Homework 2: Part I

Statistical Approaches to Learning and Discovery

Zoubin Ghahramani & Teddy Seidenfeld

Due: Mon Mar 18, 2002

Consider the multiple cause model discussed in class. This is a model with K binary latent variables, s_i, real-valued observed vector $\mathbf{y}$ and parameters $\boldsymbol{\theta} = \{\{\boldsymbol{\mu}_i, \pi_i\}_{i=1}^K, \sigma^2\}$

$$p(s_1, \ldots, s_K|\boldsymbol{\pi}) = \prod_{i=1}^K p(s_i) = \prod_{i=1}^K \pi_i^{s_i}(1-\pi_i)^{(1-s_i)}$$

$$p(\mathbf{y}|s_1, \ldots, s_K, \boldsymbol{\mu}, \sigma^2) = \mathcal{N}(\sum_i s_i \boldsymbol{\mu}_i, \sigma^2 I)$$

Assume you have a data set of N i.i.d. observations of $\mathbf{y}$, i.e. $Y = \{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(N)}\}$.

General Matlab hint: wherever possible, avoid looping over the data points. Many (but not all) of these functions can be written using matrix operations.

Warning: Each question depends on earlier questions. If you get stuck early on, send an email or come talk to Zoubin. As always, you are welcome to discuss the homework with each other, but write up and hand in your own work.

Hand in: Derivations, code and plots.

1. Implement the fully factored (a.k.a. mean-field) **variational approximation** described in class. That is, for each data point $\mathbf{y}^{(n)}$, approximate the posterior distribution over the hidden variables by a distribution:

   $$q_n(\mathbf{s}) = \prod_{i=1}^K \lambda_{in}^{s_i}(1-\lambda_{in})^{(1-s_i)}$$

   and find the λ's that maximize $\mathcal{F}$ holding $\boldsymbol{\theta}$ fixed. Specifically, write a Matlab function:

   `[lambda,F] = MeanField(Y,mu,sigma,pie,lambda0,maxsteps)`

   where `lambda` is $N \times K$, `F` is the lower bound on the likelihood, `Y` is the $N \times D$ data matrix, `mu` is the $D \times K$ matrix of means, `pie` is the $1 \times K$ vector of priors on $\mathbf{s}$, `lambda0` are initial values for `lambda` and `maxsteps` are maximum number of steps of the fixed point equations. You might also want to set a convergence criterion so that if `F` changes by less than ϵ the iterations halt.

2. Derive the conditional probability:

   $$p(s_i|s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_K, \mathbf{y}, \boldsymbol{\theta})$$

3. Using the above conditional probability, implement **Gibbs sampling** to approximate the posterior distribution over the hidden states given the data. Specifically, write a Matlab function:

`[S] = Gibbs(Y, mu, sigma, pie, S0, nsamples)`

where `S` is a $N \times K \times$ `nsamples` array of samples over the hidden variables, `S0` is an $N \times K$ matrix of initial settings for the hidden variables.

4. Derive the M step for this model in terms of the quantities: `Y`, `ES` $= E_q[\mathbf{s}]$, which is an $N \times K$ matrix of expected values, and `ESS` $= E_q[\mathbf{s}\mathbf{s}^\top]$, which is an $N \times K \times K$ array of expected values. Hint: write down the expected log of the joint probability of $\mathbf{s}$ and $\mathbf{y}$ summed over the data, take derivatives w.r.t. the parameters and set to zero. The solution should look like linear regression.

5. Using the above, implement:

   `[mu, sigma, pie] = MStep(Y,ES,ESS)`

   You might be able to implement it using `ESS` = a $K \times K$ matrix summing over N of `ESS` as defined above.

6. Put the E step and M step code together into a function:

   `[mu, sigma, pie] = LearnMultipleCause(Y,K,iterations,gibbsflag)`

   where K is the number of causes, `iterations` are maximum number of iterations of EM, and `gibbsflag` = 1 means use Gibbs, otherwise, use mean field. For the mean field algorithm, make sure `F` always increases (this is a good debugging tool).

7. For some setting of `Y, mu, sigma, pie` plot various statistics of `S` as a function of sampling iteration for Gibbs sampling. Can you assess (visually) how long it takes for the Gibbs sampler to converge? How is this affected by increases and decreases in `sigma`? Why?

8. For some setting of `Y, mu, sigma, pie`, plot `F` and `log(F(t)-F(t-1))` as a function of iteration number `t` for `MeanField`. How rapidly does it converge? How is this affected by increases and decreases of `sigma`?

9. Examine the data `images.jpg` shown on the web site (Do **not** look at `genimages.m` yet!). This shows 100 greyscale 4×4 images generated by randomly combining several features and adding a little noise. Try to guess what these features are by staring at the images. How many are there? Would you expect factor analysis to do a good job modelling this data? How about mixture of Gaussians? Explain your reasoning.

10. Run your algorithm for learning the multiple cause model on the data set generated by `genimages.m`. What features `mu` does the algorithm learn (rearrange them into 4×4 images)? How do the Gibbs and variational algorithms differ? Which do you prefer? How could you improve the algorithm and the features it finds? Explain any choices you make along the way and the rationale behind them (e.g. what to set K, how to initialize parameters, hidden states, and `lambda`s).

11. **BONUS**: Given known values of σ^2 and $\boldsymbol{\pi}$, and a sample of $\mathbf{s}$ and $\mathbf{y}$, what is the conjugate prior for the $\boldsymbol{\mu}$s? Implement a sampling procedure for $\boldsymbol{\mu}$ given σ^2, $\boldsymbol{\pi}$, $\mathbf{s}$ and $\mathbf{y}$. You might want to use Gibbs sampling or Metropolis. Show some samples from the posterior distribution of `mu`.