Comp 652: Assignment 3 Solutions

1. Exploring smoothing for discrete distributions

A, B.
```
function [retX, retY] = cross_val()
ttr = load('ttr.txt');
n = length(ttr);
m = max(ttr);

% Split the data set:
D = ttr(1:floor(n/2));
V = ttr(floor(n/2)+1:end);

% Estimate the two models on the training set and then compute the validation set probs
probPX = ones(m,1);
probPY = ones(m,1);
for s = 1 : m
pX = profX(s,D,m);
pY = profY(s,D,m);

for j=1: length(V)
probPX(s) = probPX(s) * pX( floor((V(j)-1)/s)+1 );
probPY(s) = probPY(s) * pY(V(j));
end
end

retX = -log(probPX);
retY = -log(probPY);
end

function dist = profX(s,D,m)
n = length(D);

S = zeros( m,1 );
for i = 1 : n
S(D(i)) = S(D(i)) + 1;
end

% the estimate is the # of examples in a bin divided by the total # of examples
dist = zeros( ceil(m/s), 1);
for i = 1 : length(dist)
dist(i) = sum(S((i-1)*s + 1 : min(i*s, m))) / n;
end
end


function dist = profY(s,D,m)
n = length(D);

S = zeros( m,1 );
for i = 1 : n
S(D(i)) = S(D(i)) + 1;
end
```

```
S = S / n;

# the estimate is just like prof x's, except divided by the size of the bin
dist = zeros( m, 1);
for i = 1 : length(dist)
a = floor((i-1)/s)*s + 1;
b = min(floor((i-1)/s)*s + s, m);
dist(i) = sum(S(a:b)) / (b-a+1);
end
end
```

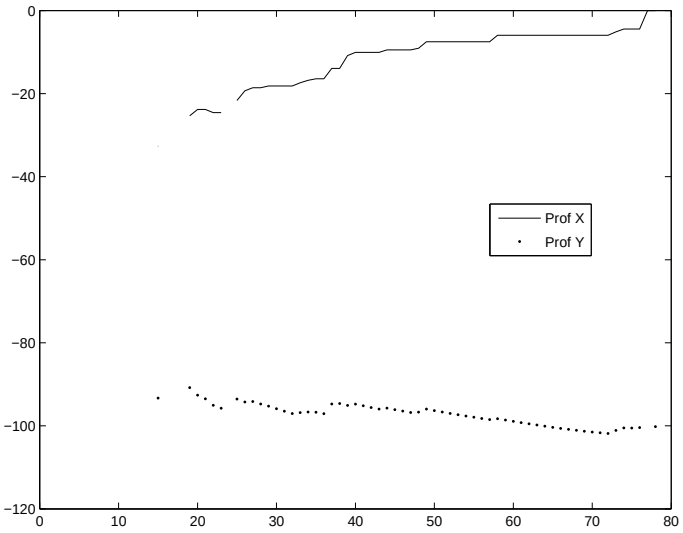

Figure 1: The log likelihood of the data under the two models.

C. The two methods compute different things. Professor X computes a distribution over bins (sets of months), whereas Professor Y computes a distribution over months. Professor Y's method can be interpreted as being the same as X's method, with one extra assumption: that all the months in a particular bin share the same probability (the distribution within a bin is uniform).

For Professor X, likelihood values under different bin sizes are not directly comparable: the bin size is not only a parameter of the method, but it also affects the semantics of the output. Under Professor Y's method, the semantics of the output are the same under all bin sizes, and the choice of $S = 20$ looks to be the best as it provides the largest likelihood for the validation set. Note that, even though for Professor X the likelihood doesn't tell us much, it is obvious for both methods that large bin sizes offer little information. In particular, if the bin size is 78, both methods will output the same result no matter what the data set looks like.

Since the two methods provide different information, they are not directly comparable. One would use Professor X's method if only interested in the probabilities of different groups of months. On the other hand, Professor Y offers a fine-grained, smoothed distribution, but this smoothing comes at the cost of adding the uniformity assumption.

2. Gaussian discriminant analisys

A.

$$\arg\max_{\sigma} l(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m | y_1, y_2, ..., y_m, \mu_0, \mu_1, \sigma) = \arg\max_{\sigma} \log l(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m | y_1, y_2, ..., y_m, \mu_0, \mu_1, \sigma) =$$

$$\log l(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m | y_1, y_2, ..., y_m, \mu_0, \mu_1, \sigma) = \log \prod_{i=1}^{m} P(\mathbf{x}_i | y_1, y_2, ..., y_m, \mu_0, \mu_1, \sigma)$$

$$= \sum_{i=1}^{m} \log \left(\frac{1}{\sqrt{(2\pi)^n |\Sigma|}} \exp \left[-\frac{1}{2} (\mathbf{x}_i - \mu(y_i))^T \Sigma^{-1} (\mathbf{x}_i - \mu(y_i)) \right] \right)$$

where

$$\mu(y_i) = \begin{cases} \mu_0, & \text{if } y_i = 0 \\ \mu_1, & \text{if } y_i = 1 \end{cases}$$

So,

$$\log l(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m | y_1, y_2, ..., y_m, \mu_0, \mu_1, \sigma) =$$

$$= -m \log \sqrt{(2\pi)^n (\sigma^2)^n} - \frac{1}{2} \sum_{i:y_i=0} (\mathbf{x}_i - \mu_0)^T \Sigma^{-1} (\mathbf{x}_i - \mu_0) - \frac{1}{2} \sum_{i:y_i=1} (\mathbf{x}_i - \mu_1)^T \Sigma^{-1} (\mathbf{x}_i - \mu_1)$$

$$= -\frac{mn}{2} \log 2\pi - mn \log \sigma - \frac{1}{2} \frac{1}{\sigma^2} \left(\sum_{i:y_i=0} (\mathbf{x}_i - \mu_0)^T (\mathbf{x}_i - \mu_0) + \sum_{i:y_i=1} (\mathbf{x}_i - \mu_1)^T (\mathbf{x}_i - \mu_1) \right)$$

Now we take the derivative and equal it to find the minimum:

$$\frac{\partial}{\partial \sigma} \log l(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m | y_1, y_2, ..., y_m, \mu_0, \mu_1, \sigma) = 0$$

$$\frac{\partial}{\partial \sigma} - \frac{mn}{2} \log 2\pi - mn \log \sigma - \frac{1}{2\sigma^2} \left(\sum_{i:y_i=0} (\mathbf{x}_i - \mu_0)^T (\mathbf{x}_i - \mu_0) + \sum_{i:y_i=1} (\mathbf{x}_i - \mu_1)^T (\mathbf{x}_i - \mu_1) \right) = 0$$

$$- \frac{mn}{\sigma} + \frac{1}{\sigma^3} \left(\sum_{i:y_i=0} (\mathbf{x}_i - \mu_0)^T (\mathbf{x}_i - \mu_0) + \sum_{i:y_i=1} (\mathbf{x}_i - \mu_1)^T (\mathbf{x}_i - \mu_1) \right) = 0$$

$$- \sigma^2 mn + \left(\sum_{i:y_i=0} (\mathbf{x}_i - \mu_0)^T (\mathbf{x}_i - \mu_0) + \sum_{i:y_i=1} (\mathbf{x}_i - \mu_1)^T (\mathbf{x}_i - \mu_1) \right) = 0$$

$$\sigma^2 = \left\{ \sum_{i=1}^{m} (1 - y_i)(\mathbf{x}_i - \mu_0)^T (\mathbf{x}_i - \mu_0) + y_i (\mathbf{x}_i - \mu_1)^T (\mathbf{x}_i - \mu_1) \right\} / (mn)$$

B.

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

$$= \frac{\left(\sqrt{(2\pi)^n |\Sigma|}\right)^{-1} \exp\left[-\frac{1}{2}(\mathbf{x} - \mu_0)^T \Sigma^{-1} (\mathbf{x} - \mu_0)\right] P(y=0)}{\left(\sqrt{(2\pi)^n |\Sigma|}\right)^{-1} \exp\left[-\frac{1}{2}(\mathbf{x} - \mu_1)^T \Sigma^{-1} (\mathbf{x} - \mu_1)\right] P(y=1)} =$$

$$= \exp\left[-\frac{1}{2}(\mathbf{x} - \mu_0)^T \Sigma^{-1} (\mathbf{x} - \mu_0) + \frac{1}{2}(\mathbf{x} - \mu_1)^T \Sigma^{-1} (\mathbf{x} - \mu_1) \right] \frac{P(y=0)}{P(y=1)} =$$

$$= \exp\left[-\frac{1}{2\sigma^2} \left((\mathbf{x} - \mu_0)^T (\mathbf{x} - \mu_0) + (\mathbf{x} - \mu_1)^T (\mathbf{x} - \mu_1) \right) + \log P(y=0) - \log P(y=1) \right] =$$

Therefore, on the boundary, where the ratio is 1,

$$-\frac{1}{2\sigma^2}\left((\mathbf{x}-\mu_0)^T(\mathbf{x}-\mu_0)+(\mathbf{x}-\mu_1)^T(\mathbf{x}-\mu_1)\right)+\log P(y=0)-\log P(y=1)=0$$

$$\sum_{j=1}^{n}(x_j-\mu_{0,j})^2+\sum_{j=1}^{n}(x_j-\mu_{1,j})^2=(\log P(y=0)-\log P(y=1))2\sigma^2$$

$$\sum_{j=1}^{n}(x_j^2-2\mu_{0,j}x_j+\mu_{0,j}^2)+\sum_{j=1}^{n}(x_j^2-2\mu_{1,j}x_j+\mu_{1,j}^2)=(\log P(y=0)-\log P(y=1))2\sigma^2$$

$$\sum_{j=1}^{n}(-2\mu_{0,j}x_j+\mu_{0,j}^2)+\sum_{j=1}^{n}(-2\mu_{1,j}x_j+\mu_{1,j}^2)=(\log P(y=0)-\log P(y=1))2\sigma^2$$

$$(\mathbf{x}-\mu_0)^T(\mathbf{x}-\mu_0)=(\mathbf{x}-\mu_1)^T(\mathbf{x}-\mu_1)+(\log P(y=0)-\log P(y=1))2\sigma^2$$

The term $\log P(y=0)-\log P(y=1))2\sigma^2$ shifts the decision boundary by increasing the probability of the class with the larger prior. The larger σ^2 is, the more weight is given to the class with the larger prior. An intuitive explanation for this is that large variance means unreliable data, so the prior should be trusted more in that case.

C. The matlab code:

```
function  [mu0,mu1,sigmaSq,correct] = lda_commonVar()
X = load('wpbc_x.txt');
[m,n] = size(X);
Y = load('wpbc_yrecur.txt');

mu0 = (X' * (1-Y)) / sum(1-Y) ;
mu1 = (X' * Y) / sum(Y);
mus = mu0 * ( 1 - Y)' + mu1 * Y';
sigmaSq = trace( (X' - mus)' * (X' - mus))/ (m * n);
p1 = sum(Y)/m;
p0 = 1 - p1;

hyp0 = -0.5 * sigmaSq * ((X - ones(m,1) * mu0' ) .^ 2 ) * ones(n,1) + log(p0);
hyp1 = -0.5 * sigmaSq * ((X - ones(m,1) * mu1' ) .^ 2 ) * ones(n,1) + log(p1);

correct = sum(abs((hyp0 > hyp1) - Y ))  ;
end
```

123 of the training examples are classified as correctly, and the MLE parameters are:

$$\mu_0 = \begin{bmatrix} 0.0171 \\ 0.0225 \\ 0.1128 \\ 0.9340 \\ 0.0001 \\ 0.0001 \\ 0.0002 \\ 0.0001 \\ 0.0002 \\ 0.0001 \\ 0.0006 \\ 0.0013 \\ 0.0041 \\ 0.0667 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0205 \\ 0.0304 \\ 0.1367 \\ 1.3290 \\ 0.0001 \\ 0.0004 \\ 0.0004 \\ 0.0002 \\ 0.0003 \\ 0.0001 \\ 0.0027 \\ 0.0027 \end{bmatrix} \times 10^3 \quad \mu_1 = \begin{bmatrix} 0.0183 \\ 0.0218 \\ 0.1211 \\ 1.0820 \\ 0.0001 \\ 0.0001 \\ 0.0002 \\ 0.0001 \\ 0.0002 \\ 0.0001 \\ 0.0007 \\ 0.0012 \\ 0.0047 \\ 0.0820 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0227 \\ 0.0296 \\ 0.1513 \\ 1.6358 \\ 0.0001 \\ 0.0004 \\ 0.0004 \\ 0.0002 \\ 0.0003 \\ 0.0001 \\ 0.0035 \\ 0.0049 \end{bmatrix} \times 10^3 \quad \sigma = 118.525$$

D. In a similar fashion, it is not hard to see that to find the maximum likelihood of σ_j we will have to maximize:

$$\begin{aligned} &\log l(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m | y_1, y_2, ..., y_m, \mu_0, \mu_1, \sigma_1, \sigma_2, ..., \sigma_n) = \\ &= -m \log \sqrt{(2\pi)^n \prod_{j=1}^{n} \sigma_j^2} - \frac{1}{2} \sum_{i:y_i=0} (\mathbf{x}_i - \mu_0)^T \Sigma^{-1} (\mathbf{x}_i - \mu_0) - \frac{1}{2} \sum_{i:y_i=1} (\mathbf{x}_i - \mu_1)^T \Sigma^{-1} (\mathbf{x}_i - \mu_1) \\ &= -\frac{mn}{2} \log 2\pi - m \sum_{j=1}^{n} \log \sigma_j - \frac{1}{2} \sum_{j=1}^{n} \frac{1}{\sigma_j^2} \left(\sum_{i:y_i=0} (x_{ij} - \mu_{0j})^2 + \sum_{i:y_i=1} (x_{ij} - \mu_{1j})^2 \right) \end{aligned}$$

Since

$$\mathbf{x}^T \Sigma^{-1} \mathbf{x} = \mathbf{x}^T \begin{bmatrix} \sigma_1^{-2} & 0 & ... & 0 \\ 0 & \sigma_2^{-2} & ... & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & ... & \sigma_n^{-2} \end{bmatrix} \mathbf{x} = \begin{bmatrix} \sigma_1^{-2} x_1 & \sigma_2^{-2} x_2 & ... & \sigma_n^{-2} x_n \end{bmatrix} \mathbf{x} = \sum_{i=1}^{n} \sigma_i^{-2} x_i^2$$

So taking derivatives with respect to each σ_j,

$$\frac{\partial}{\partial \sigma_j} \log l(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m | y_1, y_2, ..., y_m, \mu_0, \mu_1, \sigma_1, \sigma_2, ..., \sigma_n) = 0$$

$$\frac{\partial}{\partial \sigma_j} - \frac{mn}{2} \log 2\pi - m \sum_{j=1}^{n} \log \sigma_j - \frac{1}{2} \sum_{j=1}^{n} \frac{1}{\sigma_j^2} \left(\sum_{i:y_i=0} (x_{ij} - \mu_{0j})^2 + \sum_{i:y_i=1} (x_{ij} - \mu_{1j})^2 \right) = 0$$

$$\frac{m}{\sigma_j} - \frac{1}{\sigma_j^3} \left(\sum_{i:y_i=0} (x_{ij} - \mu_{0j})^2 + \sum_{i:y_i=1} (x_{ij} - \mu_{1j})^2 \right) = 0$$

$$m\sigma_j^2 = \sum_{i:y_i=0} (x_{ij} - \mu_{0j})^2 + \sum_{i:y_i=1} (x_{ij} - \mu_{1j})^2 = 0$$

$$\sigma_j^2 = \left\{ \sum_{i=1}^{m} (1 - y_i)(x_{ij} - \mu_{0j})^2 + y_i (x_{ij} - \mu_{1j})^2 \right\} / m$$

The following matlab code uses the formulas above:

```
function  [mu0, mu1, sigmaSq ,correct] = lda_distVar()
X = load('wpbc_x.txt');
[m,n] = size(X);
Y = load('wpbc_yrecur.txt');

mu0 = (X' * (1-Y)) / sum(1-Y);
mu1 = (X' * Y) / sum(Y);

sigmaSq =  ( ((X' - mu0 * ones(1,m)) .^ 2) * (1 - Y) ...
+ ( (X' - mu1 * ones(1,m)) .^ 2) * Y ) / m;
p1 = sum(Y)/m;
p0 = 1 - p1;

hyp0 = -0.5 * ((X - ones(m,1) * mu0' ) .^ 2 ) * (1 ./ sigmaSq) + log(p0);
hyp1 = -0.5 * ((X - ones(m,1) * mu1' ) .^ 2 ) * (1 ./ sigmaSq) + log(p1);

correct = sum(abs((hyp0 > hyp1) - Y );
end
```

131 of the training examples are classified as correctly, and the MLE parameters are:

$$\mu_0 = \begin{bmatrix} 0.0171 \\ 0.0225 \\ 0.1128 \\ 0.9340 \\ 0.0001 \\ 0.0001 \\ 0.0002 \\ 0.0001 \\ 0.0002 \\ 0.0001 \\ 0.0006 \\ 0.0013 \\ 0.0041 \\ 0.0667 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0205 \\ 0.0304 \\ 0.1367 \\ 1.3290 \\ 0.0001 \\ 0.0004 \\ 0.0004 \\ 0.0002 \\ 0.0003 \\ 0.0001 \\ 0.0027 \\ 0.0027 \end{bmatrix} \times 10^3 \quad \mu_1 = \begin{bmatrix} 0.0183 \\ 0.0218 \\ 0.1211 \\ 1.0820 \\ 0.0001 \\ 0.0001 \\ 0.0002 \\ 0.0001 \\ 0.0002 \\ 0.0001 \\ 0.0007 \\ 0.0012 \\ 0.0047 \\ 0.0820 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0227 \\ 0.0296 \\ 0.1513 \\ 1.6358 \\ 0.0001 \\ 0.0004 \\ 0.0004 \\ 0.0002 \\ 0.0003 \\ 0.0001 \\ 0.0035 \\ 0.0049 \end{bmatrix} \times 10^3 \quad \sigma = \begin{bmatrix} 3.1208 \\ 4.3135 \\ 21.0836 \\ 346.5807 \\ 0.0126 \\ 0.0501 \\ 0.0707 \\ 0.0337 \\ 0.0275 \\ 0.0072 \\ 0.3070 \\ 0.5247 \\ 2.1639 \\ 47.4518 \\ 0.0030 \\ 0.0177 \\ 0.0209 \\ 0.0055 \\ 0.0096 \\ 0.0019 \\ 4.1299 \\ 6.0442 \\ 28.0668 \\ 570.8065 \\ 0.0220 \\ 0.1651 \\ 0.1743 \\ 0.0453 \\ 0.0737 \\ 0.0213 \\ 1.9162 \\ 5.3864 \end{bmatrix}$$