Stat521A Spring 2009: homework 5

This week you will learn about Monte Carlo EM and variational EM, in the context of the genetic association model we have been studying. We will ignore the family structure between the genes, and hence you do not need to use PMTK.

Let $O_{ij} \in \{0, 1, 2\}$ be the observation for person i gene j, $G_{ij} \in \{0, 1, 2\}$ be the hidden gene, $\mathbf{x}_i$ be the observed covariate, and y_i be the observed response. The data was generated by sampling from the following model:

$$\begin{aligned} p(y_i|\mathbf{g}_i, \mathbf{x}_i, \boldsymbol{\beta}, \lambda) &= \mathcal{N}(y_i|\mathbf{z}_i^T\boldsymbol{\beta}, \lambda) && (1) \\ \mathbf{z}_i &= (g_{i1} - 1, \ldots, g_{iN_g} - 1, \mathbf{x}_i, 1) && (2) \\ p(O_{ig} = o|G_{ig} = k, \boldsymbol{\nu}) &= p(o|k, \boldsymbol{\nu}) && (3) \end{aligned}$$

where $p(o|k, \nu)$ is in Table 1, and the G's were generated from a family tree (the details of which are irrelevant).

There are 32 datasets, shown in Figure 1. These were generated by varying the following parameters in order: which gene(s) cause the response, the strength of the effect, the noise level $\nu \in \{0.01, 0.1\}$, the number of genes $N_g \in \{5, 10\}$, the number of families $N_f \in \{5, 50\}$. (Since there are 4 people in each family, the number of data cases is $n = 4N_f$.) The goal, as before, is to figure out which gene(s), if any, cause the response, and to estimate the strenght (and sign) of this effect.

The data is stored in files `familyTreeDataXXX.mat` on the class webpage. Each file contains the following variables:

```
nu: 0.1000
 X: [200x1 double]
 Y: [200x1 double]
 O: [200x5 double]
```

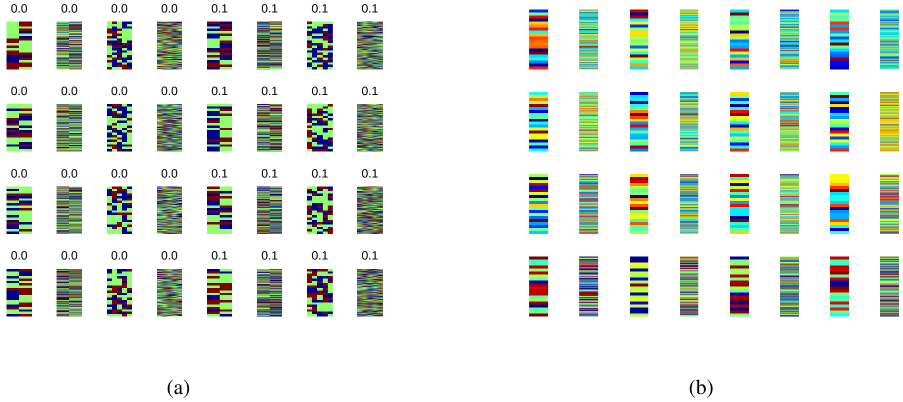
Note that the number of rows is either 20 or 200, and the number of columns of O is either 5 or 10. ν varies in each file.

1. The simplest model is to assume that there are no genotyping errors, so $O_{ig} = G_{ig}$: see Figure 2(left). In this case, there are no hidden variables, so one can compute the exact MLE, MAP or posterior of $p(\boldsymbol{\beta}|\mathcal{D})$ very easily. (You can either integrate out λ or estimate it separately.) Try this first. Combine your estimates of $\boldsymbol{\beta}$ for each dataset into a 5×32 matrix $\mathbf{B}$; only the first $N_g + 2$ rows need to be filled, where N_g differs across datasets. Plot $\mathbf{B}$ (with colorbar). It should be apparent by eye which genes cause the response.

2. The next model takes genotyping errors into account. This is called an "error in covariates" regression model. See Figure 2(right). Derive and implement an EM algorithm to fit this model. (Show your derivation and turn

G	$p(O = AA)$	$p(O = Aa)$	$p(O = aa)$
AA	$1 - \nu_2$	ν_2	0
Aa	ν_1	$1 - 2\nu_1$	ν_1
aa	0	ν_2	$1 - \nu_2$

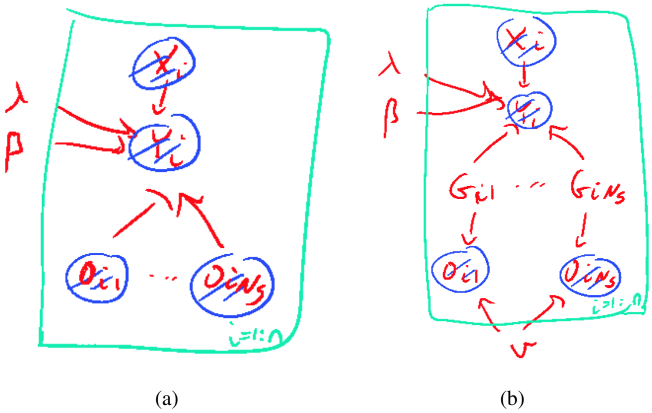
Table 1: CPD for $p(O|G)$.



(a)

(b)

Figure 1: The data. (a) Observed genes O. (b) Response Y. (X is omitted to save space.)



(a)

(b)

Figure 2: The models. (a) Assume $O = G$. (b) Errors in covriates.

in your code.) For simplicity, you may assume ν is known (it is stored in each file), and that $\lambda = 1$. Also, just compute the MLE of $\boldsymbol{\beta}$, rather than a MAP estimate. Plot $\mathbf{B}$.

3. Although the EM algorithm is simple, it is slow (especially when we model family structure). The reason is that each gene competes to explain the response, so they are all correlated in the posterior: $p(G_{i1}, \ldots, G_{iN_g}|\mathcal{D}_i, \boldsymbol{\theta})$ is a table of $K = 3^{N_g}$ numbers, where $\mathcal{D}_i = (y_i, \mathbf{x}_i, \mathbf{o}_i)$. So each E step takes $O(Kn)$ operations (assuming we do a full sweep over all the data). One simple way around this is to use Monte Carlo EM. We approximate the expected sufficient statistics as follows:

$$\sum_i \sum_{\mathbf{g}} p(\mathbf{g}|\mathcal{D}_i, \boldsymbol{\theta}_{old})(y_i - \boldsymbol{\beta}^T \mathbf{z}_i(\mathbf{g}))^2 \approx \sum_i \sum_s w_{si}(y_i - \boldsymbol{\beta}^T \mathbf{z}_i(\mathbf{g}_i^s))^2 \tag{4}$$

where $\mathbf{g}_i^s$ is a sample from $p(\mathbf{G}_i|\mathcal{D}_i, \boldsymbol{\theta}_{old})$ and w_{si} is its weight. One simple way to generate such samples is to use importance sampling, with the following proposal:

$$Q_i(G_1, \ldots, G_{N_g}) = \prod_{j=1}^{N_g} p(o_{ij}|G_j) \tag{5}$$

If the probability of observation error is small, this proposal should be close to the posterior. Derive the corresponding importance sampling weight w_{si} and then implement this algorithm. Compare the accuracy of the importance sampling approximation to the exact E step as a function of the number of samples. Then integrate this into your EM code and plot the resulting matrix $\mathbf{B}$.

4. Sampling in a discrete state-space is often unnecessary. A natural alternative in this case is to suppose that $\mathbf{G}_i$ is either equal to $\mathbf{O}_i$ or is very close to it. Hence we can approximate the posterior $p(\mathbf{G}_i|\mathcal{D}_i, \boldsymbol{\theta})$ by storing a finite number of deterministically chosen vectors, such as $\mathbf{O}_i$ and all its b-nearest neighbors in Hamming distance, along with their corresponding probabilities. Compare the accuracy of this to the exact E step as a function of the size of the Hamming ball b. Then integrate this into your EM code and plot the resulting matrix $\mathbf{B}$.

5. Finally, we come to the trickiest method, variational EM. To simplify notation, we drop the subscript i from G. We will use a mean field approximation of the form

$$p(G_1, \ldots, G_{N_g}|\mathcal{D}_i, \boldsymbol{\theta}) \approx Q(G_1, \ldots, G_{N_g}) \overset{\text{def}}{=} \prod_{j=1}^{N_g} Q_j(G_j) \tag{6}$$

where $Q_j(G_j) \overset{\text{def}}{=} \text{Mun}(G_j|\boldsymbol{\phi}_j)$ and $\boldsymbol{\phi}_{j,1:3}$ are the variational parameters for gene j. The corresponding mean values are

$$\mu_j = E_Q[G_j] = \sum_{k=1}^{3} \boldsymbol{\phi}_{j,k} \times (k-1) \tag{7}$$

Assuming the μ's are known, derive the expected complete data log-likelihood and the corresponding M step. (It should be a simple weighted least squares problem.) For the E step, follow Bishop's recipe and solve

$$Q_j(k) \propto E_{-j} \ln p(G_1, \ldots, G_{N_g}, \mathcal{D}_i|\boldsymbol{\theta}) \tag{8}$$

for $\boldsymbol{\phi}_j$, where we take expectations (wrt Q) over all G variables except G_j. (The update equation should be the softmax of a few simple matrix equations.) Give your derivation, and then implement your method. Compare the accuracy of the variational approximation to the exact E step as a function of the number of mean field updates. Then integrate this into your EM code and plot the resulting matrix $\mathbf{B}$.

Hint: I recommend you read section 10.1 of Bishop's book. If things are still not clear, also read the paper [GJ97] on factorial HMMs, which discusses a very similar approximation (if we ignore the HMM part of the story).[1]

[1] Note that the FHMM paper uses a different weight vector for each discrete state, whereas we multiply the weight vector by the discrete values, essentially treating them as ordinal.

References

[GJ97] Z. Ghahramani and M. Jordan. Factorial hidden Markov models. *Machine Learning*, 29:245–273, 1997.