

10-701/15-781 Machine Learning, Fall 2007: Homework 1

Due: Wednesday, October 3rd, beginning of the class

Instructions There are 4 questions on this assignment. The last question involves coding. Do *not* attach your code to the writeup. Instead, copy your implementation to

`/afs/andrew.cmu.edu/course/10/701/Submit/your_andrew_id/HW1`

To write in this directory, you need a kerberos instance for andrew, or you can log into, for example, `unix.andrew.cmu.edu`. Please submit each problem *separately* with your name and userid on each problem. Refer to the webpage for policies regarding collaboration, due dates, and extensions.

1 Decision Trees [Steve, 20 points]

1.1 ID3

Consider the following set of training examples for the unknown target function $\langle X_1, X_2 \rangle \rightarrow Y$. Each row indicates the values observed, and how many times that set of values was observed. For example, $(+, T, T)$ was observed 3 times, while $(-, T, T)$ was never observed.

Y	X_1	X_2	Count
+	T	T	3
+	T	F	4
+	F	T	4
+	F	F	1
-	T	T	0
-	T	F	1
-	F	T	3
-	F	F	5

Table 1:

- [3 pts] What is the sample entropy $H(Y)$ for this training data (with logarithms base 2)?
- [3 pts] What are the information gains $IG(X_1) \equiv H(Y) - H(Y|X_1)$ and $IG(X_2) \equiv H(Y) - H(Y|X_2)$ for this sample of training data?
- [2 pts] Draw the decision tree that would be learned by ID3 (without postpruning) from this sample of training data.

1.2 KL-divergence, Information Gain, and Entropy

When we discussed learning decision trees in class, we chose the next attribute to split on by choosing the one with maximum information gain, which was defined in terms of entropy. To further our understanding of information gain, we will explore its connection to *KL-divergence*, an important concept in information theory and machine learning. For more on these concepts, refer to Section 1.6 in Bishop.

The KL-divergence from a distribution $p(x)$ to a distribution $q(x)$ can be thought of as a distance measure from P to Q :

$$KL(p||q) = - \sum p(x) \log_2 \frac{q(x)}{p(x)}$$

From an information theory perspective, the KL-divergence specifies the number of additional bits required on average to transmit values of x if the values are distributed with respect to $p(x)$ but we encode them assuming the distribution $q(x)$. If $p(x) = q(x)$, then $KL(p||q) = 0$. Otherwise, $KL(p||q) > 0$. The smaller the KL-divergence, the more similar the two distributions.

We can define information gain as the KL-divergence from the observed joint distribution of X and Y to the product of their observed marginals.

$$IG(x, y) \equiv KL(p(x, y)||p(x)p(y)) = - \sum_x \sum_y p(x, y) \log_2 \left(\frac{p(x)p(y)}{p(x, y)} \right)$$

When the information gain is high, it indicates that adding a split to the decision tree will give a more accurate model.

1. [4 pts] Show that this definition of information gain is equivalent to the one given in class. That is, show that $IG(x, y) = H[x] - H[x|y] = H[y] - H[y|x]$, starting from the definition in terms of KL-divergence.
2. [1 pts] In light of this observation, how can we interpret information gain in terms of dependencies between random variables?

1.3 Controlling Complexity to Avoid Overfitting

In class, we mentioned that decision trees tend to overfit, and that in order to generalize well, we need to limit the complexity of the trees we learn.

One method of pruning decision trees (typically from the bottom up, after training) is to test the data involved in each split in the tree, to see how likely it is that we would see the observed proportions of labels in each branch if the attribute we split on was actually uncorrelated with the target labels. A commonly used test is Pearson's chi-square test, which is an example of a statistical hypothesis test. In this test, we hypothesize that the attribute is uncorrelated with the labels, and test if the observed evidence strongly supports rejecting this hypothesis. Specifically, we calculate a certain test statistic which has a chi-squared distribution if they are uncorrelated, and reject the hypothesis that they are uncorrelated if the statistic has a value unlikely to have been generated by a chi-squared distribution.

To perform the test, assume we have learned a tree, and denote by S the set of training examples whose classification paths pass through the node we want to test the significance of; say that node splits on a discrete attribute x that can take values $1, 2, \dots, k$. Let p be the number of examples in S with label $+$, and $n = |S| - p$ the number of examples in S with label $-$. Let S_i be the subset of S with $x = i$, p_i be the number of examples in S_i with label $+$, and $n_i = |S_i| - p_i$ the number of examples in S_i with label $-$. Furthermore, let $\bar{p}_i = p \cdot \frac{|S_i|}{|S|}$ and $\bar{n}_i = |S_i| - \bar{p}_i$; these are the expected values of p_i and n_i under the hypothesis that x is uncorrelated with the label.

Using the data, we calculate the test statistic χ^2 :

$$\chi^2 = \sum_{i=1}^k \frac{[p_i - \bar{p}_i]^2}{\bar{p}_i} + \frac{[n_i - \bar{n}_i]^2}{\bar{n}_i}$$

Under the hypothesis that x is uncorrelated with the label value, this obeys a χ -square distribution. A parameter to the χ -square distribution is the *degrees-of-freedom*, which, in this case is $k - 1$.¹ For example, if the features each have only two values, the degrees-of-freedom parameter would be 1.

We compute the value of χ^2 for the data, and compare it to a *critical value*: the smallest value such that the probability the statistic exceeds that value is at most α if they are uncorrelated. α is the *confidence*

¹If there were m different class labels instead of two, we would use a similar statistic and there would be $(m - 1)(k - 1)$ degrees of freedom.

level, and it is common practice to let $\alpha = 0.05$, (so we are “95% confident” in the significance of a split that passes the test). For example, with 1 degree of freedom, and $\alpha = 0.05$, the critical value is 3.841; with 2 degrees of freedom it is 5.991. We say the split is statistically significant if χ^2 exceeds the critical value.

1. [2 pts] According to the chi-square test with $\alpha = 0.05$, which of the three splits in the decision tree from part 1 of this problem are statistically significant?

We know that a tree with lower complexity will tend to have better generalization properties. So one (rather simplistic) option to help avoid overfitting is to find the simplest tree that fits the data. This is a principle known as *Occam’s Razor*. One simple way to define “simplest” is based on the *depth* of the tree. Specifically, the depth is the number of nodes along the longest root-to-leaf path. For example, the tree from part 1 would have depth 2. In this problem, we will be interested in learning the tree of least depth that fits the data.

Suppose the training examples are n -dimensional boolean vectors, where $n > 2$ is some constant integer. (For example (T, F, F, T, T) is a 5 dimensional boolean vector). We know that the ID3 decision tree learning algorithm is guaranteed to find a decision tree consistent² with any set of (not self-contradicting) training examples, but that doesn’t necessarily mean it will find a short tree.

2. [3 pts] For $n = 3$, does ID3 always find a consistent decision tree of depth ≤ 2 if one exists? If so, prove it. If not, provide a counterexample (a set of examples, similar to Table 1 above, but with 3 variables), with an explanation.
3. [2 pts] Propose your own learning algorithm that finds a shortest decision tree consistent with any set of training examples (your algorithm can have running time exponential in the depth of the shortest tree). Give the pseudocode and a brief explanation.

2 Regression [Jingrui, 25 points]

2.1 Linear Models

Suppose that you have a software package for linear regression. The linear regression package takes as input a vector of responses (Y) and a matrix of features (X), where the entry $X_{i,j}$ corresponds to the i -th data point and the j -th feature for that data point and Y_i is the i -th response of the function. The linear regression package returns a vector of weights w that minimizes the sum of squared residual errors. The j -th entry of the vector, w_j is the weight applied to the j -th feature.

For the following functions G_i of the input vector C_i , you should

EITHER

- specify how the response and features (Y_i and $X_{i,j}$) are calculated for the regression software package
- specify how parameters α can be obtained from the values returned by the regression software package w so that α is the maximum likelihood estimate

OR

- provide your reasoning for why the software can not be employed

Example. Given the function $G_i = \sum_{j=0}^3 \alpha_j C_{i,1}^j + \epsilon_i = \alpha_0 + \alpha_1 C_{i,1} + \alpha_2 C_{i,1}^2 + \alpha_3 C_{i,1}^3 + \epsilon_i$ where $C_{i,1}$ is the first component of C_i and $\epsilon_i \sim N(0, \sigma^2)$, by setting: $X_{i,j} \leftarrow C_{i,1}^j$ for $j = 0, 1, 2, 3$ and $Y_i \leftarrow G_i$ for each i , the software package then returns $w^* = \operatorname{argmin} \sum_i (y_i - w_0 - w_1 x_{i,1} - w_2 x_{i,2} - w_3 x_{i,3})^2 = \operatorname{argmin} \sum_i (G_i - \sum_{j=0}^3 w_j C_{i,1}^j)^2$. $\alpha_j \leftarrow w_j$ then is the MLE for each α_j for $j = \{1, 2, 3\}$.

1. [2 pts] $G_i = \alpha_1 C_{i,1}^2 e^{C_{i,2}} + \epsilon_i$ where $C_{i,2}$ is the second component of C_i and $\epsilon_i \sim N(0, \sigma^2)$.

²A “consistent” tree is one with zero training error.

- [2 pts] $G_i = \alpha_1 C_{i,1}^2 e^{C_{i,2}} + \epsilon_i + \gamma_i$ where $\epsilon_i \sim N(0, \sigma_1^2)$, $\gamma_i \sim N(\mu, \sigma_2^2)$, and ϵ_i and γ_i are independent. Here μ is the unknown bias and must be estimated.
- [2 pts] $G_i = \sum_j \alpha_j f_j(C_i) + \epsilon_i$ where $f_j(C_i)$ are known basis functions calculated using the input vector C_i and $\epsilon_i \sim N(0, \sigma^2)$
- [2 pts] $G_i = \sum_j \alpha_{(j\%5)} f_j(C_i) + \epsilon_i$ where “%” is the modulo operator and $\epsilon_i \sim N(0, \sigma^2)$
- [2 pts] $G_i = \sum_j \alpha_j f_j(C_i|\theta) + \epsilon_i$ where θ is a real valued unknown parameter in the basis functions and $\epsilon_i \sim N(0, \sigma^2)$. You need to estimate both α and θ
- [2 pts] $e^{G_i} = \gamma_i [\prod f_j(C_i)^{\alpha_j}]$ where $\gamma_i \sim \text{logNormal}(0, \sigma^2)$ ³ and the range of f_j is positive.

2.2 Weighted Least Squares

In class, we have learned that given instances $\langle x_j, t_j \rangle$ generated from the linear regression model $t_j = \sum_i w_i h_i(x_j) + \epsilon_j$, the least squares estimate for the coefficient vector w is given by $w^* = (H^T H)^{-1} H^T t$. If $\epsilon_1, \dots, \epsilon_n$ are independent Gaussian with mean 0 and constant standard deviation, the least squares estimate is also the MLE. In the first three questions, assume that $\epsilon_1, \dots, \epsilon_n$ are independent Gaussian with mean 0, but the variances are different, i.e. $\text{Variance}(\epsilon_i) = \sigma_i^2$.

- [3 pts] Give the formulation for calculating the MLE of w .
- [3 pts] Calculate the MLE of w .
- [3 pts] Explain why the MLE of w can also be obtained by weighted least squares, i.e. w^* is obtained by minimizing the weighted residual squared error $\sum_j a_j (t_j - \sum_i w_i h_i(x_j))^2$, where a_j is the weights. Give the weights a_j .
- [3 pts] If $\epsilon_1, \dots, \epsilon_n$ are independent Laplace with mean 0 and the same scale parameter b , i.e. the pdf of ϵ_i is $f_{\epsilon_i}(x) = \frac{1}{2b} \exp(-\frac{|x|}{b})$, give the formulation for calculating the MLE for w (closed form solution is not required).
- [1 pts] Sometimes the model in the last question is preferred because its solution tends to be more robust to noise. Explain why this is true.

3 Parameter Estimation [Sue Ann, 20 points]

The Poisson distribution is a useful discrete distribution which can be used to model the number of occurrences of something per unit time. For example, in networking, packet arrival density is often modeled with the Poisson distribution. That is, if we sit at a computer, count the number of packets arriving in each time interval, say every minute, for 30 minutes, and plot the histogram of how many time intervals had X number of packets, we expect to see something like a Poisson pmf curve.

If X (e.g. packet arrival density) is Poisson distributed, then it has pmf

$$P(X|\lambda) = \frac{\lambda^X e^{-\lambda}}{X!}$$

(For the purposes of this problem, everything you need to know about Poisson and Gamma distributions will be provided.)

³The log-Normal distribution is the distribution of a random variable whose logarithm is normally distributed.

3.1 MLE and MAP estimates

It can be shown that the parameter λ is the mean of the Poisson distribution. In this part, we will estimate this parameter from the number of packets observed per unit time X_1, \dots, X_n which we assume are drawn i.i.d from $Poisson(\lambda)$.

- [3 pts] Recall that the *bias* of an estimator of a parameter θ is defined to be the difference between the expected value of the estimator and θ .

Show that $\hat{\lambda} = \frac{1}{n} \sum_i X_i$ is the maximum likelihood estimate of λ and that it is unbiased (that is, show that $\mathbb{E}[\hat{\lambda}] - \lambda = 0$). Recall that $\mathbb{E}[a + b] = \mathbb{E}[a] + \mathbb{E}[b]$ (linearity of expectations).

- [5 pts] Now let's be Bayesian and put a prior distribution over the parameter λ .

Your friend in networking hands you a typical plot showing the counts of computers at a university cluster with different average packet arrival densities (Figure 1). Your extensive experience in statistics tells you that the plot resembles a Gamma distribution pdf. So you believe a good prior distribution for λ may be a Gamma distribution.

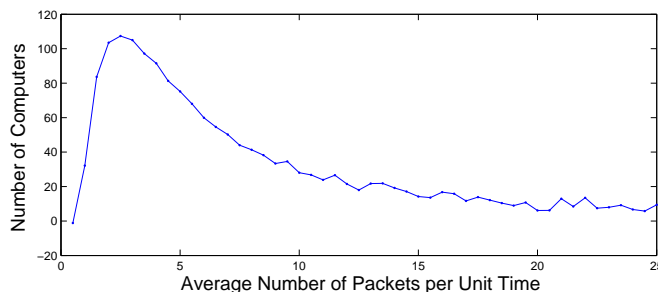


Figure 1: Just giving you some motivation. Don't take it so seriously.

Recall that the Gamma distribution has pdf:

$$p(\lambda|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\beta\lambda}, \quad \lambda > 0$$

Also, if $\lambda \sim \Gamma(\alpha, \beta)$, then it has mean α/β and the mode is $(\alpha - 1)/\beta$ for $\alpha > 1$.⁴

Assuming that λ is distributed according to $\Gamma(\lambda|\alpha, \beta)$, compute the posterior distribution over λ .

Hint:

$$\lambda^{\sum X_i + \alpha - 1} e^{-\lambda(n + \beta)}$$

looks like a Gamma distribution! Is the rest of the expression constant with respect to λ ? Working out a messy integral can lead to the answer but shouldn't be necessary.

- [2 pts] Derive an analytic expression for the maximum a posteriori (MAP) estimate of λ under a $\Gamma(\alpha, \beta)$ prior.

3.2 Estimator Bias/Variance

In class, we learned that the maximum likelihood estimator is not always unbiased. For example, we saw that the maximum likelihood estimator for the variance of a Normal distribution,

$$\hat{\sigma}_{MLE}^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \hat{\mu})^2$$

⁴ $\Gamma(\alpha)$ refers to the Gamma function, but don't worry if you don't know what this is - it will not be important for this question.

is biased - and that an unbiased estimator of variance is:

$$\hat{\sigma}_{unbiased}^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \hat{\mu})^2$$

For the Normal distribution, these estimators give similar results for large enough N , and it is unclear whether one estimator is preferable to the other. In this problem, we will explore an example in which the maximum likelihood estimate is dramatically superior to any unbiased estimator.

We will again be interested in the Poisson distribution, but instead of estimating the parameter λ , we will estimate a *nonlinear* function of λ , namely $\eta = e^{-2\lambda}$ from a single sample $X \sim \text{Poisson}(\lambda)$.

1. [3 pts] Let $\hat{\eta} = e^{-2X}$. Show that $\hat{\eta}$ is the maximum likelihood estimate of η .
2. [4 pts] Show that the bias of $\hat{\eta}$ is $e^{\lambda(1/e^2-1)} - e^{-2\lambda}$.

The following identity from Taylor expansion may be useful:

$$e^t = \sum_{n=0}^{\infty} \frac{t^n}{n!}$$

3. [3 pts] It turns out that $(-1)^X$ is the *only* unbiased estimate of η . Prove that it is indeed unbiased and briefly explain why this is a bad estimator to use. It may be instructive to plot the values of the MLE and unbiased estimate for $X = 1, \dots, 10$. (You do not need to hand in the plot.)

4 Discriminative vs. Generative Classifiers [Joseph, 35 points]

A common debate in machine learning has been over generative versus discriminative models for classification. In this question we will explore this issue, both theoretically and practically. We will consider naive Bayes and logistic regression classification algorithms.

To answer this question, you might want to read: *On Discriminative vs. Generative Classifiers: A comparison of logistic regression and Naive Bayes*, Andrew Y. Ng and Michael Jordan. In NIPS 14, 2002. <http://www.robotics.stanford.edu/~ang/papers/nips01-discriminativegenerative.pdf>

4.1 [5 points] Logistic regression and Naive Bayes

1. Briefly describe the functions Naive Bayes and Logistic-regression optimize.
2. Recall from the suggested reading that “the discriminative analog of naive Bayes is logistic regression.” This means that the parametric form of $P(Y|X)$ used by Logistic regression is implied by the assumptions of a Naive Bayes classifier, for some specific class-conditional densities. In class you will see how to prove this for a Gaussian naive bayes classifier for continuous input values. Can you prove the same for binary inputs? Assume X_i and Y are both binary. Assume that $X_i|Y = j$ is Bernoulli(θ_{ij}), where $j \in \{0, 1\}$, and Y is Bernoulli(π).

4.2 [10 points] Double counting the evidence

1. Consider the two class problem where class label $y \in \{T, F\}$ and each training example X has 2 binary attributes $X_1, X_2 \in \{T, F\}$. How many parameters will you *need* to know/evaluate if you are to classify an example using the Naive Bayes classifier?
2. Let the class prior be $P(Y = T) = 0.5$ and also let $P(X_1 = T|Y = T) = 0.8$ and $P(X_1 = F|Y = F) = 0.7$. , $P(X_2 = T|Y = T) = 0.5$ and $P(X_2 = F|Y = F) = 0.9$. So, attribute X_1 provides slightly stronger evidence about the class label than X_2 . For this problem, you should assume that the *true* distribution of X_1, X_2 , and Y satisfies the Naive Bayes assumption of conditional independence with the above parameters.

- (a) Assume X_1 and X_2 are truly independent given Y . Write down the Naive Bayes decision rule given $X_1 = x_1$ and $X_2 = x_2$.
- (b) Show that if Naive Bayes uses both attributes, X_1 and X_2 , the error rate is 0.235. Is it better than using only a single attribute (X_1 or X_2)? Why? The error rate is defined as the probability that each class generates an observation where the decision rule is incorrect.
- (c) Now, suppose that we create a new attribute X_3 , which is an exact copy of X_2 . So, for every training example, attributes X_2 and X_3 have the same value, $X_2 = X_3$. Are X_2 and X_3 conditionally independent given Y ? What is the error rate of Naive Bayes now? [Hint: The true distribution has not changed.]
- (d) Explain what is happening with Naive Bayes? Does Logistic Regression suffer from the same problem? Explain why?
- (e) (Extra credit) In spite of the above fact we will see that in some examples Naive Bayes doesn't do too badly. Consider the above example i.e. your features are X_1, X_2 which are truly independent given Y and a third feature $X_3 = X_2$. Suppose you are now given an example with $X_1 = T$ and $X_2 = F$. You are also given the probabilities $P(Y = T|X_1 = T) = p$ and $P(Y = T|X_2 = F) = q$, and $P(Y = T) = .5$. Prove that the decision rule is $p \geq \frac{(1-q)^2}{q^2+(1-q)^2}$ (Hint : use Bayes rule again). What is the true decision rule? Plot the two decision boundaries (vary q between 0 – 1) and show where Naive Bayes makes mistakes.

4.3 [20 points] Learning Curves of Naive Bayes and Logistic Regression

Compare the two approaches on the Breast Cancer dataset you can download from course webpage. You can find the description of this dataset in <ftp://ftp.ics.uci.edu/pub/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.names>. We have removed the records with missing values for you and included a short Matlab function to automatically load the data into Matlab. In this problem you will obtain the learning curves similar to those from Figure 1 in the paper.

Implement a Naive Bayes classifier and a logistic regression classifier with the assumption that each attribute value for a particular record is independently generated.

For the NB classifier, assume that $P(x_i|y)$, where x_i is a feature in the breast cancer data (that is, i is the number of column in the data file) and y is the label, of the following multinomial distribution form:

for $x_i \in \{v_1, v_2, \dots, v_n\}$,

$$p(x_i = v_k | y = j) = \theta_{jk}^i \text{ s.t. } \forall i, j : \sum_{k=1}^n \theta_{jk}^i = 1$$

where $0 \leq \theta_{jk} \leq 1$ and $I(z) = 1$ iff the condition z is true (else $I(z) = 0$). It may be easier to think of this as a normalized histogram or as a multi-value extension of the Bernoulli.

Use 2/3 of the examples for training and the remaining 1/3 for testing. Be sure to randomly split the data into training and test sets, do not just use the first 2/3 data points.

For each algorithm:

1. Briefly describe how you implement it by giving the pseudocode. The pseudocode must include equations for estimating the classification parameters and for classifying a new example. Remember, this should not be a printout of your code, but a high-level outline.

You should submit the code itself electronically to

afs/andrew.cmu.edu/course/10/701/Submit/your_andrew_id/HW1/

2. Plot a learning curve: the accuracy vs. the size of the training data. Generate six points on the curve, using [.01 .02 .03 .125 .625 1] fractions of your training set and testing on the full test set each time. Average your results over 5 random splits of the data into a training and test set (always keep 2/3 of the data for training and 1/3 for testing, but randomize over which points go to training set and

which to testing). This averaging will make your results less dependent on the order of records in the file. Plot both the Naive Bayes and Logistic Regression, learning curves on the same plot. Use the `plot(x,y)` function in Matlab since the training data fractions are not equally spaced.

Specify your choice of prior/regularization parameters and keep those parameters constant for these tests. A typical choice of constants would be to add 1 to each bin before normalizing (for NB) and $\lambda = 0$ (for LR).

3. What conclusions can you draw from your experiments? Specifically, what can you say about the speed of convergence of the classifiers? Also, you may find that the categorical Naive Bayes performs worse than Logistic Regression for small training sets and better than Logistic Regression for large training sets (opposite from what the paper suggest). Why might this happen? Think about the decision surfaces and the number of parameters for both classifiers.