MACHINE LEARNING COMS 4771, HOMEWORK 3
Assigned February 28, 2013. Due March 28, 2013 before 1:00pm.

Here are the instructions for submitting your homework. Archive/package all of the files you are submitting as a single tarball or zip archive: “UNI-HW3.tar.gz” or “UNI-HW3.zip”. For example, a compressed tarball would be “ir2322-HW3.tar.gz”. Your homework should contain:

- a writeup (PDF, TXT, or PostScript)
- code (as Matlab M files, shorter code is generally better but include comments)
- any figures/pictures not included in the writeup (PDF or PostScript)
- if you have special instructions, include them as a plain text file called README.txt.

Submit your homework through CourseWorks by doing the following:

1 Log into https://courseworks.columbia.edu/
2 Click “Assignments” on the left side.
3 Choose the appropriate HW Folder to submit to.
4 Use the filename “yourUNI-HW3.tar.gz” or “yourUNI-HW3.zip”.
5 Make sure that the “title” is yourUNI-HW3 (example: zz9999-HW3).
6 Add any special instructions in both the description and the README.txt.
7 Click “Submit” at the bottom to upload your file.
8 If you submit multiple times, only the last submission prior to the deadline will count.
9 If something goes wrong, ask the TAs for help.
10 In a dire emergency, if nothing else works, send your homework to the TAs.

Handwritten writeups are not allowed without prior approval.

All your code should be written in Matlab (other languages may be used only with prior permission from an instructor). Please submit all your souce files, each function in a separate file. Clearly denote what each function does, its inputs and outputs, and to which problem it belongs. Do not resubmit code or data provided to you. Do not submit code written by others. Identical submissions will be detected and both parties will get zero credit. Sample code is available on the Tutorials web page. Datasets are available from the Handouts web page. You may include figures directly in your write-up, or separately and refer to them by filename.

Each homework counts equally towards your grade (other than your worst which will be dropped). Points shown here for each problem indicate relative weights for this specific homework. As always, up to 10% bonus points are available for exceptional, relevant work going beyond what is asked.

1 Problem 1 (15 points)

Statistical Learning Theory (intro):

1.1 Convergence Modes (10 points)

Recall the two definitions given for convergence modes (slide 8.15). Provide an example of your own choosing of random variables r_l and r_0 such that convergence in probability is guaranteed as $l \to \infty$ but we do not have almost sure convergence.

Provide a plot or sketch to support your explanation.

1.2 Strict Consistency (5 points)

Recall the definition of strict consistency (slide 8.20). Explain why trivial cases of consistency are excluded. You may add an illustrative plot (if you find it simpler).
Hint: in class we emphasized why the convergence conditions in the definition are “fair”.

2 Problem 2 (20 points)

Chernoff Bounds: Recall the notation used in class to explain Chernoff bounds (slides 9.12, 10.14).

2.1 Estimate Close to Probability (10 points)

Suppose M is the number of trials required to ensure that, with high confidence, the estimate for p is within a multiplicative factor of 3 of the true probability. Give a lower bound on M.
In other words, given δ, for $M \geq ?$, $\frac{p}{3} \leq \hat{p} \leq 3p$ with confidence (probability) of at least $1 - \delta$.

2.2 Bound in Terms of Standard Deviation (10 points)

Show that the multiplicative Chernoff bound can be stated in terms of the standard deviation σ of the random variable S. Specifically, show that:

$$\mathbf{Pr}[|S - E[S]| \geq k\sigma] \leq 2\exp\{-\frac{k^2}{6}\} \tag{1}$$

3 Problem 3 (20 points)

VC Dimension: We consider sets of indicator functions for both parts of this problem. If you wish to add drawings to illustarte your proof, you may do this by hand (scan and make them legible however).

3.1 Rectangles (10 points)

Find the VC dimension of the admissible set (of functions) consisting of single *axis-aligned* rectangles in the plane, where the internal sign is positive and outside the rectangle it is negative. Note: a rectangle is axis-aligned if its sides are aligned with the axes. Fully justify your answer.

3.2 Convex Polygons (10 points)

Find the VC dimension of the admissible set (of functions) consisting of convex polygons with d sides in the plane, where the internal sign is positive and outside the polygon it is negative. Note: your proof/result should hold for every d (integer). Fully justify your answer.

4 Problem 4 (25 points)

SVM You will build an SVM to classify data and use cross-validation to find the best SVM kernel and regularization value. Try different polynomials and RBF kernels (varying polynomial order from 1 to 5) and varying sigma in the RBF kernel. Also, try different values of C in the SVM. First, extract the support vector machine Matlab code from Steve Gunn's software package here:

```
http://www.isis.ecs.soton.ac.uk/resources/svminfo/
In svc.m replace [alpha lambda how] = qp(...);
with             [alpha lambda how] = quadprog(H,c,[],[],A,b,vlb,vub,x0);
```

Clearly denote the various components and the function calls or scripts that execute your Matlab functions. Note, to save the current figure in Matlab as a postscript file you can type:

```
print -depsc filename.eps
```

To test your SVM, you will build a simple object recognition system. Download the data file shoesducks.mat from the Handouts web page. This loads a matrix X of size 144 by 768 and a vector Y of size 144 by 1. The X matrix contains 144 images, each is a vector of length 768. These vectors are not quite images but rather are the contour profile of the top part of the object (either a shoe or a duck). To view them as contours, just type plot(X(4,:)) which will plot a contour of the 4th object (a duck in this case with Y(4) being 1). There are a total of 72 images of ducks and 72 images of shoes at different 3D rotations. Train your SVM on half of the examples and test on the other half (or other random subsets of the examples if you see fit). Show performance of the SVM for linear, polynomial and RBF kernels with different settings of the polynomial order, sigma and C value. Try to hunt for a good setting of these parameters to obtain high recognition accuracy.

5 Problem 5 (15 points)

Lagranage Multipliers: Consider the following optimization problem,

$$\begin{aligned}
\min_{\mathbf{w},b,\mathbf{w}^*,b^*,\xi^*} \quad & \sum_{i=1}^{\ell} [(\mathbf{w}^* \cdot \mathbf{x}_i^*) + b^* + \xi_i^*] \\
& \forall i,\ y_i\left[(\mathbf{w} \cdot \mathbf{x}_i) + b\right] \geq 1 - [(\mathbf{w}^* \cdot \mathbf{x}_i^*) + b^*] \\
& \forall i,\ (\mathbf{w}^* \cdot \mathbf{x}_i^*) + b^* + \xi_i^* \geq 0,\ \xi_i^* \geq 0 \\
& (\mathbf{w} \cdot \mathbf{w}) + \gamma(\mathbf{w}^* \cdot \mathbf{w}^*) \leq \frac{1}{\Delta^2}
\end{aligned} \tag{2}$$

where we assume that $\gamma > 0, \Delta > 0$ are some positive constants, $\mathbf{x}_i, \mathbf{x}_i^*, \mathbf{w}, \mathbf{w}^*$ are n-dimensional vectors, and b, b^* are constants.

1. Construct (define) the Lagrangian function L for the problem above.
2. Find the stationary point of L. You should minimize L with respect to some variables and maximize with respect to others.
3. Write down the conditions obtained from the partial derivatives of L.

BONUS (10 points, not easy!): Can you provide a brief explanation how the form above is related to the SVM optimization problem? In other words, intuitiviely (not technically), what could be the point behind using this particular form?