

Midterm Examination

Wednesday November 10, 5pm to 6:20pm

Your name:

Instructions: Look through the whole exam and answer the questions that you find easiest first. Answer each question in the space below the question, using the backs of the pages for extra space as necessary.

If necessary, you may make assumptions that are reasonable, and that do not make a question trivial. If you do make an assumption, state it clearly.

You may bring and use the following materials: Russell and Norvig, the published lecture and section notes, your own *personal* hand-written notes, and a calculator. You may not use any other materials.

(Question 1) [20 points]

(a) [5 points] Explain why the following is true: The most probable label of a test example $\langle v_1, v_2, \dots, v_p \rangle$ is the label c that maximizes

$$\frac{P(X_1 = v_1, \dots, X_p = v_p | C = c)P(C = c)}{P(X_1 = v_1, \dots, X_p = v_p)}.$$

—

We want to find the class label c that maximizes $P(C = c | X_1 = v_1, \dots, X_p = v_p)$. The above formula is found by applying Bayes rule to this formula.

—

(b) [5 points] Now explain why the following is true: If all labels are equally likely in advance, then the most probable label of a test example $\langle v_1, v_2, \dots, v_p \rangle$ is the label c that maximizes

$$P(X_1 = v_1, \dots, X_p = v_p | C = c).$$

—

If all classes are equally likely, $P(C = c)$ is the same for all classes. It is the constant $\frac{1}{|C|}$ where $|C|$ is the number of classes.

The denominator $P(X_1 = v_1, \dots, X_p = v_p)$ does not depend on class, and thus is constant for all classes.

In this case, since $P(C = c)$ and $1/P(X_1 = v_1, \dots, X_p = v_p)$ are constants in the Equation from Part (a), the most probable label of a test example will only depend on the factor $P(X_1 = v_1, \dots, X_p = v_p | C = c)$.

—

(c) [From <http://www.cs.nyu.edu/courses/spring04/G22.2560-001/sample-fx-qns.html>, 5 points] Consider the following five training examples with three binary features W, X, Y and binary label C .

W	X	Y	C
T	T	T	T

T	F	T	F
T	F	F	F
F	T	T	F
F	F	F	T

We now encounter a test example with $W = F, X = T, Y = F$. If we apply the naive Bayes method, what probability is assigned to the two values of C ?

—

The probability of predicting $C = t$ given the assignment:

$$\begin{aligned}
 P(C = t | W = f, X = t, Y = f) &= \frac{P(W = f, X = t, Y = f | C = t)P(C = t)}{P(W = f, X = t, Y = f)} \\
 &= \frac{P(W = f, X = t, Y = f | C = t)P(C = t)}{\sum_{c \in C} P(W = f, X = t, Y = f | C = c)P(C = c)}
 \end{aligned}$$

Under the naive Bayes Assumption of class conditional independence, we have

$$P(C = t | W = f, X = t, Y = f) = \frac{P(W = f | C = t)P(X = t | C = t)P(Y = f | C = t)P(C = t)}{\sum_{c \in C} P(W = f | C = c)P(X = t | C = c)P(Y = f | C = c)P(C = c)}$$

.

By examining the table, we can calculate all of these values:

$$P(C = t) = 2/5$$

$$P(C = f) = 3/5$$

$$P(W = f | C = t)P(X = t | C = t)P(Y = f | C = t) = (1/2)(1/2)(1/2) = (1/8)$$

$$P(W = f | C = f)P(X = f | C = f)P(Y = f | C = f) = (1/3)(1/3)(1/3) = (1/27)$$

Then,

$$P(C = t | W = f, X = t, Y = f) = \frac{(1/8)(2/5)}{(1/8)(2/5) + (1/27)(3/5)} = \frac{1/20}{13/180} = 0.692$$

and

$$P(C = f | W = f, X = t, Y = f) = 1 - P(C = t | W = f, X = t, Y = f) = 0.308$$

—

(d) [5 points] Suppose we had a very large training set (say $N = 1,000,000$) but only two classes ($r = 2$), only a few features (say $p = 4$) and only a few values per feature (say $q = 5$). Explain how we could train a Bayesian classifier without making the naive Bayes assumption. (Hint: How many parameters do we need to learn, if we don't make the naive Bayes assumption?)

—

Without making the naive Bayes assumption, the number of parameters we would have to estimate is $O(q^p r)$.

Consider estimating the first parameter where X_i can take on values $\{1, 2, 3, 4, 5\}$ and C can take values $\{1, 2\}$:

$$P(X_1 = 1, X_2 = 1, X_3 = 1, X_4 = 1 | C = 1)$$

The next parameter we would have to estimate would be

$$P(X_1 = 1, X_2 = 1, X_3 = 1, X_4 = 2|C = 1)$$

and so on. If we were to enumerate all assignments to all variables, we would have $q \cdot q \cdot q \cdot q \cdot (r - 1) = 5^4 = 625$ parameters to estimate. Note that we have the factor $(r - 1)$ instead of r since we can always calculate the the probability for one class using the probabilities from all the other classes:

$$P(X_1 = 1, \dots, X_4 = 1|C = r) = 1 - \sum_{i=1}^{r-1} P(X_1 = 0, \dots, X_4 = 0|C = i)$$

Given that we have 1,000,000 training points and 625 parameters to estimate, we will have on average 1600 training points, which is enough for accurate parameter estimation.

—

(Note: The answers given in parts c and d are complete. Your answers did not have to be as detailed to receive full credit.)

(Question 2) [30 points] For each statement below, clearly write “True” if it is mostly true, or “False” if it is mostly false. Then in the space below, write one or two sentences explaining why or how the statement is true or false. The maximum score for each answer is two points.

1. Asymptotically, iterative deepening and breadth-first search have the same big-O space complexity.
FALSE - Breadth-First Search has a space complexity of $O(b^{d+1})$ and iterative deepening (which reuses space) has a space complexity $O(bd)$. See table 3.17 in AIMA for more details.
2. Asymptotically, iterative deepening and breadth-first search have the same big-O time complexity.
TRUE - Both have time complexity $O(b^d)$. Note that $O(b^d)$ and $O(b^{d+1})$ are the same assuming b is fixed and d varies. This is a valid assumption in that the branch factor does not change across various problem instances of a problem (ie eight puzzle). However, there are some problem (ie Robot Navigation) in which the branch factor does depend on problem instance.
3. Breadth-first search is a special case of the A* algorithm.
TRUE - Let $g = \text{number of edges}$ and let $h = 0$. This is the special case of A* that is the same as BFS.
4. If the branching factor $b > 1$ then A* uses exponential time (i.e. $O(c^d)$ time where c is a constant and d is the depth of the solution found by A*
TRUE - A* is an exponential time search algorithm when $b > 1$. A* does uses heuristics to prune branches for the search tree but this does not result in a reducing the theoretical time bounds.
5. Asymptotically, A* always has the same big-O time complexity with or without duplicate elimination.
FALSE - Duplicate elimination effectively reduces the branch factor. Consider the case where the search algorithm is allowed to return to the node that it had just visited. With duplicate elimination, it will never return to this node.

Without duplicate elimination, suppose that the branch factor was b , the depth of the solution was d , and the algorithm had a big-O time complexity of $O(b^d)$. With duplicate elimination, the new branch factor is less the $b - 1$, and the time complexity is $O((b - 1)^d)$.

6. The class scheduling problem is a CSP.

The class scheduling problem: Given fixed sets of classes, professors, and class times, and that each professor can only teach certain classes, find which professors should teach which classes at which times.

TRUE - Our states are a (complete or partial) assignment of $\langle class, professor, time \rangle$ for each class. Our constraints consist of professors not teaching class that they are not qualified to teach and that a professor does not teach two classes at the same time.

7. The classroom planning problem is a CSP.

The classroom planning problem: Given fixed sets of classes, professors, and class times, and that each professor can only teach certain classes, find the minimum number of classrooms needed.

FALSE - This formulation involves finding an optimal assignment. CSPs involve finding any satisfying assignment.

8. The worst-case running time of DPLL is exponential in the number of Boolean variables.

TRUE - DPLL is an exponential runtime search algorithm. If you refer to the pseudo code for DPLL (Figure 7.16 in AIMA), in the worst case, the algorithm recursively calls itself twice having assigned only one variable. This means that, in the worst case, DPLL is called 2^n where n is the number of boolean variables.

9. The worst-case running time of Walksat is exponential in the number of Boolean variables.

FALSE - Walksat may not terminate even if there is a satisfying assignment and will not terminate if there is no satisfying assignment.

10. It is possible that $P(A, B|C) \neq P(B, A|C)$.

FALSE - $P(A, B|C) = \frac{P(A, B, C)}{P(C)} = P(B, A|C)$ since both $P(X, Y) = P(Y, X)$ and by the definition of conditional probability.

11. The following equation is valid: $P(E, F|G) = P(E|G)P(F|E, G)$.

TRUE -

$$P(E, F|G) = \frac{P(E, F, G)}{P(G)}$$
$$P(E|G)P(F|E, G) = \frac{P(E, G)}{P(G)} \frac{P(F, E, G)}{P(E, G)} = \frac{P(E, F, G)}{P(G)}$$

This is according to the product rule. See the course lecture notes.

12. If P and Q are independent given R , then $\neg P$ and $\neg Q$ are also independent given R .

TRUE - Intuition: If knowing P gives no information about Q , then it also gives no information about $\neg Q$. Conditioning everything on R is a red herring. (i.e. It serves to distract.)

Proof:

$$\begin{aligned}Pr(\neg P, \neg Q|R) &= 1 - Pr(P \vee Q|R) \quad (\text{by DeMorgan's law}) \\&= 1 - [Pr(P|R) + Pr(Q|R) - Pr(P, Q|R)] \\&= 1 - [Pr(P|R) + Pr(Q|R) - Pr(P|R)Pr(Q|R)] \quad (\text{since } P \text{ and } Q \text{ are independent given } R) \\&= (1 - Pr(P|R))(1 - Pr(Q|R)) \\&= Pr(\neg P|R)Pr(\neg Q|R)\end{aligned}$$

Thus, $\neg P$ and $\neg Q$ are independent given R .

13. Suppose student A attends 60% of lectures, student B attends 50% of lectures, A and B both attend 30% of lectures, and student C attends precisely when neither A nor B attends. In this scenario, A and B are not independent.

FALSE - Since $P(A, B) = P(A)P(B) = 0.3$, the events A and B are independent by definition.

14. In the scenario above, $P(C|\neg B) = 0.25$.

FALSE - Note that from question 13, we can deduce that $P(A, \neg B) = .3$, $P(B, \neg A) = .2$, $P(A, B) = .3$. This means that $P(C, \neg A, \neg B) = P(C, \neg B) = .2$. (To realize this, a Venn diagram will be helpful.) Then, $P(C|\neg B) = \frac{P(C, \neg B)}{P(\neg B)} = \frac{.2}{.5} = 0.4 \neq 0.25$.

15. A naive Bayes classifier is especially useful if the features are uncorrelated within each class.

TRUE - This is the case in which the assumption is fully justified. The output of the classifier will be an *exact* probability of the class given the feature vector.