Hint: The professor likes very short and to the point answers.

Tang Alexander
Van de Staey Zimcke
Xavier Goas Aguililla

Exam: Machine Learning and Inductive Inference.
January 8, 2015, 14:00

Guidelines — important!
1. This is an **open book** exam: you are allowed to consult the course text and copies of the slides used in the lectures (not the exercise sessions). Limited handwriting on these texts is OK. You are not allowed to consult any other materials.
2. Read each question carefully. Just answer the question, do not provide information that is not asked (if you do, it may affect your score). 3. Answer each question in a clear, structured way. Be **concise**, **precise** and **to the point**. It is not always necessary to write full sentences, bulleted lists may suffice.
4. All questions should be answered on these pages only. Sufficient space is provided. When answer boxes are provided, write the requested answer in the box.
5. When asked to explain some concept, give a coherent explanation in your **own words**; do not just copy formulas or text from your course text.
6. When a maximum length (number of words) is mentioned, do not ignore it. Too lengthy answers may reduce you score. Drawings and mathematical formulas do not count as words.
7. You have 3 hours to complete the exam.

Good luck!
Prof. Blockeel

Question 1 Software for organizing pictures on a computer, nowadays, often contains AI methods that can, for instance, detect faces in a picture, and label those faces with the name of a person (if that person's face was labeled on earlier pictures). Assume your software already contains a good face detector, but it has to learn how to associate faces with names. To that aim, it can present the user with pictures from the users collection, with faces indicated on them, and ask the user which person this is. Characterize this learning problem as precisely as possible.

The learning task is (draw a circle around the terms that apply)...

Predictive / descriptive

Supervised / unsupervised

Classification / regression / clustering /
multi-instance learning / reinforcement learning

Among the three settings in which we studied learning complexity (examples are presented randomly, teacher selects examples, learner selects examples), which one is the most suitable here (draw a circle around it)?

Random / teacher selects / learner selects

Arguments for your answers (optional):

Descriptive analytics provides insight into the past and answer "What has happened?" whereas predictive analytics try to understand the future and answer "What could happen?".

There's some doubt here. The end goal of the software is to predict the correct name associated with the face, but the learner itself might describe faces based on user selection.

Supervised learning is the machine learning task of inferring a function from labeled training data. Unsupervised learning is the machine learning task of inferring a function to describe hidden structure from unlabeled data.

The data (pictures) are labeled. => supervised

Classification: categorize pictures per name.

Learner selects: Software chooses which examples to learn.

Which of the following learning methods are suitable for this task? Below, briefly explain why (not).

method	YES	NO
decision trees	X	
rule learning	X	
association rules		X
neural networks	X	
support vector machines	X	
nearest neighbor methods	X	
Q-learning		X
inductive logic programming		X

Arguments for your answers:

DT: Decision trees may prove useful but depending on how the tests are chosen, the classification results may differ greatly. To remove this level of uncertainty, it's advised to apply random forests.

RL: Can be converted to decision trees and back, so equivalent to DT.

AR: Too low flexibility (true, false) on variables to determine complex facial structure in an efficient manner.

NN: Used to estimate or approximate functions that can depend on a large number of inputs and are generally unknown. Given enough data, almost anything can be described by neural networks.

SVM: An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. Using kernel and loss functions, SVM can be adapted to both allow non-linear separators and a degree of error.

features.

NN method: Label picture based on k nearest pictures, which should possess common

Q-learning: There are no consistent states and actions since it's fair to assume no two pictures will be the exactly same.

ILP: The difference in facial features cannot be represented in logic programming very well.

Question 2 Below are a number of terms (1–4), and a number of descriptions. Match the descriptions to each term (one number per description, each number occurs once). Write your answer in the boxes.

1. decision tree learners
2. support vector machines
3. k-nearest neighbors
4. Bayesian networks

description	**matching term (1–4)**
rely strongly on the definition of a similarity relationship	3
partition the dataset into subsets based on the value of specific attributes	1
construct a linear separator such that the smallest distance between the separator and any data point is maximal	2
explicitly take assumptions about probabilistic independence among attributes into account	4

Question 3 A classifier A has TP=0 and FP=1. Interpret this. Can you define a classifier B that makes use of the outputs of A to provide better predictions? In what range do you expect the predictive accuracy of B to be?

See p100 in handbook.

Classifier A is situated on the bottom right of the ROC diagram. In other words, the classifier A only makes incorrect predictions: if it should have been positive A will predict negative and if it should have been negative A will predict positive.

A classifier B which would make use of A could simply predict the reverse: predict positive if A predicts negative and predict negative if A predicts positive. This means B will now be on the upper left of the ROC diagram with TP = 1 and FP = 0. B makes only positive predictions.

$$\text{precision} = \frac{\text{number of true positives}}{\text{number of true positives} + \text{false positives}}$$

Classifier B:
TP = 1; FP = 0

precision = 100%

accuracy = TP . Pr(f(x) = pos) + (1 - FP) . Pr(f(x) = neg)
= Pr(f(x) = pos) + Pr(f(x) = neg) = 100%
Explanation: An instance is either positive or negative, so the combined chance of occurrence is 100%. Another way of looking at it: TP is the probability of predicting any positive instance correctly while (1 - FP) is the probability of predicting any negative instance correctly. Since both these variables equal 1, all positive and negative instances get predicted correctly which gives us the maximum accuracy.

Question 4 Naive Bayes is often used for text classification. Reformulate the main assumption that Naive Bayes makes in the concrete context of text classification. Is this assumption realistic? If not, how does this affect the quality of Naive Bayes' predictions?

Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

Exercise session 4, exercise 2 answers this question pretty well. Here's a concise solution:

In standard Naive Bayes we would consider every word in the vocabulary, whether or not it occurs in the sentence. For large vocabularies and document collections, a probability of a word not occurring in a document of a class is usually very high (most words do not occur in most documents). Instead, it's easier to concatenate all documents of a particular class and count word occurrences in the resulting document. This simplification makes computations easier.

Question 5 Assume given a dataset with n predictive attributes $x_1 \ldots x_n$, each of which has 5 nominal values; and a class attribute y with 3 possible values. Discuss the difference in sample complexity when (a) learning the joint probability distribution without making any assumptions at all; (b) using a Bayesian network with a given structure; (c) using Naive Bayes. Provide formulas that are as concrete as possible (i.e., with specific numbers filled in when known).

The amount of hypotheses spanned by JPD equals 3 . 5^n (permutation of 3 y values and n times 5).
Using the formula for finite hypothesis space (p171 in the book), we get:

$$n \geq \frac{log|H| + log1/\delta}{\epsilon} = \frac{log(5^n 3) + log1/\delta}{\epsilon}$$

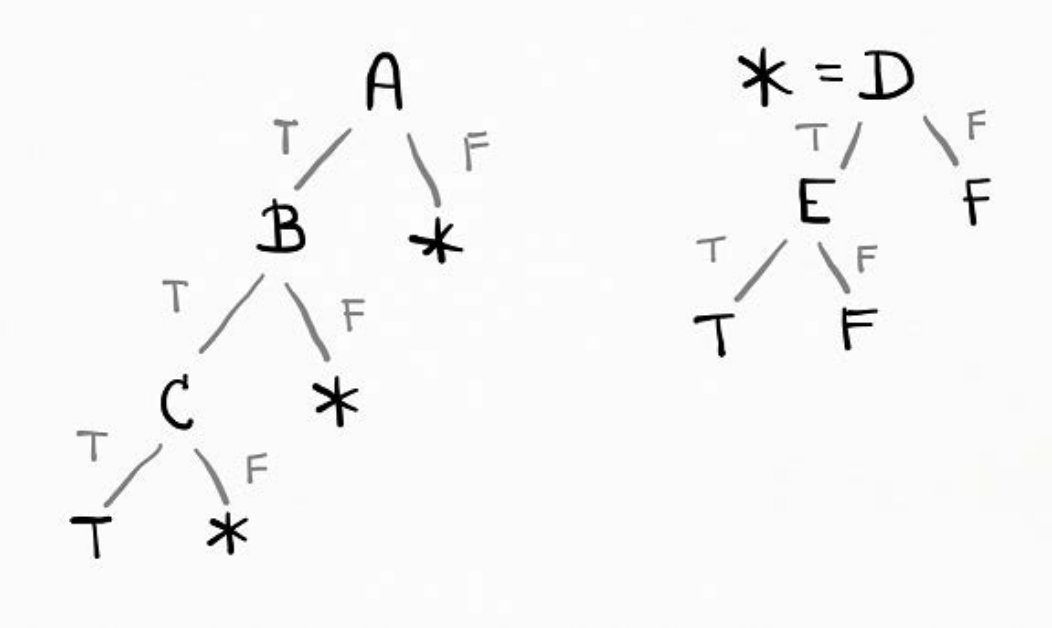
For the Bayesian network (with and without Naive Bayes) there are an infinite amount of hypotheses since there are an infinite amount of parametrizations possible. This is because a Bayesian network is a chance distribution in the [0;1] interval. So use the formula in section 9.6.2 in the book p171.

The chance distribution is more prominent with Naive Bayes since there are no conditional dependencies amongst the n attributes. This means the VC dimension of the Bayesian network without Naive Bayes should be smaller than the VC dimension of the Bayesian network with Naive Bayes. I can't figure out how to prove this numerically though.

Question 6 Answer using at most 50 words: In what way can the use of regression trees within reinforcement learning lead to more efficient reinforcement learning?

Regression trees can be merged with the exploration-exploitation phases to more accurately predict results in uncharted areas. Reinforcement learning will converge to accurate results faster this way.

Question 7 Show a decision tree that expresses the following boolean function: $(A \wedge B \wedge C) \vee (D \wedge E)$



Question 8 A tavern serves the following beers: Jupiler, Hoegaarden, Duvel. You can drink these at a table inside or outside. Jef is a regular customer in this tavern, and the waiter has the following data on him:

weather	beer	place
cloudy	Duvel	outside
rainy	Jupiler	inside
cloudy	Duvel	inside
sunny	Hoegaarden	outside
rainy	Hoegaarden	inside
sunny	Duvel	inside
rainy	Duvel	inside
cloudy	Jupiler	outside

On a rainy day, Jef sits down at a table inside. Use the following methods to predict which beer he will order:
(a) Naive Bayes (estimate probabilities without Laplace or m-estimate)
(b) 3 nearest neighbors, using the Hamming distance (distance between two vectors = number of components that differ) as dissimilarity measure.

(a) Naive Bayes

The most likely hypothesis is: $argmax_H P(rainy|H)P(inside|H)P(H)$

$$P(rainy|Duvel) = \frac{1}{4} \qquad P(inside|Duvel) = \frac{3}{4} \qquad P(Duvel) = \frac{1}{2}$$

$$P(rainy|Hoegaarden) = \frac{1}{2} \qquad P(inside|Hoegaarden) = \frac{1}{2} \qquad P(Hoegaarden) = \frac{1}{4}$$

$$P(rainy|Jupiler) = \frac{1}{2} \qquad P(inside|Jupiler) = \frac{1}{2} \qquad P(Jupiler) = \frac{1}{4}$$

$$\begin{aligned} Likelihood(Duvel) &= P(rainy|Duvel)P(inside|Duvel)P(Duvel) \\ &= \frac{1}{4}\frac{3}{4}\frac{1}{2} = \frac{3}{32} \\ Likelihood(Hoegaarden) &= \frac{1}{2}\frac{1}{2}\frac{1}{4} = \frac{1}{16} \\ Likelihood(Jupiler) &= \frac{1}{2}\frac{1}{2}\frac{1}{4} = \frac{1}{16} \end{aligned}$$

The Naive Bayes is predicts a Duvel.

(b) 3NN using Hamming distance

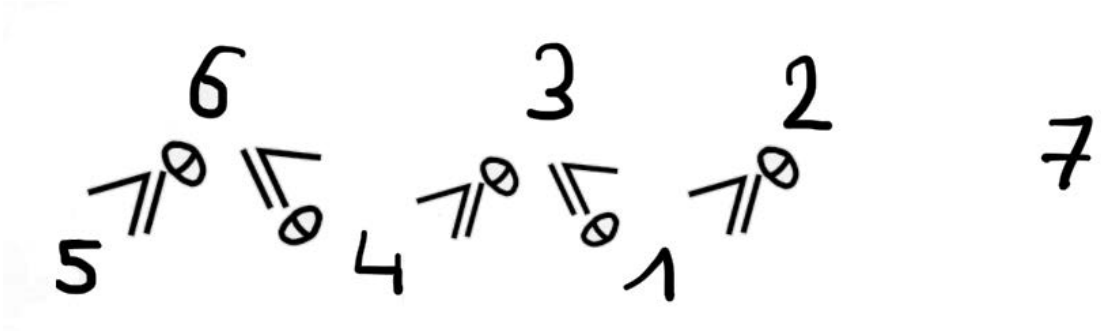
First compute the Hamming distances:

$$d(e_1, (rainy, inside)) = \sqrt{1+1} = \sqrt{2} \qquad d(e_2, (rainy, inside)) = \sqrt{0+0} = \sqrt{0}$$
$$d(e_3, (rainy, inside)) = \sqrt{0+1} = \sqrt{1} \qquad d(e_4, (rainy, inside)) = \sqrt{1+1} = \sqrt{1}$$
$$d(e_5, (rainy, inside)) = \sqrt{0+0} = \sqrt{0} \qquad d(e_6, (rainy, inside)) = \sqrt{0+1} = \sqrt{1}$$
$$d(e_7, (rainy, inside)) = \sqrt{0+0} = \sqrt{0} \qquad d(e_8, (rainy, inside)) = \sqrt{1+1} = \sqrt{2}$$

The three minimum distances are e_2 (Jupiler), e_5 (Hoegaarden) and e_7 (Duvel).
In other words, 3NN predicts that the 3 beers are equally likely to get chosen.

Question 9 Order the following clauses according to their generality under theta-subsumption:

1: p(X) ← q(X), r(X)
2: p(X) ← q(Y), r(Y)
3: p(X) ← q(Y), r(X)
4: p(a) ← q(X), r(a)
5: p(a) ← q(a), r(b)
6: p(a) ← q(X)
7: q(a) ← q(X)



Question 10 Consider an input space of 10 boolean variables, and consider as a hypothesis space the set of all decision trees of depth at most 2 (that is, there is at most one internal node between the root and any leaf). Give a lower and upper bound for the VC-dimension of this hypothesis space. (The tighter the bounds, the better, but you must give convincing arguments.)

Lower bound:	1
Upper bound:	3

Reasoning leading to these answers:

First consider 10 decision trees of depth 1. To shatter these 10 points, you need 2^{10} labels. So $|H| = 10$ and $VC(H) \leq \log_2(10) \approx 3.32$ (see proposition 9.2 for finite hypothesis spaces p169). So the upper bound for VC(H) is 3.

To find the lower bound, consider a set T with $|T| = 3$. The decision trees of depth 1 are classified as true if the attribute is true and false if the attribute is false. The following set T can be shattered by H:

$$\begin{aligned} x_1 = (A_1 = true, A_2 = false, A_3 = false, A_4 = true, A_5 = false, \\ A_6 = true, A_7 = true, A_8 = false, A_9 = -, A_{10} = -) \\ x_2 = (A_1 = false, A_2 = true, A_3 = false, A_4 = true, A_5 = true, \\ A_6 = false, A_7 = true, A_8 = false, A_9 = -, A_{10} = -) \\ x_3 = (A_1 = false, A_2 = false, A_3 = true, A_4 = false, A_5 = true, \\ A_6 = true, A_7 = true, A_8 = false, A_9 = -, A_{10} = -) \end{aligned}$$

In other words, all permutations of (boolean, boolean, boolean) as seen in figure 9.2 p 168 in the book. We know all 8 permutations can be shattered (with two excess trees for A_9 and A_{10}). Since this subset can be shattered, VC(H) must be at least 3. So we have determined that VC(H) for the decision trees of depth 1 equals exactly 3.

Consider the hypothesis space spanned by 3 decision trees with 3 variables (a variable in the root and 2 variables in each of the leaves of the root node) and one leftover decision tree of depth 1. Then we have a decision tree with 4 target leaves. Such a decision tree has 4 possible permutations: (true, true), (true, false), (false, true), (false, false). The amount of hypotheses in this set equals 4. So the upper bound for VC(H) is computed as follows: $VC(H) \leq \log_2(4) = 2$.

Trying to shatter 2 points:

$$\begin{aligned} x_1 = (A_1 = (true, true), A_2 = (true, true), A_3 = (true, true), A_4 = true) \\ x_2 = (A_1 = (true, true), A_2 = (true, false), A_3 = (false, true), A_4 = false) \end{aligned}$$

Way too many combinations needed; cannot be shattered. It is trivial to shatter 1 point, so the VC(H) equals 1.

Note that any combination of decision trees in between these two extremes will result in a VC(H) that's also in between. This can be easily checked by computing the total amount of hypotheses available in each scenario.

Everything combined gives us a lower bound of 1 and an upper bound of 3.