Elements of Statistical Learning - Questions and Solutions

Edwin Fennell

1.1 Suppose each of K-classes has an associated target t_k, which is a vector of all zeros, except a one in the kth position. Show that classifying to the largest element of $\hat{y}$ amounts to choosing the closest target, $\min_k \|t_k - \hat{y}\|$, if the elements of $\hat{y}$ sum to one.

Suppose that the largest element of $\hat{y}$ is the i-th one, s.t. $t_i \cdot y \geq t_j \cdot y \; \forall j$. Then it is clear that

$$\|t_k - \hat{y}\|^2 = (t_k - \hat{y}) \cdot (t_k - \hat{y}) = 1 + \hat{y} \cdot \hat{y} - 2t_k \cdot y$$

is minimised for $k = i$, and thus so is $\|t_k - \hat{y}\|$.

1.2 Show how to compute the Bayes decision boundary for the simulation example in Figure 2.5.

The decision boundary is where the generating densities for the two classes are equal. Given that we know the exact generating desities for our two classes across the entire state space, this is easy to determine.

1.3 Derive equation 2.24

Imagine that we have N points uniformly distributed on the p-dimensional unit ball. The probably that a point lies at a distance at most r from the origin is given by r^p. The p.d.f. of this function is thus pr^{p-1}. The p.d.f. of the distance of the closest point to the origin is thus

$$Npr^{p-1}(1 - r^p)^{N-1}$$

This is derived from the p.d.f. for a single point conditioned on the fact that all other points are further away than this point, and then with a factor of N due to symmetry between points. To get the median value of this distance, we take the corresponding c.d.f., equate to $\frac{1}{2}$ and solve for r. The equation is

$$1 - (1 - r^p)^N = \frac{1}{2}$$

which rearranges to

$$r = \left(1 - \left(\frac{1}{2}\right)^{\frac{1}{N}}\right)^{\frac{1}{p}}$$

as required.

1.4 The edge effect problem discussed on page 23 is not peculiar to uniform sampling from bounded domains. Consider inputs drawn from a spherical multivariate normal distribution $\boldsymbol{X} \sim \boldsymbol{N(0, I_p)}$. The squared distance from any sample point to the origin has a $\boldsymbol{\chi^2_p}$ distribution with mean p. Consider a prediction point $\boldsymbol{x_0}$ drawn from this distribution, and let $\boldsymbol{a = \frac{x0}{\|x0\|}}$ be an associated unit vector. Let $\boldsymbol{z_i = a^T x_i}$ be the projection of each of the training points on this direction. Show that the $\boldsymbol{z_i}$ are distributed $\boldsymbol{N(0, 1)}$ with expected squared distance from the origin 1, while the target point has expected squared distance p from the origin. Hence for p = 10, a randomly drawn test point is about 3.1 standard deviations from the origin, while all the training points are on average one standard deviation along direction a. So most prediction points see themselves as lying on the edge of the training set.

Our spherical multivariate normal distribution has total spherical symmetry - density is purely a function of distance from the origin. Therefore rotating our frame of reference after choosing a yields WLOG $a_i = \delta_{0i}$. Therefore $a^T x_i$ is just distributed as $(x_i)_0$, which from the definition of our multivariate normal is distributed as $N(0, 1)$. I'm not really sure that this is at all meaningful.

1.5

- **Derive equation (2.27). The last line makes use of (3.8) through a conditioning argument.**
- **Derive equation (2.28), making use of the cyclic property of the trace operator [trace(AB) = trace(BA)], and its linearity (which allows us to interchange the order of trace and expectation).**

The context here is that we have an independent variable X and a dependent variable Y with the relationship

$$Y = X^T \beta + \epsilon$$

where $\epsilon \sim N(0, \sigma^2)$. We can jointly observe X, Y but don't know β. We make a bunch of observations x_i, y_i and then calculate a value $\hat{\beta}$ for β that minimises the L2 error. Given our model $\hat{\beta}$ and a test point x_0, we want to find the expected (squared) prediction error from our predicted value

$$\hat{y}_0 = x_0^T \hat{\beta}$$

from the actual "noisy" observed value

$$y_0 = x_0^T \beta + \epsilon$$

(A bit of notation abuse here - we defined ϵ as a random variable, and are here using it to mean a specific observation of that random variable)

The quantity we want is therefore

$$EPE(x_0) = E((y_0 - \hat{y_0})^2)$$

but what exactly are we taking the expectation over? The way we have set up our system allows our training inputs and test input to vary freely, and we already have training and test noise. If all of these are held constant than all the values above are completely determined, so these four things can be used to completely parametrise our state space. Here we are conditioning on a fixed value for our test point, x_0, so we reduce our state space to three variables - our training points $\mathbf{X}$, our training error $\mathbf{e}$, and our test error $y_0|x_0$, all of which are mutually independent of each other.

We can simplify our expression for the EPE by noting that

$$\hat{\beta} = (\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}(\mathbf{X}^T\beta + \mathbf{e}) = \beta + (\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{e}$$

and therefore

$$y_0 - \hat{y}_0 = \epsilon - x_0^T(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{e}$$

Note that the first term is dependent only on $y_0|x_0$ and the second term is dependent only on $\mathbf{X}, \mathbf{e}$. In addition the expectation of each term is 0 since $E(\epsilon) = 0$, $E(\mathbf{e}) = \mathbf{0}$, and $\mathbf{X}, \mathbf{e}$ are indpendent of each other. Therefore

$$EPE(x_0) = E(\epsilon^2) + E((x_0^T(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{e})^2)$$

The first term simply evaluates to σ^2 We can simplify the tensor composition by writing

$$(x_0^T(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{e})^2 = x_0^T(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{e}\mathbf{e}^T\mathbf{X}^T(\mathbf{X}^T\mathbf{X})^{-1}x_0$$

When we expand this expression, we end up with an $\mathbf{e}\mathbf{e}^T$ term in the middle surrounded by things independent of $\mathbf{e}$. Therefore we have

$$E(x_0^T(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{e}\mathbf{e}^T\mathbf{X}^T(\mathbf{X}^T\mathbf{X})^{-1}x_0) = E(x_0^T(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}E(\mathbf{e}\mathbf{e}^T)\mathbf{X}^T(\mathbf{X}^T\mathbf{X})^{-1}x_0)$$

$E(\mathbf{e}\mathbf{e}^T)$ is just $\sigma^2\mathbf{I}$, and so we can cancel out a bunch of the middle terms to get

$$E(\sigma^2 x_0^T(\mathbf{X}^T\mathbf{X})^{-1}x_0)$$

The only variable left that we are taking the expectation over is $\mathbf{X}$, and so we finally have

$$EPE(x_0) = \sigma^2 + \sigma^2 x_0^T E_{\mathbf{X}}((\mathbf{X}^T\mathbf{X})^{-1})x_0$$

as required. For the second part, we assume $E(\mathbf{X}) = 0$, take the expectation of this expression over x_0, and consider what happens if we pick a very large number of training samples. Assuming some sane regularity conditions on the distribution of X, we obtain

$$\frac{1}{N}\mathbf{X}^T\mathbf{X} \xrightarrow[a.s./\text{in probabilty}]{} Cov(X)$$

by the strong law of large numbers. We make an appeal to suffix notation to get

$$E_{x_0}(EPE(x_0)) \sim \sigma^2 + \sigma^2 E((x_0)_i(x_0)_j)(Cov(X)^{-1}/N)_{ij}$$

But the matrix with (i,j)-th element $E((x_0)_i(x_0)_j)$ is just $Cov(x_0) = Cov(X)$. Therefore

$$E_{x_0}(EPE(x_0)) \sim \sigma^2(p/N) + \sigma^2$$

as required.

1.6 Consider a regression problem with inputs x_i and outputs y_i, and a parameterized model $f_\theta(x)$ to be fit by least squares. Show that if there are observations with tied or identical values of x, then the fit can be obtained from a reduced weighted least squares problem.

Consider the expression

$$\sum_{i=1}^{n}(y_1 - f_\theta(x))^2$$

for positive a_i When we write this out in full as a quadratic in terms of $f_\theta(x)$ we obtain

$$\sum_{i=1}^{n}(y_i^2 - 2y_i f_\theta(x) + f_\theta(x)^2)$$

We complete the square and take out a constant factor to obtain

$$n\left(\frac{\sum_{i=1}^{n} y_i}{n} - f_\theta(x)\right)^2 + c$$

where c is a constant we don't care about.

This shows that in any sum of squared errors, we can replace any sum of terms with the same x-value by a single term weighted by multiplicity and the $y - value$ replaced by the average y-value for those terms, plus a constant. Since we are trying to maximise the expression over $\theta \in \Theta$, we can just ignore the constants and are left with a weighted least squares.