Machine Learning 10-701 November 4, 2013
http://alex.smola.org/teaching/cmu2013-10-701x/ due November 13, 2013
Carnegie Mellon University

Homework 3

START HERE: Instructions

- The homework is due at 10:30am on Wednesday November 13, 2013. Anything that is received after that time will not be considered.
- Answers to everything will be submitted electronically through the submission website: http://alex.smola.org/teaching/cmu2013-10-701x/submission.html. Let us know if you have any problems.
- **Read this before handwriting or LaTeXing your solutions:** In HW1 some students reported difficulty with submitting large image files to the discussion/handin server. So, we recommend that, to the extent possible, you should not handwrite or LaTeXyour solutions; instead use "plain text" or "markup text" mode and type or paste your solutions into the compose box. We will make our best effort to provide support for image-based handins, but until further notice they should be considered an experimental feature.
- Collaboration on solving the homework is allowed (after you have thought about the problems on your own). However, when you do collaborate, you should list your collaborators! You might also have gotten some inspiration from resources (books or online etc...). This might be OK only after you have tried to solve the problem, and couldn't. In such a case, you should cite your resources.
- If you do collaborate with someone or use a book or website, you are expected to write up your solution independently. That is, close the book and all of your notes before starting to write up your solution.

1 Probability Inequalities [Jing; 25 pts]

In machine learning we use algorithms to make predictions. It is often important to characterize the probability of making an incorrect prediction. In this question, we will investigate different probability inequalities and compare their ability to bound the probability that the value of X is far from its expectation. We will first write down the bounds for a setting where the samples are coin flips with a biased coin. Then, we will plot the value of the bounds as we vary the sample size n.

For this problem, you may use any software environment that you like; some good ones include Matlab, Octave, SciPy, and R. You do not need to hand in your code, only the plots, derivations, and explanations requested below.

1.1 Finding the Probability Inequalities for a Biased Coin

Consider the following experiment: we have a biased coin. We want to determine whether it is biased heads or tails. To do so we use the following simple algorithm: Flip the coin n times, and record h, the number of heads. If $h > \frac{n}{2}$ heads, we predict the coin is biased heads. If $h < \frac{n}{2}$ heads, we predict the coin is biased tails. However, we would like to know the probability that we make the *wrong* prediction. Intuitively as n increases, the probability of error should decrease. Because it is difficult to determine the exact probability of error, we use probability inequalities to bound the probability of error.

A biased coin lands heads with probability $\frac{1}{5}$ each time it is flipped (i.e. the coin is biased tails). Let $X_1, \ldots, X_n$ represent n consecutive coin flips with $X_i = 0$ if the coin lands tails and $X_i = 1$ if the coin lands heads. Now, let $\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$. Note that we will incorrectly predict that the coin is biased heads if $\bar{X}_n > \frac{1}{2}$.

Homework 3

(a) Using Markov's Inequality, give an upper bound on the probability that the coin lands heads at least 50% of the time across n flips. The general form of the Markov's Inequality is given below.

Markov's Inequality
Suppose that X is a random variable taking only non-negative values. Then, for any $a > 0$ we have

$$P(X \geq a) \leq \frac{E[X]}{a}.$$

(b) Assuming that $X_1, \ldots, X_n$ are iid observations, using Chebyshev's Inequality, give an upper bound on the probability that the coin lands heads at least 50% of the time across n flips.

Hint: For this, you will need to compute the variance of a single coin flip. Note that

$$\mathsf{Var}[\bar{X}_n] = \left(\frac{1}{n}\right)^2 \sum_{i=1}^{n} \mathsf{Var}[X_i].$$

Chebyshev's Inequality
Let X be a random variable with finite mean $E[X] = \mu$ and $\mathsf{Var}[X] = \sigma^2$. Then, for any $k > 0$ we have

$$P(|X - \mu| \geq k) \leq \frac{\sigma^2}{k^2}$$

(c) Next, use Hoeffding's Inequality to give an upper bound on the probability that the coin lands heads at least 50% of the time across n flips.

Hoeffding's Inequality
Let $X_1, \ldots, X_n$ be iid observations such that $E[X_i] = \mu$ and $a \leq X_i \leq b$. Then for any $\epsilon \geq 0$, we have

$$P(\left|\bar{X}_n - \mu\right| \geq \epsilon) \leq 2e^{-2n\epsilon^2/(b-a)^2}$$

(d) On a single plot, vary the sample size from $n = 1$ to $n = 100$ and plot the Markov, Chebyshev and Hoeffding bounds. Which one is the tightest?

(e) **Sanity Check** On the same plot as above, plot the exact probability using the binomial distribution and check that Markov, Chebyshev and Hoeffding bounds that you've plotted are indeed upper bounds. Make sure you hand in your plot with your solution (a single plot for this part and the previous part together).

Hint: In Matlab, `1-binocdf(floor(0.5.*n-0.5), n, 0.2)` will give you the exact probabilities.

(f) On a separate figure, plot (d) and (e) on $\log(\text{Probability})$ vs. n axes.

(g) On a separate figure, plot (d) and (e) on $\log(\text{Probability})$ vs. $\log(n)$ axes.

(h) For each of the three figures that you have plotted (original space, semi-log space, and log-log space), identify which bounds (Markov, Chebyshev, or Hoeffding) generate linear or near-linear relationships in each space. Using the functional forms of these bounds, explain why you see the linear relationships you identified, and where possible, write down the slope of each line.

2 Choosing an SVM kernel [Leila; 25 pts]

In this problem, you will learn to use LIBSVM for classification and regression. Your goal will be to select the right kernel to build a model for each of the provided datasets. Although this question requires implementation, you will not be required to submit your code on Autolab; instead you should submit just the plots and explanations requested below to the coursework server.

The handout with the data for this question is at http://alex.smola.org/teaching/cmu2013-10-701x/assignments/assignment_3_handout.zip.

2.1 LIBSVM with Gaussian Kernels

(a) Install LIBSVM from http://www.csie.ntu.edu.tw/~cjlin/libsvm/. You many use any language or IDE which has an interface to LIBSVM.
If you want to use MATLAB you can either compile the mex files yourself and then add the files "svmtrain.mex..." and "svmpredict.mex..." to your working directory or if you find it problematic you can download the binary files for Mac 32-bit, Mac 64-bit, Linux 32-bit, Linux 64-bit and Windows 64-bit. To see the help type "svmtrain" or "svmpredict" in the command window (MATLAB has its own implementation of svmtrain, so typing "help svmtrain" will give you the wrong function.)

(b) Using the data in `artificial.csv`, train a Gaussian (radial basis) Kernel for each of the following values of the parameter γ

$$\gamma \in \{1, 10, 100, 1000, 10000\}$$

The γ parameter of the Gaussian kernel in LIBSVM dictates how wide or narrow the kernel is. For instance $\gamma = 0.1$ represents a very smooth kernel, while $\gamma = 10000$ is a narrow one. A narrow kernel can better separate the training points, but it is more prone to overfitting.For each of the models obtained

- Plot the decision boundary on top of a scatterplot of the data. You can create your own function (for which you might use MATLAB's `contour` function), or modify the function `svmtoy` if you like: http://home.caltech.edu/~htlin/program/libsvm/doc/svmtoy.m. Please make the boundary lines easy to see. The model structure returned by svmtrain also contains the support vectors, you should label the support vectors differently from the other points.
- State—based on the plot—whether the model overfits, underfits, or performs well.

(c) Perform 10-fold cross validationto pick the appropriate γ for this dataset. Show how the testing and training errors averaged across folds change with γ. What's the value of γ you would choose and what are its corresponding test/training errors?

(d) Let's assume you were submitting a paper on the usefulness of the Gaussian Kernel. Is it ok to report the test error you obtained after 10-fold cross validation as the test error of your model? Why or why not?

For 2.1(b) use *all* the data to train the model. For 2.1(c) , split the data into 10 folds, and use 9 of the folds for training and the remaining fold for testing each time. Keep the data in the same order it appears in the file: i.e., the first 10% is the first fold, the next 10% is the next fold, etc. Do NOT randomly shuffle the data.

2.2 Unbiased Model Evaluation

Compare the following two ways of model evaluation on `dataset2.csv`. Use the same kernel described in 2.1, but try values

$$\gamma \in \{1, 5, 10, 50, 100, 500, 1000, 5000, 10000\}.$$

- (1-stage CV) Split your data into 2 sets. Do 10-fold CV separately on each set and average the two cross-validation test results. Report the best kernel and the test error obtained by averaging over the two sets.
- (2-stage CV) Split your data into 2 sets. Do 10-fold CV on Set_1 and pick the best kernel K_1. Compute the test error of K_1 on Set_2, thus obtaining the value E_1. Do 10-fold CV on Set_2; pick the best kernel K_2. Evaluate K_2 on Set_1, obtaining the test error E_2. Report $E = \frac{E_1 + E_2}{2}$.

Compare the two values. Which one do you think is a more accurate estimator of how your model would behave on hold-out data?

Do K_1 and K_2 match in the 2-stage CV?

Suppose that K_1 and K_2 don't match in the 2-stage CV. What conclusion can you draw? In this case, how would you report the results about error rate and kernel selection in a paper?