Assignment 1

Unsupervised & Probabilistic Learning

Maneesh Sahani & Yee Whye Teh

Due: Monday 18 Oct, 2010

Note: all assignments for this course are to be handed in to the Gatsby Unit, **not** to the CS department. Please hand in all assignments at the beginning of lecture on the due date to the lecturer. Late assignments will be penalised. If you are unable to come to class, you can also hand in assignments to Rachel Howes in the Alexandra House 4th floor reception.

1. **[10 points] Consistent beliefs**. A friend (perhaps not for much longer) reports that he is willing to bet on the following beliefs about two events A and B.

$$\begin{aligned} b(A) &= 0.5 \\ b(A \cap B) &= 0.5 \\ b(B) &= 0.6 \\ b(A \cup B) &= 0.7 \end{aligned}$$

   (a) Show that these beliefs are inconsistent. [3 points]

   (b) Construct a Dutch Book, which he would accept according to his beliefs, but which will guarantee that he will lose money. [7 points]

2. **[28 points] Statistics and Distributions**.

   In the coming weeks we will be making extensive use of the following distributions, which you should know. For each one of the exponential family distributions listed below look up or calculate:

   (a) the conventional probability distribution or density function as appropriate,

   (b) the mean and variance in terms of the conventional parameters,

   (c) the standard exponential form of the distribution or density function,

   (d) the natural parameters in terms of the conventional parameters (i.e., the function $\phi(\theta)$), and the sufficient statistic function (i.e., $\mathbf{T}(x)$).

   The distributions to consider are:

   Binomial, Multinomial, Poisson, Beta, Dirichlet, Gaussian, Gamma

   [4 points each]

3. **[27 points] Models for binary vectors**. 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 $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ and each image has D pixels, where D is (number of rows $\times$ number of columns) in the image. For example, image $\mathbf{x}^{(n)}$ is a vector $(x_1^{(n)}, \ldots, x_D^{(n)})$ where $x_d^{(n)} \in \{0, 1\}$ for all $n \in \{1, \ldots, N\}$ and $d \in \{1, \ldots, D\}$.

   (a) Explain why a multivariate Gaussian is not an appropriate model for this data set of images. [3 points]

Assume that the images were modelled as independently and identically distributed samples from a D-dimensional **multivariate Bernoulli distribution** with parameter vector $\mathbf{p} = (p_1, \ldots, p_D)$, which has the form

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

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

(b) How many bits would it take on average to code this data set, using the optimal code for the specified distribution? [2 points]

(c) What is the equation for the maximum likelihood (ML) estimate of $\mathbf{p}$? Note that you can solve for $\mathbf{p}$ directly. [5 points]

(d) 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 $\mathbf{p}$. [5 points]

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

(e) Write code to learn the ML parameters of a multivariate Bernoulli from this data set and display these paramteres as an 8×8 image. Hand in your code and the learned parameter vector. (Matlab or Octave code is preferred, but C, Java or R are acceptable). [7 points]

(f) 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. [5 points]

4. **[10 points] Model selection**. In the binary data model above, how would you calculate the (relative) probability of the three different models:

   (a) all D components are generated from a Bernoulli distribution with $p_d = 0.5$

   (b) all D components are generated from Bernoulli distributions with unknown, but identical, p_d

   (c) each component is Bernoulli distributed with separate, unknown p_d

5. **[10 points] Latent Variable Models**.

   (a) Describe a real-world data set which you believe could be modelled using factor analysis. Argue why factor analysis is a sensible model for this data. What do you expect the factors to represent? How many factors do you think there would be? Are the linearity and Gaussianity assumptions reasonable, and if not, how would you modify the model? [5 points]

   (b) Describe a real-world data set which you believe could be modelled using a mixture model. Argue why a mixture model is a sensible model for your real world data set. What do you expect the mixture components to represent? How many components (or clusters) do you think there would be? What parametric form would each component have? [5 points]

6. **[15 points] Principal Components Analysis.**

The conventional latent variable model for Probabilistic Principal Components Analysis has a standard normal latent $\mathbf{y}$ and an arbitrary *loading matrix* Λ.

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{0}, I)$$
$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\Lambda\mathbf{y}, \psi I)\,.$$

An alternative model would be to draw $\mathbf{y}$ from a normal with diagonal covariance (say Υ), and then restrict Λ to be orthogonal:

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{0}, \Upsilon)\,; \qquad \Upsilon_{ij} = 0 \text{ for } i \neq j$$
$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\Lambda\mathbf{y}, \psi I)\,; \qquad \Lambda^{\mathsf{T}}\Lambda = I.$$

(a) Show that this alternative model is equivalent to the standard one. [5 points]

(b) Derive the mean and covariance of $p(\mathbf{y}|\mathbf{x})$ within the alternative model in the PCA limit, $\psi \to 0$. [10 points]

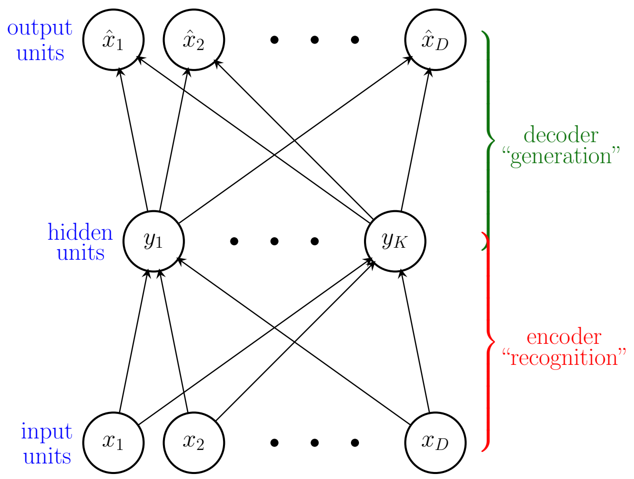
BONUS QUESTION: please attempt the questions above before answering this one.

7. **[Bonus: 20 points] Linear Autoencoders.** A linear autoencoder is a "neural network" implementation of PCA. The network has three "layers" of "units" — each unit represents a single scalar variable, and so each layer represents a vector:

$\hat{x}_j$	$j = 1 \ldots D$	output units
y_k	$k = 1 \ldots K < D$	hidden units
x_i	$i = 1 \ldots D$	input units

The mappings from input to hidden, and hidden to output layers, are linear:

$$y_k = \sum_i P_{ki}\, x_i$$
$$\hat{x}_j = \sum_k Q_{jk}\, y_k$$



Given a set of N observed "input" vectors $\{\mathbf{x}_n\}$, the weight matrices P and Q are set to minimise the "autoencoding error" $\sum_n \|\hat{\mathbf{x}}_n - \mathbf{x}_n\|^2$, often using iterative gradient descent. Assume the input vector distribution has zero mean.

(a) Show that after the weights have converged, they obey the identities:

$$P = (Q^{\mathsf{T}}Q)^{-1}Q^{\mathsf{T}}$$
$$Q = \Sigma_X Q^{\mathsf{T}}(Q\Sigma_X Q^{\mathsf{T}})^{-1}$$

where $\Sigma_X = \sum_n \mathbf{x}_n \mathbf{x}_n^{\mathsf{T}}$ and we assume that Q is rank K. [5 points]

(b) It is clear that if P and Q minimize the error, then, for any invertible $K \times K$ matrix C the matrices $P_* = CP$ and $Q_* = QC^{-1}$ also achieve the same minimum. Show that you can always find a matrix C such that $Q_*^{\mathsf{T}}Q_* = I$ and so $P_* = Q_*^{\mathsf{T}}$. [2 points]

(c) Show that the minimisation of the error is equivalent to the maximisation of

$$\mathsf{Tr}\left[Q_* Q_*^{\mathsf{T}} \Sigma_X\right]$$

and that this maximum is achieved by choosing the columns of Q_* proportional to the first K eigenvectors of Σ_X. (You may need to look up the relationship between the singular value decomposition—or eigendecomposition of a symmetric matrix—and low-rank approximations). [7 points]

This demonstrates the assertion made in lecture that the linear autoencoder will find the subspace of the leading K principal components.

(d) How would you adapt the algorithm to implement factor analysis in the case that the uniquenesses (another term for the output noise variances Ψ_{dd}) are known? [3 points]

(e) Is it possible to use an autoencoder for FA in the case of unknown uniquenesses? How, if so; or why not, if not? [3 points]