CS 340 Fall 2007: Homework 2

1 The Monty Hall problem

On a game show, a contestant is told the rules as follows:

> There are three doors, labelled 1, 2, 3. A single prize has been hidden behind one of them. You get to select one door. Initially your chosen door will *not* be opened. Instead, the gameshow host will open one of the other two doors, and *he will do so in such a way as not to reveal the prize.* For example, if you first choose door 1, he will then open one of doors 2 and 3, and it is guaranteed that he will choose which one to open so that the prize will not be revealed.
>
> At this point, you will be given a fresh choice of door: you can either stick with your first choice, or you can switch to the other closed door. All the doors will then be opened and you will receive whatever is behind your final choice of door.

Imagine that the contestant chooses door 1 first; then the gameshow host opens door 3, revealing nothing behind the door, as promised. Should the contestant (a) stick with door 1, or (b) switch to door 2, or (c) does it make no difference? You may assume that initially, the prize is equally likely to be behind any of the 3 doors. Hint: use Bayes rule.

2 Reject option in classifiers

In many classification problems one has the option either of assigning $\mathbf{x}$ to class j or, if you are too uncertain, of choosing the **reject option**. If the cost for rejects is less than the cost of falsely classifying the object, it may be the optimal action. Let α_i mean you choose action i, for $i = 1 : C+1$, where C is the number of classes and $C+1$ is the reject action. Let $Y = j$ be the true (but unknown) **state of nature**. Define the loss function as follows

$$\lambda(\alpha_i|Y=j) = \begin{cases} 0 & \text{if } i=j \text{ and } i,j \in \{1,\ldots,C\} \\ \lambda_r & \text{if } i = C+1 \\ \lambda_s & \text{otherwise} \end{cases} \tag{1}$$

In otherwords, you incur 0 loss if you correctly classify, you incur λ_r loss (cost) if you choose the reject option, and you incur λ_s loss (cost) if you make a substitution error (misclassification).

1. Show that the minimum risk is obtained if we decide $Y = j$ if $p(Y = j|\mathbf{x}) \geq p(Y = k|\mathbf{x})$ for all k (i.e., j is the most probable class) *and* if $p(Y = j|\mathbf{x}) \geq 1 - \frac{\lambda_r}{\lambda_s}$; otherwise we decide to reject.
2. Describe qualitatively what happens as λ_r/λ_s is increased from 0 to 1 (i.e., the relative cost of rejection increases).

3 Fun with entropy

Consider the joint distribution $p(X, Y)$

		x			
		1	2	3	4
y	1	1/8	1/16	1/32	1/32
	2	1/16	1/8	1/32	1/32
	3	1/16	1/16	1/16	1/16
	4	1/4	0	0	0

1. What is the joint entropy $H(X, Y)$?
2. What are the marginal entropies $H(X)$ and $H(Y)$?
3. The entropy of X conditioned on a specific value of y is defined as

$$H(X|Y = y) = -\sum_x p(x|y) \log p(x|y) \tag{2}$$

   Compute $H(X|y)$ for each value of y. Does the posterior entropy on X ever increase given an observation of Y?
4. The conditional entropy is defined as

$$H(X|Y) = \sum_y p(y)H(X|Y = y) \tag{3}$$

   Compute this. Does the posterior entropy on X increase or decrease when averaged over the possible values of Y?
5. What is the mutual information between X and Y?

4 Bayesian concept learning

In this question, you will implement the Bayesian concept learning framework for the "number game" we discussed in class. You are provided the following functions

- `hypSpace = mkHypSpace()` which creates the hypothesis space (a structure). To extract the set of integers defined by the h'th hypothesis (this is called the support or extension of the hypothesis), and its name, use the following:

  ```
  hypSpace.hyps{h}
  hypSpace.names{h}
  ```

  There are `hypSpace.Nmath=23` mathematical hypotheses, and `hypSpace.Nint =5050` interval hypotheses, stored in order in order of increasing size. For example,

  ```
  hypSpace.hyps{2} = [1 3 5 ... 99]; hypSpace.names{2} = 'evens';
  hypSpace.hyps{24} = 1; hypSpace.names{24}= 'interval 1..1';
  hypSpace.hyps{25} = 2; hypSpace.names{25}= 'interval 2..2';
  hypSpace.hyps{124} = [1,2]; hypSpace.names{124}= 'interval 1..2';
  ```

  etc.

- `prior = mkPrior(hypSpace)`, which creates a (row) vector, in which $prior(h) = p(h)$, for h=1:5073. This assigns probability $\lambda/|H_{math}| = 0.029$ to the mathematical hypotheses, and $1 - \lambda/|H_{int}| = 6.6 \times 10^{-5}$ to the interval hypotheses, where $\lambda = 2/3$. (This distribution is a mixture of two uniform distributions (over different ranges), where λ is the mixing weight.)

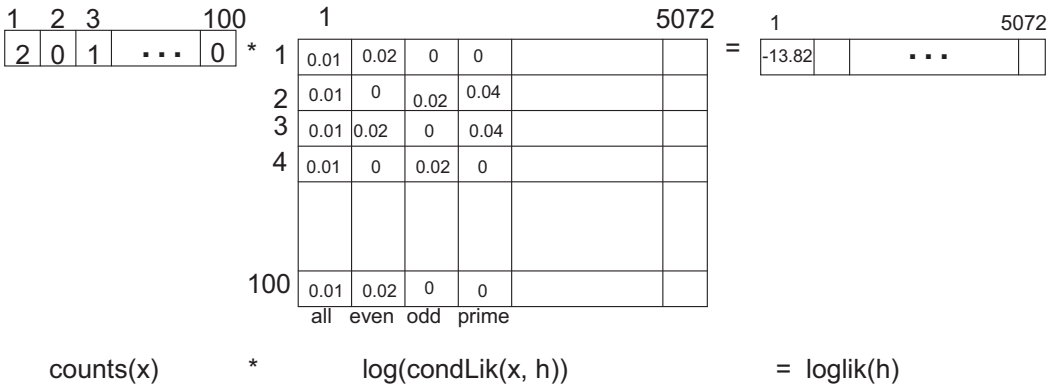

Figure 1: Computation of the likelihood function. In this example, $D = [1, 1, 3]$, so the count vector is $[2, 0, 1, 0, ...]$. The condLik matrix has $1/100$ in every row of column 1, $1/50$ in even rows of column 2, $1/50$ in odd rows of column 3, $1/25$ in column 4 in locations $[2, 3, 5, 7, 11, ..., 89, 97]$, representing the prime numbers, etc.

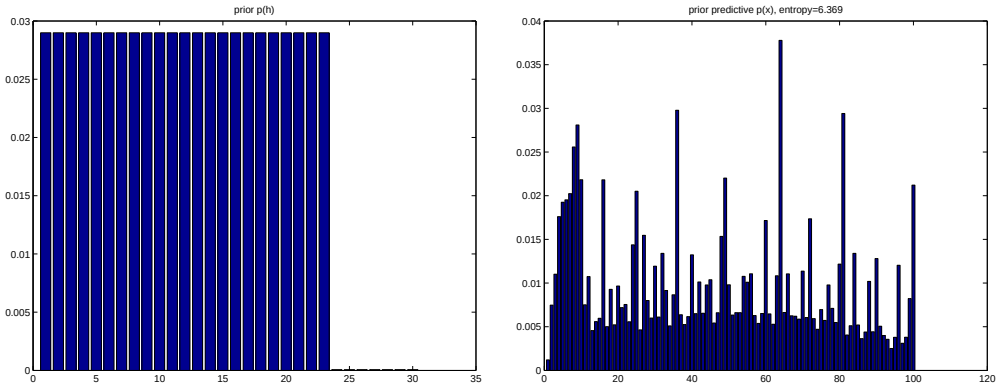

Figure 2: Prior and prior predictive distributions.

- `cl = mkCondLik(hypSpace)` generates a 100×5072 matrix, where $cl(x, h) = p(x|h)$ is the likelihood assigned to integer x by hypothesis h, given by

$$p(x|h) = \begin{cases} \frac{1}{|size(h)|} & \text{if } x \in h \\ 0 & \text{if } x \notin h \end{cases}$$

- `loglik = mkLogLik(hypSpace, D)` which computes

$$loglik(h) = \log p(D|h) = \sum_{n=1}^{N} \log p(x_n|h)$$

  where N is the number of examples in D. This can be computed using a vector-matrix multiply, as shown in Figure 1.

Use these to answer the following questions

1. Plot the prior over the first 30 hypotheses. The result should look like Figure 2(left). Turn in your code and plot.
2. Write a function to compute the prior predictive distribution

$$pred(x) = p(y(x) = 1) = \sum_{h \in \mathcal{H}} p(y(x) = 1|h)p(h)$$

where $pred(x)$ is a vector, with one element for each $x = 1 : 100$, Plot pred(x) as a histogram. What is its entropy? (Use log base 2.) The result should look like Figure 2(right). Turn in your code and plot. Hint: you can use the `mkCondLik` function.

3. Write a function `post = mkPost(hypSpace, D)` which computes

$$post(h) = \frac{p(D|h)p(h)}{\sum_{h'} p(D|h')p(h')}$$

where $post$ is a vector. Turn in your code.

4. Suppose $D = [32]$. Compute the posterior $post(h) = p(h|D)$. Plot the posterior over the mathematical hypotheses, $post(1 : 23)$, as a histogram. Turn in your plot.

5. Sort the hypotheses into decreasing order of posterior probability. What are the top 5 most probable hypotheses? Return their names and their probabilities.

6. Compute the posterior predictive distribution

$$pred(x) = p(y(x) = 1|D) = \sum_{h \in \mathcal{H}} p(y(x) = 1|h)p(h|D)$$

(Obviously $y(x) = 1$ for all $x \in D$; the goal is to generalize beyond the training set, i.e., to predict which other numbers are in the concept class.) Plot pred(x) as a histogram. What is its entropy? Turn in your code and plot.

7. Write a function to compute the maximum likelihood estimate

$$\hat{h}_{ML} = \arg\max_h p(D|h)$$

Turn in your code. What is $\hat{h}_{ML}$?

8. Write a function to compute the plug-in estimate

$$predML(x) = p(y(x) = 1|\hat{h}_{ML}(D))$$

Plot predML(x) as a histogram. Turn in your code and plot.

9. Now repeat steps 4–8 using $D = [32, 24]$. Turn in your new plots and numbers. What are the main qualitative differences?

10. Now repeat steps 4–8 using $D = [32, 16, 2]$. Turn in your new plots and numbers. What are the main qualitative differences?