School of Electronics and Computer Science
University of Southampton

COMP6229(2017/18): Machine Learning Lab 3 $\boxed{\text{Not for assessment}}$

Issue	17 Oct 2016
Deadline	27 Oct 2016 (17:00)

Spend no more than 15 hours on this task. Please work independently. Please complete Lab 2 before starting this one.

1. Define a two class pattern classification problem in two dimensions, in which the two classes are Gaussian distributed with means $\boldsymbol{m}_1 = [0 \;\; 2]^t$ and $\boldsymbol{m}_1 = [1.7 \;\; 2.5]^t$, and have a common covariance matrix

$$C_1 = C_2 = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}.$$

Plot contours on the density by adapting the code snippet below.

```
numGrid = 50;
xRange = linspace(-6.0, 6.0, numGrid);
yRange = linspace(-6.0, 6.0, numGrid);
P1 = zeros(numGrid, numGrid);
P2 = P1;
for i=1:numGrid
    for j=1:numGrid;
        x = [yRange(j) xRange(i)]';
        P1(i,j) = mvnpdf(x', m1', C1);
        P2(i,j) = mvnpdf(x', m2', C2);
    end
end
Pmax = max(max([P1 P2]));
figure(1), clf,
contour(xRange, yRange, P1, [0.1*Pmax 0.5*Pmax 0.8*Pmax], 'LineWidth', 2);
hold on
plot(m1(1), m1(2), 'b*', 'LineWidth', 4);
contour(xRange, yRange, P2, [0.1*Pmax 0.5*Pmax 0.8*Pmax], 'LineWidth', 2);
plot(m2(1), m2(2), 'r*', 'LineWidth', 4);
```

2. Draw 200 samples from each of the two distributions and plot them on top of the contours.

```
N  = 200;
X1 = mvnrnd(m1, C1, N);
X2 = mvnrnd(m2, C2, N);
plot(X1(:,1),X1(:,2),'bx', X2(:,1),X2(:,2),'ro');grid on
```

3. Compute the Fisher Linear Discriminant direction using the means and covariance matrices of the problem, and plot the discriminant direction:

```
wF = inv(C1+C2)*(m1-m2);
xx = -6:0.1:6;
yy = xx*wF(2)/wF(1);
plot(xx,yy,'r', 'LineWidth', 2);
```

4. Project the data onto the Fisher discriminant directions and plot histograms of the distribution of projections:

```
p1 = X1*wF;
p2 = X2*wF;

plo = min([p1; p2]);
phi = max([p1; p2]);
[nn1, xx1] = hist(p1);
[nn2, xx2] = hist(p2);
hhi = max([nn1 nn2]);
figure(2),
subplot(211), bar(xx1, nn1);
axis([plo phi 0 hhi]);
title('Distribution of Projections', 'FontSize', 16)
ylabel('Class 1', 'FontSize', 14)
subplot(212), bar(xx2, nn2);
axis([plo phi 0 hhi])
ylabel('Class 2', 'FontSize', 14)
```

5. Compute a Receiver Operating Characteristic (ROC) curve, by sliding a decision threshold, and computing the True Positive and False Positive rates:

```
thmin = min([xx1 xx2]);
thmax = max([xx1 xx2]);

rocResolution = 50;
thRange = linspace(thmin, thmax, rocResolution);
ROC = zeros(rocResolution,2);
for jThreshold = 1: rocResolution
    threshold = thRange(jThreshold);
    tPos = length(find(p1 > threshold))*100 / N;
    fPos = length(find(p2 > threshold))*100 / N;
    ROC(jThreshold,:) = [fPos tPos];
end
figure(3), clf,
plot(ROC(:,1), ROC(:,2), 'b', 'LineWidth', 2);
axis([0 100 0 100]);
grid on, hold on
plot(0:100, 0:100, 'b-');
xlabel('False Positive', 'FontSize', 16)
ylabel('True Positive', 'FontSize', 16);
title('Receiver Operating Characteristic Curve', 'FontSize', 20);
```

6. Compute the area under the ROC curve (Hint: try `>help trapz`)

7. For a suitable choice of decision threshold, compute the classification accuracy.

8. Plot the ROC curve (on the same scale) for

   - A random direction (instead of the Fisher discriminant direction).
   - Projections onto the direction connecting the means of the two classes.

   Compute the area under the ROC curve for these two cases.

   (Optional: Since you need to call the ROC computing code several times, tidy up your code by writing this part as a function to which you pass the data as parameters and have the ROC curve returned as answer.)

9. Implement a nearest neighbour classifier (1-NN) on this data, and compare its accuracy with that of the Fisher Discriminant Analyzer.

```
% Nearest neighbour classifier
% (Caution: The following code is very inefficient -- why?)
X = [X1; X2];
N1 = size(X1, 1);
N2 = size(X2, 1);
y = [ones(N1,1); -1*ones(N2,1)];
d = zeros(N1+N2-1,1);
nCorrect = 0;
for jtst = 1:(N1+N2)
    % pick a point to test
    %
    xtst = X(jtst,:);
    ytst = y(jtst);

    % All others form the training set
    %
    jtr = setdiff(1:N1+N2, jtst);
    Xtr = X(jtr,:);
    ytr = y(jtr,1);

    % Compute all distances from test to training points
    %
    for i=1:(N1+N2-1)
        d(i) = norm(Xtr(i,:)-xtst);
    end

    % Which one is the closest?
    %
    [imin] = find(d == min(d));

    % Does the nearest point have the same class label?
    %
    if ( ytr(imin(1)) * ytst > 0 )
        nCorrect = nCorrect + 1;
    else
        disp('Incorrect classification');
    end
end

% Percentage correst
%
pCorrect = nCorrect*100/(N1+N2);
disp(['Nearest neighbour accuracy:  ' num2str(pCorrect)]);
```

10. For the dataset you have generated, construct a distance-to-mean classifire using (a) Euclidean distance and (b) Mahalanobis distance as distance measures and compare their classification accuracies.

11. For the above classification problem, compute and plot a three dimensional graph of the posterior probability of one of the two classes for the Bayes optimal classifier. Does the graph match your expectations from theory?

12. What do we expect the Baye's optimal class boundary to be, if C_1 and C_2 are not identical? Write out the algebra to show this. Change C_2 to a different covariance

matrix, *e.g.* $\boldsymbol{C}_2 = 1.5\boldsymbol{I}$ and illustrate the theoretical prediction.

Upload a report of **no more than four pages** on your work as a *pdf* file. Please write your name on it. Pay attention to style and clarity of presentation. Do not cut and paste equations (typeset them yourself); if you include a graphs, make sure the axes are labelled with a readable size font. Use LaTeXif you do not already do so.

Mahesan Niranjan October 2017