Computer Science 791DD, Learning to See

http://www.cs.umass.edu/$\sim$elm/learning2see/

Assignment 4

1. Suppose a continuous random variable has a distribution whose support is limited to the interval $[a,b]$. Answer the following questions (5 points each)

   (a) Give an upper bound, in terms of a and b, on the differential entropy of this random variable.
   As I have said many times in class, the upper bound on the entropy of a distribution over a finite interval $[a,b]$ *is achieved by the uniform distribution over that interval, and is* $\log(b-a)$.

   (b) What is the tightest lower bound?
   Imagine a uniform distribution over $[a, a+\frac{b-a}{k}]$ *where* $k=2$. *This distribution covers half of the interval between a and b and will have entropy of* $\log(b-a)-1$. *Each time we double k, the entropy drops by 1. Since there is no limit to the number of times we can double k, there is no lower limit to the differential entropy. In other words, the lower bound is* $-\infty$.

   (c) Now suppose you are told that the maximum height of the probability density is d, while the support is still limited to $[a,b]$. Can you put a tighter lower bound on the differential entropy? What is it?
   Remember that the differential entropy is the negative of the average log probability density. Thus to minimize the entropy we should maximize the average log probability density. If the density height is bounded by d, then the log probability density at any point can not be higher than $\log(d)$. *Hence, the average log probability density cannot be higher than* $\log(d)$, *and the entropy can not be lower than* $-\log(d)$ *or equivalently* $\log(\frac{1}{d})$. *Since this bound might be achievable, by having a uniform distribution over the interval* $[a, a+\frac{1}{d}]$, *it is a tight bound. That is, the tightest lower bound is* $-\log(d)$.

   (d) If you remove the restriction on the support, but the density is still limited in height to d, what is the tightest upper bound on the differential entropy? The tightest lower bound?
   As in part (b), we can continue doubling the width of the interval over which we define a uniform distribution, and continue adding to the differential entropy. Hence, there is no limit to the entropy, and the upper bound is ∞.

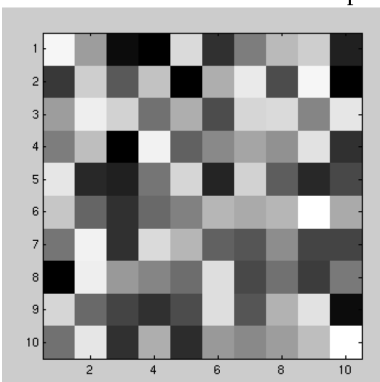
2. Matlab functions. (10 points each)

   (a) Write a Matlab function discreteEntropy.m that computes the entropy of a discrete probability distribution, i.e. a probability mass function. Assume that the input is a vector of probabilities of events. Make sure the function can handle zero probabilities. Return the entropy in bits. Show the output of a function for a particular example.
   See Shaolei's homework posted on the web-page for a good example.

   (b) Use the function you just wrote to write another function, mutualInformation.m, which computes the mutual information between two random variables. Assume that the joint probability distribution, in the form of a two-dimensional matrix, is given as the input. Hint: the body of this function can easily be written in a single line in Matlab.
   See Shaolei's homework posted on the web-page for a good example.

   (c) Finally, write a function to compute the KL-divergence between two discrete distributions. It should take the two distributions as inputs (as vectors). If the true KL-divergence is infinity, then this is what your function should return.
   See Shaolei's homework posted on the web-page for a good example.

3. The following figure shows the joint distribution of two random variables X and Y, each of which takes on 10 different values. The probability of each joint event, $P(X = x, Y = y)$ is represented by its brightness, where black is 0 and white is the highest possible value. Are X and Y independent? Give an airtight argument for your answer. Remember that this is a probability distribution, not a *sample* from a probability distribution. (5 points)



There are many possible arguments. Here is one. If X and Y were independent, then $P(X = a_1|Y = b)$ should be equal to $P(X = a_1|Y = c)$. Thus, $d_1 = P(X = a_2|Y = b) - P(X = a_1|Y = b)$ should be equal to $d_2 = P(X = a_2|Y = c) - P(X = a_1|Y = c)$. Letting $a_1 = 1, a_2 = 2, b = 4$, and $c = 5$, we see that d_1 would be negative (the lower square is brighter than the upper square) but d_2 would be positive (the lower square is darker than the upper square). Hence, assuming we can judge relative brightnesses between adjacent squares correctly, this joint distribution is inconsistent with X and Y being independent.

4. George Bush has an algorithm that only runs on "grayscale" images, meaning images in which each pixel is an integer value in the interval $[0, 255]$ inclusive, representing the brightness of that pixel. That is, each pixel needs to be an 8-bit value. The algorithm cannot use color information. George has an image to which he wants to apply his algorithm, but it is a color image, with 24-bits per pixel. The first 8 bits of a pixel represent red, the next 8 green, and the next 8 blue. He uses a matlab command (rgb2hsv) to convert the red-green-blue image to hue-saturation-value, in which the last channel, "value", can be interpreted as the brightness of the image. He uses this last channel as his gray-scale image.
Unfortunately, after converting the image to grayscale, many of the brightness values in the image are exactly the same. In particular, the distribution of brightness values is as follows:

$$P(0) \quad \frac{1}{2} \tag{1}$$
$$P(63) \quad \frac{1}{16} \tag{2}$$
$$P(127) \quad \frac{1}{16} \tag{3}$$
$$P(253) \quad \frac{1}{8} \tag{4}$$
$$P(254) \quad \frac{1}{8} \tag{5}$$
$$P(255) \quad \frac{1}{8} \tag{6}$$

(a) The number of "bits" of information in this image can be approximated by the number of pixels times the entropy of the distribution of pixel values. Calculate this number. (2 points)
2.125 bits.

(b) George decides he wants his image to contain more "information", so he takes the image pixels whose values are 0 and changes them randomly to have values from 0-62. He changes the values in the other bins similarly. Will the entropy of the distribution of brightness values go up, go down, or stay the same? Why?

(2 points).
The entropy will go up. In the original configuration, the image pixels that have a value of 0 have average log probability of -1. When these pixels are spread out, their average log probability can only go down. The same is true for the pixels in the other bins. Thus all of the average log probabilities for each bin must go down, and thus the entropy must go up.

(c) George claims he has increased the amount of "information" in the image. What do you think of this argument? (5 points).
Though the entropy has gone up, this increase in entropy was due to meaningless randomness and not to additional structure in the image. Thus, it is bogus to say that the amount of information in the image has increased.

(d) George's friend Ralph notices that Matlab's rgb2hsv code uses the command $v = \max(r, g, b)$ to calculate the brightness of an rgb pixel. He argues that the function is throwing away information about the pixel brightness since the pixels $(237, 0, 0)$, $(237, 25, 42)$, and $(237, 0, 1)$ would all get mapped to the same brightness value, even though the second and third pixels are clearly brighter than the first pixel. He proposes using the formula $v = \max(r, g, b) + \min(r, g, b)/256$ to produce a pixel's brightness value. Of course, truncating such a value, he points out, would produce the same result as the original brightness function. Instead, he does histogram equalization on the values *before* truncating them. His procedure results in an image whose values are distributed uniformly across brightness. He argues that his procedure does indeed preserve more information in the image. Is he deluding himself? Be as clear as you can be in your answer. (10 points)
One way to look at the goal of obtaining a brightness image is to distinguish among as many different brightness values in the original scene as possible. Since Ralph's procedure preserves more different brightness values from the original color images (but simply relabels them to fit in the range 0-255), he has indeed preserved more information.