear model is. When only a small number of examples (4 or 5) are available, the actively learned model seeks out the most informative training examples. It is thus more resistant to overfitting than the passively learned model. However, this resistance is gained at the cost of a strict assumption of linearity of the underlying model. If the data truly fit a non-linear model, the passive learning algorithm will eventually find the best linear fit. The active learning algorithm will not, as it will repeatedly query extreme training inputs in an effort to minimise variance (for the assumed linear model), and never query the large set of points lying in intermediate regions. Thus, for a large number of training examples (when both models have low output variance), the passive model is expected to have a lower 'bias' and hence lower generalisation error than the actively learned model.

There are secondary effects caused by the facts that querying the same input produces exactly the same output, and the noise distribution is not truly Gaussian. However, these effects are not very significant in the present context, since the test error of the actively learned linear model would still be expected to be greater than that for the passive model with infinite possible queries from a non-linear model with true Gaussian noise. (Our conclusion of the non-linearity of the model is supported by plotting the outputs against each of the input features and observing the nature of these (two) plots.) ■

Problem 2: Optimality of the Linear Discriminant

1. **Solution:** As mentioned in the text of the problem, the points on the decision boundary satisfy:

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

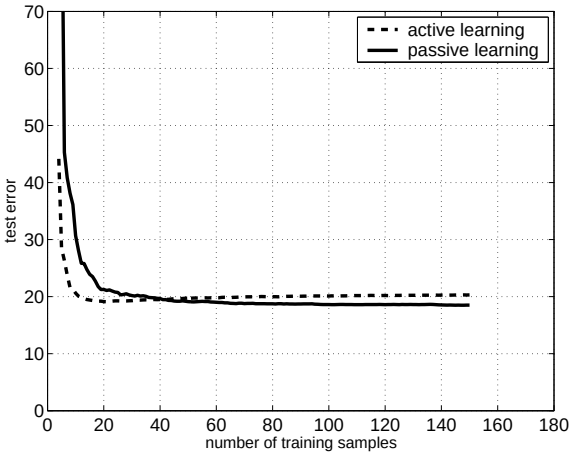

Figure 2: Plot for Problem 1.5.

The $\sqrt{2\pi|\mathbf{\Sigma}|}$ in the normal distributions cancel because of the ratios, while the exponentials are canceled by the logarithm:

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

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

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

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

and the assertion follows.

The decision boundary of logistic regression is also linear, and with the appropriate sample of training points, any linear separation can be the result of training logistic regression on some data. Thus the optimal decision boundary can be the decision boundary of a linear logistic regression model.

In fact, if the two Gaussians have the same covariance and we sample a large number of points as training data, the linear logistic regression decision boundary converges to the optimal decision. ■

2. **Solution:** The following optimization problem:

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

can be easily written as:

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} \frac{\left[(\hat{\boldsymbol{\mu}}_1 - \hat{\boldsymbol{\mu}}_0)^T \mathbf{w}\right]^2}{\mathbf{w}^T \left[n_0 \hat{\boldsymbol{\Sigma}}_0 + n_1 \hat{\boldsymbol{\Sigma}}_1\right] \mathbf{w}}$$

Therefore

$$\begin{aligned} \mathbf{m} &= \hat{\boldsymbol{\mu}}_1 - \hat{\boldsymbol{\mu}}_0 \\ \mathbf{S} &= n_0 \hat{\boldsymbol{\Sigma}}_0 + n_1 \hat{\boldsymbol{\Sigma}}_1 \end{aligned}$$

$\mathbf{S}$ is symmetric and positive semi-definite because $\hat{\boldsymbol{\Sigma}}_0$ and $\hat{\boldsymbol{\Sigma}}_1$ are and $n_0, n_1 \geq 0$. ■

3. **Solution:** To write the criterion in terms of $\mathbf{v}$, we substitute $\mathbf{w}$ by $\mathbf{R}^{-1}\mathbf{v}$:

$$\frac{(\mathbf{m}^T\mathbf{w})^2}{\mathbf{w}^T\mathbf{S}\mathbf{w}} = \frac{(\mathbf{m}^T\mathbf{R}^{-1}\mathbf{v})^2}{(\mathbf{R}^{-1}\mathbf{v})^T\mathbf{S}\mathbf{R}^{-1}\mathbf{v}} = \frac{((\mathbf{R}^{-T}\mathbf{m})^T\mathbf{v})^2}{\mathbf{v}^T\mathbf{v}} = \left[(\mathbf{R}^{-T}\mathbf{m})^T \frac{\mathbf{v}}{\|\mathbf{v}\|}\right]^2$$

where we have used $\mathbf{S} = \mathbf{R}^T\mathbf{R}$.

The criterion takes the form of the square of a dot product between the fixed vector $\hat{\mathbf{v}} = \mathbf{R}^{-T}\mathbf{m}$ and the vector of norm 1 given by $\mathbf{v}/\|\mathbf{v}\|$. The only degree of freedom over which to optimize is the angle between the two vectors. But if two vectors have fixed norms, their dot product is maximized when they have the same direction (the inequality $\mathbf{v}^T\mathbf{u} \leq \|\mathbf{v}\| \cdot \|\mathbf{u}\|$ holds). Thus $\mathbf{v} \equiv \hat{\mathbf{v}}$ maximizes the criterion (as well as any scalar multiple of $\hat{\mathbf{v}}$). Moreover:

$$\hat{\mathbf{w}} = \mathbf{R}^{-1}\hat{\mathbf{v}} = \mathbf{R}^{-1}\mathbf{R}^{-T}\mathbf{m} = \mathbf{S}^{-1}\mathbf{m} = \left(n_0\hat{\boldsymbol{\Sigma}}_0 + n_1\hat{\boldsymbol{\Sigma}}_1\right)^{-1}(\hat{\boldsymbol{\mu}}_1 - \hat{\boldsymbol{\mu}}_0)$$

■

Problem 3: Linear Discriminant vs. Logistic Regression

1. **Solution:**

	Set 1	**Set 2**	**Set 3**
logistic	0.067	0.17	0.2125
Fisher disc	0.07	0.1650	0.22

For the plots see Figure 3 at the end. ■

2. **Solution:** Yes, the performance of logistic regression and classification with the Fisher linear discriminant should be similar if data truly comes from Gaussian classes

of equal covariance. In this situation in the limit of infinite training data both logistic regression and the Fisher discriminant converge to the decision-theoretical optimal boundary, thus the only differences should arise because of the randomness of the finite training sample. ■

3. **Solution:**

	Trained on `train1`	**Trained on** `train1_2`
logistic	0.067	0.067
Fisher disc	0.07	0.1510

We observe that the addition of the outlier does not affect logistic regression, but has a strong negative impact on the performance of the Fisher linear discriminant classifier.

Since the new point is correctly classified and far from the boundary, the logistic model assigns to the outlier a $P(y|\mathbf{x})$ probability exponentially close to 1. Adding this probability to the likelihood has almost not effect on the criterion, because the probability is already almost maximum at the point. Thus logistic regression is not affected.

On the other hand the Fisher discriminant sees the data as if each class is Gaussian, and the addition of a single point very far from the current mean can greatly affect the estimate of the mean and variance of that Gaussian. We can distinguish two effects on the decision boundary:

- a translation of the decision boundary because the location of the mean of one class shifts with the addition of the outlier
- a rotation of the decision boundary because the variance in one direction increases by a large amount, while the variance in the perpendicular direction remains small. Thus the projection performed by the Fisher discriminant has to be rotated to keep the variance small.

■

4. **Solution:**

	64 features	**2D projection**
logistic	0.1425	0.2125
Fisher disc	0.23	0.22

We observe that Fisher discrimination and logistic regression achieve similar classification performance on the reduced 2D representation, but while the performance of logistic regression improves significantly on the full set of features, that of Fisher discrimination remains at best the same (if not even worse than that on 2D features).

Several properties of the given representation of digits violate the Gaussian assumption. Think about averaging together all digit images in one class as if they were

printed on transparent playing cards and placed in a deck. If digits were Gaussian, the average image should be well defined (the mean of the multivariate Gaussian), and the variability around that mean image should be distributed in all directions as if its noise. In reality:

- some digits, like 4 and 7, are commonly written in more than one way, so when we look through the transparent deck of cards we see more than one defined outline
- the same basic outline can be transformed by slight rotations, shear, scaling, or translation, and while the digit remains the same, the Gaussianity is violated by such operations

Logistic regression is not that sensitive to the Gaussian assumption. If fact, in logistic regression we only model $P(y|\mathbf{x})$ as if it is the decision boundary between Gaussians, but we do not assume that $P(\mathbf{x})$ is a Gaussian distribution.

The Fisher discriminant is more sensitive to the Gaussian assumption because it is derived by modeling directly the means and covariances of the training data as if classes were Gaussian. If data is not Gaussian the means and covariances alone do not fully capture data structure. ■

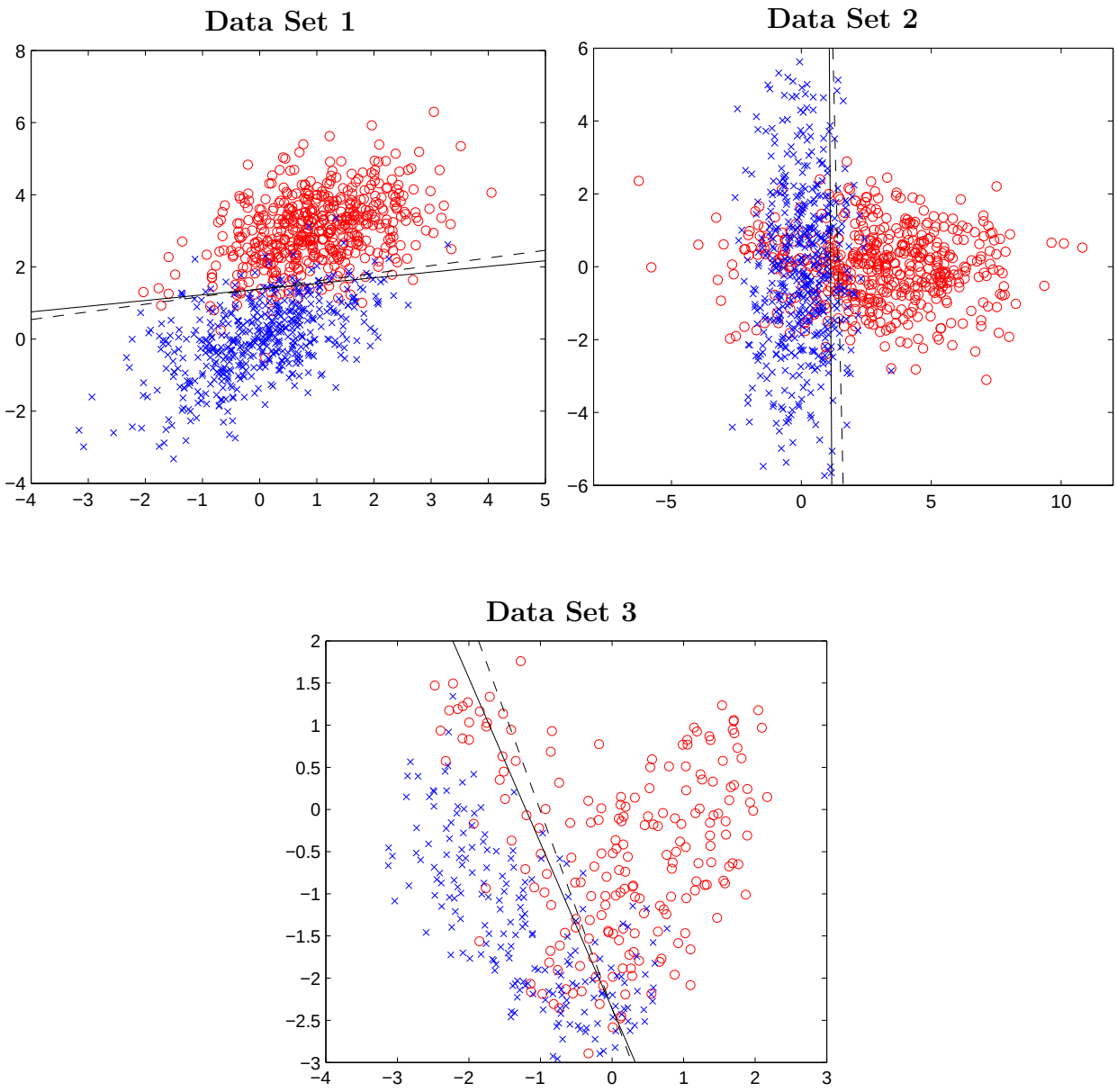

Figure 3: Plots for Problem 3 Part 1