

CSC2535 Assignment 1

Due: Feb 6, 2008, 11am at START of class

January 25, 2008

Late assignments will have 25% subtracted from the total out of which they are graded for each day or part of a day that they are late. They will be more than one day late if they are not slipped under Prof. Zemel's office door before 11.00am the next day.

1 Conditional independence in graphical models

Each subsection is worth 1 point.

(a). Consider the model with input variable x , output (target) variable t , hidden (unobserved) variable y , and three parameters $\theta_1, \theta_2, \theta_3$:

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

Use Bayes' Rule to compute the distribution $p(y|x, t)$, where the final expression must be in terms of the $p_{\theta_1}, p_{\theta_2}, p_{\theta_3}$ distributions only.

(b). Draw the graphical model and determine whether x and t are conditionally independent given y . Check this by explicit calculation (i.e., does $p(t|y, x) = p(t|y)$?).

(c). We now receive an i.i.d. dataset $\{t_i, x_i\}, i = 1, \dots, N$. Write the expression for the maximum likelihood objective function (over both t and x) in terms of $p_{\theta_1}, p_{\theta_2}, p_{\theta_3}$ and the data. Note that we did not receive any data for y , so the final expression cannot depend on it.

(d). Now someone provides you with priors for the parameters. The expression for the joint distribution now becomes:

$$p(t, y, x, \theta_1, \theta_2, \theta_3) = p(t|y, \theta_1)p(y|x, \theta_2)p(x|\theta_3)p(\theta_1)p(\theta_2)p(\theta_3)$$

We observe x and t . Draw the corresponding graphical model, with appropriate shading. Check if the following statements are true:

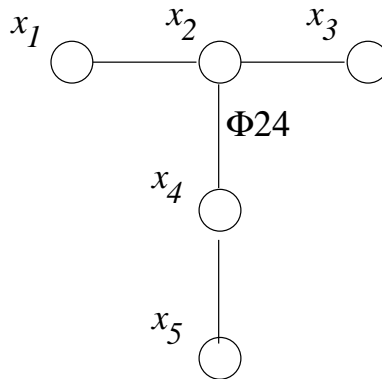
$$\begin{aligned}\theta_1 &\perp \theta_2 | \{t, x\} \\ \theta_1 &\perp \theta_3 | \{t, x\}\end{aligned}$$

(e). Use Bayes' Rule to express the posterior distribution $p(\theta_1, \theta_2, \theta_3 | \{t_i, x_i\})$ in terms of the distributions $p(t|y, \theta_1)p(y|x, \theta_2)p(x|\theta_3)p(\theta_1)p(\theta_2)p(\theta_3)$ and the data $\{t_i, x_i\}, i = 1, \dots, N$.

2 Inference via message-passing

Each subsection is worth 1 point.

Consider the MRF model below, where associated with every connected pair of nodes is a potential, with parameter(s) w : $\Phi_{jk}(x_j, x_k, w_{jk})$.



(a). Draw a factor graph corresponding to this MRF, and write down the messages into and out of node x_2 , as calculated by the sum-product algorithm.

(b). Calculate the marginal distribution for the factor associated with x_2 and x_4 . Here you can make use of the fact that the marginal distribution associated with a set of variables belonging to a factor s is

$$p(\mathbf{x}_s) = \frac{1}{Z} f_s(\mathbf{x}_s) \prod_{k \in n(f_s)} \mu_{k \rightarrow s}(x_k)$$

(c). Check that this marginal matches what you would calculate via direct enumeration on the MRF.

3 Learning using a variational approximation

A chief goal of unsupervised learning algorithms is to discover the underlying causes or factors that can characterize the observed data. One natural way to formulate this problem is using a directed acyclic graph, with latent variables representing the underlying causes, and observable variables to represent the data. Inference, and learning, is difficult in these models, as the latent variables are highly dependent given the observables. In this section you will explore a variational approach to this learning problem.

Consider formulating a directed acyclic graphical model, in which there are J latent variables $\mathbf{y} = \{y_j\}, j = 1, \dots, J$ and D real-valued observables, $\mathbf{x} = \{x_i\}, i = 1, \dots, D$. Values of \mathbf{x} are contained in greyscale images. We will treat the latent variables as binary variables, and the inputs as Gaussian variables.

The parameters are \mathbf{a} , the priors for the latent variables, and W , which defines their pre-

ditions for \mathbf{x} :

$$p(\mathbf{y}|\mathbf{a}) = \prod_j p(y_j|a_j) = \prod_j a_j^{y_j} (1 - a_j)^{(1-y_j)}$$

$$p(\mathbf{x}|\mathbf{y}, W) = \mathcal{N}(\sum_j W_j y_j, \sigma^2 I)$$

where I is a $D \times D$ identity matrix; we are assuming that W is a $J \times D$ matrix, and all the vectors are column vectors.

A simple variational approximation for this problem is:

$$Q(\mathbf{y}|\mathbf{x}, \mathbf{m}) = \prod_{j=1}^J m_j^{y_j} (1 - m_j)^{(1-y_j)}$$

The objective function is then (see tutorial notes):

$$F = \sum_j [\langle y_j \rangle \log(a_j / \langle y_j \rangle) + (1 - \langle y_j \rangle) \log((1 - a_j) / (1 - \langle y_j \rangle))] - \frac{1}{2\sigma^2} \left[\mathbf{x}^T \mathbf{x} - 2 \sum_j \langle y_j \rangle W_j \mathbf{x} + \sum_{j,k} \langle y_j y_k \rangle W_j W_k^T \right] - \frac{D}{2} \log \sigma^2 + const.$$

As explained in tutorial, the sufficient statistics are:

$$\langle y_i \rangle = m_i$$

$$\langle y_i y_j \rangle = m_i m_j + \delta_{ij} (m_i - m_i^2)$$

The updates for the E step of EM, for each input case c are:

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

where $g(z) = 1 / (1 + \exp(-z))$, and $\hat{\mathbf{x}}^c = \sum_j W_j m_j^c$.

For the M step, the updates involve summing over the training cases:

$$W = [\sum_c M^{cT} M^c]^{-1} [\sum_c M^c \mathbf{m}^c \mathbf{x}^{cT}]$$

$$a_j = \frac{1}{C} \sum_{c=1}^C m_j^c$$

where $M_{i,j}^c = \langle y_i y_j \rangle$.

Note that we are not updating σ here; you may choose to do so if you'd like.

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/~zemel/csc2535/matlab/assign1.mat>

This should work for all versions of matlab from version 6 up.

1. 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.
2. 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., $a = p(y_j = 1) = .3; p(y_j = 0) = 1 - a$. 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.
3. 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:

- (4 points) 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 .
- (5 points) 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?
- (3 points) One page or less discussing what you think these results show and describing **exactly** three other experiments you would do to make sure that your conclusions were correct.

You do not need to submit your code.