CS 194-10, Fall 2011
Assignment 2

1. (8 pts) In this question we briefly review the expressiveness of kernels.

   (a) (Question 18.17 from Russell & Norvig) Construct a support vector machine that computes the XOR function. Use values of +1 and -1 (instead of 1 and 0) for both inputs and outputs, so that an example looks like $([-1, 1], 1)$ or $([-1, -1], -1)$. Map the input $[x_1, x_2]$ into a space consisting of x_1 and x_1x_2. Draw the four input points in this space, and the maximal margin separator. What is the margin? Now draw the separating line back in the original Euclidian input space.

   (b) (Question 18.16 from RN) Recall that the equation of the circle in the 2-dimensional plane is $(x_1 - a)^2 + (x_2 - b)^2 - r^2 = 0$. Expand out the formula and show that every circular region is linearly separable from the rest of the plane in the feature space (x_1, x_2, x_1^2, x_2^2).

   (c) Recall that the equation of an ellipse in the 2-dimensional plane is $c(x_1 - a)^2 + d(x_2 - b)^2 - 1 = 0$. Show that an SVM using the polynomial kernel of degree 2, $K(\mathbf{u}, \mathbf{v}) = (1 + \mathbf{u} \cdot \mathbf{v})^2$, is equivalent to a linear SVM in the feature space $(1, x_1, x_2, x_1^2, x_2^2, x_1x_2)$ and hence that SVMs with this kernel can separate any elliptic region from the rest of the plane.

2. (12 pts) Logistic regression is a method of fitting a probabilistic classifier that gives soft linear thresholds. (See Russell & Norvig, Section 18.6.4.) It is common to use logistic regression with an objective function consisting of the negative log probability of the data plus an L_2 regularizer:

$$L(\mathbf{w}) = \sum_{i=1}^{N} \log\left(\frac{1}{1 + e^{y_i(\mathbf{w}^T\mathbf{x}_i + b)}}\right) + \lambda||\mathbf{w}||_2^2$$

   (Here $\mathbf{w}$ does not include the "extra" weight w_0.)

   (a) Find the partial derivatives $\frac{\partial L}{\partial w_j}$.

   (b) Find the partial second derivatives $\frac{\partial^2 L}{\partial w_j \partial w_k}$.

   (c) From these results, show that $L(\mathbf{w})$ is a convex function.
   Hint: A function L is convex if its Hessian (the matrix $\mathbf{H}$ of second derivatives with elements $H_{j,k} = \frac{\partial^2 L}{\partial w_j \partial w_k}$) is positive semi-definite (PSD). A matrix $\mathbf{H}$ is PSD if and only if

$$\mathbf{a}^T\mathbf{H}\mathbf{a} \equiv \sum_{j,k} a_j a_k H_{j,k} \geq 0$$

   for all real vectors $\mathbf{a}$.

3. (8 pts) Consider the following training data,

class	x_1	x_2
+	1	1
+	2	2
+	2	0
−	0	0
−	1	0
−	0	1

   (a) Plot these six training points. Are the classes $\{+, -\}$ linearly separable?

(b) Construct the weight vector of the maximum margin hyperplane by inspection and identify the support vectors.

(c) If you remove one of the support vectors does the size of the optimal margin decrease, stay the same, or increase?

(d) (Extra Credit) Is your answer to (c) also true for any dataset? Provide a counterexample or give a short proof.

4. (12 pts) Consider a dataset with 3 points in 1-D:

(class)	x
$+$	0
$-$	-1
$-$	$+1$

(a) Are the classes $\{+,-\}$ linearly separable?

(b) Consider mapping each point to 3-D using new feature vectors $\boldsymbol{\phi}(x) = [1, \sqrt{2}x, x^2]^T$. Are the classes now linearly separable? If so, find a separating hyperplane.

(c) Define a class variable $y_i \in \{-1,+1\}$ which denotes the class of x_i and let $\mathbf{w} = (w_1, w_2, w_3)^T$. The max-margin SVM classifier solves the following problem

$$\min_{\mathbf{w},b} \tfrac{1}{2}||\mathbf{w}||_2^2 \text{ s.t.} \tag{1}$$

$$y_i(\mathbf{w}^T\boldsymbol{\phi}(x_i) + b) \geq 1,\ i = 1,2,3 \tag{2}$$

Using the method of Lagrange multipliers show that the solution is $\hat{\mathbf{w}} = (0,0,1)^T$, $b = 1$ and the margin is $\frac{1}{||\hat{\mathbf{w}}||_2}$.

(d) Show that the solution remains the same if the constraints are changed to

$$y_i(\mathbf{w}^T\boldsymbol{\phi}(x_i) + b) \geq \rho,\ i = 1,2,3$$

for any $\rho \geq 1$.

(e) (Extra Credit) Is your answer to (d) also true for any dataset and $\rho \geq 1$? Provide a counterexample or give a short proof.

5. (60 pts) In this question you will attempt to classify the **phase** of seismic waves by measurements of the arriving signal. This is a difficult problem that we will return to later in the course using other methods; however, it should not be hard to improve on the currently deployed phase classifier, whose overall accuracy is lower than that of the classifier that always chooses the most common class (P).

This assignment is also about learning to use an off-the-shelf package as opposed to rolling your own. We will use the SVM Light package by Thorsten Joachims (Cornell). Start by downloading and unzipping the relevant files from **http://svmlight.joachims.org/** and proceed by running the "getting started" example and reading the specifications on the site. (Note that SVM Light may run slowly on large data sets. You may want to work initially with small subsamples to make sure everything is behaving itself before moving up to larger sizes.)

Next, download the **training data** for this assignment.

(a) The first thing to do when you get a new data set to work on is *look at it.*[1] (Witten's book has some good material on this, and Weka provides nice tools.) Begin by writing a function

[1] In statistics, this is called *exploratory data analysis*; so if someone asks you what you are doing when they see you staring at the screen with a glazed expression, say "exploratory data analysis."

`discrete-histogram` that takes a data file and a discrete attribute (this could be an attribute explicitly recorded in the data record, or one computed from those attributes) and returns a list of value/absolute frequency/relative frequency triples. Apply this to examine the `phase` and `iphase` attributes. Here `phase` is the true phase as assigned by expert seismic analysts and `iphase` is the initial phase assigned by the UN's automated classifier (a neural network, as it happens). For `phase`, the phases of interest are Lg, P, PKP, Pg, Pn, Rg, S, Sn. For `iphase`, the phases include N, Px, Sx, tx as well.

(b) Next, write a function `confusion-matrix` that takes two attributes X and Z and returns a matrix $\mathbf{C}$ indexed by the values of each attribute such that C_{kl} is the fraction of records that have $Z = z_l$ out of all records having $X = x_k$. Apply your function to compute the confusion matrix for `phase` and `iphase`. Then use the results to calculate the accuracy of `iphase` as follows: If the iphase exactly matches the phase, it is considered correct. Additionally an iphase of Sx is considered correct if the true phase is S or Sn, and an iphase of Px is considered correct if the true phase is P, PKP, Pg, or Pn.

(c) Determine the station IDs (`sta`) of the top 10 stations in the data set (i.e., those with the most detection records).

(d) A common task in many real applications of machine learning is *data munging*, i.e., messing with the data to get it into the right input format, drop irrelevant features, etc. This assignment is no exception. You could do a lot of this work using, say, emacs keyboard macros, but that work would have to repeated from scratch each time. If you build the right tools, a lot of the busy-work of dealing with multiple formats, multiple tasks, multiple data sets, etc., is eliminated.

For this assignment, to enable comparative evaluation on our hidden test set, we will require the following input features *in this order* (the numbers in the brackets are the indices of the corresponding features): ddet60 (58), dtime60 (59), hmxmn (39), htov0.25 (48), htov0.5 (49), htov1 (50), htov2 (51), htov4 (52), hvrat (38), hvratp (37), inang1 (44), inang3 (40), per (8), plans (35), rect (34), arrival_slow (4), ddet100 (62), dtime100 (63), ddet300 (66), dtime300 (67).

With this input feature ordering, generate 10 new data files, one per top-10 station, suitable for use with SVM Light. The binary target variable should distinguish Px phases (P, PKP, Pg, Pn) as +1 from the other phases as −1. Use SVM Light with default parameters to obtain ten classifiers; estimate their accuracy and compare to the accuracy of `iphase` for the same stations on this binary task.

(e) Choose an optimal c parameter (controlled by the $-c$ option) for each of the 10 stations and determine whether it results in improved accuracy.

(f) We can solve the multi-class problem of predicting the phase by learning 8 binary classifiers, so-called "one-against-all" classifiers. Then, the class for a new example is given by whichever learned model (prior to the ±1 mapping) has the highest value. Use this approach to determine the best multi-class classifier you can for each station; report your estimated accuracy for each station and averaged over all 10 stations, and give the averaged confusion matrix.

(g) Submit your learned models as 80 (sorry about that) files, each named `model_`s`_`p, where s is the station ID and p is the phase (case-sensitive). With the right tools, this shouldn't be too painful.