MACHINE LEARNING COMS 4771, HOMEWORK 1
Assigned January 31, 2013. Due February 14, 2013 before 1:10pm.

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-HW1.tar.gz" or "UNI-HW1.zip". For example, a compressed tarball would be "ir2322-HW1.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 "CLASS FILES" on the left side, then click "POST FILE" at the top.
3 Choose the Dropbox as the folder to submit to.
4 Use the filename "yourUNI-HW1.tar.gz" or "yourUNI-HW1.zip".
5 Make sure that the "title" is yourUNI-HW1 (example: zz9999-HW1).
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

As in lecture 2, in this problem we consider tossing a coin with two sides labeled H and T. Assume all tosses are iid with $p(H) = \mu$.

1.1 Discrete possible parameter values (10 points)

There are 3 possible values for μ. It could be fair $\mu = \frac{1}{2}$, biased tails $\mu = \frac{1}{4}$, or biased heads $\mu = \frac{3}{4}$. Assume our prior is that each of these possibilities is equally likely.

Looking forward, what is the minimum number of tosses we'd need to see in order to conclude that $p(\mu = \frac{1}{2}) > \frac{1}{2}$ (strictly greater)? Give an example of a sequence of tosses with this length which would lead to this conclusion. Prove your results.

Instead, assume no tosses have yet occurred. What is the minimum number of tosses we'd need to see in order to conclude that $p(\mu = \frac{3}{4}) > \frac{1}{2}$ (strictly greater)? Give an example of a sequence of tosses with this length which would lead to this conclusion. Prove your results.

1.2 Continuous possible parameter values (10 points)

You may find results on the Beta distribution given in lecture 2 useful and may quote them without proof.

Consider two possible prior distributions: (A) $\mu \sim$ Uniform$[0, 1]$; (B) we have some reason to think the coin is likely to be fair so assume μ has a probability distribution which has the form of a concave parabola with its maximum at $\frac{1}{2}$ and falls to 0 at 0 and 1.

Along with 2 possible realizations: (1) $\mathcal{D}_1 = \{H, T\}$; (2) $\mathcal{D}_2 = \{T, T, T\}$.

For each of the 4 combinations of priors and realizations, derive with proof:

- $p(H)$ given the prior
- the posterior distribution $p(\mu|\mathcal{D})$ [ensure this is properly normalized]
- the maximum likelihood estimate μ_{ML} given the data $\mathcal{D}$
- the MAP estimate μ_{MAP} given the data $\mathcal{D}$
- $p(H|\mathcal{D})$, i.e. the full Bayesian probability that the next toss is H
- the variance of the posterior distribution $p(\mu|\mathcal{D})$

Give one reason why the maximum likelihood estimate might not be good in this context.

2 Problem 2 (10 points)

A coin is tossed and then based on the outcome, a single sample data point x is taken from one of two one-dimensional normal distributions. Your prior distribution on the coin is that $p(H) = p$, $p(T) = 1 - p$ for some fixed $p \in [0, 1]$. If the toss is H then the sample point $\sim N(\mu_H, \sigma^2)$ and if it's T then the sample $\sim N(\mu_T, \sigma^2)$, so the normal distributions have different means but the same variance.

Using Bayesian decision theory, derive the decision boundary on the observed sample point x that minimizes the misclassification rate for estimating whether the coin landed H or T given the observation x.

Under what conditions can the decision boundary *not* lie between the two means?

3 Problem 3 (10 points)

The entropy of a distribution $p(x)$ is given by the integral over all possible x:

$H(x) = -\int p(x) \ln p(x) \, \mathrm{d}x$

Show that the entropy of the multivariate Gaussian $N(x|\mu, \Sigma)$ is given by:

$H(x) = \frac{1}{2} \ln |\Sigma| + \frac{D}{2}(1 + \ln(2\pi))$

where D is the dimensionality of x.

4 Problem 4 (30 points)

Cross validation for Polynomial Fitting: Recall the problem of over-fitting. In particular consider the example of regression we introduced in class. Assume the feature space $X = [0, 1]$ is just the unit interval, the label space is $\Re$ and consider the squared error loss function. Suppose you want to carry out prediction in this setting, that is, you want to consider a mapping $f : [0, 1] \to \Re$ that predicts a label given the corresponding feature. In this homework we are going to consider models that are polynomials. In other words, at the end of the learning procedure you should have a prediction rule of the form: $f(x) = w_0 + w_1 x + w_2 x^2 + \ldots + w_d x^d$, where d is the degree of the polynomial, that you need to choose adequately.

Consider the dataset in the file **regression_dataset.txt**. The first column corresponds to the features $\{X_i\}_{i=1}^n$ and the second column corresponds to the labels $\{Y_i\}_{i=1}^n$.

1. Fit a polynomial model to this data. Experiment with various choices for d, the degree of the polynomial. Which value(s) of d seems somewhat more reasonable, and why?
2. Show a few plots illustrating your work, and for each also show the values of the weights w. What can be observed by considering both the plots and the weights?
3. Suggest two or more automated procedures to select the complexity of the model (i.e., the degree of the polynomial) based on the data.
4. Implement and experiment with these procedures and comment on your results. Include a few plots illustrating your work.

5 Problem 5 (20 points)

RBF Basis Regression:

1. Modify the Matlab code for **polyreg.m** so that it performs RBF basis curve fitting instead of polynomial regression.
2. Fit the data in **"dataset1.mat"** with the first 100 points as training and the second 100 points as testing, setting the RBFs sigma parameter equal to 1.0.

3. Compute and show the training and testing error for this model, and show the fit graphically by saving the Matlab plot of $f(x)$ overlayed on the data.

4. Next try to experiment with the sigma parameter. How does sigma parameter affect overfitting/underfitting? Support your observations with appropriate graphical illustrations.