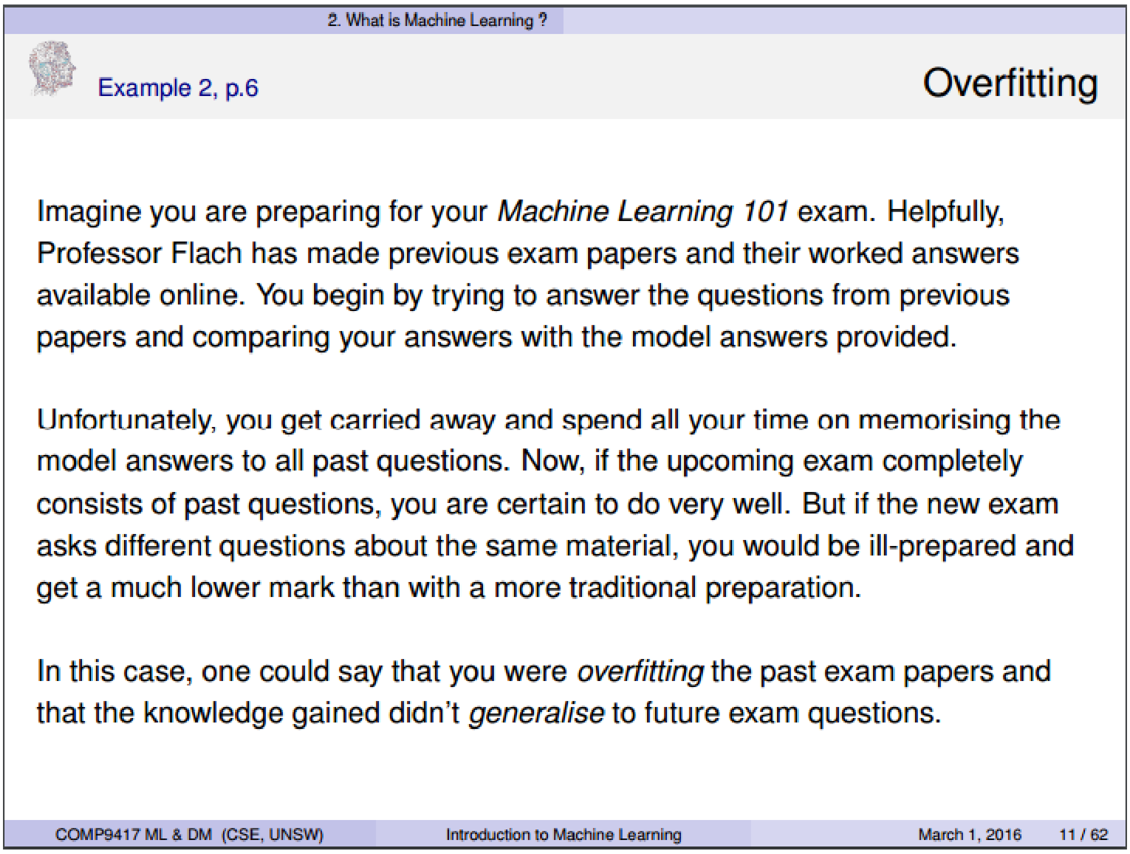
9417 Exam questions



2. What is Machine Learning ?

Example 2, p.6

Overfitting

Imagine you are preparing for your *Machine Learning 101* exam. Helpfully, Professor Flach has made previous exam papers and their worked answers available online. You begin by trying to answer the questions from previous papers and comparing your answers with the model answers provided.

Unfortunately, you get carried away and spend all your time on memorising the model answers to all past questions. Now, if the upcoming exam completely consists of past questions, you are certain to do very well. But if the new exam asks different questions about the same material, you would be ill-prepared and get a much lower mark than with a more traditional preparation.

In this case, one could say that you were *overfitting* the past exam papers and that the knowledge gained didn't *generalise* to future exam questions.

COMP9417 ML & DM (CSE, UNSW) Introduction to Machine Learning March 1, 2016 11 / 62

9417 Exam questions

Sample paper

Question 1 [20 marks] Association Rule Learning[a][b] & and ILP[c]

- **A) [7 marks] An association rule has the form A → C where A is the antecedent and C is the consequent. Suppose you have a database of one million transactions.**

1) If an association rule has 90% support, how many transactions contain all the items in A ?

Support(A) $\geq$ Support(A$\wedge$C):

$\geq$900 000 transactions[d][e]

2) For the same rule, how many transactions contain all the items in C ?

Support(C) $\geq$ Support(A$\wedge$C):

$\geq$900 000 transactions

3) For the same rule, how many transactions contain all the items in both A and C?

Support(A$\wedge$C) $=$ Support(A$\wedge$C):

900 000 transactions

4) If another rule has 40% support and 100% confidence, how many transactions contain all the items in A ?

Confidence(A$\wedge$C) = Support(A$\wedge$C) $\div$ Support(A):

400 000 transactions

5) If yet another rule has 40% support and 50% confidence, how many transactions contain all the items in A ?

Confidence(A$\wedge$C) = Support(A$\wedge$C) $\div$ Support(A):

800 000 transactions

B) [3 marks] You are mining a data warehouse. In three sentences or less propose a method to derive all candidate item sets of size k + 1 from the k-item sets with at least minimum support.

Use an Apriori algorithm:

- Generate all k-item sets, store those which have at least minimum support

9417 Exam questions

Learning in Logic- 2016

C) [4 marks] Consider the following two clauses: C = Q(A, x, B) ∨ S(y, B) and C1 = S(w,B) ∨ ¬R(z) Using inverse resolution, provide at least one solution for C2. [Show all substitutions].

C1 = S(w,B) V ¬R(z)
C = Q(A,x,B) V S(y,B)

L = ~R(z)
C2 = (C - (C1-{L})) V {¬L}
= Q(A,x,B) V R(z)
(not sure if correct)
second one is = Q(A,x,B) V R(z) V S (y,B) for this to be correct, you need to mention the unification y/w I think!
[f]

$C_2=(C-(C_1-\{L_1\})\theta_1)\theta_2^{-1}\cup\{\neg L_1\theta_1\theta_2^{-1}\}$

We will choose $L_1=\neg R(z)$[g][h][i][j]
$\theta_1=\{w/y\}$ i.e. substitute all w variables with y
$\theta_2^{-1}=$
$C_2=(C-\{S(y,B)\})\theta_2^{-1}\cup\{R(z)\}=Q(A,x,B)\vee R(z)$

Boris:

We will choose $L_1=S(w,B)$[k][l][m][n] to replace on the previous result.
$Q(A,x,B)\vee R(z)\vee$[?][?]$S(w,B)$

D) [6 marks] Construct the Relative Least General Generalisation (RLGG) of two observations: likes(alan,sushi) and likes(alan,curry), given the background predicates food(sushi) and food(curry). Now suppose you are given two more observations: likes(bettina,sushi) and likes(bettina,curry). Will the RLGG of the four observations, given the same background predicates, change ? If you think the answer is yes, give the new RLGG, otherwise give an argument why it will not have changed. [Show all working].

I would guess the final answer would be something on the realm of likes(X,Y) and food(Y), but not sure.
^^ someone elses answer
==
1) (Given likes(alan,sushi), likes(alan,curry), food(sushi) and food(curry))
likes(alan, X),
food(X)

2) (Adding likes(bettina,sushi) and likes(bettina,curry))
person(A),   //D: I don’t think you can just make your own predicates.
likes(A, B),
food(B)

likes(lgg(A,food(B)) //Can we do this?
Or just for fun:
food(sushi, curry) <- // maybe food(X) then?
  likes(Alan, X)
  likes(Bettina, X)

(For 1) I think it’s alan because there’s only one value possible for the 1st parameter so it’s kinda like likes(lgg(alan), lgg(sushi, curry)) => likes (alan, X))
not sure though just correct me if I’m wrong. Also is the question missing a “person” predicate/are we allowed to just create one in 2) ?)
~p35 http://webapps.cse.unsw.edu.au/webcms2//course/showfile.php?cid=2405&color=orange&addr=Notes/l12_ILP_1up.pdf

Boris: another solution…
Food {<sushi>, <curry>}
Likes {<alan, sushi>, <alan, curry>}

likes(lgg(alan), lgg(sushi, curry)) ← food(lgg(sushi, curry))

likes(A, B) ← food(B)

If we add Bettina’s likes there’s no change…
[o]

likes(lgg(alan, bettina), lgg(sushi, curry)) ← food(sushi, curry)   #SAME
likes(A, B) ← food(B)

Another Answer?… Please if you find something, please comment.

E) Explain how the *generality* order on hypotheses can be expressed for hypotheses that are atoms in first-order logic. Suggest refinement operator for such atoms that could be used to search the hypothesis space ?

The most popular framework for generality within inductive logic programming is θ-subsumption. It provides a generalization relation for clausal logic and it extends propositional subsumption to first order logic

These systems typically employ a specialization or refinement operator to traverse the search space. To guarantee the systematic enumeration of the search space, the specialization op-erator ρ_s can be employed.
$\rho_s(c)$ is obtained by applying the adding condition or substitution rule with the following restrictions.

- the adding condition rule only adds atoms of the form $p(V_1, \cdots, V_n)$ where the V_i are variables not yet occurring in the clause c;

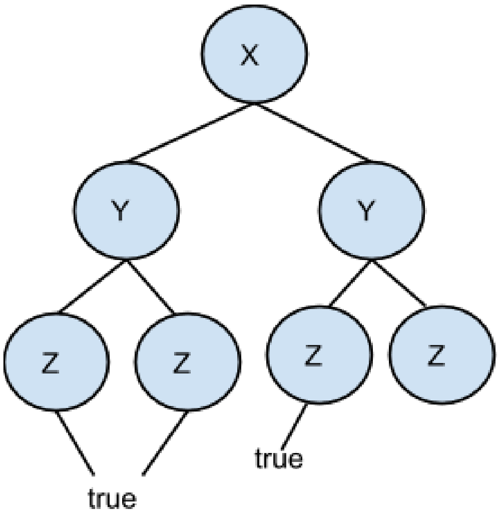
- the substitution rule only employs *elementary substitutions*, which are of the form.

  - ❖ {V/Y }, where X and Y are two variables appearing in c
  - ❖ {V /ct}, where V is a variable in c and ct a constant
  - ❖ $\{V/f(V_1,\cdots,V_n)\}$, where V is a variable in c, f a functor of arity n and the V_i are variables not yet occurring in c.

Question 2 [20 marks] Comparing Lazy and Eager Learning
The following truth table gives an "m–of–n function" for three Boolean variables, where "1" denotes true and "0" denotes false. In this case the target function is: "exactly two out of three variables are true".

X	Y	Z	Class
0	0	0	false
0	0	1	false
0	1	0	false
0	1	1	true
1	0	0	false
1	0	1	true
1	1	0	true
1	1	1	false

A) [4 marks] Construct a decision tree which is complete and correct for the examples in the table. [Hint: draw a diagram.]



Right implies 0, left 1 unlabeled leafs are obviously "false"

B) [4 marks] Construct a set of ordered classification rules which is complete and correct for the examples in the table. [Hint: use an if–then–else representation.]

If X=1 ∧ Y=1 ∧ Z=1 then false
else if X=1 ∧ Y=1 then true

]

```
R
(If x = 1 ∧ y = 1 ∧z = 0 then true
else if x = 1 ∧ y = 0 ∧ z = 1 then true
else if x = 0 ∧ y = 1 ∧ z = 1 then true
else false)            <- is this correct?
```

From my understanding, **Complete and correct** has all positive examples and None of Negative examples. Therefore, we must consider only the cases where True is the outcome. False should not be present.

```
IF X=1
    IF Y=1 THEN
            IF Z=0 THEN
                True[p][q]
    ELSE IF Z=1 THEN
        True
ELSE IF Y=1 AND Z=1 THEN
    True
```

C) [10 marks] Suppose we define a simple measure of distance between two equal length strings of Boolean values, as follows. The distance between two such strings B1 and B2 is:

distance(B1, B2) = |(ΣB1) − (Σ B2)|

where ΣBi is simply the number of variables with value 1 in string Bi . For example: distance(<0, 0, 0>,<1, 1, 1>) = |0 − 3| = 3

and

distance(<1, 0, 0>,<0, 1, 0>) = |1 − 1| = 0

What is the LOOCV (“Leave–one–out cross-validation”) error of 2–Nearest Neighbour using our distance function on the examples in the table ? [Show your working.]

My basic understanding of LOOCV when applied to K Nearest Neighbours (KNN) is that you test each training instance using KNN (in this case k =2) using their distance measurement with all the other instances in the dataset except the instance being tested and give it a value of 0 if is correctly classified and 1 if incorrectly classified. You repeat this for every instance in the dataset and then sum up all the 1s (i.e. the misclassified ones) and divide it by n (the dataset size) to get the mean error.

Therefore, using the distance measurement that they give, all examples are correctly classified except for the last one:

1. False (Correct -> 0 error)
2. False (Correct -> 0 error)
3. False (Correct -> 0 error)
4. True (Correct -> 0 error)
5. False (Correct -> 0 error)
6. True (Correct -> 0 error)
7. True (Correct -> 0 error)
8. True (Incorrect -> 1 error)

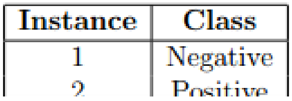
LOOCV error = mean error = ⅛ = 12.5%

D) [2 marks] Compare your three models. Which do you conclude provides a better representation for this particular problem ? Give your reasoning (one sentence).

Tree, 100% accurate and more comprehensible.

Question 3 [20 Marks] Mistake Bounds

Consider the following learning problem on an instance space which has only one feature, i.e., each instance is a single integer. Suppose instances are always in the range [1, 5]. The hypothesis space is one in which each hypothesis is an interval over the integers. More precisely, each hypothesis h in the hypothesis space H is an interv al of the form a ≤ x ≤ b, where a and b are integer constants and x refers to the instance. For example, the hypothesis 3 ≤ x ≤ 5 classifies the integers 3, 4 and 5 as positive and all others as negative.



Instance	Class
1	Negative
2	Positive

A) [15 marks] Apply the Halving Algorithm to the five examples in the order in which they appear in the table above. Show each class prediction and whether or not it is a mistake, plus the initial G and S sets and those at the end of each iteration.

I have been trying to follow the answers here: http://www.danielgoldbach.com/files/9417notes.html#consistent-and-agnostic-learners can anyone confirm if they get the same answer as Daniel here?
I have:
Initially: G = {[1,5]} S = {}
1: (-) : 1 vote + , 1 vote -, prediction: + so INCORRECT
Prune G to exclude this case
As it is negative and S is empty leave S
G = {[2,5]} S = []

2: (+) 1 vote +, 1 vote -, prediction + so CORRECT
Leave G as it is consistent
Add to S all hypotheses h consistent with the instance
G = {[2,5]} S = {[2,2] [2,3] [2,4] [2,5]}[r][s][t][u][v][w][x][y][z]

3: (+) 4 votes +, 1 vote -, prediction: + so CORRECT
Leave G as it is consistent
Prune S to remove and inconsistent examples
G = {[2,5]} S = {[2,3] [2,4] [2,5]}

4: (+) 3 votes +, 1 vote -, prediction + so CORRECT
Leave G as it is consistent
Prune S to remove and inconsistent examples
G = {[2,5]} S = {[2,4] [2,5]}

5: (-) 2 votes +, 1 vote -, prediction + so INCORRECT
Prune G as it is inconsistent
Remove any h from S that is inconsistent
Here is where Daniel and I differ:
My answer:
G = {[2,4]} S = {[2,4]}
Daniel's Answer:
G = {[2,5]} S = {[2,5]}

L: Which is right? Can someone explain why it would be [2,5] for both instead of [2,4]?
D: I think it should be [2,4]
L: Awesome, thanks! :)
U: I agree with [2,4] as well.

U: However I have a concern regarding the methodology. If |H|=16, how do you get that? and in halving algorithm, aren't we supposed to list all the hypothesis first and then eliminate the ones that are wrong? (That's the impression I got from reading the book)?

D:|H| = all possible combinations including the empty set. ([] ,[1,1-5], [2,2-5], [3,3-5], [4,4-5],[5,5])
1+5+4+3+2+1 = 16

C: In step 2: S = {[2,2] [2,3] [2,4] [2,5]}[aa][ab][ac][ad][ae][af][ag][ah][ai], but aren't [2,3], [2,4] and [2,5] more general then [2,2]?

Djole's answer:

Initial: G = P(|H|) S={}
Check for 1: Vote: 5 positive, 11 negative
Vote result NO; Correct
1 Removed from concept
G = {[], [2,2], [2,3], [2,4], [2,5], [3,3], [3,4], [3,5], [4,4], [4,5], [5,5]| S={}
Check for 2: Vote 4 Yes, 7 No
Vote result NO; WRONG, remove everything that voted no
G=[2,2],[2,3],[2,4],[2,5] S = {[2,2],[2,3],[2,4]}
Check for 3: vote 3 Yes, 1 No

Vote result: Yes;Correct, Add all sets which agree
G=[2,3],[2,4],[2,5] S = {[2,3],[2,4]} <- shouldn't [2, 5] be removed as it is more general than [2,4] (according to the candidate elimination algorithm?)
Check for 4: vote 2 Yes, 2 No

Check for 5: vote 1 Yes, 1 no
Vote result: NO; Correct, remove 5 from concept
G=[2,4] S=[2,4]
.
BRANCH 2:
G=[2,2],[2,3],[2,4],[2,5] S = {[2,3],[2,4],[2,5]}
Check for 4: vote 2 Yes, 2 No
Vote Result   Yes
“No” removed from the S set.
G=[2,2],[2,3],[2,4],[2,5] S = {[2,4],[2,5]}
Check for 5: vote 1 Yes, 3 no;
Vote result: NO; Correct, 5 removed from concept.
G=[2,2],[2,3],[2,4] S = {[2,4]}

Please correct me if you think I’m wrong I’m 51% sure I’m correct. According to Bayes, that’s enough. (52% now)

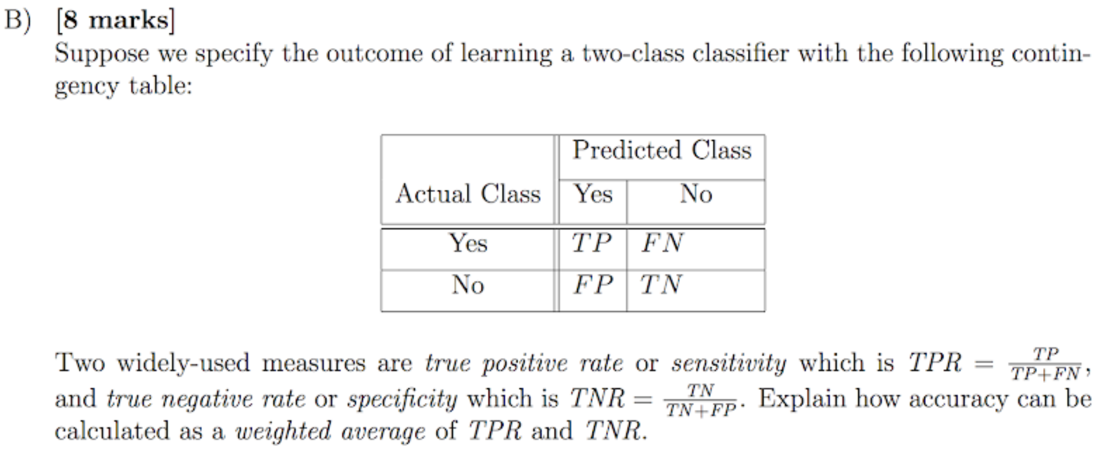
L: Djole I like yours more, I’m going with it - thanks for your help :)
LA: Me too, makes more sense to me
U: This is kinda what I’ve been doing as well :D
HD: don’t agree. Draw the diagram of hypothesis space, according to the definition on relationship of more general than or equal to, G should be initialized as {[1,5]}.

B) [5 marks] What is the worst-case mistake bound for the Halving Algorithm given the hypothesis space described above ? Give an informal derivation of your bound.
$\log_2|H|$ I think |H|=16 in this case! It does

Question 4 [20 marks]
Evaluation of Learning - 2016

A) [3 marks] The AUC (area under the ROC curve) measure originated in signal detection theory. For the evaluation of classifier learning on a two-class prediction problem, can you think of a probabilistic interpretation of this measure?

AUC is the probability that the classifier will rank a randomly-drawn positive example higher than a randomly-drawn negative one.
AUC will be 1 if all positive examples were ranked above all negative examples (compare with the ROC curve diagram)



B) **[8 marks]**
Suppose we specify the outcome of learning a two-class classifier with the following contingency table:

Actual Class	Predicted Class	
	Yes	No
Yes	TP	FN
No	FP	TN

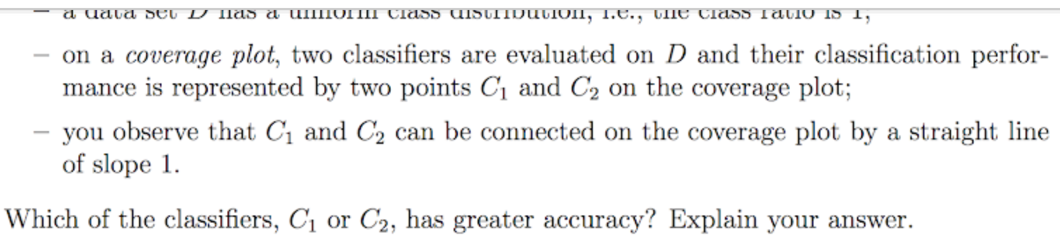
Two widely-used measures are *true positive rate* or *sensitivity* which is $TPR = \frac{TP}{TP+FN}$, and *true negative rate* or *specificity* which is $TNR = \frac{TN}{TN+FP}$. Explain how accuracy can be calculated as a *weighted average* of TPR and TNR.

$$\frac{TP}{TP+FN}w+\frac{TN}{TN+FP}(1-w)$$

Let $w=\frac{(TP+FN)}{N}$ **, then** $(1-w)=\frac{(TN+FP)}{N}$

So that $\frac{TP}{TP+FN}w+\frac{TN}{TN+FP}(1-w)=\frac{TP}{N}+\frac{TN}{N}=\frac{TP+TN}{N}=accuracy$

9417 Exam questions



– a data set D has a uniform class distribution, i.e., the class ratio is 1;

– on a *coverage plot*, two classifiers are evaluated on D and their classification performance is represented by two points C_1 and C_2 on the coverage plot;

– you observe that C_1 and C_2 can be connected on the coverage plot by a straight line of slope 1.

Which of the classifiers, C_1 or C_2, has greater accuracy? Explain your answer.

They have the same accuracy. Accuracy of C1 = Accuracy of C2.
Explain ?

Evaluation of Learning -2015

A) [3 marks] You have implemented a new two-class classifier learning algorithm that outputs a "score" to indicate how strongly it believes an input instance to be in the positive class. Explain how you could use a margin-based approach to penalise incorrect classifications.
Let the score given by the classifier be denoted by s^(x)
Let the correct classification be c(x) where + = 1, - = -1
Therefore if we take margin = c(x)s^(x) and examine the sign of the margin, it should give us an indication of whether it was correctly classified or not: + if correct, - if incorrect. We can then use a loss function to determine the amount to which the algorithm is penalised for an incorrect classification; the loss function maps a margin to a penalty weight.

Short answer: Multiply them together (Score * Classification)

C) [14 marks] Suppose the following decision tree has been learned on a training set of 10 positive and 10 negative examples for a "play sport" task.

Outlook = sunny:
    Temperature = warm: [play (5); don't play (2)]
    Temperature = cold: [play (4); don't play (3)]
  Outlook = rainy: [play (1); don't play (5)]

The three leaves of the tree each show the number of positive (play) and negative (don't play) examples in each leaf. Using a scoring function based on the ratio of positive to negative examples in each leaf node, generate the ranking of the training set produced by this scoring function. Then compute the ranking error for the tree on the training data.

Sunny, Warm, Play (5)	2.5
Sunny, Warm, Don't (2)	2.5
Sunny, Cold, Play (4)	1.3
Sunny, Cold, Don't (3)	1.3
Rainy, Play (1)	0.2
Rainy, Don't (5)	0.2

To calculate the AUC, for every positive example we count the number of negative examples ranked below it. If a negative example is equal to it, it gets a 1/2 score. We take the sum of all these and divide by the number of positives * number of negatives.
Therefore AUC =(5 * (0.5*2 +3+5) + 4*(0.5*3 + 5) + 1 * (0.5 * 5)) / (10 * 10)
= 73.5 / 100 = 0.735
Therefore the probability that a positive is ranked over a negative is 73.5%,
ie. the error is 100% - 73.5% = 26.5%.

Question 5 [20 Marks] Computational Learning Theory 2015-2016

A) [8 marks] An instance space X is defined using m Boolean attributes. Let the hypothesis space H be the set of decision trees defined on X (you can assume two classes). What is the largest set of instances in this setting which is shattered by H ? [Show your reasoning.]

Because the hypothesis space H spans all of X, we can say that all of X is shattered by H. The vapnik chervonenkis dimension is the size of the largest finite subset of X that is shattered by H. As all of X is shattered by H, VC(H) = =|X| 2^m (as there m boolean attributes there are **2^m** different possible instances)

B) [10 marks] Suppose we have a consistent learner with a hypothesis space restricted to conjunctions of exactly 8 attributes, each with values {true, false, don't care}. What is the size of this learner's hypothesis space ? Give the formula for the number of examples sufficient to learn with probability at least 95% an approximation of any hypothesis in this space with error of at most 10%. [Note: you are not required to

9417 Exam questions

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

We set:
epsilon = 0.1 (10% error)
H = 3^8
delta = 1 - 0.95 (100% - 95%) = 0.05

C) [2 marks] Informally, which of the following are consequences of the No Free Lunch theorem:

a) averaged over all possible training sets, the variance of a learning algorithm dominates its bias

b) averaged over all possible training sets, no learning algorithm has a better off-training set error than any other

c) averaged over all possible target concepts, the bias of a learning algorithm dominates its variance

d) averaged over all possible target concepts, no learning algorithm has a better off- training set error than any other

e) averaged over all possible target concepts and training sets, no learning algorithm is independent of the choice of representation in terms of its classification error

B and D ?
A and B (NFL Theorem considers training set only)

Question 6 [20 marks] There are 8 attributes with three values. Therefore there can be 3^8 separate instances in the learner's hypothesis space. (ie. the size = 3^8) plus the null hypothesis, so 3^8 + 1

I think the answer is this formula from the Algorithms and Independent Machine Learning
Ensemble Learning 2015-2016
A) [8 marks] As model complexity increases from low to high, what effect does this have on:
1) Bias ?
decrease
2) Variance ?
increase
3) Predictive accuracy on training data ?
Increase[aj][ak] - Gets better at predicting training data
4) Predictive accuracy on test data ?
increase to a certain point as complexity getting higher and higher(bias reducing), then decrease(variance dominated, overfitting)
B) [3 marks] Is decision tree learning relatively stable ? Describe decision tree learning in terms of bias and variance in no more than two sentences.

Like if u train the algo L on 2 training sets T1 and T2 taken from D, and the models for both sets are very similar or the same, then it has LOW variance (coz the same or similar) AND high bias, while if they are very dissimilar, then unstable so it has HIGH variance and low bias.

not stable, it is a high variance and low bias learning algorithm as the root node very dependent on the training data set and since u use diff sets even though from same D, the model produced will be different and hence unstable/high variance.
C) [3 marks] Is nearest neighbour relatively stable ? Describe nearest neighbour in terms of bias and variance in no more than two sentences.
Stable, high bias and low variance as when number of neighbours k increase bias increase, **BUT** 1nn perfectly separates training data so it has low bias with high variance (unstable).

D) [3 marks] Bagging reduces bias. True or false ? Give a one sentence explanation of your answer.
No. by voting/averaging among different models, bias actually increase (and decrease variance).

E) [3 marks] Boosting reduces variance. True or false ? Give a one sentence explanation of your answer.

True. Variance reduced by voting/averaging among different models that were trained over different training set, and thus reduce the overall expected error

Rohit : The boosting algorithm takes a weighted average of many weak models, and hence the final model has

9417 Exam questions

- In later iterations, it appears to be primarily a variance-reducing method

I am pretty sure boosting reduces bias not variance, as it will increase the weight for misclassified instance and decrease the weight for correct instance, therefore the bias decrease. (according to tut)

However boosting can reduce variance as well. If you look at the lecture notes, they explicitly suggest that bagging does nothing to bias. However it only says boosting reduces variance. I've read elsewhere that boosting may also reduce variance.

Question 7 [20 marks] Bayesian Learning 2015-2016

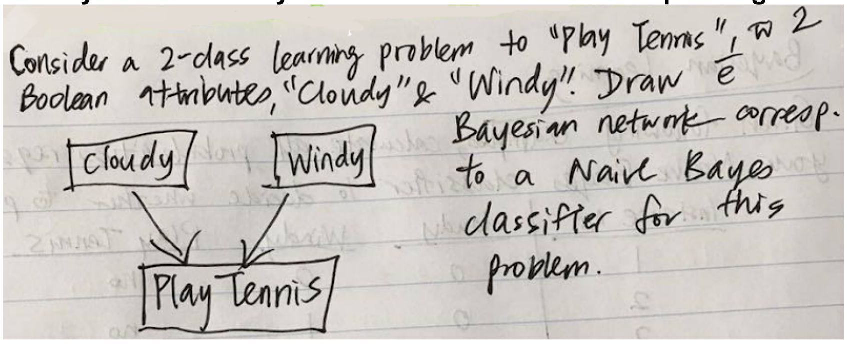
A) [4 marks]000000000000000000000 posteriori hypothesis HMAP and the maximum likelihood hypothesis HML.

H_{MAP}=argmax(P(D|h)P(h))

H_{ML}=argmax(P(D|h))

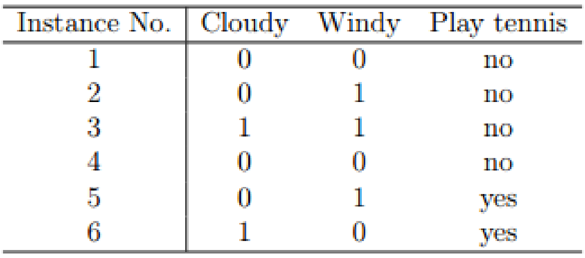
H_{ML} is deducted from H_{MAP} when all priors are considered uniform and hence doesn't make a difference in getting argmax. Which is P(h)=P(D), then both can be simplified.

B) [2 marks] Consider a two-class learning problem to "Play tennis", with two Boolean attributes, "Cloudy" and "Windy". Draw the Bayesian network[al][am][an] **corresponding to a Naive Bayes classifier for this problem.**



Consider a 2-class learning problem to "Play Tennis", w 2 ē Boolean attributes, "Cloudy" & "Windy". Draw ē Bayesian network corresp. to a Naive Bayes classifier for this problem.

C) [10 marks] Given the following examples, calculate all the probabilities required for your Naive Bayes classifier to be able to decide whether to play or not:



Instance No.	Cloudy	Windy	Play tennis
1	0	0	no
2	0	1	no
3	1	1	no
4	0	0	no
5	0	1	yes
6	1	0	yes

P(Play tennis) = 1/3
P(Cloudy) = 1/3
P(Windy) = 1/2
P(Cloudy | Play tennis) = 1/2

[ao][ap][aq]

P(Cloudy | ~Play tennis) = 1/4
P(Windy | Play tennis) = 1/2
P(Windy | ~Play tennis) = ½

D) [4 marks] To which class would your Naive Bayes classifier assign each of the following instances ?

Instance No.	Cloudy	Windy	Play tennis

9417 Exam questions

P(Play tennis | ~Cloudy, ~Windy) = P(~Cloudy | Play tennis) * P(~Windy | Play tennis) * P(Play tennis) / P(~Cloudy) * P(~Windy)
= (1/2 * 1/2 * 1/3) / (2/3 * 1/2) = (1/4) / P(evidence[ar][as])
P(~Play tennis | ~Cloudy, ~Windy) = P(~Cloudy | ~Play tennis) * P(~Windy | ~Play tennis) * P(~Play tennis) / P(~Cloudy) * P(~Windy)
= (3/4 * 1/2 * 2/3) / (2/3 * 1/2) = (3/4) / P(evidence)
3/4 > 1/2 -> Classify Play tennis as no.0

P(Play tennis | ~Cloudy, Windy) = P(~Cloudy | Play tennis) * P(Windy | Play tennis) * P(Play tennis) / P(~Cloudy) * P(Windy)
= (1/2 * 1/2 * 1/3) / (2/3 * 1/2) = (1/4) / P(evidence)
P(~Play tennis | ~Cloudy, Windy) = P(~Cloudy | ~Play tennis) * P(Windy | ~Play tennis) * P(~Play tennis) / P(~Cloudy) * P(Windy)
= (3/4 * 1/2 * 2/3) / (2/3 * 1/2) = (3/4) / P(evidence)
3/4 > 1/2 -> Classify Play tennis as no.

My take on this question …:[at]

Naive Bayes classifier: $v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_i P(a_i|v_j)$

#7:
P(Play) * P(!Cloudy | Play) * P(!Windy | Play) = ⅓ * ½ * ½ = 1/12
P(!Play) * P(!Cloudy | !Play) * P(!Windy | !Play) = ⅔ * ¾ * ½ = 3/12
3/12 > 1/12 --> Play Tennis = No
#8:
P(Play) * P(!Cloudy | Play) * P(Windy | Play) = ⅓ * ½ * ½ = 1/12
P(!Play) * P(!Cloudy | !Play) * P(Windy | !Play) = ⅔ * ¾ * ½ = 3/12
3/12 > 1/12 → Play Tennis = No

Calculate likelihood of PT and !PT (Slide 23 - 24)
Instance 7
Cloud - FALSE
Windy - FALSE
Yes = P(Play) * P(!Cloudy | Play) * P(!Windy | Play)   = ⅓ * (½ * ½) =~ 0.083
No = P(!Play) * P(!Cloudy | !Play) * P(!Windy | !Play) = ⅔ * (¾ * ½) =~ 0.25
Normalise (Slide 25)
Yes = 0.083 / (.083 + 0.25) = 0.249
No = 0.25 / (.083 + 0.25) = 0.75
Max Arg = Noeh
Repeat for Instance 8
Cloud - FALSE
Windy - TRUE

Yes = P(Play) * P(!Cloudy | Play) * **P(Windy | Play)**   = ⅓ * (½ * ½) =~ 0.083
No = P(!Play) * P(!Cloudy | !Play) * **P(Windy | !Play)** = ⅔ * (¾ * ½) =~ 0.25
These should turn out to be the same prediction since,
Cloudy for instance 7 and 8 is identical
WIndy True/False has the same probability in the training dataset whether true or fals

We first calculate the **likelihood of the two cases** and then the **probability** of Playing…
Ref. to example in slide 24 of ProbLearn2.
(7)
PlayTennis|¬Cloudy, ¬Windy <Likelihood>
P(PlayTennis) * P(¬Cloudy|PlayTennis) * P(¬Windy|PlayTennis)
½*½*⅓ = 0.[au][av]0833
¬PlayTennis|¬Cloudy, ¬Windy <Likelihood>
P(PlayTennis) * P(¬Cloudy|¬PlayTennis)*P(¬Windy|¬PlayTennis)
¾*½*⅓ = 0.375
Now we compute the probability of Playing…
P(PlayTennis|¬Cloudy, ¬Windy) = 0.25/(0.25+0.375) = 0.4
P(¬PlayTennis|¬Cloudy, ¬Windy) = 0.375/(0.25+0.375) = 0.6 #WeDon'tLikeTennisMate

(8)
PlayTennis|¬Cloudy, Windy <Likelihood>
P(¬Cloudy|PlayTennis)*P(Windy|PlayTennis)
½*½ = 0.25
¬PlayTennis|¬Cloudy,Windy <Likelihood>

9417 Exam questions

Now we compute the probability of Playing…
P(PlayTennis|¬Cloudy, Windy) = 0.25/(0.25+0.375) = 0.4
P(¬PlayTennis|¬Cloudy, Windy) = 0.375/(0.25+0.375) = 0.6 #WeDon’tLikeTennisMate

If this is incorrect we can remove it ^

[a]Couldn't find association rules in 2016 sample paper. Does that mean its not included?

[b]It was tested in our mid sem... I think?

[c]what is ILP ? is it covered in the exam?

[d]maybe precisely 0.9m<=x<=1m

[e]same as question 1.2

[f]can someone explain how S(y.b) came about?

In addition, doesn't this now mean that C1 and C2 share a common literal?

[g]why S(w, B) is not considered? that clause have different variables, then that clause will be different right?

[h]I think another step is needed, like make again L1 = S(w, B).
Otherwise won't make sense that C doesn't have a S(w,B) and that new L1 is also different, right?

[i]according to the formulation C2 = (C-(C1-{not L})) union { L}
C1-{notL} here = S(w,B)
but C- S(w,B) = Q(A,x,B) union S(y,B)

for example A = {a,b,c} B={d,e}
A-B=A

correct me if I am wrong

[j]Actually, I think that's for C1 instead. Lecture slide 22 shows...
C2 = (C - (C1 - {L})) U {¬L}

[k]why S(w, B) is not considered? that clause have different variables, then that clause will be different right?

[l]I think another step is needed, like make again L1 = S(w, B).
Otherwise won't make sense that C doesn't have a S(w,B) and that new L1 is also different, right?

[m]according to the formulation C2 = (C-(C1-{not L})) union { L}
C1-{notL} here = S(w,B)
but C- S(w,B) = Q(A,x,B) union S(y,B)

for example A = {a,b,c} B={d,e}
A-B=A

correct me if I am wrong

[n]Actually, I think that's for C1 instead. Lecture slide 22 shows...
C2 = (C - (C1 - {L})) U {¬L}

[o]BTW: I am not sure if we can assume that we can get all possible combinations from alan and sushi, curry or we strictly have to declare alan, alan for sushi, curry.

[p]What about z = 1?

[q]Sorry, You were right. Z was missing. Now is correct.

[r]And why are there only these hypotheses, what happens to [3,3][3,4] [4,4] etc

[s] Marked as resolved

[t] Re-opened

[u]Oops didnt mean to resolv - I don't know, that's what David did. I had assumed we don't add them because they weren't consistent with the previous example (2) and if they were they would have been added in that step. Thoughts?

[v]*daniel lol

[w]Ok so, the number at the start is the concept we are predicting? so whys is it 1 v 1 at the start when there are 16 hypotheses?

[x]I think even though there are 16 hypotheses in the hypothesis space, there is only 1 each in S and G (so 2 votes in total).

[y](S is empty ie it votes 'false' because it fits nothing)

[z]Hey Djole while that was the way I rationalised Daniel's answer I like the look of yours better. Going to read over it now.

[aa]And why are there only these hypotheses, what happens to [3,3][3,4] [4,4] etc

[ab] Marked as resolved

[ac] Re-opened

[ad]Oops didnt mean to resolv - I don't know, that's what David did. I had assumed we don't add them because they weren't consistent with the previous example (2) and if they were they would have been added in that step. Thoughts?

hypotheses?

[ag]I think even though there are 16 hypotheses in the hypothesis space, there is only 1 each in S and G (so 2 votes in total).

[ah](S is empty ie it votes 'false' because it fits nothing)

[ai]Hey Djole while that was the way I rationalised Daniel's answer I like the look of yours better. Going to read over it now.

[aj]why?

[ak]reduces bias, so it's better at prediciting the training set

[al]He never went through this. Do we have to know it ?

[am]I think yes, and I think he did go through it. Last lecture before Edwin

[an]http://webapps.cse.unsw.edu.au/webcms2//course/index.php?cid=2405

[ao]Do we need to add the cases for ¬Windy or ¬Cloudy? or those will be enough?

[ap]Up to u I think. Coz it's obvious that it will be 1- whatever probability you find.

[aq]Question asks for required probabilities only so not really, however it doesn't specify the training instances used, so maybe all of them

[ar]Where is this P(evidence) from

[as]:Previous part C

[at]I think we need to calculate first the likelihood of Play and ¬Play and the calculate the probability of Play as in example in slide 24 of ProbLearn2.

[au]You need to multiply by P(y) of PlayTennis for the whole thing I think, as in 1/3 x (1/2 x 1/2)

[av]True! Sorry :(