CS 760
Machine Learning
Fall 2012 Exam

Name _____

Write your answers on these pages and show your work. You may use the back sides of pages as necessary. Before starting, make sure your exam has every page (numbered **1** through **10**).

Problem	**Score**	**Max Score**
1.	_____	14
2.	_____	14
3.	_____	24
4.	_____	22
5.	_____	10
6.	_____	16
Total		100

1. *k*-Nearest Neighbor (14 points): A weakness of the standard *k*-NN method is that the time required to classify a test instance increases as the size of the training set grows.

(a) Briefly explain why this is the case.

(b) Describe how you could speed up the classification time for *k*-NN by using ideas from decision-tree learning.

(c) Briefly describe one advantage of the approach you devised for part (b) relative to a standard decision-tree learner such as ID3.

2. Information gain (14 points): Consider the following training set with two Boolean features and one continuous feature.

	A	*B*	*C*	*Class*
Instance 1	F	T	115	neg
Instance 2	T	F	890	neg
Instance 3	T	T	257	pos
Instance 4	F	F	509	pos
Instance 5	T	T	733	pos

(a) How much information about the class is gained by knowing whether or not the value of feature *C* is less than 383?

(b) How much information about the class is gained by knowing whether or not features *A* and *B* have the same value?

3. Bayesian networks (24 points): Consider the following training set with two Boolean features, and one 3-valued feature, *C*, that has possible values {red, blue, green}.

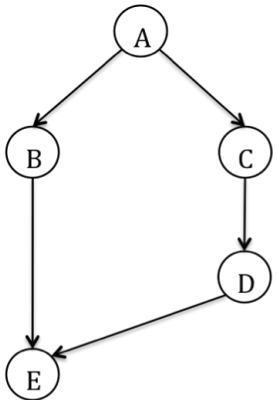
	A	*B*	*C*	*Class*
Instance 1	F	T	red	neg
Instance 2	T	F	blue	neg
Instance 3	T	T	red	pos
Instance 4	F	F	blue	pos
Instance 5	T	T	green	pos

(a) Draw the structure of a naïve Bayes network for this task.

(b) Using *m*-estimates with $m = 6$ and uniform priors for each feature, estimate the parameters in this naïve Bayes network.

(c) Consider the Bayes net that would result from reversing every edge in the naïve Bayes model above. How many parameters are in this model? Explain your answer.

(d) Assume that we are doing a hill-climbing structure search using the *add-edge*, *delete-edge* and *reverse-edge* operators. List or show all of the operator applications that we could consider for the current network shown below.



(e) Now assume that we are using the *sparse-candidate* algorithm for our structure search, and the candidate parents for each node are those listed below. List or show all of the operator applications that we could consider for the first change in the maximize step that starts with the network from part **(d)**.

node	candidate parents
A	{B, E}
B	{A, C}
C	{A, D}
D	{C, E}
E	{B, D}

4. Learning Theory (22 points):

(a) Consider the concept class that consists of disjunctions of <u>exactly</u> two literals, where each literal is a feature or its negation and there are *n* Boolean features. For example, one concept in this class is: $(x_1 \vee x_5)$. Another one is $(\neg x_1 \vee x_3)$. According to the PAC model, how many training examples do we need in order to be 90% confident that we will learn a Boolean target concept in this class to 80% accuracy when $n = 100$.

(b) What else would we need to show to prove that this concept class is PAC learnable?

(c) Now consider a similar class of concepts that consists of disjunctions of exactly two literals, <u>only one</u> of which can be <u>negated</u>. Suppose that the number of features $n = 3$. Show what the *Halving* algorithm would do with the following two training instances in an on-line setting.

x_1 x_2 x_3	y
⟨T, F, F⟩	pos
⟨F, T, T⟩	neg

(d) Suppose the learner can pick the next training instance it will be given. (The learner can pick the feature vector part of the instance; the class label will be provided by the teacher). Which instance should it ask for next?

(e) How many mistakes will the *Halving* algorithm make for this concept class in the worst case?

(f) List <u>two</u> key reasons why we couldn't use the *Halving* algorithm to learn a more realistic concept class such as decision trees.

5. Error Correcting Output Codes (10 points): Consider using an ECOC approach for a classification task with four classes.

(a) Which code (**A** or **B**) shown below would be better to use in an ECOC ensemble? Explain why.

	Code A				
class	f_1	f_2	f_3	f_4	f_5
1	0	1	0	1	1
2	1	0	1	0	1
3	0	1	1	0	0
4	1	0	0	1	0

	Code B				
class	f_1	f_2	f_3	f_4	f_5
1	1	1	1	0	0
2	1	0	0	1	0
3	1	1	0	0	0
4	0	0	0	1	1

(b) Briefly describe how you would use SVMs in concert with Code A. Specifically address how many SVMs would you train and how would you form the training set for each?

(c) If you weren't using an ECOC, could you use just one SVM for this task? Why or why not?

6. Short Answer (16 points): Briefly define each of the following terms.

overfitting

inductive bias

confusion matrix

recurrent neural networks

values that are missing systematically

VC dimension

large margin classification

kernel functions