CS340 Fall 2006: Homework 7

Out Fri 10 Nov, back Mon 17 Nov

1 Variable elimination

Consider the MRF in Figure 1.

1. Suppose we want to compute the partition function using the elimination ordering $\prec = (1, 2, 3, 4, 5, 6)$, i.e.,

$$Z = \sum_{x_6} \sum_{x_5} \sum_{x_4} \sum_{x_3} \sum_{x_2} \sum_{x_1} \psi_{12}(x_1, x_2)\psi_{13}(x_1, x_3)\psi_{24}(x_2, x_4)\psi_{34}(x_3, x_4)\psi_{45}(x_4, x_5)\psi_{56}(x_5, x_6) \tag{1}$$

   If we use the variable elimination algorithm, we will create new intermediate factors. What is the largest intermediate factor?

2. Add an edge to the original MRF between every pair of variables that end up in the same factor. (These are called fill in edges.) Draw the resulting MRF. What is the size of the maximal clique in this graph?

3. Now consider elimination ordering $\prec = (4, 1, 2, 3, 5, 6)$, i.e.,

$$Z = \sum_{x_6} \sum_{x_5} \sum_{x_3} \sum_{x_2} \sum_{x_1} \sum_{x_4} \psi_{12}(x_1, x_2)\psi_{13}(x_1, x_3)\psi_{24}(x_2, x_4)\psi_{34}(x_3, x_4)\psi_{45}(x_4, x_5)\psi_{56}(x_5, x_6) \tag{2}$$

   If we use the variable elimination algorithm, we will create new intermediate factors. What is the largest intermediate factor?

4. Add an edge to the original MRF between every pair of variables that end up in the same factor. (These are called fill in edges.) Draw the resulting MRF. What is the size of the maximal clique in this graph?

5. Suppose each variable is binary. What is the approximate cost of computing Z using $\prec = (1, 2, 3, 4, 5, 6)$? What is the approximate cost of computing Z using $\prec = (4, 1, 2, 3, 5, 6)$?

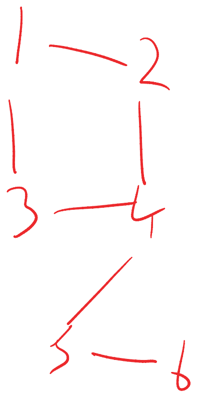

Figure 1: An MRF.

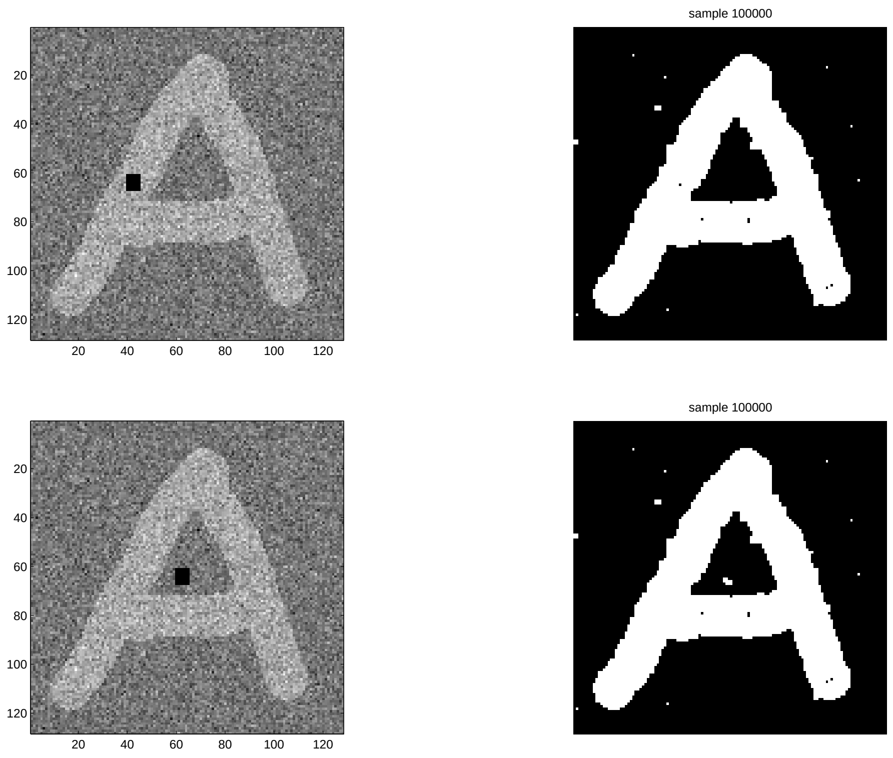

Figure 2: An example of image denoising. If the hole is in a white area, it gets filled in to white. If the hole is in a black area, it (mostly) gets filled in to black.

2 Gibbs sampling for image denoising

The function `gibbsDemoNoBlindspot` implements the image denoising we discussed in class. This works on the image `imageWithoutHole.mat`, but fails on `imageWithBlackHole.mat`, and `imageWithWhiteHole.mat`, which are missing some pixels (i.e., they have NaN values (not a number) in certain locations). Use the `isnan` command to find these pixels. In this question, you must modify the `gibbsDemoNoBlindspot` function so it can handle images with missing information. You need to modify the code that computes the local evidence and the initial state Xinit. You cannot just assume that all missing pixels are 0, because sometimes they may be on, sometimes off. Instead, you need to ignore the image evidence for the missing pixels, and just rely on the prior. Once you have modified the code, apply it the black and white hole images. Your results should look like Figure 2. Turn in your code and denoised images.