10-701 Machine Learning - Spring 2012
Problem Set 1

Out: February 1st
In: February 15th

TA: Yang Xu (yx1@cs.cmu.edu)
School of Computer Science, Carnegie Mellon University

Homework will be done individually: each student must hand in their own answers. It is acceptable for students to collaborate in figuring out answers and helping each other solve the problems. We will be assuming that, as participants in a graduate course, you will be taking the responsibility to make sure you personally understand the solution to any work arising from such collaboration. You also must indicate on each homework with whom you collaborated. Homework is due at the beginning of class on the due date.

1 Basic probability and statistics [30 points]

1.1 Probability distribution [2 points]

Mickey is a novice in probability theory and is facing a seeming paradox. Consider a uniform probability distribution of variable X on the interval $[0, \frac{1}{2}]$, namely $f(x) = 2$, for $x \in [0, \frac{1}{2}]$. Given that probabilities under a distribution should sum to 1, Mickey is puzzled that $f(x)$ exceeds their sum. Explain the paradox. In other words, what is confusing Mickey?

1.2 Mean and variance [4 points]

Compute the mean and variance of X described in Section 1.1. from first principles (i.e. write down the definitions and compute their values).

1.3 Sampling [6 points]

Mickey knows how to sample from a canonical uniform distribution $g(y) = 1$, for $y \in [0, 1]$. Explain how Mickey can simulate the distribution described in Section 1.1 based on samples of Y, i.e. simulate a uniform distribution on $[0, \frac{1}{2}]$.

Show analytically that with your proposal, the new distribution would be identical in mean and variance as obtained in Section 1.2—in other words, prove that the simulated distribution converges to the desired distribution (*hint: use transformation of variables*).

1.4 More on expectations [6 points]

The odds that a particular face of a fair six-sided die is thrown can be approximated by a discrete uniform distribution. Equipped with his knowledge in probability, Mickey goes to gamble and hedges his bet on throwing a 6. What is the expected number of times he must throw the die to get a 6? Show your workings in clear steps (*hint: use recursion*).

1.5 Detective Bayes [12 points]

Mickey tosses a die multiple times, hoping for a 6. The sequence of his 10 tosses is $1, 3, 4, 2, 3, 3, 2, 5, 1, 6$. Mickey is suspicious whether the die is biased towards 3 (or fair). Conduct a simple analysis based on the Bayes theorem to inform on Mickey—to what degree is the die biased? Explain your reasoning. Assume in general every 100 dice contain 5 unfair dice that are biased towards 3 with the probability distribution of the six faces $(1, 2, 3, 4, 5, 6)$ as $P = [0.1, 0.1, 0.5, 0.1, 0.1, 0.1]$.

2 Linear regression [15 points]

In regression, the input variable X is mapped to the response variable Y via a function $Y = f(X) + E$ subject to some noise E. Linear regression assumes a linear form of the function and finds weight W that minimizes the sum of squared errors:

$$\min \sum_{i=1}^{n} (y_i - wx_i)^2 \tag{1}$$

2.1 Probabilistic interpretation [5 points]

Suppose there are n indenpendent pairs of input and response $\{(x_1, y_1), ..., (x_n, y_n)\}$. Assume these are linearly related subject to standard Gaussian noise $y_i = wx_i + \epsilon_i$, $\epsilon_i \sim N(0, 1)$. Using maximum likelihood, show that maximizing the log-likelihood function $L(Y|X) = \log(P(Y|X))$ is equivalent to minimizing the sum of squared errors in Equation 1 (*hint: first derive the likelihood expression*).

2.2 Geometric interpretation [3 points]

Now think of the responses $\{y_1, ..., y_n\}$ as a vector $\mathbf{y} = [y_1, ..., y_n]$ in n-dimensional space, similarly for the estimates in linear regression $\hat{\mathbf{y}} = [wx_1, ..., wx_n]$. What does Equation 1 mean geometrically to vectors $\mathbf{y}$ and $\hat{\mathbf{y}}$? Provide a simple proof.

2.3 Regularization [7 points]

Real-world problems often involve input variables with multiple dimensions and limited data samples. In these scenarios, a common technique is often used to regularize the weights in linear regression and hence provides smoothing and improves generalization performance. Using matrix notations, the sum of squared errors in Equation 1 can be re-written as

$$\arg\min ||\mathbf{y} - \mathbf{X}\mathbf{w}|| + \lambda||\mathbf{w}||^2 \tag{2}$$

where $\mathbf{X} \in \mathbb{R}^{n\times m}$, $\mathbf{y} \in \mathbb{R}^{m\times 1}$ and λ regularizes the norm of the weights. Derive an analytical expression that minimizes the term in Equation 2 with respect to $\mathbf{w}$ using vector derivatives. Explain briefly what happens to the weights $\mathbf{w}$ when λ is increased.

3 Density estimation [15 points]

In K-nearest neighbors (KNN), the classification is achieved by majority vote in the vicinity of data. Suppose there are two classes of data each of $\frac{n}{2}$ points overlapped to some extent in a 2-dimensional space.

1. Describe what happens to the training error (using all available data) when the neighbor size k varies from n to 1. [**2 points**]

2. Predict and explain with a sketch how the generalization error (e.g. holding out some data for testing) would change when k varies? Explain your reasoning. [**3 points**]

3. Propose a method to determine an appropriate value for k. [**3 points**]

4. In KNN, once k is determined, all neighbors within the perimeter of k are weighted equally in deciding the class label. Suggest a modification to the algorithm that improves this caveat. [**3 points**]

5. Give two reasons why KNN may be undesirable when the input dimension is high. [**4 points**]

4 Naive Bayes and classification [40 points]

In this section, you will implement the full and naive Bayes classifiers and compare their performances on several data sets.

4.1 The rationale of "naive" [4 points]

Suppose you are modeling some N-dimensional data X with a full Gaussian distribution $X \sim N(\mu, \Sigma)$, what is the number of parameters to be estimated? Now instead using the naive Bayes assumption, what is the number of parameters to be estimated? Show your calculations clearly.

4.2 Programming task [36 points]

First, simulate 10 data sets each containing two classes of data from the 2-dimensional Gaussian distributions specified in Table 1. Plot separately sets 5 and 10 on a 2-D plane with the two classes coded in different symbols and colors. Save the simulated 10 sets. **[2 points]**

Parameter 1: $\mu_0 = [-2\ \ 0]$; $\Sigma_0 = [2\ \ 0; 0\ \ 3]$; $\mu_1 = [2\ \ 0]$; $\Sigma_1 = [2\ \ 0; 0\ \ 3]$.

Parameter 2: $\mu_0 = [-2\ \ 0]$; $\Sigma_0 = [2\ \ -1.8; -1.8\ \ 3]$; $\mu_1 = [2\ \ 0]$; $\Sigma_1 = [2\ \ -1.8; -1.8\ \ 3]$.

Set	# data per class	Gaussian parameters
1	20	Parameter 1
2	40	Parameter 1
3	100	Parameter 1
4	300	Parameter 1
5	1200	Parameter 1
6	20	Parameter 2
7	40	Parameter 2
8	100	Parameter 2
9	300	Parameter 2
10	1200	Parameter 2

Table 1: Specification for the data sets.

Now write two separate routines that implement the full Gaussian and naive Bayes classifiers respectively. Apply each of these classifiers to the simulated 10 data sets using 10-fold cross validated classification—use equal number of data points from class 0 and 1 as training or testing (held-out) data in each fold by writing a generic routine that performs classification with k-fold cross validation. Save the learned parameters of each classifier in each fold (for full Gaussian Bayes, these are the means and covariances; for naive Bayes, these are the means and variances at each dimension).

1. Plot the cross-validated classification accuracies with error bars under Parameter 1 (sets $1-5$) and Parameter 2 (sets $6-10$) on separate figures. For each figure, overlay the accuracies of the two classifiers for comparison. **[5 points]**

2. Carefully examine and report the parameters learned for the two classifiers under Parameter 1 and 2. How do these vary when the data size increases from 20 to 1200? For full Gaussian and naive Bayes, how close are the estimated means and covariances to the ground truth—you may use the parameters averaged across 10-fold validation as the estimated parameters. **[8 points]**

3. Compare and comment on the performance of these classifiers under different parameterizations. What is your prediction about their performances?

Do the results reflect your prediction? Explain your reasoning. [**5 points**]

4. Submit the complete (MATLAB) code in the appendix of your write-up in compact format (e.g. double columns). These include 1) the routines for the two classifiers 2) the routine for k-fold cross validation. Do not attach or print out the simulated data sets. [**16 points**]

5 Bonus question: a small real-world challenge [10 points]

Note: this question is OPTIONAL. However, extra credits will be given if you attempt at it. In addition, if your classifier achieves the highest accuracy in the class, another 10 *points will be granted.*

The US postal service uses machine learning methods to robustly identify zip codes. In this section, you will design and implement a classifier that does a simpler job—classifying hand-written digits of “0”s and “1”s. You can load these digits in MATLAB with the command “load -ascii digitX” where X= 0 or 1. Each of the “.mat” files contains 700 samples of digits in 8 × 8 pixels. To get a feel for the data, you can visualize these samples with the command “imagesc(reshape(digitX(i,:),8,8)); colorbar gray”, where i is the row index.

1. Your task is to perform a leave-one-out cross validated classification on the digit data—we recommend the Naive Bayes classifier although you are free to develop your own algorithm (we do not recommend you spending too much time on developing an over-complex model). In any case, specify clearly the assumptions and formulations of your algorithm. For example, for the Naive Bayes classifier, you should specify the prior, the likelihood function and any manipulation or transformation on the data. [**3 points**]

2. Implement and apply your classifier on the data. Report the cross-validated accuracy (this should be a single number). Did your classifier work well—explain why or why not. [**2 points**]

3. Submit your MATLAB code in compact format and attach it in the write-up. [**5 points**]