School of Electronics and Computer Science
University of Southampton

COMP6229(2017/18): Machine Learning Lab 2 Not for assessment

Issue	9 October 2017
Deadline	16 October 2017

This exercise supplements material taught in the lectures. It is a mandatory part of the module, but is not for assessment contributing to your grade in the module; spend about 10 hours on it in timetabled lab sessions and afterwards.

1. Draw a line: A straight in two dimensional coordinate geometry is defined as $a_1x + b_1y + c_1 = 0$. It intersects the x and y axes at $-c/b_1$ and $-c/a_1$ respectively. Therefore, points $(-c/b_1, 0)$ and $(0, -c/a_1)$ are points on this line. So, if we have $3x + 4y + 12 = 0$, we can plot this in MATLAB by:

```
plot([0 -4], [-3 0], 'b', 'LineWidth', 2);
axis([-5 5 -5 5]); grid on;
```

   The perpendicular distance from the origin to the straight line above is $-c/\sqrt{a_1^2 + b_1^2}$. Confirm this is correct on the plot you have drawn (simple visual inspection on the graph is fine).

   (In vector notation, we will write the equation of the straight line as $\boldsymbol{w}^t\,\boldsymbol{x} + b = 0$ in which the vector $\boldsymbol{w}$ contains information about the slope and the *bias* term b relates to the offset of the line. The distance from the origin to the line is thus $-b/|\boldsymbol{w}|$. We will return to this when we study support vector machine later in the module.)

2. Generate 100 samples each from two bi-variate Gaussian densities with distinct means $\boldsymbol{m}_1 = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$ and $\boldsymbol{m}_2 = \begin{bmatrix} 1.5 \\ 0.0 \end{bmatrix}$, and identical covariance matrix $\boldsymbol{C} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$.

   Hints:

   - See work in `Lab One` in which we used cholesky decomposition (`chol`) of the covariance matrix and turned samples from an isotropic distribution to samples from a distribution with the desired covariance matrix.
   - If `X` is an `N` by `2` array variable containing zero mean bivariate samples, one way of shifting the mean of the distribution is (also see `repmat`):

```
> m1 = [0 2];
> X1 = X + kron(ones(N,1), m1);
> help kron
```

   Plot the data as a scatter plot.

3. Compute the Bayes' optimal class boundary (see slides of lecture in Week 2) between the two distributions and draw it on the same graph (Hint: Use `> hold on` to retain a plot and display a second graph on top of it).

$w_1 \cdot x_1 + w_2 \cdot x_2 + b = 0$ $\quad \frac{w_1}{w_2} \cdot x_1 + x_2 + \frac{b}{w_2} = 0$

4. Implement a perceptron learning algorithm to classify this data and compare its solution with the Bayes' optimal boundary. Hint: please see Appendix for illustrative snippet of code (with no guarantees of correctness) which might help you get started. Explore how convergence of the learning algorithm changes with (a) different values of the learning rate; and (b) separation of the two classes.

Appendix

Perceptron Learning: Code snippet

```
> % Set up your own data in the following form
> % X: N by 2 matrix of data
> % y: class labels -1 or +1
> % include column of ones for bias
> X = [X ones(N,1)];
>
> % Separate into training and test sets (check: >> doc randperm)
> ii =  randperm(N);
> Xtr = X(ii(1:N/2),:);
> ytr = X(ii(1:N/2),:);
> Xts = X(ii(N/2+1:N),:);
> yts = y(ii(N/2+1:N),:);
>
> % initialize weights
> w = randn(3,1);
>
>
> % Error correcting learning
> eta = 0.001;
> for iter=1:500
>     j = ceil(rand*N/2);
>     if ( ytr(j)*Xtr(j,:)*w  < 0 )
>          w = w + eta*ytr(j)*Xtr(j,:)';
>     end
> end
>
> % Performance on test data
> yhts = Xts*w;
> disp([yts yhts])
> PercentageError = 100*sum(find(yts .* yhts < 0))/Nts;
>
```

Submit a short report (upload a pdf file into the ECS handin system) of no more than two pages (in total); please make sure your work is neat and include your name, student number and email in the submission.