Homework 1. Due 11PM Sunday Feb 1 (a little less than 2 weeks from now).
To be submitted electronically as a pdf to a dropbox in Angel.

Consider a "categorical" distribution, which is a generalization of the bernoulli distribution to K classes. The likelihood function for a single sample can be written as

$$P(x_i|u_1,u_2,...u_K) \;=\; \prod_{k=1}^{K} u_k^{z_{ik}}$$

where u_k is the probability of a sample being drawn from the kth class, $u_1+u_2+\cdots+u_K=1$, and z_{ik} is a "1 of K" representation of the class label of sample x_i, e.g. if x_i belongs to class 3, then $z_{i3}=1$ and z_{im} for all other $m\neq 3$ would be 0.

Problem 1. Consider N samples drawn *i.i.d.* from a categorical distribution with K classes. Go through the derivation to show me that the maximum likelihood estimate for each parameter u_k is

$$\hat{u}_k^{\text{MLE}} \;=\; \frac{N_k}{\sum_{m=1}^{K} N_m} \qquad .$$

Now, consider a generalization of the Beta distribution called the "Dirichlet" distribution. The Dirichlet distribution is a conjugate prior for performing Bayesian estimation of the parameters $u1\ldots u_K$ of a categorical distribution in the same way that the Beta distribution is a conjugate prior for estimating parameters of the bernoulli distribution. We can write the Dirichlet prior over K classes as

$$P(u_1,\ldots,u_k) \;=\; \frac{\Gamma(\sum_{k=1}^{K} a_k)}{\prod_{k=1}^{K}\Gamma(a_k)} \;\prod_{k=1}^{K} u_k^{a_k-1}$$

for k arbitrary parameters $a_k>0$ and with the understanding that each $u_k\in[0,1]$ and that $\sum_k u_k=1$. It might be helpful at this point to convince yourself on some scratch paper that the Dirichlet formula simplifies to the formula for a beta distribution when there are only two classes.

Problem 2. Consider N samples drawn *i.i.d.* from a categorical distribution with K classes, and assume you are also given a set of k values $a_k>0$ defining a Dirichlet prior on the parameters of that categorical distribution. Show me what the posterior distribution $P(u_1,\ldots,u_k|x_1,\ldots,x_n)$ is, and work through the derivation to show that the maximum a posterior estimate for each parameter u_k is

$$\hat{u}_k^{\text{MAP}} \;=\; \frac{N_k+a_k-1}{\sum_{m=1}^{K}(N_m+a_m-1)} \qquad .$$

Problem 3. This is a programming assignment where we explore a simple image-based example where MAP estimation is superior to MLE. Consider the problem of image retrieval where, based on features extracted from some query image, we rank a library of images according to how similar their distribution of features are to the query image, with the goal of eventually selecting the most similar ones. We are going to represent each image by a "bag of features" model, defined as follows. Lets say there are K distinct types of features that can be measured at each pixel in an image. Assume we have N_1 pixels where a feature of type 1 exists, N_2 pixels where a feature of type 2 exists, and so on, then we can represent any image by a categorical distribution with K parameters $u_i = N_i/N$, where $N = N_1 + \ldots + N_K$. Note that these values are essentially MLE estimates, derived by forming a histogram of feature counts N_i over the whole image and then dividing by the sum of all counts to get a probability mass function that sums to one.

Now consider a scoring function that computes how similar the categorical distribution $[u_1, u_2, \ldots, u_k]$ acquired for image I is to the categorical distribution $[q_1, q_2, \ldots, q_k]$ describing our query image Q. One way to define this score is to consider the probability that the features in Q were generated from the categorical distribution representing image I. Assuming features in Q are drawn independently from the categorical distribution of I, we can compute this score as

$$\text{score}(Q,I) \;=\; c\,P(Q|u_1,\ldots,u_k) \;=\; u_1^{q_1}\,u_2^{q_2}\,\cdots\,u_K^{q_K}$$

where c is a postive constant that we will ignore. Also, note that the q_i exponents in this formula are not integer counts like we typically see when discussing a joint likelihood function formed from a categorical distribution. However, each q_i is proportional to an integer count N_i, since $q_i = N_i/N$, for some value N. As long as we are going to use the computed scores to rank different images I with respect to the same query image Q, it will be OK to use the q_i values rather than N_i (in fact it will be numerically beneficial to do so because raising a number $u_i < 1$ to a large integer power could easily result in numerical underflow).

Having defined a score for comparing two images, we can now compare the query image Q with each of M images $I_1, I_2, \ldots, I_M$ in the library. Sorting the resulting M scores from highest to lowest then allows us to rank order the images in the library, in the hope that images with the highest scores will be the most similar to Q.

3a) OK, here comes the first part of the programming assignment. Consider greyscale images, and define four different feature types (K=4) computed by thresholding a pixel's grey value g. Specifically, for each pixel, the type of feature found at that pixel is

$$\text{feature type} = \begin{cases} 1 & 0 \le g < 64 \\ 2 & 64 \le g < 128 \\ 3 & 128 \le g < 192 \\ 4 & 192 \le g < 256 \end{cases}$$

I have uploaded a small, contrived dataset containing 6 images to use as the library $I_1, \ldots, I_6$ and a single test image to use as a query Q, as illustrated in Figure 1. Three library images are very bright

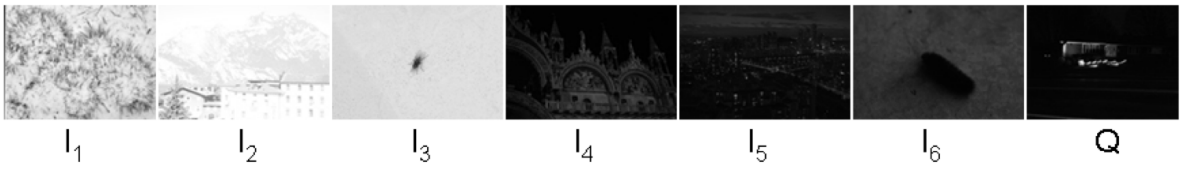

Figure 1: A library of 6 images $I_1, \ldots, I_6$ are to be ranked with respect to a single query image Q.

and three are very dark. The query image is also very dark, so we would expect a rank ordering to rank the dark images more highly than the bright ones. Go ahead and do the experiment, by computing MLE estimates of categorical distribution parameters for all images, computing the comparison scores between Q and each image I, and then sorting scores from highest to lowest to see which images are deemed to be most similar (meaning having the highest scores). The results should surprise you, as the method fails miserably. Go back and look at the image histograms, the categorical parameters computed from them, and the scoring function, and try to explain why the scoring/ranking fails so badly.

3b) Now it is time for MAP estimation to come to the rescue. When estimating the categorical distribution parameters for an image, we will use a Dirichlet prior to impose knowledge about the expected distribution of feature values across the population of all images in the library. In the context of retrieval, this process is known as *Dirichlet smoothing*. Indeed, in the field of text retrieval, from which the bag of features representation idea arises (in text retrieval it is called "bag of words"), Dirichlet smoothing is known to be crucial.[1] To apply this method, start by computing the parameters $\{\rho_1, \rho_2, \ldots, \rho_K\}$ of a categorical distribution of features across all the images in the library, using MLE, to get a population distribution. This distribution tells us the overall frequency with which to expect any given feature to appear, apriori. To form a Dirichlet prior to use for MAP estimation, recall we must come up with parameters a_i that can be thought of as virtual observation counts that get pooled with actual observation counts N_i accumulated from the observed data. We therefore will define some virtual number of pixels η and set $a_i = 1 + \eta\rho_i$ to form our Dirichlet prior. The variable η acts as a variable smoothing parameter. If you set η to be of roughly equal magnitude to the number of pixels in any image, then the prior information will be of roughly equal importance as the data, if you set η to be very large, the prior will swamp the data, and if η is very low, the data will dominate (if $\eta = 0$ the process reduces to MLE estimation). I set η to be 2000 in my experiments, but play around with different values to see how it affects the results. Go ahead now and rerun the experiment in part 3a) using MAP estimation to determine the categorical distributions representing each image. Now when you compute similarity scores between query image Q and images in the library, the sorted scores should appear to do a better job of ranking images in terms of similarity with Q. Verify this result, and discuss.

[1] "When estimating a language model based on a limited amount of text, such as a single document, smoothing of the maximum likelihood model is extremely important. ... In the language modeling approach to retrieval, the accuracy of smoothing is directly related to the retrieval performance." Zhai and Lafferty, A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval, 2001.