CISC859 Pattern Recognition, Winter 2019

Assignment 3, due February 11

Estimating classifier performance (Course reader pp 25-34)

1) As discussed on page 30 of the course reader: DHS Equation (39) on page 484 gives the probability of getting k errors when using n′ test samples, if the true error rate is p.

a) Informally explain this equation: how do the three factors arise?

Here is a review of "n choose k" notation. $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ is the number of different ways of choosing a subset of size *k* from a set of size *n*. For example, create a committee of *k*=7 people from *n*=100 candidates by lining up the 100 candidates and choosing the first 7 people. There are 100! ways to form the line. Divide by 7! because all permutations of the first seven people result in the same committee, and divide by 93! because all permutations of the last 93 people result in the same committee.

b) The number of classes, c, does not appear in equation (38). Is this a mistake, or is it true that it doesn't matter how many classes there are?

c) Refer to DHS Fig 9.10 on page 485. Someone tests a classifier by running 50 tests and finding that there are 10 errors. The estimated error probability is 20%. Use DHS Fig 9.10 to find the 95% confidence interval: the true P(error) has a 95% chance of lying within this interval.

Parametric estimation of a probability density (course reader pp 37-38)

2) Consider a recognizer for upper case characters (26 classes, ω_1 to ω_{26}). We choose 5 features, x_1 to x_5. We decide to use a Bayes classifier, and assume that all the $p(\mathbf{x} \mid \omega_i)$ densities have a multivariate normal form. We use parameter estimation to determine the parameters of $p(\mathbf{x} \mid \omega_i)$ from the training samples.

a) Describe the contents of the parameter vector θ. How many parameters have to be estimated, in total?

b) Describe the operation of the classifier at "run time". We are given a feature vector $\mathbf{x}'$ for an unknown sample. How does the classifier process this feature vector? How does it make use of the estimated θ in this process?

3) [This problem is adapted from DHS page 140, problem 2 "Let *x* have a uniform density ..."]

Let x be distributed uniformly between 0 and θ. In other words, $p(x \mid \theta) = 1/\theta$ when $0 \le x \le \theta$, and $p(x \mid \theta)$ is zero when x is outside the range $0 \le x \le \theta$.

(a) Sketch $p(x \mid \theta)$ versus x, for a fixed value of the parameter θ. (This is the "usual sketch" for a uniform density.)

(b) Sketch $p(x_1 \mid \theta)$ versus θ (where $\theta > 0$) for a fixed value of x_1. This sketch depicts the situation where some sample x_1 has been observed, and we are trying out various guesses for the value of θ. (If we guess that θ is smaller than x_1 then we have made a mistake. It is impossible for θ to be less than x_1 because that would mean zero probability of obtaining x_1 as a sample drawn from this density, but we are told that we *did* obtain x_1 as a sample.) This sketch allows you to compare what happens if the guessed θ is equal to x_1 versus guessing that θ is larger or smaller than x_1.

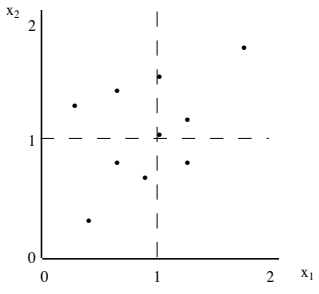
(c) Suppose that *n* samples x_1, ..., x_n are drawn independently according to $p(x \mid \theta)$. Argue that the maximum likelihood estimate for θ is the largest of the *n* samples. An intuitive discussion suffices: informally argue why it is plausible that the best estimate for θ is the largest observed sample. You can refer to the sketch from part (b) in your argument.

Here the single parameter θ is sufficient for characterizing a density uniform from 0..θ. More generally, a uniform density in one-dimensional space requires estimating two parameters: estimate mean and variance, or alternatively estimate *a*, *b* where the density is uniform in the range [*a*, *b*].

A note about notation: Here DHS uses $x_1 .. x_n$ to denote *n* samples that are used for estimating θ. Do not confuse this with similar-looking notation elsewhere in DHS where $(x_1, ..., x_d)$ are the elements of a feature vector.

Nonparametric estimation of a probability density (course reader pp 39-41)

4) Here we write $p(\mathbf{x})$ instead of $p(\mathbf{x} \mid \omega_i)$, since the density estimation is done one class at a time.
We have the following data for making a nonparametric estimate of $p(\mathbf{x})$.



a) What is the estimate at $\mathbf{x}=(1, 1)$ if we use a "volume" (in this case, an area) with side-length 1?

b) What is the estimate at $\mathbf{x}=(1, 1)$ if we use a volume with side-length 0.5?

It's fine to estimate k by eye: "It appears that k of the 10 points lie in a volume centered at (1, 1)".

Note that it is possible for $p(\mathbf{x})$ to be greater than 1. For example, if a one-dimensional density f(x) is uniform in the range $1.1 \le x \le 1.2$, this means that $f(x)=10.0$ in that range.

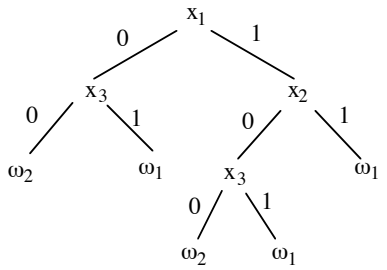
Nearest neighbor classifier (course reader pp 41-43)

5a) Under what conditions would you expect the kNN (k-Nearest Neighbor) classifier to give better results than the NN (Nearest Neighbor) classifier?

b) How are the prior probabilities $P(\omega_i)$ reflected in the NN and kNN classification methods? For example, consider character recognition: in English, the letter *e* is much more frequent than the letter *q*. A Bayes' classifier is affected by this, because it uses the value of $P(\omega_i)$ in the process of making a classification decision. In contrast, an NN classifier for OCR does not make explicit reference to $P(\omega_i)$; it just looks for the nearest neighbor. So the question is: how do the prior probabilities have an effect on an NN classifier?

Decision trees (course reader p 46)

6) Here is a decision tree for a two-class problem with three binary features (x_1, x_2, x_3). Class ω_1 is defined by the Boolean formula "$(x_1$ AND $x_2)$ OR x_3". This means that for samples in class ω_1, either both features x_1 and x_2 are present, or feature x_3 is present. Everything else is in class ω_2.



a) Find the expected length from the root to a leaf for this tree. Assume that the values 0 and 1 are equally likely for each of the three features x_1, x_2, and x_3. This means that each of the 8 feature vectors (000, 001, 010, etc) is equally likely, and as a result, **some leaves are reached more frequently than other leaves**. The expected path length tells us "how many features, on average, have to be measured when using this decision tree to come up with a classification".

b) Create a decision tree that performs the same classification with shorter expected path length. State the expected path length of your tree.

Classifier combination (course reader pp 46-48)

7) Classifiers A, B, and C each perform digit recognition (10 classes). Assume that the classifiers are independent: if classifier A makes an error on a certain input, that does not affect the probability of classifier B making an error on this input. Also, there is no correlation among the wrong answers: if A and B both make an error, the chance that they both give the same wrong answer is 1/9.

Classifier D is defined as a combination of the outputs produced by classifiers A, B, C. It operates as follows:

- If two or three of A, B, C produce the same answer ω_i, then classifier D answers ω_i
- If all three of A, B, C produce different answers, then classifier D rejects the input.

(a) Classifiers A, B, C are correct 50% of the time. What is P(correct), P(reject), P(error) for classifier D? [See discussion on the next page to help you get started.]

(b) Classifiers A, B, and C are correct 70% of the time. What is P(correct), P(reject), P(error) for classifier D?

(c) *Optional -- you don't have to do this if parts (a) and (b) took you a long time*. Find P(correct) for majority-voting combination of 5 classifiers, where each classifier is correct 70% of the time.

Discussion for problem 7

There are many ways to calculate the answers. Some analysis is given on page 48 of the course reader and here is further discussion about parts (a) and (b).

You need to find the probability of the various situations that cause classifier D to be right, wrong, or reject. Since P(correct) + P(reject) + P(error) = 1, you only need to figure out two of these three values. (Or you can be thorough and figure out all three values; then check the correctness of your work by making sure that they sum to 1.) Here is a summary of the situations.

Classifier D rejects if

- A, B, C produce three different answers. All three might be wrong, or one of the answers might be right.

Classifier D is correct if

- All three of ABC give the right answer.
- Two of ABC are right. This can happen 3 ways: AB right and C wrong; BC right and A wrong; AC right and B wrong.

Classifier D is wrong if

- All three of ABC give the wrong answer AND at least two of these wrong answers agree.
- Two of ABC give the same wrong answer, and the third classifier gives the right answer.

In summary, the performance of classifier combination can be improved in two ways:

1. Improve the component classifiers, as illustrated by going from 50% to 70% correct in parts (a) and (b).
2. Combine more classifiers, as illustrated by going from 3 to 5 classifiers in parts (b) and (c).