CSE 573 – Artificial Intelligence I
Autumn 2001
Instructor: Henry Kautz

Assignment #4

Part I (written problems) due Wednesday Dec 12th in class.
Part II (programming problem) due Monday Dec 17th, turned in to Don Patterson in the manner he specifies.

Part I – selected questions from the attached problem set labeled "CS 221".

1. (Bayes Nets) – Problem 1 from attached set.
2. (Neural Nets) – Problem 3 from attached set.
3. (Decision Trees) – Problem 5 from attached set.

Note that you do not need to do the attached problems 2 or 4.

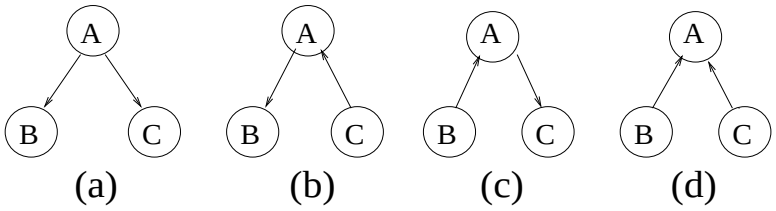
CS 221, Autumn 2000
Problem Set #4 - Bayesian Nets, Machine Learning
Handout #29

---

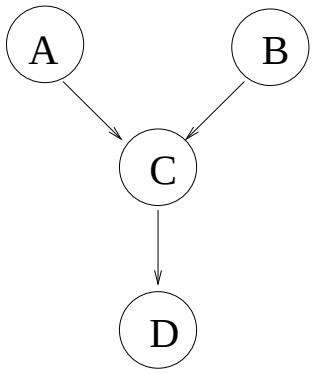
1. **[17 points] I-Equivalence**

   Given two Bayesian networks over the same variables, bn_1 and bn_2, bn_1 and bn_2 are said to be *I-equivalent* if all d-separation properties of bn_1 also hold for bn_2 and vice versa.

   (a) **[1 points]** In the figure below, which of the four networks are I-equivalent?

   (b) **[2 points]** In the network pictured below, enumerate all of the conditional independencies represented by the network. Each independence should be of the form $I(X, Y \mid \mathbf{Z})$ where X and Y are variables and $\mathbf{Z}$ is a (possibly empty) set of variables. Note that if your list contains $I(X, Y \mid \mathbf{Z})$ then it need not contain $I(Y, X \mid \mathbf{Z})$.

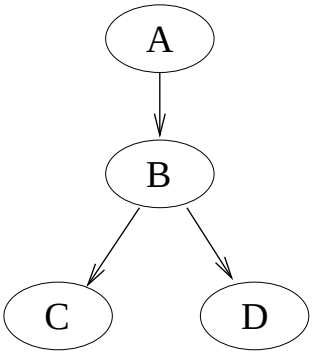
   

   (c) **[6 points]** Is there another I-equivalent network to this network (except itself)? If so, draw the equivalent network. If not, **briefly** explain using high level reasoning based on the independencies why no other network over these four variables can be equivalent.

   (Hint: certain independencies that you listed in (b) will allow you to eliminate, in one shot, many of the possible structures as clearly being not I-equivalent to this one. Thus, your answer could have the following format: list an independence assertion from part (b), and conclude certain properties that the structure will have to satisfy in order to make this independence assertion true. Repeat this process until you have

found a structure satisfying all the independencies, or until you show that none can exist.)

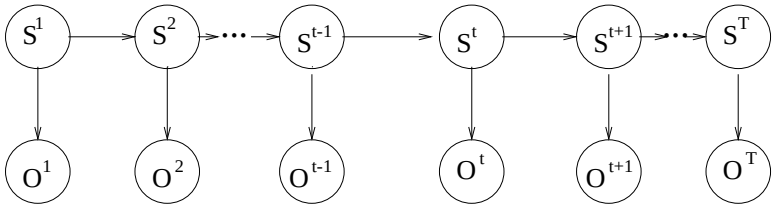
(d) **[2 points]** In the network pictured below, enumerate all of the conditional independencies represented by the network.



(e) **[6 points]** Is there another I-equivalent network to this network (except itself)? If so, draw the equivalent network. If not, **briefly** explain using high level reasoning based on the independencies why no other network over these four variables can be equivalent.

2. **[20 points] Hidden Markov Models**

Consider the following simple Bayesian network:



This type of network is often used to represent and track the state of a system that evolves over time. At each time point i, the system is in some state, which is one of $s_1, \ldots, s_n$. The state at time t is denoted using the random variable S^t. The state of the system at time $t+1$ depends only on the state at time t, as reflected by the arcs in the network. At each time point t, the user gets some observation O^t, which is one of $o_1, \ldots, o_m$. The observation at time t depends only on the state S^t (again, as implied by the network structure).

As implied by the structure of this network, we have the following parameters. For the initial state, we have $P(S^1 = s_i)$ for all $i = 1, \ldots, n$ (recall that there are n possible states that the system can be in at any point). For each time slice $t = 2, \ldots, T$, and each pair of states s_i and s_j, we have $P(S^t = s_i \mid S^{t-1} = s_j)$; this is the probability that if the process is in state s_j at time $t-1$, then it will be in state s_i at time t. Finally, for each t, each state s_i and each possible observation o_k, we have a parameter $P(O^t = o_k \mid S^t = s_i)$.

Our goal in this problem is to come up with a simple algorithm for tracking the state of the system as it evolves. I.e., we will show how we can efficiently compute $P(S^t \mid o^1, \ldots, o^t)$,

where o^i is the actual observation received at time i.

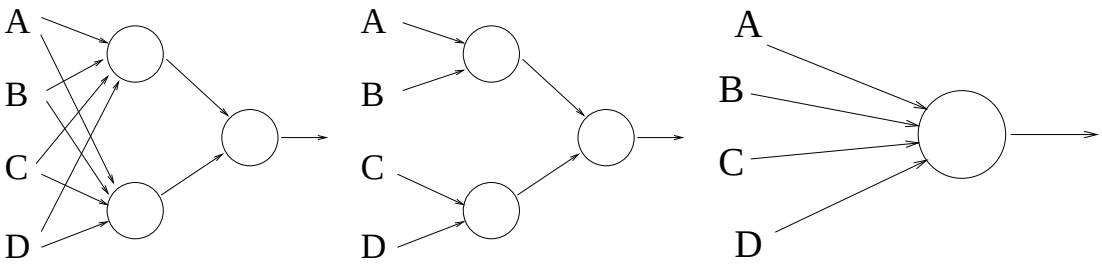
(a) **[4 points]** Let o^1 be the value of the observation obtained at time 1, i.e., it is the observed value of the node O^1 in the network. Because of the structure of this network, we can find $P(S^1 \mid o^1)$ *without* generating the entire joint distribution over $S^1, \ldots, S^T, O^1, \ldots, O^T$. State the independence assumptions (implied by the network structure) that allow us to do this, and derive a formula for $P(S^1 \mid o_1)$ in terms of network parameters (but not in terms of the full joint distribution). Note that $P(S^1 \mid o^1)$ is a probability distribution over $s_1, \ldots, s_n$, so that your formula should allow us to compute $P(S^1 = s_i \mid o^1)$ for any i, and these n numbers should sum to 1.

(b) **[12 points]** Now, assume that we have computed $P(S^t \mid o^1, \ldots, o^t)$ (part (a) shows us how to do this for $t = 1$). How could we use that to compute $P(S^{t+1} | o^1, \ldots, o^{t+1})$?

(c) **[4 points]** What is the complexity of your algorithm for computing every $P(S^t \mid o^1, \ldots, o^t)$ for $t = 1, \ldots, T$ from the network parameters? Your answer should be a formula in terms of some of: n (the number of possible states), m (the number of possible observations), and T (the number of time slices).

3. **[21 points]** EXPRESSIVITY OF NEURAL NETWORKS

This question investigates the relative expressivity of neural networks for the class of boolean functions, i.e., functions for which *true* is represented as the numerical value 1 (both in the inputs to the network and in its output) and *false* is represented as 0. Consider the three networks presented in Figure 3 over the four *distinct* boolean variables A, B, C, and D.



(1) (2) (3)

All three networks use the step function as the activation function for all units. Network 1 is a standard 3-layer neural network with two hidden units. Network 2 is a restricted 3-layer neural network, where each of the two hidden units only gets to see two of the inputs. Network 3 is a standard perceptron.

It should be clear that network 1 is at least as expressive as networks 2 and 3. We also know, from class, that it is strictly more expressive than network 3, since there is a Boolean function — XOR — that network 1 can represent but network 3 cannot. In this exercise, you will have to determine the other relations between the expressive power of these networks.

(a) **[7 points]** We know that network 1 is at least as expressive as network 2. Is it strictly more expressive? If so, provide an example of a Boolean function that can be represented by network 1 and not by network 2, and explain why it cannot be represented by network 2. If not, prove that any function that can be represented by network 1 can also be represented by network 2.

(b) **[7 points]** Is network 2 as expressive as network 3? If so, prove that any function that can be represented by network 3 can also be represented by network 2. Otherwise, provide an example of a Boolean function that can be represented by network 3 and not by network 2, and explain why it cannot be represented by network 2.

(c) **[7 points]** Is network 3 as expressive as network 2? If so, prove that any function that can be represented by network 2 can also be represented by network 3. Otherwise, provide an example of a Boolean function that can be represented by network 2 and not by network 3, and explain why it cannot be represented by network 3.

4. **[12 points] Learning rules**

Numerical overfitting in neural networks occurs when the weights are overtrained and become too large. One way to avoid that phenomenon is to train with a different error function, one which penalizes large weights. For example, consider a single sigmoid perceptron, parameterized by a vector of weights $\vec{w} = (w_0, \ldots, w_k)$, and let $h_{\vec{w}}$ be the current hypothesis. For a given target instance $\vec{x}, t$ (with t the target value), we define the error function to be

$$E(\vec{w}) = \frac{1}{2}[(t - h_{\vec{w}}(x))^2 + \lambda \sum_{i=1}^{k} w_i^2]$$

In other words, we add to the traditional squared error function a term that penalizes for large weights.

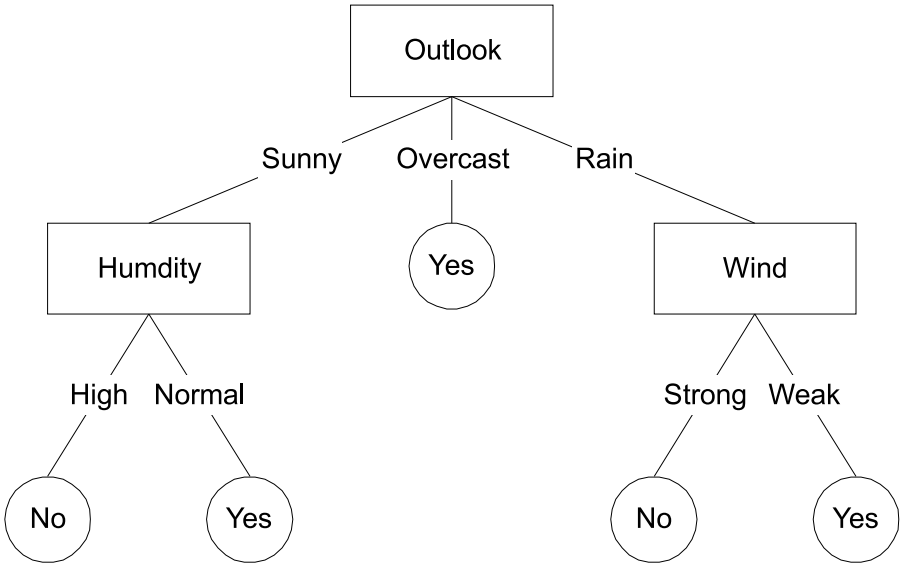
(a) **[4 points]** Derive an update rule for the weights of a sigmoid perceptron relative to this error function. Show your derivation, in addition to the final rule.

(b) **[8 points]** Now, consider a similar error function for the case of a multi-layer neural network. Write down the update rule for the weights w_{ij} of a neural network relative to this error function. In this case, it is enough to write down the rule; there is no need to show a derivation. (Hint: In fact, you shouldn't need to do another derivation.)

5. **[20 points]** Decision Trees

In this question we assume that there is some decision tree generating the data. This question investigates whether we can accurately reconstruct the original decision tree by using the decision tree learning algorithm on data generated from the original tree.

Suppose Beatrice uses the following decision tree to decide whether she will practice her archery:



We wish to reconstruct her original decision tree by observing when and when she does not go to practice her archery. Suppose we gather the following observations over the course of two weeks:

Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*Practice Archery*
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Notice that ID3 (the decision tree learning algorithm presented in class that uses information gain to split on attributes) actually learns the original tree perfectly when presented with these 14 training instances. In the following questions we will require that all of the training data are consistent with Beatrice's original tree, meaning the PracticeArchery label on each of the training data should be what Beatrice herself would do if she were to observe those settings of the attributes. (Recall that we assumed that Beatrice is using the tree shown above.) Duplicate training examples are allowed; they count as separate individual examples in the information gain computations.

(a) [**4 points**] Is it possible for a training set to get ID3 to learn a tree that is identical

to Beatrice's tree except that the learned tree further elaborates the tree below the rightmost leaf? Justify your answer.

(b) [**9 points**] Is it possible to add more training examples to the original 14 examples listed above that will cause ID3 to initially split on the attribute *Temperature* at the root node even though the original tree that Beatrice uses is independent of *Temperature*? You should explain your answer at the level of arguing about information gain, either by using formulæ or explaining clearly in words.

(c) [**7 points**] We will say that a tree is *incorrect* if there is a setting of the attributes { *Outlook, Temperature, Humidity, Wind* } that will cause this tree to classify *Practice Archery* differently from Beatrice's tree.

Is it possible to get ID3 to learn an incorrect tree by adding new correct examples to the original fourteen. Justify your answer. **Hint:** you may like to see whether your answer to part 5b can help you answer this question.