FINAL: CS 6375 (Machine Learning) Fall 2014

The exam is closed book. You are allowed a one-page cheat sheet. Answer the questions in the spaces provided on the question sheets. If you run out of room for an answer, use an additional sheet (available from the instructor) and staple it to your exam.

- **NAME** _____
- **UTD-ID if known** _____

Solutions

Question 1: Decision Trees (10 points)

Consider the training dataset given below. In the dataset, X_1, X_2, and X_3 are the attributes and Y is the class variable.

Example#	X_1	X_2	X_3	Y
E1	0	0	0	+ve
E2	0	0	1	-ve
E3	0	1	0	-ve
E4	0	1	1	+ve
E5	1	0	0	-ve

(a) (3 points) Which attribute has the highest information gain? Justify your answer?

Solution:

$$IG(X_1) = \texttt{Entropy}(2,3) - \frac{1}{5}\texttt{Entropy}(0,1) - \frac{4}{5}\texttt{Entropy}(2,2) = 0.1709$$

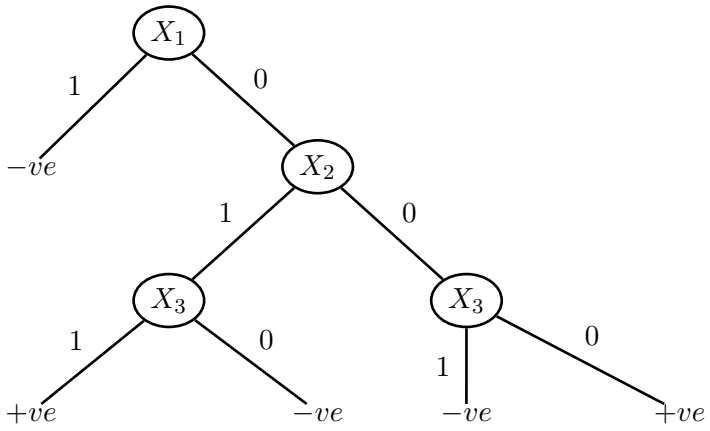
$$IG(X_2) = \texttt{Entropy}(2,3) - \frac{2}{5}\texttt{Entropy}(1,1) - \frac{3}{5}\texttt{Entropy}(1,2) = 0.0199$$

$$IG(X_3) = \texttt{Entropy}(2,3) - \frac{2}{5}\texttt{Entropy}(1,1) - \frac{3}{5}\texttt{Entropy}(1,2) = 0.0199$$

Therefore, X_1 has the highest information gain.

(b) (3 points) Draw the (full) decision tree for this dataset using the information gain criteria.

Solution:



Note that X_3 and X_2 are interchangeable.

Suppose that we have the following validation dataset:

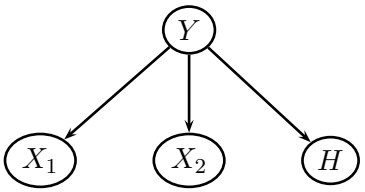
Example#	X_1	X_2	X_3	Y
E6	1	1	0	+ve
E7	0	1	1	-ve
E8	1	1	1	-ve

(c) (4 points) Can you construct a decision tree that will have 100% accuracy on the validation dataset as well as the training dataset. Answer Yes or No. If your answer is yes, then draw the decision tree that will have 100% accuracy. If your answer is no, then explain why it cannot be done. No credit will be given if the explanation is incorrect.

Solution: It is not possible because examples $E4$ and $E7$ have the same feature vectors but different classes.

Question 2: Naive Bayes and the EM algorithm (10 points)

In this question, we will develop EM algorithm for Naive Bayes with hidden variables. Consider the Naive Bayes model given below. X_1 and X_2 are always observed and H is a hidden variable. Y is the class variable.



Assume that all variables are binary and all parameters are initialized as follows:

Parameter	$P(Y)$	$P(X_1\|Y)$	$P(X_1\|\overline{Y})$	$P(X_2\|Y)$	$P(X_2\|\overline{Y})$	$P(H\|Y)$	$P(H\|\overline{Y})$
Initial Value	θ	α_1	α_0	β_1	β_0	γ_1	γ_0

We are given a dataset having 1000 examples and is such that all tuples in the truth-table over X_1, X_2 and Y occur exactly 125 times in it. Thus, for example $\#(X_1, X_2, Y) = 125$; $\#(\overline{X_1}, X_2, \overline{Y}) = 125$; etc.

(a) (7 points) After running the EM algorithm until convergence, what will be the values of the parameters θ, α_1, α_0, β_1, β_0, γ_1 and γ_0. Explain your answer. No credit without correct explanation. To figure out the parameters at convergence, try running EM for 1-2 iterations and the pattern will be clear to you.

Solution: All parameters will converge to 0.5. In the first iteration of EM, they will become 0.5 and they will not change in the second iteration.

(b) (3 points) Your Boss claims that those hidden variables in the Naive Bayes model, assuming that they are children of the class variable, are useless. Is your Boss correct? Explain your answer.

Solution: Your Boss is right. H does not affect the class of the example because

$$\begin{aligned} P(Y, X_1, X_2) &= P(Y, X_1, X_2, H = 0) + P(Y, X_1, X_2, H = 1) \\ &= P(Y)P(X_2|Y)P(X_3|Y)P(H = 0|Y) + P(Y)P(X_2|Y)P(X_3|Y)P(H = 1|Y) \\ &= P(Y)P(X_2|Y)P(X_3|Y)[P(H = 0|Y) + P(H = 1|Y)] \\ &= P(Y)P(X_2|Y)P(X_3|Y) \times 1 \\ &= P(Y)P(X_2|Y)P(X_3|Y) \end{aligned}$$

Question 3: Linear Methods (10 points)

(a) (2 points) Does a 2-class Gaussian Naive Bayes classifier with parameters $\mu_{1,k}$, $\sigma_{1,k}$, $\mu_{2,k}$ and $\sigma_{2,k}$ for attributes $k = 1, \ldots, m$ have exactly the same representational power as logistic regression, given no assumptions about the variance values $\sigma^2_{1,k}$ and $\sigma^2_{2,k}$? True or False. Explain your answer (no credit if the explanation is incorrect).

Solution: No. Gaussian Naive Bayes classifier has more expressive power than a linear classifier if we dont restrict the variance parameters to be class-independent. In other words, Gaussian Naive Bayes classifier is a linear classifier if we assume $\sigma_{1,k} = \sigma_{2,k}$.

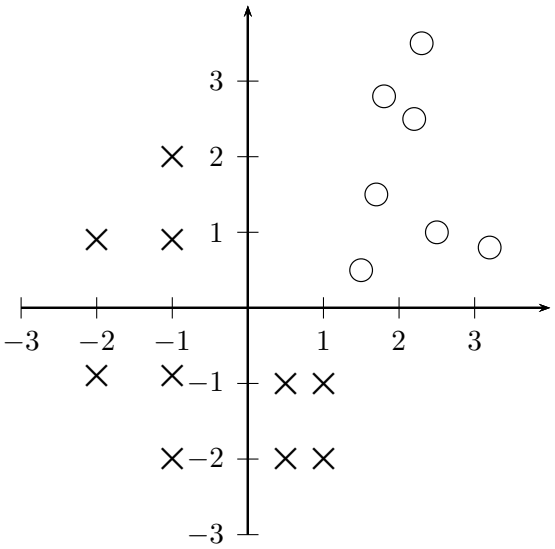
(b) (8 points) Consider the 2-D dataset given below. Suppose we are going to use logistic regression to train a classifier on this dataset and let us further assume that we are going to apply regularization to all the weights. Formally, we will train a classifier that maximizes the following expression:

$$\max \sum_{i=1}^{n} \log(P(y^{(i)}|x_1^{(i)}, x_2^{(i)})) - (\lambda_0 w_0^2 + \lambda_1 w_1^2 + \lambda_2 w_2^2)$$

where i indexes the i-th datapoint given by the three tuple $(x_1^{(i)}, x_2^{(i)}, y^{(i)})$. x_1 and x_2 are the features and y is the class variable. Recall that $P(y|x_1, x_2)$ is given by:

$$P(y|x_1, x_2) = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + w_2 x_2)}}$$

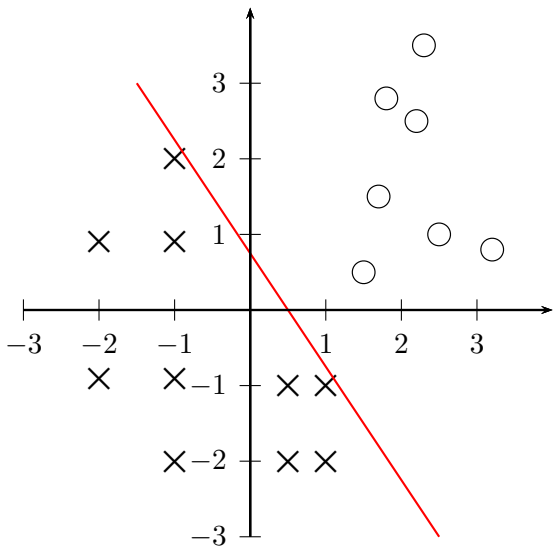
Dataset:



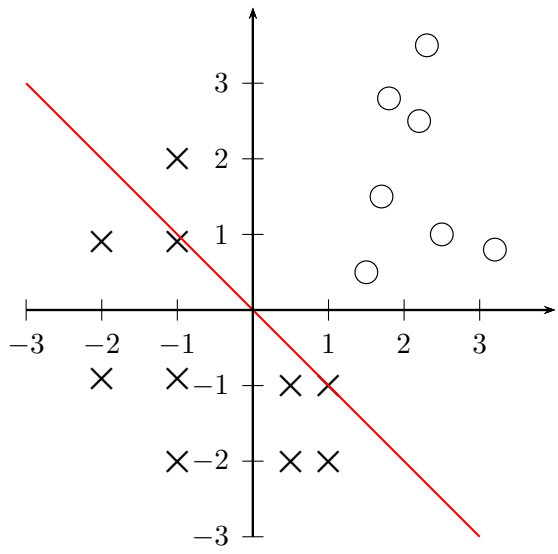
(Question continues to the next page.)

Draw the decision boundary for each of the following cases:

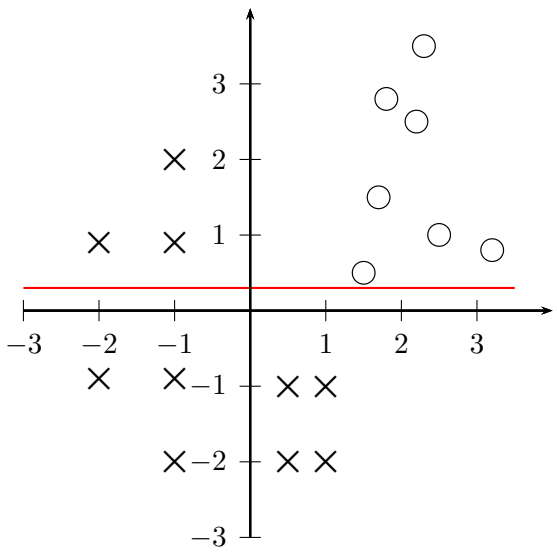
Case 1: $\lambda_0 = 0$, $\lambda_1 = 0$ and $\lambda_2 = 0$



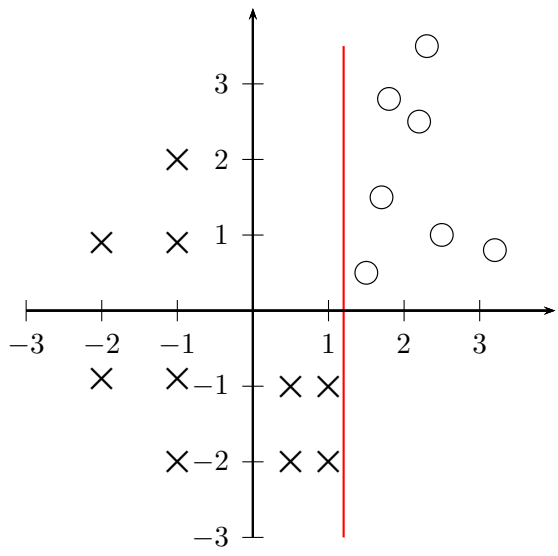
Case 2: $\lambda_0 = \infty$, $\lambda_1 = 0$ and $\lambda_2 = 0$



Case 3: $\lambda_0 = 0$, $\lambda_1 = \infty$ and $\lambda_2 = 0$

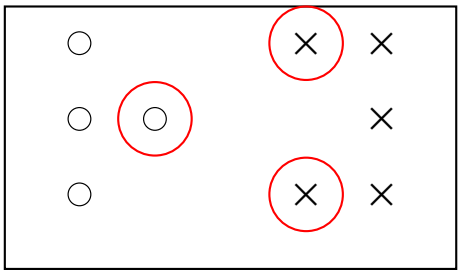


Case 4: $\lambda_0 = 0$, $\lambda_1 = 0$ and $\lambda_2 = \infty$



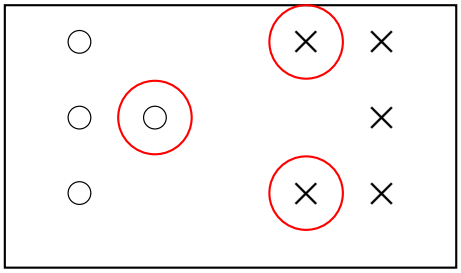
Question 4: SVMs and Kernels (10 points)

(a) (3 points) In the following image, circle the points such that removing that example from the training set and retraining linear SVM, we would get a different decision boundary than training on the full sample. Also, give a one or two sentence explanation for why you circled the points.



Explanation: Circled points are support vectors.

(b) (3 points) Suppose you used a perceptron instead of linear SVMs. Circle the points such that removing that example from the training set and running a perceptron, we **may** get a different decision boundary than training with perceptron on the full sample. Note the word "may" and therefore you should take the worst case view. Also, give a one or two sentence explanation for why you circled the points.



Explanation: If we remove an uncircled point, the decision boundary does not change enough to misclassify the point.

(c) (4 points) You are trying to use SVMs to build a classifier for a dataset that your boss gave you. In the dataset, there are only a few positive training examples and you are certain that they are always classified correctly by your SVM classifier. The dataset also has a large number of negative training examples and your boss tells you that it is OK if we misclassify some of them. Modify the basic dual form of the SVM optimization problem given below:

$$\text{maximize} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

$$\text{subject to} \quad \alpha_i \geq 0, \text{ and } \sum_{i=1}^{n} \alpha_i y_i = 0$$

to better solve this type of problem. Namely, you would like to ensure that you won't misclassify any of the positive examples but could misclassify some of the negative examples. Introduce additional parameter(s) (or constants for the purpose of solving the quadratic programming problem) in order to achieve this.

In your solution, please use $I+$ to index positively labeled examples ($y_i = +1$) and $I-$ for negative examples ($y_i = -1$). In other words, $i \in I+$ means that $y_i = +1$ and $i \in I-$ means that $y_i = -1$.

Write down your solution using the dual formulation.

Solution: Add the constraint that

$$0 \leq \alpha_i \leq C, \text{ for all points } i \in I-$$

Question 5: Ensemble Methods (10 points)

For your convenience, AdaBoost is given on the last page.

(a) (1 point) Circle the method or methods that are slower in terms of training time than the decision tree classifier. More than one answer is possible. No explanation needed.

BAGGING BOOSTING

(b) (1 point) Circle the method or methods that can be parallelized easily (after all we live in the age of multi-core machines). More than one answer is possible. No explanation needed.

BAGGING BOOSTING

(c) (1 point) Circle the method or methods that attempt to reduce the variance. More than one answer is possible. No explanation needed.

BAGGING BOOSTING

(d) (1 point) Circle the method or methods that hurt the performance of stable classifiers. To recap, a stable classifier is a classifier whose decision boundary does not change much as we add new data points. More than one answer is possible. No explanation needed.

BAGGING BOOSTING

(e) (3 points) Answer True or False. Explain your answer. A weak learner with less than 50% accuracy does not present any problem to the Adaboost algorithm.

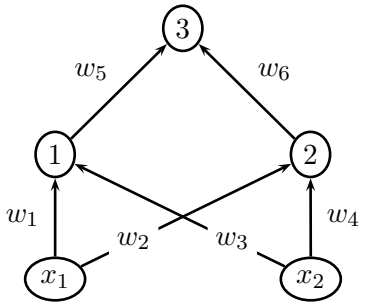
Solution: False. If the error $\epsilon > 0.5$, then the weight assigned to the mis-classified points will be smaller than the weight assigned to the points that are classified correctly. Thus, subsequent iterations will not try to classify these points correctly and thus the algorithm is likely to exhibit poor performance.

(f) (3 points) True or False. AdaBoost is not susceptible to outliers. If your answer is true, explain why. If your answer is false, then describe a simple heuristic to fix Adaboost so that it is not susceptible to outliers.

Solution: False. Adaboost is susceptible to outliers. A possible heuristic is to put a threshold on the weight and remove all points that have very large weights (typically outliers may have large weights because they will be consistently mis-classified).

Question 6: Neural Networks (10 points)

Consider the Neural network given below.



Assume that all internal nodes and output nodes compute the $tanh$ **function**. In this question, we will derive an explicit expression that shows how back propagation (applied to minimize the least squares error function) changes the values of w_1, w_2, w_3, w_4, w_5 and w_6 when the algorithm is given the example (x_1, x_2, y) with y being class variable (there are no bias terms). Assume that the learning rate is η. Let o_1 and o_2 be the output of the hidden units 1 and 2 respectively. Let o_3 be the output of the output unit 3.

Hint: Derivative of $tanh(x) = 1 - tanh^2(x)$.

(a) (3 points) Forward propagation. Write equations for o_1, o_2 and o_3.

Solution:
$o_1 = tanh(w_1 x_1 + w_3 x_2)$
$o_2 = tanh(w_2 x_1 + w_4 x_2)$
$o_3 = tanh(w_5 o_1 + w_6 o_2)$

(b) (3 points) Backward propagation. Write equations for δ_1, δ_2 and δ_3 where δ_1, δ_2 and δ_3 are the values propagated backwards by the units denoted by 1, 2 and 3 respectively in the neural network.

Solution:
$\delta_3 = (1 - o_3^2)(y - o_3)$
$\delta_1 = (1 - o_1^2)\delta_3 w_5$
$\delta_2 = (1 - o_2^2)\delta_3 w_6$

(c) (4 points) Give an explicit expression for the new (updated) weights w_1, w_2, w_3, w_4, w_5 and w_6 after backward propagation.

Solution:

$w_1 = w_1 + \eta\delta_1 x_1$
$w_2 = w_2 + \eta\delta_2 x_1$
$w_3 = w_3 + \eta\delta_1 x_2$
$w_4 = w_4 + \eta\delta_2 x_2$
$w_5 = w_5 + \eta\delta_3 o_1$
$w_6 = w_6 + \eta\delta_3 o_2$

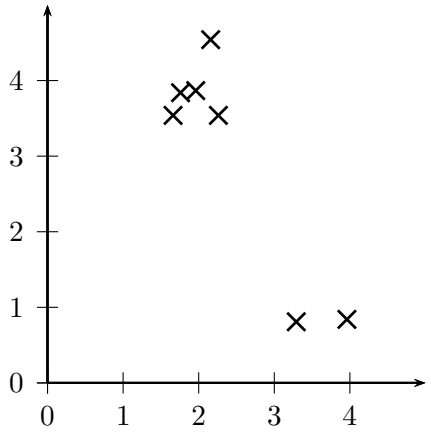
Question 7: Nearest Neighbors and K-means (10 points)

(a) (3 points) As we increase k, the k nearest neighbor algorithm begins to overfit the training dataset. True or False. Explain your answer.

Solution: False. In fact, as we increase k, the algorithm underfits because the decision surface becomes simpler. (for example, when $k = \#points$, the induced function is just the majority class function).

(b) (3 points) What is the time complexity of the K-means algorithm. Use the O-notation. Explain your notation clearly.

Solution: As we saw in class, the time complexity is $O(nkdi)$, where n is the number of data points, d is the number of features, k is the number of clusters and i is the number of iterations needed until convergence.



(c) (4 points) Draw the agglomerative clustering tree for the dataset given above. You must use single link clustering.

Solution: Skipped. Several judgment calls.

Question 8: Learning Theory (10 points)

(a) (5 points) Prove that the VC dimension of the concept class of all axis parallel rectangles is greater than or equal to 4.

Solution: See class slides.

(b) (5 points) Consider the class of concepts that can be expressed as an axis parallel rectangle. Assume further that the rectangles can be defined in terms of 4 integers: a, b, c, and d that lie between 1 and 100 (both included). Namely, $a, b, c, d \in \{1, \ldots, 100\}$. We assume that a consistent algorithm is available.

i. What is the size of the hypothesis space.

Solution: The hypothesis space is 100^4

ii. How many randomly obtained examples are needed for PAC learning with accuracy parameter $\epsilon = 0.1$ and confidence parameter $\delta = 0.05$?

Solution: Since a consistent learner is available, we use the formula:

$$m \geq \frac{1}{\epsilon}\left(\ln(1/\delta) + \ln(|H|)\right)$$

$$m \geq \frac{1}{0.1}\left(\ln(1/0.05) + \ln(100^4)\right)$$

$$m \geq 214.16$$

iii. What is the accuracy level (what is ϵ) if it is known that 1000 randomly chosen examples were used, and the desired confidence level is $\delta = 0.05$?

Solution: Rearraging the formula given above,

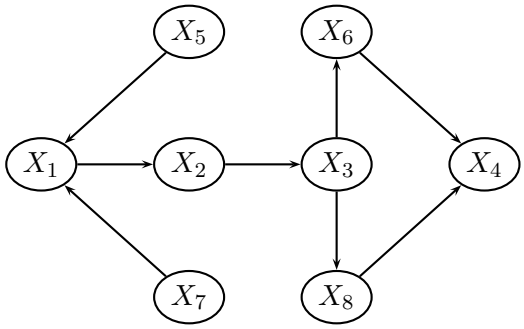
$$\epsilon \geq \frac{1}{m}\left(\ln(1/\delta) + \ln(|H|)\right)$$

$$\epsilon \geq \frac{1}{1000}\left(\ln(1/0.05) + \ln(100^4)\right)$$

$$\epsilon \geq 0.0214$$

Question 9: Bayesian networks (20 points)

Consider the Bayesian network given below:



(a) (7 points) Let X_4 be evidence. Consider the elimination order $(X_1, X_2, X_3, X_5, X_6, X_7, X_8)$. Write down the factors that you will encounter during this elimination for computing $P(X_4 = x_4)$ (namely, trace the variable elimination algorithm). What is the time and space complexity along this ordering.

Solution:

Var	Function created	Complexity
X_1	$\phi(X_7, X_5, X_2) = \sum_{X_1} P(X_1\|X_7, X_5)P(X_2\|X_1)$	exp(4)
X_2	$\phi(X_5, X_7, X_3) = \sum_{X_2} \phi(X_7, X_5, X_2)P(X_3\|X_2)$	exp(4)
X_3	$\phi(X_5, X_7, X_6, X_8) = \sum_{X_3} \phi(X_5, X_7, X_3)P(X_6\|X_3)P(X_8\|X_3)$	exp(5)
X_5	$\phi(X_7, X_6, X_8) = \sum_{X_5} \phi(X_5, X_7, X_6, X_8)P(X_5)$	exp(4)
X_6	$\phi(X_7, X_8) = \sum_{X_6} \phi(X_7, X_6, X_8)P(X_4 = x_4\|X_6, X_8)$	exp(3)
X_7	$\phi(X_8) = \sum_{X_7} \phi(X_7, X_8)P(X_7)$	exp(2)
X_8	$P(X_4 = x_4) = \sum_{X_8} \phi(X_8)$	exp(1)

Time Complexity: sum of the last column = $O(7\exp(5))$

Space Complexity: size of the largest function created = $O(7\exp(4))$.

(b) (7 points) Let X_4 be evidence for the Bayesian network given on the previous page. Consider the elimination order $(X_5, X_7, X_1, X_2, X_3, X_6, X_8)$. Write down the factors that you will encounter during this elimination for computing $P(X_4 = x_4)$. What is the time and space complexity along this ordering.

Solution:

Var	Function created	Complexity
X_5	$\phi(X_7, X_1) = \sum_{X_5} P(X_1\|X_7, X_5)P(X_5)$	exp(3)
X_7	$\phi(X_1) = \sum_{X_7} \phi(X_7, X_1)P(X_7)$	exp(3)
X_1	$\phi(X_2) = \sum_{X_1} \phi(X_1)P(X_2\|X_1)$	exp(2)
X_2	$\phi(X_3) = \sum_{X_2} \phi(X_2)P(X_3\|X_2)$	exp(2)
X_3	$\phi(X_6, X_8) = \sum_{X_3} \phi(X_3)P(X_6\|X_3)P(X_8\|X_3)$	exp(3)
X_6	$\phi(X_8) = \sum_{X_6} \phi(X_6, X_8)P(X_4 = x_4\|X_6, X_8)$	exp(2)
X_8	$P(X_4 = x_4) = \sum_{X_8} \phi(X_8)$	exp(1)

Time Complexity: sum of the last column = $O(7\exp(3))$
Space Complexity: size of the largest function created = $O(7\exp(2))$.

(c) (6 points) Suppose we want to learn a Bayesian network over two binary variables X_1 and X_2. You have n training examples $\{(x_1^{(1)}, x_2^{(1)}), \ldots, (x_1^{(n)}, x_2^{(n)})\}$. Let $BN1$ denote the Bayesian network having no edges and $BN2$ denote the Bayesian network having an edge from X_1 to X_2. Suppose you are learning both networks using the maximum likelihood estimation approach.

i. Describe a situation in which you will prefer $BN1$ over $BN2$ for modeling this data?

Solution: X_1 and X_2 are independent.

ii. Describe a situation in which you will prefer $BN2$ over $BN1$ for modeling this data?

Solution: X_1 and X_2 are not independent.

Entropy

```
Entropy(0,0) = 0.0
Entropy(0,1) = 0.0
Entropy(0,2) = 0.0
Entropy(0,3) = 0.0
Entropy(0,4) = 0.0
Entropy(1,1) = 1.0
Entropy(1,2) = 0.918295834054
Entropy(1,3) = 0.811278124459
Entropy(1,4) = 0.721928094887
Entropy(2,2) = 1.0
Entropy(2,3) = 0.970950594455
Entropy(2,4) = 0.918295834054
Entropy(3,3) = 1.0
Entropy(3,4) = 0.985228136034
Entropy(4,4) = 1.0
```

Useful formulas for Learning Theory

$$m \geq \frac{1}{\epsilon}\left(\ln(1/\delta) + \ln(|H|)\right)$$

$$m \geq \frac{1}{2\epsilon^2}\left(\ln(1/\delta) + \ln(|H|)\right)$$

$$m \geq \frac{1}{\epsilon}\left(4\log_2(2/\delta) + 8VC(H)\log_2(13/\epsilon)\right)$$

AdaBoost

1. Initialize the data weighting coefficients $\{w_n\}$ by setting $w_n^{(1)} = 1/N$ for $n = 1, \ldots, N$.
2. For $m = 1, \ldots, M$:
   (a) Fit a classifier $y_m(\mathbf{x})$ to the training data by minimizing the weighted error function

$$J_m = \sum_{n=1}^{N} w_n^{(m)} I(y_m(\mathbf{x}_n) \neq t_n) \tag{14.15}$$

   where $I(y_m(\mathbf{x}_n) \neq t_n)$ is the indicator function and equals 1 when $y_m(\mathbf{x}_n) \neq t_n$ and 0 otherwise.
   (b) Evaluate the quantities

$$\epsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} I(y_m(\mathbf{x}_n) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}} \tag{14.16}$$

   and then use these to evaluate

$$\alpha_m = \ln \left\{ \frac{1 - \epsilon_m}{\epsilon_m} \right\}. \tag{14.17}$$

   (c) Update the data weighting coefficients

$$w_n^{(m+1)} = w_n^{(m)} \exp \{\alpha_m I(y_m(\mathbf{x}_n) \neq t_n)\} \tag{14.18}$$

3. Make predictions using the final model, which is given by

$$Y_M(\mathbf{x}) = \text{sign} \left(\sum_{m=1}^{M} \alpha_m y_m(\mathbf{x}) \right). \tag{14.19}$$