# CS 380 Machine Learning - Spring 2011
# Homework Assignment 2
## Due at the start of class on Feb 21$^{st}$

1. (Linear Discriminants – 14 pts) [Duda et al., Ex. 5.1 a-b] Sketch two multimodal distributions (i.e., each class should have multiple areas of concentration) for which a linear discriminant could give excellent (or even optimal) classification accuracy. Sketch two unimodal distributions (i.e., each class is concentrated in a single area) for which even the best linear discriminant would give poor classification accuracy. You may need to look up the ideas "multimodal distributions" and "unimodal distributions" to complete this problem.

2. (Linear Separability – 6 pts) [Rephrase of Alpaydin Ex.10.9] Consider data characterized by a single attribute $x \in \mathbb{R}$ and class label $y \in \{C_1, C_2\}$ with the following class assignments:

$$y_i = \begin{cases} C_1 & \text{if } x < 2 \text{ or } x > 4 \\ C_2 & \text{otherwise} \end{cases} \quad (1)$$

   Describe how you can use a linear discriminant to separate the two classes.

3. (Gradient Descent – 10 pts) Let $k$ be a counter for the iterations of gradient descent. What are the implications of using a constant value for $\eta_k$ in gradient descent? What are the implications for setting $\eta_k$ as a function of $k$?

4. (Support Vectors – 15 pts) For an SVM, if we remove one of the support vectors from the training set, does the size of the maximum margin decrease, stay the same, or increase for that dataset? Why? Also justify your answer by providing a simple dataset (no more than 2-D) in which you identify the support vectors, draw the location of the maximum margin hyperplane, remove one of the support vectors, and draw the location of the resulting maximum margin hyperplane.

5. (Cubic Kernels – 15 pts) We showed in class that the quadratic kernel $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \mathbf{x}_j)^2$ was equivalent to mapping each $\mathbf{x}$ into a higher dimensional space where

$$\Phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1 x_2)$$

   for the case where $\mathbf{x} = (x_1, x_2)$. Now consider the cubic kernel $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i, \mathbf{x}_j + 1)^3$. Note that this kernel adds 1 to the dot product. What is the corresponding $\Phi$ function, again for the case where $\mathbf{x} = (x_1, x_2)$?

6. (Non-linear kernels – 15 pts) Consider the 2-bit XOR problem for which the entire instance space is as follows:

| $y$ | $x_1$ | $x_2$ |
|---|---|---|
| -1 | -1 | -1 |
| +1 | -1 | 1 |
| +1 | 1 | -1 |
| -1 | 1 | 1 |

   These instances are not linearly separable, but they are separable with a polynomial kernel. Recall that the polynomial kernel is of the form $K(\mathbf{x_i}, \mathbf{x_j}) = (\mathbf{x_i x_j} + \mathbf{c})^{\mathbf{d}}$ where $c$ and $d$ are integers. Select values for $c$ and $d$ that yield a space in which the instances above are linearly separable. Write down the mapping $\Phi$ to which this kernel corresponds, write down $\Phi(x)$ for each instance above, and write down the parameters of any hyperplane in the expanded space that perfectly classifies the instances.

7. (SVM tools – 25 pts) In this exercise you will get some experience with a state-of-the-art SVM package, and will compare the performance of various kernels. You will need to do the following:

   - Download SVMlight from `http://www.cs.cornell.edu/People/tj/svm_light/`. This web page has links to source code and binaries, and contains information on how to build and run the various SVMlight tools.
   - Download the ionosphere dataset from the course web page. This dataset comes from the UC Irvine Machine Learning Repository. The file ionosphere.names describes the contents of the dataset. The file ionosphere.data contains the actual data in a format that is accepted by SVMlight. You'll need to manually divide the data into training and testing sets. One split will suffice.
   - Are the data linearly separable? Describe how you used SVMlight to determine this.
   - Compare the performance of the SVM using each of the following kernels - linear, polynomial, radial basis. The polynomial and radial basis kernels each take parameters. Experiment with various values of these parameters and $C$, which controls the tradeoff between training error and margin. Write a short report detailing your experiments. Be sure to report the margin and number of support vectors for various settings of the parameters. What kernel performed the best? What does this say, if anything, about the data?