VICTORIA UNIVERSITY OF WELLINGTON

Te Whare Wananga o te Upoko o te Ika a Maui

EXAMINATIONS — 2003

END-YEAR

COMP 421

Machine Learning

Time Allowed: 3 Hours

Instructions: There are 12 questions: each question is worth 20 marks.

Answer NINE (9) questions (180 marks). If you answer more, only your best 9 questions will be taken.

TWO pages (i.e. two sides only) of hand-written notes are permitted.

Non-electronic Foreign-English language dictionaries are permitted.

Question 1. Various topics [20 marks]

(a) [7 marks] A mixture of Gaussians clustering model comes up with the following log (base 10) probability of the data $\mathbf{D}$, when run with different numbers of clusters, K:

K	$\log_{10} p(\mathbf{D}\|K)$
2	-103
3	-100

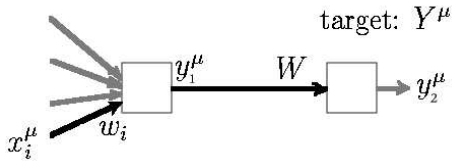
In order for the peak *posterior* probability to be at $K = 2$, how strong would your *prior* preference need to be for believing that $K = 2$ rather than 3? That is, how high would $\frac{P(K=2)}{P(K=3)}$ have to be?

(b) [7 marks] Describe the "Baldwin effect", an interaction between learning and evolution.

(c) [6 marks] A stochastic binary neuron sees an input pattern $\mathbf{x}$ and outputs a 1 with probability y. If the binary target output is Y, give an expression (in terms of y and Y) for the expected log probability that the neuron's output will be correct for this $\mathbf{x}$.

Question 2. Network training [20 marks]

Consider a network consisting of just two model 'neurons' as shown:



The connection from (say) the i^{th} input x_i^μ to the first neuron has weight w_i. The connection from the first neuron to the second has weight W. Both neurons are continuous-valued 'sigmoid units', meaning their output for the μ^{th} input pattern is given by

$$y^\mu = \frac{1}{1 + \exp(-\phi^\mu)}$$

where ϕ^μ is their weighted sum of inputs. This miniature network is to be trained to minimize the squared-error cost function

$$C = \sum_\mu C^\mu \qquad \text{where} \quad C^\mu = \frac{1}{2}(Y^\mu - y_2^\mu)^2$$

on a set of input-output pairs. Derive the batch update rule for changing w_i so as to perform gradient descent of C.

Question 3. Reinforcement learning [20 marks]

(a) [10 marks] Outline the major assumptions made by *Value Iteration* (a form of Dynamic Programming) as an approach to arriving at an optimal policy, and contrast these with the approach taken by the algorithm SARSA.

(b) [10 marks] A drawback of 'vanilla' Q-learning is that the look-up table $Q_{s,a}$ (where s is a discrete state and a is a discrete action) is inapplicable to continuous state or action spaces, where one or both of s and a is a real-valued vector. One approach to continuous states and actions is to use a function approximator - for example we might assume that $Q_{s,a}$ is a linear function of s and a and attempt to learn the parameters of this function. Describe how you could do this, including a sensible online learning rule for improving the parameters.

[Hint: it might help to think of $Q_{s,a}$ as the output of a linear model neuron with learnable weights.]

Question 4. Gradient descent [20 marks]

Consider a slightly non-standard model 'neuron' which, when given the input vector $\mathbf{x}^\mu$, produces the output

$$y^\mu \;=\; \exp(\phi^\mu) \quad \text{where } \phi^\mu \;=\; \sum_i w_i \, x_i^\mu$$

Notice the inclusion of exp. This neuron is exposed to a training set of input-output pairs $\{\mathbf{x}^\mu,\ Y^\mu\}$ where Y is the 'target' output value. The following cost function is proposed for this neuron to minimize:

$$C \;\;=\;\; \sum_\mu (Y^\mu - y^\mu)^4$$

Note the unusual exponent in the cost.

(a) [14 marks] Derive the gradient descent learning rule for updating the weights of this neuron.

(b) [6 marks] For my own strange reasons, I also wish to "decay" all weights towards some constant, a. One could do this by making an additional change to each weight of

$$-\beta \, (a - w_{ij})$$

where β is some small positive number. This is equivalent to adding a new term to C penalising weights that are far from a. What is this penalty term?

Question 5. Probabilistic inference [20 marks]

(a) [10 marks] Describe the three Cox-Jaynes axioms for beliefs or "plausabilities", and discuss their significance for probabilistic inference.

(b) [5 marks] According to the Bayesian point of view, in making predictions we should integrate over models and parameters. Explain what is meant by this statement,

(c) [5 marks] Relate this theory (*i.e.* the Bayesian point of view given above) to ML (maximum likelihood) and MAP (maximum a posteriori) learning.

Question 6. Mixtures of Gaussians [20 marks]

Suppose you are provided with an unlabelled data set of real-valued vectors (*i.e.* there are no "targets", just input vectors). You are to assume a mixture of Gaussians model for the distribution of these vectors.

(a) [10 marks] If you know the number of components (say K) in the mixture, describe how you would go about estimating the probability density at some particular point $\mathbf{x}$,

$$p(\mathbf{x}|\mathbf{D}, K)$$

as *accurately as possible*. You may assume the efficiency of this calculation is not an issue.

(b) [10 marks] Describe how you would go about estimating $p(\mathbf{x}|\mathbf{D})$ in the case where K is *not* known.

Question 7. Markov Chain Monte Carlo [20 marks]

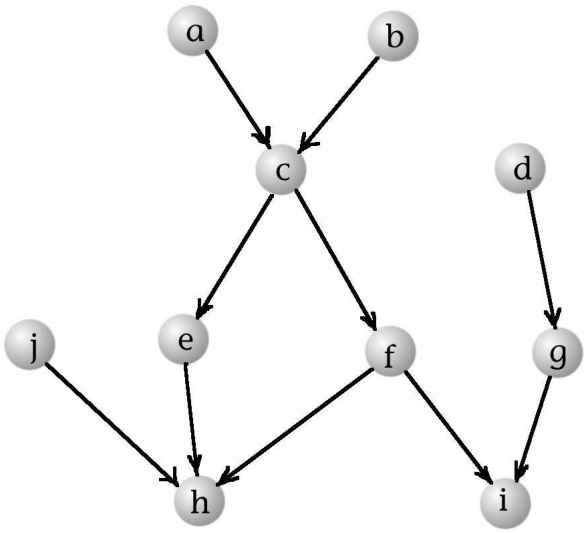
(a) [5 marks] MCMC methods were put forward as a possible solution to a fundamental difficulty with sampling methods such as rejection and importance sampling. What is this difficulty and why is it such a problem?

(b) [5 marks] Describe the Metropolis-Hastings algorithm, with mention of the proposal and acceptance distributions.

(c) [10 marks] Show why virtually *any* proposal distribution still leads to detailed balance in the Metropolis-Hastings algorithm.

Question 8. Belief net semantics [20 marks]

Consider the following belief network:



(a) [4 marks] Write out the factorization of the joint probability distribution expressed by this graph.

(b) [4 marks] Draw the factor graph.

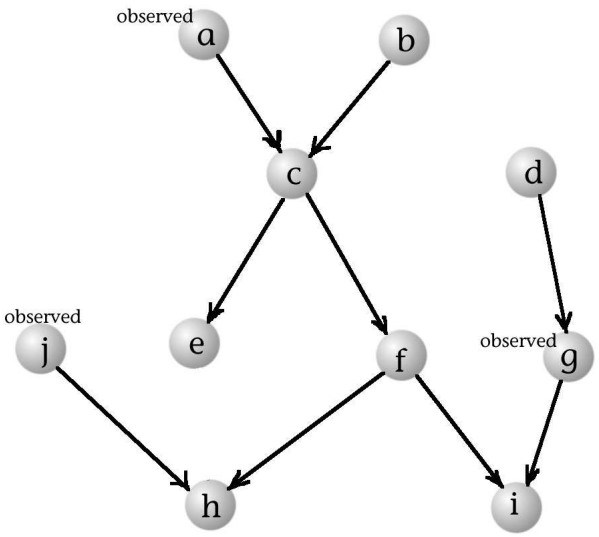
(c) [4 marks] If all the variables are binary, how many independent probability values need to be specified in order to make inferences using this belief network? Show your working.

(d) [4 marks] Is variable g conditionally dependent or conditionally independent of c, given i? Justify your answer.

(e) [4 marks] Is variable e conditionally dependent or conditionally independent of i, given c? Justify your answer.

Question 9. Probability propagation [20 marks]

Consider the following belief net:



Note this is **not** quite the same belief network as in the previous question: the arrow from e to h has been deleted. Assume further that the following observations have been made:

- $a = 1$
- $g = 1$
- $j = 0$

The following questions relate to the use of the probability propagation message-passing algorithm on this new graph.

(a) [3 marks] What is in the message arriving at variable node f *from above*?

(b) [3 marks] What is in the (sole) message leaving node i?

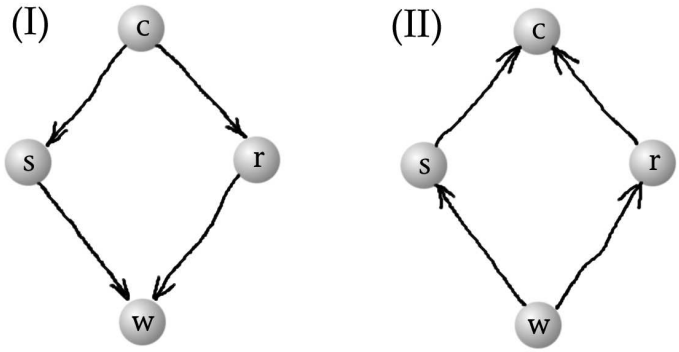
(c) [3 marks] What is in the message leaving f, going to the factor between f and h?

(d) [3 marks] What is in the message from the factor between f and h, going to f?

(e) [8 marks] You have met the Viterbi algorithm in the context of HMMs. Describe what it is used for and how it could be applied to belief networks in general.

Question 10. Loopy graphs, and causality [20 marks]

Consider the following two belief nets:



(a) [5 marks] For network (I), describe in words how the technique of *grouping* (or *merging*) variables can be used to allow use of the probability propagation algorithm.

(b) [5 marks] For network (II), describe in words how the technique of *cut-set conditioning* can be used to allow use of the probability propagation algorithm.

(c) [5 marks] Explain why no amount of passively observed data (for example, records of many concurrent observations of all four variables) would allow one to infer which of the above networks most correctly represents the *casual* relationships between variables.

(d) [5 marks] Explain how you could go about testing which network best captured the true causal relationships.

Question 11. Hidden Markov models [20 marks]

The HMM architecture discussed in the course involved observations that were each chosen from a fixed number of discrete possibilities.

However in many applications (such as speech recognition) we'd like to model observations that are real-valued vectors.

Describe how the HMM architecture could be extended to handle the case where the hidden state is discrete and Markovian, but the observation corresponding to each state is a fixed vector corrupted by Gaussian noise. Assume that the number of clusters (*i.e.* the number of hidden states) is known beforehand.

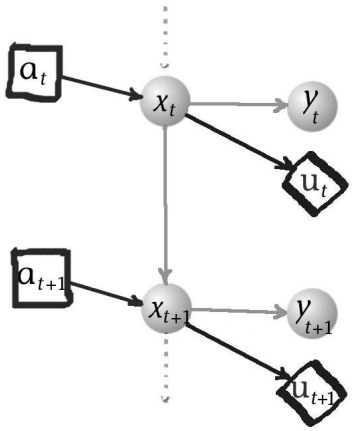
You might like to address issues such as

- what form do messages in the graph take?
- how might the means and variances be learned from data?
- would the Viterbi algorithm still work in this architecture?

Provide as much detail as you are able to.

Question 12. Decision networks [20 marks]

Consider the following graph, in which the usual HMM structure has been augmented with decision (or action) nodes labelled a, and utility nodes labelled u. The utilities are (estimated) instant rewards for each of the possible states the system can be in. You may assume that all look-up tables are the same for each time-step: for example the utility table for time t is the same as it is at $t+1$. States x are hidden (never observed).



(a) [5 marks] Briefly describe the type of process this network could be used to model.

(b) [5 marks] The utility table starts off as a set of estimated values (one for each state), but these could be improved by learning from actual sequences in which rewards r_t are received at each time-step. Suggest an algorithm for learning the utility values of states.

(c) [5 marks] Change the previous algorithm so that it learns the long term discounted reward instead of the instant one.

(d) [5 marks] For time $t+1$, describe how to decide which action a_{t+1} it is best to take, in the light of actions and observations up to and including t. Note that to make this decision you may need to consider the impact of future actions as well. You may ignore concerns over the tractability of this algorithm.

****