MA 589 — Computational Statistics

Project 5

(Due: Tuesday, 11/29/16)

1. A genome can have a number of alterations to its DNA sequence. A common type of alteration is a *copy number variation*: some regions might have been *amplified* or *deleted*. To measure possible changes in copy number, we can collect data from a comparative genomic hybridization (CGH) array. Roughly, for each probe i in the array that corresponds to a position in a genomic sequence we measure Y_i, the log of the ratio between the copy number of an individual at position i and a reference for a normal copy number.

   Suppose you have a simple CGH assay with only $n = 200$ probes. You decide to model the states of each probe as "deleted" (state 1), "normal" (state 2), and "duplicated" (state 3), and set a Markov chain for the transitions between states with probabilities

$$P = \begin{bmatrix} 0.50 & 0.50 & 0 \\ 0.05 & 0.90 & 0.05 \\ 0 & 0.5 & 0.5 \end{bmatrix}$$

   The chain starts at "normal", i.e., state 2. For the emission Y_i at state X_i of the chain, you assume that

$$Y_i \,|\, X_i = s \overset{\text{iid}}{\sim} N(\mu_s, \sigma_s^2), \quad i = 1, \dots, n,$$

   with

State s	1	2	3
μ_s	-1	0	1
σ_s	0.7	0.5	0.7

   Your task is to infer, given the observations Y in the file `cgh.txt`[1], which positions are duplicated and deleted.

   (a) Using the forward algorithm, compute $\log \mathbb{P}(Y)$.

   (b) Using the Viterbi algorithm, obtain the MAP estimate $\widehat{X}$ for X. What is $\log \mathbb{P}(\widehat{X} \,|\, Y)$?

   (c) Plot Y as points and, for each probe i, the mean μ at $\widehat{X}_i$. It's better to connect the means using a solid, thicker line. Comment on the plot; for instance, does $\widehat{X}$ seem to provide a reasonable fit? Are there regions that you would believe to differ from $\widehat{X}$, say, as being amplified instead of normal according to $\widehat{X}$?

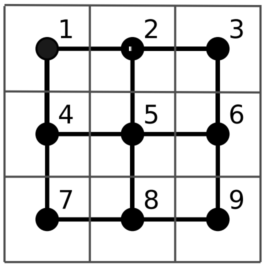
   (d) What is the probability that the *last* probe has a normal copy number given Y? How much more likely is it, again given Y, for the last probe to be in a deleted region instead of a duplicated region?

[1] You can read Y in `R` using `scan`.

2. An important application of satellite image data is to classify land cover. Suppose that you observe data in a small 3×3 image and wish to classify each pixel in the image as either "forest" (state 1) or "water" (state $-\mathbf{1}$.) To this end, based on the data you observed you then compute, for each pixel i, a vegetation index Y_i represented below:

Site i	1	2	3	4	5	6	7	8	9
Y_i	2	2	2	2	0	0	1	2	1

The interest is in assessing for each pixel i the states X_i. The marginal on X is given by a graphical model on the whole grid G,



given by

$$\mathbb{P}(X) = \frac{1}{Z_X(J)} \prod_{(i,j)\in G} \exp(J \cdot X_i X_j) \propto \prod_{(i,j)\in G} \exp(J \cdot X_i X_j),$$

where $(i, j) \in G$ means that i and j are neighbors in G, J is a parameter measuring the strength to which neighboring values X_i and X_j agree, and $Z_X(J)$ is a normalizing constant. We assume a Gaussian likelihood: the data Y are conditionally independent given X,

$$Y_i \mid X_i = s \overset{\text{iid}}{\sim} N(\mu_s, \sigma_s^2), \quad i = 1, \ldots, 9,$$

with $\mu_1 = 2$, $\sigma_1 = 1$, $\mu_{-1} = 0.5$ and $\sigma_{-1} = 0.5$.

We want to estimate the land cover classification given the vegetation indices, but computing on the lattice is challenging so we develop a Markov chain Monte Carlo procedure. We start with a Gibbs sampler that has $\mathbb{P}(X \mid Y)$ as target. This sampler is similar to the one we discussed in class for the Ising model, but it uses the data Y.

(a) Implement a Gibbs sampler that has $\mathbb{P}(X \mid Y)$ as target. Your sampler should be a function `gibbsJ` that takes as parameters J and n, the number of sample cycles over X (that is, $X_1, X_2, \ldots, X_9$) and returns an array indexed by sample t and pixel i containing $X_i^{(t)}$ at the end of the t-th cycle.

Start by showing that at each step of the cycle over pixels, you sample from pixel i according to

$$\mathbb{P}(X_i \mid X_{[-i]}, Y) \propto \exp\left\{ J X_i \sum_{j \in N_i} X_j + \log \mathbb{P}(Y_i \mid X_i) \right\}$$

where N_i is the neighborhood of i in G.

(b) Run your sampler for $J = 0.2$ and $n = 1{,}000$. To assess convergence, compute for each sample $X^{(t)}$ the log conditional density (up to a normalizing constant)

$$f(X^{(t)} \,|\, Y) = \sum_{(i,j)\in G} J X_i^{(t)} X_j^{(t)} + \log \mathbb{P}(Y \,|\, X^{(t)}),$$

and plot the trace and autocorrelation (for different lags) of f for three chains. In addition, compute the Brooks-Gelman-Rubin scale reduction factor for the chains. What can you conclude? How many samples are needed for (assumed) convergence?

(c) Obtain MCMC estimates for $\mathbb{P}(X_i \,|\, Y)$ for each pixel $i \in G$, and for the *entropy* (modulo a constant)

$$\sum_X f(X \,|\, Y)\mathbb{P}(X \,|\, Y).$$

(d) $*$ What is the effect of J in the results? To answer this question, let's conduct a simulation study on J. First, let's define the *concordance* of X as $C(X) = \sum_{i>j} I(X_i = X_j)$, that is, the number of pairs of pixels that agree according to the states in X. Now, for each J in `seq(-1, 1, length=n1)`, with `n1`=20, we run `n2`=5 simulations, compute the mean concordance across $n = 1{,}000$ Gibbs samples of X, and plot the results. The following `R` code summarizes the study:

```
n1 <- 20; n2 <- 5; n <- 1000
Js <- seq(-1, 1, length=n1)
C <- matrix(nrow=n1, ncol=n2) # store mean concordance
for (i in 1:n1) { # each J in Js
  for (j in 1:n2) { # each replication
    X <- gibbsJ(n, Js[i])
    C[i,j] <- mean(apply(X, 1, concordance))
  }
}
boxplot(C, use.cols=F, names=format(Js, digits=2))
```

Discuss the boxplots; in particular, how would you explain the behavior of the mean concordance with respect to J? Are the results expected given the model?[2]

(e) $*$ Now let's try and compute the *evidence*, the marginal log probability $\log \mathbb{P}(Y)$, exactly. First, use a suitable visitation schedule and a forward procedure to compute the normalizing constant $Z_X(J) = \sum_X \prod_{(i,j)\in G} \exp(J \cdot X_i X_j)$; next, using the same order and another forward procedure, compute $\log \mathbb{P}(Y)$. [3]

[2] Here are some ideas that would make a nice final project: set $\mu_s \overset{\text{iid}}{\sim} N(0, \tau^2)$ and come up with an MCMC sampler for $\mathbb{P}(\mu \,|\, Y)$ using Gibbs sampling and/or a MH random walk; conduct a simulation study on the effect of varying τ^2.

[3] If you're feeling adventurous, you can also use dynamic programming to obtain the posterior mode.

Here are more ideas for a final project:

- Compare the posterior distribution under the full lattice with an approximate posterior distribution using a *spanning tree* of the lattice; the comparison can use Kullback-Leibler divergences, entropies, and should include a discussion on how well it works as a function of J.
- Devise an EM algorithm to estimate the likelihood parameters μ_1, μ_{-1}, σ_1, and σ_{-1}, and the hyper-prior parameter J.