6.824 Artificial Intelligence, Fall 1993
Midterm Exam

name ______________________________

There are four problems. Write your answeres in the space provided or on the back of the sheets. The point value of each problem is shown. You have three hours.

Problem 1. (25 pojnts) This problem is on PAC learning. Consider objects with n linear attributes where each object assigns each attribute x_i an integer value from 1 to 2^k (k is the number of bits used to represent the integer x_i). We consider conjunctive concepts defined by conditions of the form

$$x_1 \in [l_1,\ u_1] \wedge x_2 \in [l_2,\ u_2] \wedge \ldots \wedge x_n \in [l_n,\ u_n]$$

where each l_i and u_i is an integer in the range $[1,\ 2^k]$ and $l_i \leq u_i$.

part a. (8 points) From the cardinality of this concept space give a number of examples that suffices to ensure PAC learning, i.e., such that with probability $1 - \delta$ all concepts consistent with the examples have error rate no larger than ϵ. Your answer should be given in terms of n, k, δ and ϵ.

part b. (8 points) Derive a number of examples sufficient for PAC learning using the VC dimension of this concept space. Your answer should depend only on n, δ, and ϵ (and not on k).

part c. (9 points) The answers to both parts a and b should be linear in n (holding the other parameters fixed). Give a relationship between k and ϵ which guarantees that the sample complexity from part a is smaller than the sample complexity from part b (Hint: you only need to look at the linear term containing n in your answers to parts a and b). Assume ϵ is $\frac{1}{32}$ (or 2^{-5}) and give an approximate value for k such that if we use no more than k bits per number then the bound from part a is no smaller than the bound from part b.

Problem 2. (25 points) This problem is on the backpropagation learning procedure. Suppose that we represent a function as a mixture of Gaussians.

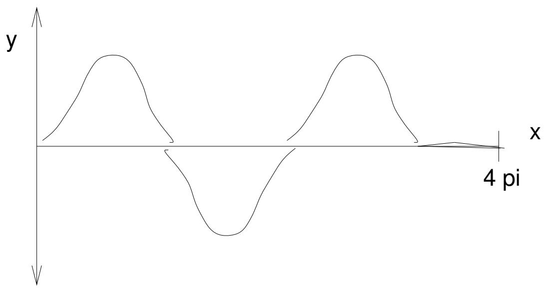
$$f(x) = \sum_{1 \leq j \leq k} a_j e^{\frac{-(x-m_j)^2}{2\sigma_j^2}}$$

Assume that you are given the parameters a_j, m_j, and σ_j that define the function f. Also suppose that you are given a set $\{\langle x_1,\ y_1\rangle,\ \ldots,\ \langle x_n,\ y_n\rangle\}$ of desired input/output pairs.

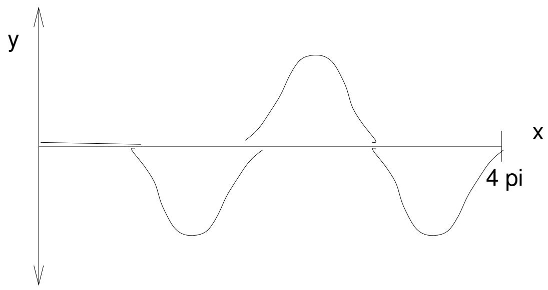
part a. (9 points) Give gradiant descent equations for updating the parameters a_j, m_j, and σ_j to reduce the error $\sum_i (y_i - f(x_i))^2$.

part b. (6 points) Suppose that the function we are trying to learn is actually a sine wave. The data points $\langle x_i, y_i\rangle$ consist of 100 points where the values x_i are evenly spaced from 0 to 4π and that $y_i = \sin x_i$. Also suppose that we are using three Gausians ($k = 3$). Draw graphs of two different functions f_1 and f_2 each of which are a sum of three Guasians and each of which are possible functions that the gradiant descent procedure could converge on given this data.

f1



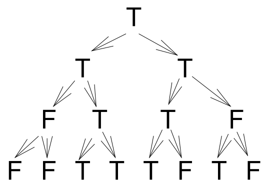
f2



part b. (9 points) This form of gradiant descent is a special case of a least squares fit (we are trying to minimize the sum squared error). Is a least squares fit in general best viewed as a maximum likelihood (ML) technique or a maximum a-posteriori probability (MAP) technique. Explain your answer.

Problem 3. (25 points) Consider a Bayesian network which is an n level binary tree with arcs directed from parents to children and ending in the 2^n leaves of the tree. Each node in this tree is a Boolean variable where the probability that a node has the same value as its parent is $(1 - \epsilon)$ with $0 < \epsilon < \frac{1}{2}$. The prior probability that the root node is true is $\frac{1}{2}$. We use P_n to represent a node that is n arcs above the leaf nodes.

part a. (6 points) Consider the execution of the tree network shown below. What is the probability of the given execution, i.e., what is the probability of this assignment of values to nodes. Your answer should be given as a function of ϵ.



part b. (9 points) Consider the case where all leaf nodes are observed to have value T. Recall that $\beta_{P_n}(T)$ is the probability of the oservations below the node P_n given that $P_n = T$. $\beta_{P_n}(F)$ is defined similarly. For each n these numbers can be written as polynomials in ϵ. Since ϵ is in the interval $(0,\ \frac{1}{2})$ we take the order of a polynomial in ϵ to be *smallest* exponent of ϵ in a nonzero term. For example, the order of $1 + \epsilon$ is 0 and the order of $3\epsilon^2 + \epsilon^3$ is 2. The polynomial 0, i.e., the function that is always 0, should be considered to have infinite order (note that $0 = \epsilon^\infty$). Give the orders of the polynomials for $\beta_{P_n}(T)$ and $\beta_{P_n}(F)$ as a function of n. (The answer is a very simple function.)

part c. (9 points) Now consider a set of observations O at the leaves of a tree with n levels of arcs and root node n. We allow O to be arbitrary (some of the observed values may be F). For the fixed observation set O consider all the possible ways that the network can be executed starting with root value $P_n = T$ and ending in the observed values O. A given execution corresponds to a particular set of values for all the internal nodes of the tree. For a given execution we consider the number of "changes" in the execution, i.e., the number of nodes whose values are different from the value of its parent. Define $B_{P_n}(T)$ to be the minimum over all executions of the number of changes of that execution. Circle the true statement in the following list and explain your answer.

- The order of the polynomial for β_{P_n} is always less than $B_{P_n}(T)$.
- The order of the polynomial for β_{P_n} is always larger than $B_{P_n}(T)$.
- The order of the polynomial for β_{P_n} is always equal to $B_{P_n}(T)$.
- For trees and data sets the order of the polynomial for β_{P_n} is less than $B_{P_n}(T)$ but for other trees and data sets the order of the polynomial for β_{P_n} is greater than $B_{P_n}(T)$.

Hint: $\beta_{P_n}(T)$ is the sum over all executions E, consistent with the oservations O, of $P_r(E \mid P_n = T)$.

Problem 4. (25 points) This problem is about using inference rules to reason about Bayesian networks. Suppose we are given a tree structured Bayesian network with observed values at all of the leaves. The tree may have nonuniform depth. We want to find the most likely execution of the network given that the root node has value T and the leaf nodes have their observed values. We assume that the Bayesian network is a binary tree. Every node other than the leaves has exactly two children. We suppose that the Bayesian network and the observations O are represented by assertions in a data base of the following forms.

- The assertion (`LEFT-CHILD` n m) means that m is the left child of n.
- The assertion (`RIGHT-CHILD` n m) means that m is the right child of n.
- The assertion (`CONDITIONAL-COMPLEXITY` X x Y y w) means that Y is a child node of the node X and $w = -\log_2 P_r(Y = y \mid X = x)$.
- The assertion (`OBSERVED` Z z) means that Z is a leaf node which has been observed to have value z.

To find the most likely execution given the data we let E_X be the set of possible executions of the nodes at or below X, i.e., the set of all possible assignments of values to X and the nodes below X. For a given node X and possible value x for X we define $\delta_X(x)$ as follows.

$$\delta_X(x) = \max_{e \in E_x} P_r(e \wedge O | X = x)$$

In English, $\delta_X(x)$ is the probability of the most likely execution e of the nodes at or below X such that e is consistent with the observations O. If Z is a leaf node observed to have value z then $\delta_Z(z) = 1$. If Z is a leaf node and w is some possible value for Z other than the observed value then $\delta_Z(w) = 0$. We define $C_X(x)$ to be $-\log_2 \delta_X(x)$. Note that if w is a possible value for Z other than the observed value for z then $C_Z(w) = \infty$. In this case we can ignore the value w for Z. If z is the observed value for leaf X then $C_Z(z) = 0$.

We wish to infer assertions of the following form.

- The assertion ($\leq$ (`C` X x) w) means that $C_X(x) \leq w$

part a. (13 points) Give two inference rules for infering bound assertions such as the above. The inference rules should be capable of infering an exact value for each complexity $C_X(x)$ in the network.

part b. (12 points) Assume that there are n nodes in the network and that each node has d possible values. Give the order of running time for executing the rules you gave in part a according to the "Dijkstra tightest bounds" algorithm. Your answer should be in terms of n and d.