36-350: Data Mining

Homework 2
Date: September 4, 2001 **Due: start of class September 9, 2001**

---

1. Let the (R,G,B) cube be quantized into 8 prototypical colors:

Color	Red	Green	Blue	Color	Red	Green	Blue
Black	0.25	0.25	0.25	Red	0.75	0.25	0.25
Green	0.25	0.75	0.25	Yellow	0.75	0.75	0.25
Blue	0.25	0.25	0.75	Magenta	0.75	0.25	0.75
Cyan	0.25	0.75	0.75	White	0.75	0.75	0.75

   Suppose an image has six pixels, with values (0.22,0.37,0.8), (0.19,0.8,0.19), (0.6,0.1,0.05), (0.8,0.3,0.22), (0.7,0.32,0.8), and (1,0.4,0.34). What is the color count representation of the image? What is the representation after dividing by Euclidean length?

2. Suppose we search for the above image via the color (0.7,0.2,0.2). What is the color-count representation for this query, and what is the Euclidean distance from the image, after both have been divided by Euclidean length?

3. Describe a potential problem if we use too many prototypical colors in the representation (besides increased computation and storage). Describe a potential problem if we use too few prototypical colors in the representation.

4. Suppose we have a collection of 50 flower and 50 ocean images. The flower images have red and green. The ocean images have blue and green. What are the inverse-picture-frequency (IPF) weights for the colors (red,green,blue)?

5. Describe a simple image search that could not be carried out effectively using a color-count representation.

6. An alternative way to represent images is to arbitrarily divide the image into four parts, and separately compute the color counts for each part. Putting these vectors together gives a representation which is four times as big as using one set of color counts. These vectors are then normalized and compared by Euclidean distance, as before. Does this approach have more or less invariance than one set of color counts? If more, describe the additional invariance. If less, describe the additional sensitivity. How might this affect similarity searching?

7. In a typical data mining application, a news agency wants to search through video archives and detect all frames depicting an event, e.g. fireworks at night, given some examples. You can regard a video as a sequence of images. How could this be implemented using the tools in lab? (Don't give code, just a brief description of the method.) Note that this is not a simple classification problem, because the complementary set (images not of fireworks) is not well-defined.