EEL709: Minor Test II

March 20, 2015

Maximum Marks: 25

Note: Please follow a consistent notation to distinguish between vectors and scalars. The suggested notation is to use an underbar for vectors; if you wish to use something else, please specify it at the start of your answer script. In the below questions, vectors are indicated via bold font.

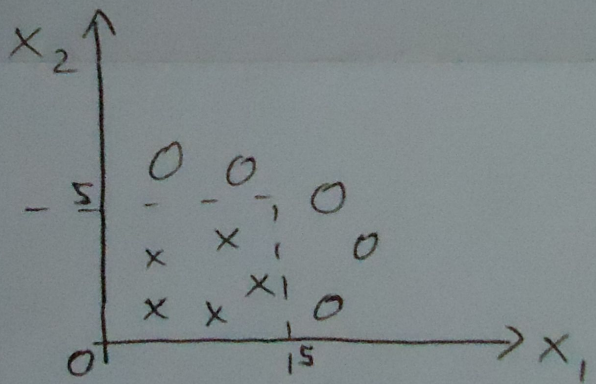
1. Are the following valid kernel functions? Justify either way, making use of Mercer's theorem or otherwise. **[1+1+3]**

   (a) $K(\mathbf{x}, \mathbf{x}') = \mathbf{x}$

   (b) $K(\mathbf{x}, \mathbf{x}') = f(\mathbf{x})f(\mathbf{x}') + f(\mathbf{x})g(\mathbf{x}') + g(\mathbf{x})f(\mathbf{x}') + g(\mathbf{x})g(\mathbf{x}')$, where $f()$ and $g()$ are real-valued functions.

   (c) $K(\mathbf{x}, \mathbf{x}') = e^{\frac{-||\mathbf{x}-\mathbf{x}'||^2}{\sigma^2}}$, the Gaussian kernel. [Hint: use $e^x = lim_{i\to\infty}\left(1 + x + ... + \frac{x^i}{i!}\right)$.]

2. Consider the following two-class data set:



   We would like to design a neural network whose decision boundary is (approximately) the dashed line depicted. Let us proceed as follows:

   (a) Design a single neuron (using the sigmoid activation function) which gives the decision boundary $x_1 = 5$. Depict the inputs and the weights clearly. You may use a bias input, i.e., have an x_0 whose value is fixed to 1. (Note that you are not being asked to obtain the weights via backpropagation or learning of any kind. You just have to write down the weights that will give the desired boundary.) **[2]**

   (b) Now design a neural network which gives (at least approximately) the dashed line as its decision boundary. Clearly draw the entire network, showing all input, hidden, and output units, and all weights. You may use a bias unit in each layer, i.e., $x_0 = 1$ for the input layer as well as $z_0 = 1$ for the hidden layer. Assume a sigmoid activation function for all neurons. Explain why your network gives the desired decision boundary. **[5]**

3. In using neural networks for binary classification, we specified a cross-entropy error function, of the form:

$$E(\mathbf{w}) = -\sum_{n=1}^{N} t_n \log y_n + (1 - t_n) log(1 - y_n)$$

Here $y_n = y(\mathbf{x}_n, \mathbf{w})$, the network output for the n^{th} training sample.

Now, suppose we attempt to use the same error function for regression, i.e., the t_n values are no longer binary. Assume, for simplicity, that $t_n \in [0, 1]$. Show that the cross-entropy error function will *still* be minimised when $y_n = t_n \; \forall n$. Derive an expression for this minimum value. (Hint: the minimum error will no longer be 0, unless $t_n \in \{0, 1\} \; \forall n$.) **[5]**

4. Consider the following data set:

Weight (kg)	Sex (M/F)	Has Diabetes? (Y/N)
82.7	F	Y
67.4	M	N
72.7	F	N
88.3	M	Y
70.3	M	N
79.7	F	Y
78.4	F	N
83.1	M	Y

We would like to construct a Naïve Bayes model to predict the occurrence of Diabetes, using Weight and Sex as features.

(a) Specify the structure of your model, using appropriate notation. Write down clearly how the full joint distribution factors (having made the Naïve Bayes assumption), and your assumed form of distribution for each factor, with parameters. You should choose appropriate parameterised distributions for each feature. How many parameters does your model have in total? **[3]**

(b) Give (derivation not required) general expressions for the maximum likelihood estimates for all your parameters, again using appropriate notation consistent with the above. Now plug in the values from the above data to obtain specific estimates for all parameters. **[3]**

(c) Based on these, can you comment on the usefulness or otherwise of the two features for the prediction task, in the context of a Naïve Bayes model? (Please link your answer to the estimates in part (b), rather than just inferring directly from the data.) **[2]**