STAT 37601 Midterm

Albert Chu

May 2025

1. (a) We are given that $f(x|Y=k) \sim N(\mu_k, \sigma^2 I)$ for $k = 0, 1$ with $P(Y = 0) = P(Y = 1) = 1/2$. Bayes rule gives us the decision rule:

$$P(Y = k|X = x) = \frac{f(x|Y = k)P(Y = k)}{f(x)}$$

But since the denominator is common to both classes and the priors are the same too, we are essentially just maximizing $f(x|Y = k)$. In particular, we have the decision boundary, which clearly reduces to

$$\exp -\frac{1}{2\sigma^2}||x - \mu_1||^2 = \exp -\frac{1}{2\sigma^2}||x - \mu_0||^2$$

$$||x - \mu_1||^2 = ||x - \mu_0||$$

$$x^T x - x^T \mu_1 - \mu_1^T x + \mu_1^T \mu_1 = x^T x - x^T \mu_0 - \mu_0^T x + \mu_0^T \mu_0$$

$$-2\mu_1^T x + \mu_1^T \mu_1 = -2\mu_0^T x + \mu_0^T \mu_0$$

$$(\mu_1 - \mu_0)^T x = \frac{1}{2}(\mu_1^T \mu_1 - \mu_0^T \mu_0)$$

This is the equation of a hyperplane with normal vector $\mu_1 - \mu_0$. Intuitively, our decision boundary is exactly this hyperplane that evenly bisects the line from μ_0 to μ_1, as defined by the midpoint on this line plus a vector that is orthogonal to the line.

(b) We plug into our equation to see

$$x = \frac{\mu_0 + \mu_1}{2} + u = \frac{(\mu - \mu, 0, ..., 0)}{2} + u = \frac{(0, ..., 0)}{2} + u$$

for u such that $u^t(\mu + \mu, 0, ..., 0) = 0$. This implies that u is either all zeros (if $\mu \neq 0$) or has its first coordinate as 0 and the rest as any real values. In particular, since the midpoint is the origin, the boundary hyperplane divides the connecting line in half.

(c) The error rate here is defined as $\frac{1}{2}P(X \in R_0|Y = 1) + \frac{1}{2}P(X \in R_1|Y = 0)$. We can substitute our regions to get

$$\frac{1}{2}P(x_1 < 0|Y = 1) + \frac{1}{2}P(x_1 > 0|Y = 0)$$

We have $X_0 \sim N(-\mu, \sigma^2), X_1 \sim N(\mu, \sigma^2)$ and so the probability of mis-classifying a class-1 point (or a class-0 point by symmetry), is $\Phi(-\frac{|\mu|}{2})$. The Bayes error is therefore $2 \cdot \frac{1}{2}\Phi(-\frac{|\mu|}{2})$.

2. (a) We start with images of shape (3, 32, 32). After the first convolutional layer, each output is now (32, 32, 32) (since we had a kernel size of 5 with a padding of 2 and used the default stride of 1). We then pool with kernel size 2 and stride 2 to get an output size of (32, 16, 16). This is passed into the second convolutional layer, which outputs a shape of (64, 16, 16). We pool again to get (64, 8, 8), which is reshaped to ("conv out size") and then passed into the final fully connected layer. We therefore see that conv out size must be 64*8*8=4096.

   (b) The first convolutional layer has $3*5*5*32$ parameters (no bias), the pooling layers don't have parameters, the second convolutional layer has $32*5*5*64$ and the final fully connected layer has $4096*10$, which all sums to $2400 + 51200 + 40960 = 94560$ total parameters.

   (c) Aside from the syntax error on the second line of forward (extra parentheses at the end), the error is in the calculation of O. We don't want to call relu on the output of the final linear layer since we typically want the raw logits to use in calculating the cross entropy loss. Calling RELU here would clip all values to >= 0, which would inhibit our loss calculation.

   (d) In code, it would simply be $loss = self.criterion(O, Y)$. Mathematically, it would be something like

$$L(O, Y) = -\sum_i^N \sum_c^C Y_{i,c}\left(O_{i,c} - \log\sum_k^C e^{O_{i,k}}\right)$$

3. (a)

$$O = f(x) = v\sigma(w_1x_1 + w_2x_2 + w_3x_3) + vp(w_1x_1 + w_2x_2 + w_3x_3)$$

   (b)

$$\epsilon_O := \frac{\partial L}{\partial O} = \frac{-y\exp(-yO)}{1 + \exp(-yO)}$$

   (c)

$$\frac{\partial L}{\partial v} = \frac{\partial L}{\partial O}\frac{\partial O}{\partial v}$$

   We can calculate

$$\frac{\partial O}{\partial v} = \sigma(w_1x_1 + w_2x_2 + w_3x_3) + p(w_1x_1 + w_2x_2 + w_3x_3)$$

   Therefore,

$$\frac{\partial L}{\partial v} = \epsilon_O \cdot (\sigma(w_1x_1 + w_2x_2 + w_3x_3) + p(w_1x_1 + w_2x_2 + w_3x_3))$$

(d)

$$\frac{\partial L}{\partial h_1} = \frac{\partial L}{\partial O}\frac{\partial O}{\partial h_1}, \frac{\partial L}{\partial h_2} = \frac{\partial L}{\partial O}\frac{\partial O}{\partial h_2}$$

For $z = w_1x_1 + w_2x_2 + w_3x_3$,

$$\frac{\partial O}{\partial h_1} = v \cdot \sigma'(z), \frac{\partial O}{\partial h_2} = v \cdot p'(z)$$

Therefore,

$$\frac{\partial L}{\partial h_1} = \epsilon_O \cdot v \cdot \sigma'(z), \frac{\partial L}{\partial h_2} = \epsilon_O \cdot v \cdot p'(z)$$

(e) For $i = 1, 2, 3$,

$$\frac{\partial L}{\partial w_i} = \sum_{k=1}^{2} \frac{\partial L}{\partial h_k}\frac{\partial h_k}{\partial w_i}$$

We have for $k = 1, 2$,

$$\frac{\partial h_k}{\partial w_i} = x_i$$

Therefore,

$$\frac{\partial L}{\partial w_1} = (\epsilon_1 + \epsilon_2) \cdot 2x_1, \frac{\partial L}{\partial w_2} = (\epsilon_1 + \epsilon_2) \cdot 2x_2, \frac{\partial L}{\partial w_3} = (\epsilon_1 + \epsilon_2) \cdot 2x_3$$

4. (a)

$$P_k(X|\theta_k) = \alpha^{X_1}(1-\alpha)^{1-X_1} \cdot \prod_{j=2}^{d} \theta_{kj}^{X_j}(1-\theta_{kj})^{1-X_j}$$

(b) i.

$$\pi_1^{new} = \frac{1}{n}\sum_{i}^{n} w_{i1}, \pi_2^{new} \sum_{i=1}^{n} w_{i2}$$

ii. Since $w_{i1} + w_{i2} = 1$, we see that in the log likelihood estimate, we can simply sum the weight terms, which gives us a normal Bernoulli MLE.

$$\alpha^{new} = \frac{1}{n}\sum_{i}^{n} X_{i1}$$

Note that since α is common to both components and depends only on the empirical mean of X_{i1}, we would get this same estimate if we knew each sample's component label.

iii. By maximizing the expected log likelihood, we have the new estimates as

$$\theta_{kj}^{new} = \frac{1}{\sum_{i}^{n} w_{ik}} \sum_{i}^{n} w_{ik}X_{ij}$$