# CSC 2535 - Assignment 1.

Due: **Feb 6, 2013**, 1pm at START of class

## Graphical models

**Q. 1)** [7 marks] Section 8.21 in Bishop shows three example graphs that form the basis of D-separation, using 3 variables. Here we want to show how these hold in the slightly more general case of four variables.

(a) [1 mark] Consider a model with four variables - $w$, $x$, $y$ and $z$, and four parameters - $\theta_1$, $\theta_2$, $\theta_3$, and $\theta_4$ whose joint distribution is given by:

$$p(w, x, y, z) = p_{\theta_1}(w)p_{\theta_2}(x|w)p_{\theta_3}(z|x)p_{\theta_4}(y|z)$$

Mathematically, show that $w \perp y|z$, i.e. $p(w, y|z) = p(w|z)p(y|z)$, or, equivalently, that $p(y|w, z) = p(y|z)$. You can use the conditional independence property $p(a, c|b) = p(a|b)p(c|b)$, for a chain of 3 variables, A $\rightarrow$ B $\rightarrow$ C as given.

(b) [1 mark] Consider the alternative model given by:

$$p(w, x, y, z) = p_{\theta_1}(z)p_{\theta_2}(x|z)p_{\theta_3}(y|z)p_{\theta_4}(w|x)$$

Mathematically, show that $w \perp y|z$. Again, you can use the corresponding independence property for the model of three variables (A $\leftarrow$ B $\rightarrow$ C) as given.

(c) [2 marks] Consider the alternative model given by:

$$p(w, x, y, z) = p_{\theta_1}(x)p_{\theta_2}(y)p_{\theta_3}(w|x, y)p_{\theta_4}(z|w)$$

Show that $x \perp y$ marginally. Give a counterexample to the conditional independence assertion: $x \perp y|z$.

(d) [3 marks] Draw the corresponding graphical models for (a), (b), (c) above, with variable $z$ as the observed node in each case, and the other variables as unobserved.

## Variational Learning

**Q. 2)** [3 marks] Consider the following directed model. Let $\mathbf{z} = \{z_j\}, j = 1 \cdots J$ be the J latent binary variables (i.e. $z_j \in \{0, 1\}$), and $\mathbf{x} = \{x_i\}, i = 1 \cdots D$ be the D real-valued observed variables.
The probability distributions governing this model are as follows:

$$p(\mathbf{z}) = \prod_j a_j^{z_j} (1 - a_j)^{(1-z_j)}$$
$$p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\sum_j W_j z_j, \sigma^2 I)$$

where, $a_1, \cdots, a_J$ are the parameters of the Bernouilli prior on $\mathbf{z}$, $\mathbf{W}$ is a JxD matrix, defining the conditional distribution of $\mathbf{x}$ given $\mathbf{z}$, and $W_j$ is the $j^{th}$ row of $\mathbf{W}$. Data are assumed to be column vectors.
Remember that in the E.M. algorithm, we maximize the marginal distribution of the visible variables using the following equality (see Bishop equations 10.2-10.4):

$$\ln p(\mathbf{x}) = \mathcal{L}(q) + KL(q\|p) \tag{1}$$

where $q(\mathbf{z})$ is the posterior distribution, $p(\mathbf{z}|\mathbf{x}, \mathbf{\Theta}^t)$ of $\mathbf{z}$ using the current set of parameters, $\mathbf{\Theta}^t$.

(a) [1 mark] What is the number of possible states for the latent variable $\mathbf{z}$ in the above problem ? Are the components of $\mathbf{z}$ independent in the posterior $p(\mathbf{z}|\mathbf{x}, \mathbf{\Theta}^t)$? What does that imply about the computation in equation 1 above ?

(b) [1 mark] Instead of using $q(\mathbf{z}) = p(\mathbf{z}|\mathbf{x}, \mathbf{\Theta}^t)$, we will use $q$ as being another Bernouilli distribution:

$$q(\mathbf{z}) = \prod_j m_j^{z_j} (1 - m_j)^{(1-z_j)}$$

Show the following:

$$\begin{aligned} \mathcal{L}(q) = \quad & \textstyle\sum_j \left[ <z_j> \log \frac{a_j}{<z_j>} + (1 - <z_j>) \log \frac{1-a_j}{1-<z_j>} \right] \\ & - \frac{1}{2\sigma^2} \left[ \mathbf{x}^T \mathbf{x} - 2 \textstyle\sum_j <z_j> W_j \mathbf{x} + \sum_{j,k} <z_j z_k> W_j W_k^T \right] \\ & - \frac{D}{2} \log \sigma^2 + const \end{aligned} \tag{2}$$

where, the expectations <.> are computed wrt q($\mathbf{z}$).

(c) [1 mark] Using the following properties of equation 2:

$$<z_j> = m_j$$
$$<z_j z_k> = m_j m_k + \delta_{jk} \left( m_j - m_j^2 \right)$$

show that $\mathcal{L}(q)$ can be maximized by using the following update rule, for each case $c$:

$$m_j^c = g\left(\log \frac{a_j}{1-a_j} + \frac{1}{\sigma^2}\left[W_j\left(\mathbf{x}^c - \hat{\mathbf{x}^c}\right) + \left(m_j^c - \frac{1}{2}\right) W_j W_j^T\right]\right) \quad (3)$$

where, $g(u) = \frac{1}{1+\exp(-u)}$ is the sigmoid function, and $\hat{\mathbf{x}^c} = \left(\sum_j W_j m_j^c\right)^T$.

**Q. 3)** [10 marks] In this question we will fit the model we discussed in question 2 to some data.

The E step for EM was given in equation 3. The M step involves the following updates, summing over the training cases:

$$\begin{aligned} \mathbf{W} = &\quad \left[\sum_c M^{cT} M^c\right]^{-1} \left[\sum_c M^c \mathbf{m}^c \mathbf{x}^{cT}\right] \\ a_j = &\quad \frac{1}{C}\sum_{c=1}^{C} m_j^c \end{aligned}$$

where, $M_{i,j}^c = <z_i z_j>$. We assume that $\sigma$ is fixed.

The dataset is a set of 5x5 greyscale images, each consisting of a horizontal and a vertical bar, and some additive Gaussian noise. To load the training and test sets, invoke matlab and load the file `http://www.cs.toronto.edu/~hinton/csc2535/matlab/assign1.mat` This should work for all versions of matlab from version 6 up.

- Run the learning algorithm with different numbers of latent variables J. You should try J = 3, 10, 15. Examine the parameters W in each case, as well as the optimum values of the cost function.
- Try setting the prior for the values of each latent variable to be identical and constant, i.e., $a_j = a$ for all $j$. You can then set this constant value of $a$ to favor sparse solutions, e.g., $a = p(z_j = 1) = .3$; $p(z_j = 0) = 1 - a$. Modify your EM algorithm accordingly, and train the system with different values of $a$, such as 0.5 and 0.2. Try this for each setting of J above.
- For the model with J = 3, you can evaluate the true posterior by direct enumeration. Compare this to the variational approximation, for the test cases.

You should submit:

(a) [5 marks] One page or less containing a clear description of the results you obtained for each of the values of J, and the different settings of a.

(b) [5 marks] One page or less discussing what your results show about the variational approximation to the posterior. You should show how you computed the true posterior. Also, look at the data examples, and the model. Do you think this is a good model for this data?

You do not need to submit your code.