CompSci 527 Final Exam Sample— Sample Solution

1. Let x be a real-valued random variable and let

$$U_x[\mu, w] = \begin{cases} \frac{1}{w} & \text{for } \mu - \frac{w}{2} \le x \le \mu + \frac{w}{2} \\ 0 & \text{elsewhere} \end{cases}$$

be the uniform distribution of width w centered at μ. One can define a mixture of uniform distributions as

$$\Pr(x|\boldsymbol{\theta}) = \sum_{k=1}^{K} \lambda_k U_x[\mu_k, w_k]$$

where the values of λ_k define a categorical distribution over the values $1, \ldots, K$, and

$$\boldsymbol{\theta} = (\lambda_1, \mu_1, w_1, \ldots, \lambda_K, \mu_K, w_K) \; .$$

(a) If you are given $\boldsymbol{\theta}$, how do you draw from $\Pr(x|\boldsymbol{\theta})$? Describe briefly but precisely an algorithm that draws a single sample x from this distribution. Assume that the only random generator at your disposal is a function `rand` that takes no arguments and returns a single scalar drawn from $U_x[0, 1]$, and explain precisely how you use `rand` to draw from the distribution(s) you need. You may use MATLAB notation if you like, but a clear, plain-language explanation or pseudo-code listing are OK as well.

Answer: *To draw a sample $x \sim Pr(x|\boldsymbol{\theta})$ do the following:*

- *Draw k from the given categorical distribution with parameters $(\lambda_1, \ldots, \lambda_K)$ as follows:*
  - *Draw $u \sim U[0, 1]$ using `rand`*
  - *Return k if $c_{k-1} \le u < c_k$ where*

$$c_k = \begin{cases} 0 & \text{for } k = 0 \\ \sum_{i=1}^{k} \lambda_k & \text{for } k = 1, \ldots, K \end{cases}$$

  is the cumulative sum of the λ_k (with a 0 at the beginning).
- *Draw x from the uniform distribution $U_x[\mu_k, w_k]$ as follows:*
  - *Draw $u \sim U[0, 1]$ using `rand`*
  - *Return*

$$x = \mu_k + w_k \left(u - \frac{1}{2} \right) \; .$$

(b) Explain clearly and succinctly why using the idea of Expectation Maximization (EM) for parameter estimation may be problematic for mixtures of uniform distributions.

Answer: *The sources of trouble are that the uniform distribution is (i) discontinuous and (ii) piecewise constant. Intuitively, moving μ_k left or right—that is, adjusting μ_k in an attempt to increase the likelihood L—does not change the value of L until a data point crosses a discontinuity, and then the likelihood function changes in a discontinuous manner. Discontinuities also occur when adjusting w_k.*

The same problem can be described from a more technical point of view: The M step is based on differentiating the likelihood function L with respect to $\boldsymbol{\theta}$. Discontinuities cause technical difficulties in this differentiation, as the derivative of a discontinuous function is not defined everywhere. More importantly, the piecewise-constant nature of U implies that the zeros of the gradient may not be isolated points, so EM will give unpredictable results.

2. Let $h_I(u)$ and $h_J(v)$ for $u, v \in \{0, \ldots, 255\}$ be the histograms of a gray-level image I and of the gray-level image J obtained by applying histogram equalization to I. Histogram equalization maps pixel value u of I to pixel value

$$v = f(u) = \left[\frac{256}{n} H_I(u) - 1 \right]$$

of J. In this expression, n is the number of pixels in the image, H_I is the cumulative histogram of I,

$$H_I(u) = \sum_{i=0}^{u} h_I(i) \; ,$$

and the brackets $[\cdot]$ denote rounding to the nearest integer.

Construct a histogram h_I for an image I with $n > 256$ pixels such that

$$h_I(u) \neq h_J(v) \quad \text{for all} \quad u \in \{0, \ldots, 255\} \quad \text{and} \quad v \in \{0, \ldots, 255\} \; .$$

That is, no value in the output histogram h_J is a copy of any value in the input histogram h_I.

Answer: *There are of course many possible answers, and here is one.*

Since we are asked to construct a histogram, we get to choose its parameter n. To simplify reasoning, we make n a multiple of 256, say,

$$n = 256 \times 4 = 1024 \; .$$

Then, histogram equalization maps pixel value u in I to pixel value

$$v = f(u) = \left\lceil \frac{1}{4} H_I(u) - 1 \right\rceil$$

in J. We construct h_I with no empty bins and in such a way that for each pair of consecutive (even, odd) values of u,

$$u = 2i \quad \textit{and} \quad u + 1 = 2i + 1 \; ,$$

the following two properties hold:

(a) Both pixel values $2i$ and $2i + 1$ in I map to the same pixel value $v = f(2i) = f(2i + 1)$ in J.

(b) All the histogram values $h_J(f(2i)) = h_J(f(2i + 1))$ are the same number, call it h, for every i.

In this way, the output histogram h_J has only values 0 (because of collisions) and h, and neither of these values exists in the input histogram h_I.

To make this construction specific, let

$$h_I(2i) = 7 \quad \textit{and} \quad h_I(2i + 1) = 1 \quad \textit{for} \quad i = 0, \ldots, 127 \; ,$$

so the histogram of I keeps oscillating between 7 and 1. Then, the cumulative input histogram is

$$H_I(0) = 7 \, , \; H_I(1) = 8 \, , \; H_I(2) = 15 \, , \; H_I(3) = 16 \, , \; \ldots \, , \; H_I(254) = 1023 \, , \; H_I(255) = 1024$$

that is,

$$H_I(2i) = 7 + 8i \quad \textit{and} \quad H_I(2i + 1) = 8(i + 1)$$

and

$$f(2i) = \left\lceil \frac{7 + 8i}{4} - 1 \right\rceil = \left\lceil 2i + \frac{3}{4} \right\rceil = 2i + 1 \quad \textit{and} \quad f(2i + 1) = \left\lceil \frac{8(i + 1)}{4} - 1 \right\rceil = [2i + 1] = 2i + 1$$

so that

$$f(2i) = f(2i + 1) = 2i + 1 \; .$$

This quantization rule yields the histogram

$$h_J(2i) = 0 \quad \textit{and} \quad h_J(2i + 1) = 8 \quad \textit{for} \quad i = 0, \ldots, 127$$

because no pixel value u maps to even pixel values $v = 2i$ and each odd-valued bin $v = 2i + 1$ accumulates the sum

$$h_J(2i + 1) = h = h_I(2i) + h_I(2i + 1) = 7 + 1 = 8 \; .$$

To summarize, h_I has only values 1 and 7, and h_J has only values 0 and 8, and the requirements of the construction are met.

[Note: This problem is not entirely straightforward to solve, and the questions on the exam are likely to be easier. However, this problem is a good preparatory exercise.]

3. The image C on the right below was obtained by convolving the image I on the left with a 2×2 kernel H whose origin (we also called this the "hot spot") is in the bottom right pixel (marked by the dot). The `'same'` option was used in Matlab, so the sizes of I and C are the same.

Answer:

$$I = \begin{array}{|c|c|c|c|c|c|} \hline 1 & 0 & 5 & 0 & 2 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 & 9 \\ \hline 0 & 7 & 0 & 3 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 & 1 \\ \hline 2 & 0 & 2 & 0 & 6 & 0 \\ \hline \end{array} \qquad H = \begin{array}{|c|c|} \hline 2 & 3 \\ \hline 1 & 1_{\bullet} \\ \hline \end{array} \qquad C = \begin{array}{|c|c|c|c|c|c|} \hline 3 & 8 & 5 & 2 & 20 & 27 \\ \hline 15 & 22 & 6 & 9 & 9 & 9 \\ \hline 10 & 7 & 3 & 3 & 2 & 3 \\ \hline 7 & 4 & 6 & 12 & 19 & 1 \\ \hline 2 & 2 & 2 & 6 & 6 & 0 \\ \hline \end{array}$$

Fill in the four values of the kernel. You may want to briefly explain your reasoning if you are not sure about your answer. [Hint: if you are doing a lot of of calculations, think again.]

Answer: The pixel $I(3,4)$ in the input image has value 3, and is surrounded by zeros. Because of this, that pixel is the only nonzero contributor to the values in the 2×2 box

$$C(2:3,3:4) = \begin{bmatrix} 6 & 9 \\ 3 & 3 \end{bmatrix}$$

of the output image. These four values must be equal to

$$C(2:3,3:4) = I(3,4)H = 3H \ ,$$

that is, three times the kernel. This yields

$$H = \frac{1}{3}C(2:3,3:4) = \begin{bmatrix} 2 & 3 \\ 1 & 1 \end{bmatrix} \ ,$$

as shown above.

[Note: The trick here is to realize that to solve this "inverse" problem (given input and output, find the kernel) you do not flip the kernel, as you would do for a convolution. If you did not get this answer right, work through the example above: Suppose that H is given, and ask the question "what pixels in C does pixel $I(3,4)$ contribute to, given the formula for convolution?" This may take a bit of patience and a drawing or two to work out, unless you want to reason algebraically (not my preferred option, but it may work for you).]

4. *Is the following convolution kernel separable? If so, separate it. If not, prove that it is not.*

$$H = \begin{array}{|c|c|} \hline 2 & 3 \\ \hline 1 & 1 \\ \hline \end{array}$$

Answer: No it is not. If it were, the matrix H would have rank 1 and therefore a zero determinant. Instead,

$$\det(H) = 2 \times 1 - 3 \times 1 = -1 \ .$$

[Note: Different proofs are possible.]

5. *What is the gradient of the following function at $x = y = 0$?*

$$f(x,y) = (x-2)^3 \sin y$$

Answer:

$$\begin{bmatrix} 3(x-2)^2 \sin y \\ (x-2)^3 \cos y \end{bmatrix}_{x=y=0} = \begin{bmatrix} 0 \\ -8 \end{bmatrix}$$

6. *A SIFT descriptor for a 16×16 image window W is a vector of 128 numbers, which can be thought of as being grouped in 16 consecutive groups of 8 numbers each. What does each group of 8 numbers represent? Explain in one or two brief, clear, and accurate sentences.*

Answer: The window W is split into a 4×4 grid of 4×4 square sections. Each group of 8 numbers is a histogram of orientations of the image gradient in one of 16 sections, each orientation weighted by the corresponding gradient magnitude. The gradient magnitude for each orientation is split into two adjacent bins of the histogram, proportionally to how close the orientation is to the center value of each bin.

7. *The K-means algorithm takes as input N points and a positive integer K. It returns a partition of the N input points into K sets with certain properties, and the centroids of the points in each of these sets. State two different ways in which you could initialize the K means algorithm. Just describe briefly in English, no formulas are needed.*

Answer: Initialize the means by choosing K points at random out of the points being clustered. Initialize the clusters by partitioning the points into K sets at random.

8. *In a 2003 paper, Sivic and Zisserman describe an image retrieval system based on SIFT descriptors and visual words. In your answers to the questions below, you may use terminology either from computer vision (images, features, ...) or from document retrieval (documents, words, ...).*

(a) Give a formula for t_{id}, the so-called "term-frequency–inverse document frequency" weight used to evaluate the contribution of a word to a bag-of-words descriptor. Use the following variables in your formula:

- *n_{id} is the number of occurrences of word i in document d*
- *n_d is the total number of words in document d*
- *n_i is the number of occurrences of word i in the whole training set*
- *N is the number of documents in the whole training set*

If you do not remember the formula, describe clearly and succinctly the criteria for it.

Answer:

$$t_{id} = \frac{n_{id}}{n_d} \log \frac{N}{n_i} \, .$$

This measure emphasizes words that occur frequently in a document (the first fraction) but not so frequently in the training set (the logarithm of the second fraction). A word that is rare in general but occurs often in a particular document conveys a significant amount of information about the topic of that document.

(b) If applied blindly, the formula for t_{id} above gives unsatisfactory results. Describe some of the pre-processing Sivic and Zisserman recommend before the weights t_{id} are computed, and explain their rationale.

Answer: They recommend creating a "stop-list" of words that includes the 5 percent most frequent visual words in the training set and the 10 percent least frequent ones. Eliminating very frequent words in the training set removes uninformative visual words, such as textureless windows or edges. Very infrequent words are likely to be unhelpful, as they may not occur consistently even across images that are related to each other.

(c) What is the principle of spatial consistency? Explain clearly and succinctly, and in broad lines. You need not explain the details of how the principle is used in the paper.

Answer: A set of visual words in the query image Q are spatially consistent with a matching set of visual words in a retrieved image R if the spatial arrangement of those words in Q is similar to that in R. A range of options is proposed in the paper for implementing this principle.