

## Assignment 3: Latent Variable Models

### Unsupervised Learning

Zoubin Ghahramani

Due: Thurs Oct 30, 2003

Consider a data set of binary (black and white) images. Each image is arranged into a vector of pixels by concatenating the columns of pixels in the image. The data set has  $N$  images  $\{\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(N)}\}$  and each image has  $D$  pixels, where  $D$  is (number of rows  $\times$  number of columns) in the image. For example, image  $\mathbf{y}^{(n)}$  consists of a vector  $(y_1^{(n)}, \dots, y_D^{(n)})$  where  $y_d^{(n)} \in \{0, 1\}$  for all  $n \in \{1, \dots, N\}$  and  $d \in \{1, \dots, D\}$ .

Recall that a **Bernoulli random variable** has the following form  $P(y = 1|p) = p$  and  $P(y = 0|p) = 1 - p$  which we can write as  $P(y|p) = p^y(1 - p)^{(1-y)}$ .

A  $D$ -dimensional **multivariate Bernoulli variable** has the following form

$$P(\mathbf{y}|\mathbf{p}) = \prod_{d=1}^D p_d^{y_d} (1 - p_d)^{(1-y_d)}$$

where both  $\mathbf{y}$  and  $\mathbf{p}$  are  $D$ -dimensional vectors

5% Explain why a multivariate Gaussian is not an appropriate model for this data set of images.

Assume that the images were modelled as independently and identically distributed samples from a multivariate Bernoulli with parameter vector  $\mathbf{p} = (p_1, \dots, p_D)$ .

5% How many bits would it take on average to code this data set?

5% What is the equation for the maximum likelihood (ML) estimate of  $\mathbf{p}$  (recall assignment 1)? Note that you can solve for  $\mathbf{p}$  directly.

10% Assuming independent Beta priors on the parameters  $p_d$

$$P(p_d) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} p_d^{\alpha-1} (1 - p_d)^{\beta-1}$$

and  $P(\mathbf{p}) = \prod_d P(p_d)$  What is the maximum a posteriori (MAP) estimate of  $\mathbf{p}$ ? Hint: maximise the log posterior with respect to  $\mathbf{p}$ .

Download the data set `binarydigits.txt` which contains  $N = 100$  images with  $D = 64$  pixels each, in an  $N \times D$  matrix. These pixels can be displayed as  $8 \times 8$  images by rearranging them. Display the data set in Matlab by running `bindigit.m` (almost no Matlab knowledge required to do this).

- 10% Write code to learn the ML parameters of a multivariate Bernoulli from this data set and display these parameters as an  $8 \times 8$  image. Hand in your code and the learned parameter vector. (Matlab code is preferred, but C or Java are acceptable).
- 5% Modify your code to learn MAP parameters with  $\alpha = \beta = 3$ . What is the new learned parameter vector for this data set? Explain why this might be better or worse than the ML estimate.

#### Mixture Models:

- 10% Write down the likelihood for a model consisting of a mixture of  $K$  multivariate Bernoulli distributions. Use the parameters  $\pi_1, \dots, \pi_K$  to denote the mixing proportions ( $0 \leq \pi_k \leq 1; \sum_k \pi_k = 1$ ) and arrange the  $K$  Bernoulli parameter vectors into a matrix  $\mathbf{P}$  with elements  $p_{kd}$  denoting the probability that pixel  $d$  takes value 1 under mixture component  $k$ .

Just like in a mixture of Gaussians we can think of this model as a latent variable model, with a discrete hidden variable  $s^{(n)} \in \{1, \dots, K\}$  where  $P(s^{(n)} = k | \boldsymbol{\pi}) = \pi_k$ .

- 5% Write down the expression for the responsibility of mixture component  $k$  for data vector  $\mathbf{y}^{(n)}$ , i.e.  $r_{nk} \equiv P(s^{(n)} = k | \mathbf{y}^{(n)}, \boldsymbol{\pi}, \mathbf{P})$
- 20% Implement the EM algorithm for a mixture of  $K$  multivariate Bernoullis. The algorithm should take as input  $K$ , a matrix  $Y$  containing the data set, and a number of iterations. The algorithm should run for that number of iterations or until the log likelihood converges (does not increase by more than a very small amount). Beware of numerical problems as likelihoods can get very small, it is better to deal with log likelihoods. Also be careful with numerical problems when computing responsibilities — it might be necessary to multiply the top and bottom of the equation for responsibilities by some constant to avoid problems. Hand in code and a high level explanation of what your algorithm does.
- 15% Run your algorithm on the data set for varying  $K = 2, 3, 4$ . Verify that the log likelihood increases at each step of EM. Report the log likelihoods obtained (measured in *bits*) and display the parameters found.
- 10% Comment on how well the algorithm works, whether it finds good clusters (look at the responsibilities and try to interpret them), and how you might improve the model.