

**CS 540: Introduction to Artificial Intelligence
Homework Assignment #3**

Assigned: Nov. 3, 2011

Due: Nov. 16, 2011 before class

Hand in your homework:

This homework assignment includes written problems and programming in Java. Hand in hardcopy of the requested written parts of the assignment in class. All pages should be stapled together, and should include a cover sheet on top of which includes your name, login, class section, HW #, date, and, if late, how many days late it is. Electronically hand in files containing the Java code that you wrote for the programming part. See course Web page for instructions.

Late Policy:

All assignments are due **at the beginning of class** on the due date. One (1) day late, defined as a 24-hour period from the deadline (weekday or weekend), will result in 10% of the total points for the assignment deducted. So, for example, if a 100-point assignment is due on a Wednesday 11 a.m., and it is handed in between Wednesday 11 a.m. and Thursday 11 a.m., 10 points will be deducted. Two (2) days late, 25% off; three (3) days late, 50% off. No homework can be turned in more than three (3) days late. Written questions and program submission have the same deadline. A total of two (2) free late days may be used throughout the semester without penalty.

Assignment grading questions must be raised with the instructor within one week after the assignment is returned.

Collaboration Policy:








You are to complete this assignment individually. However, you are encouraged to discuss the general algorithms and ideas with classmates, TAs, and instructor in order to help you answer the questions. You are also welcome to give each other examples that are not on the assignment in order to demonstrate how to solve problems. But we require you to:

- not explicitly tell each other the answers
- not to copy answers or code fragments from anyone or anywhere
- not to allow your answers to be copied
- not to get any code on the Web

In those cases where you work with one or more other people on the general discussion of the assignment and surrounding topics, we suggest that you specifically record on the assignment the names of the people you were in discussion with.

Question 1: Search Algorithms [30]

In the figure to the right is a 5 x 5 maze filled with mouse traps (fixed in cells D, G, K, O and R). There is a poor hungry mouse at one corner of the maze which smells the presence of cheese somewhere in the maze but does not know the exact location. Your task is to help the mouse find its food (fixed in square M) without getting trapped anywhere. Assume that the mouse can sense the presence of a mouse trap from any of its horizontal and vertical cells. Also, the mouse can move only in the four directions namely left, right, up and down. It cannot move diagonally. Assume the successor function will cause legal moves to be examined in a clockwise order: up; right; down; left. Note that not all of these moves may be possible from a given square.

A 	B	C	D 	E
F	G 	H	I	J
K 	L	M 	N	O 
P	Q	R 	S	T
U	V	W	X	Y

- [10] Using **Depth-First Search**, list the squares in the order they are expanded (including the goal node if it is found). Square **A** is expanded first (hint: State **B** will be examined next). Assume **cycle checking** is done so that a node is *not* generated in the search tree if the grid position already occurs on the path from this node back to the root node (i.e., Path Checking DFS). Write down the list of states you expanded in the order they are expanded. Write down the solution path found (if any), or explain why no solution is found.
- [10] Using **Iterative Deepening Search**, draw the trees built at each depth until a solution is reached. Use the same cycle checking as in Part *a*.
- [5] Let each move (up/down/left/right) of the mouse have cost 1. Consider the heuristic function $h(n) = |xn - xg| + |yn - yg|$, where the grid square associated with node n is at coordinates (xn, yn) on the board, and the goal node **M** is at coordinates (xg, yg) . That is, $h(n)$ is the Manhattan distance between n and **M**. Is $h(n)$ admissible? Why?
- [10] Regardless of your answer to Part *c*, Perform **A* Search** (as defined in lecture notes) using the heuristic function $h(n)$ with a slight modification that $h(n) = \infty$, if node n has a mouse trap. In the case of ties, expand states in alphabetical order. List each square in the order they are added to the *OPEN* list, and mark it with $f(n)=g(n)+h(n)$ (show f , g and h separately). When expanded (including node **M**), label a state with a number indicating when it was expanded (position **A** should be marked “1”). Highlight the solution path found (if any), or explain why no solution is found.

- e) [5] Repeat Part *d* using a new heuristic function $h_2(n)$, which is defined using the old heuristic $h(n)$ as follows:

$h(n)$	0	1	2	3	4	5	6	7	8	9	∞
$h_2(n)$	0	1	3	1	2	3	1	2	1	1	∞

Question 2: Probability

The following table gives probabilities for three Boolean random variables, X , Y , and Z :

	Y		$\neg Y$	
	Z	$\neg Z$	Z	$\neg Z$
X	0.5	0.03	0.02	0.1
$\neg X$	0.2	0.05	0	0.1

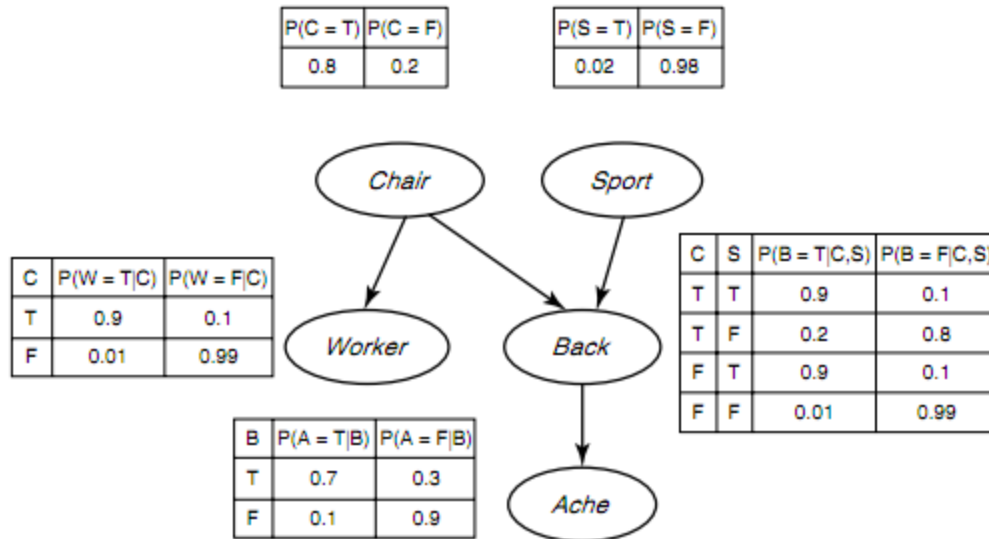
- What is $P(\neg X)$? Show your work.
- What is $P(Y | \neg X)$? Show your work.
- What is $P(\neg Z | X, Y)$? Show your work.
- What is $P(\neg Z | X, \neg Y, Z)$? Show your work.
- Is X and Z independent? Briefly explain.

Question 3: Answer Questions using a Bayesian Network

Consider five Boolean events (random variables): Consider the Bayesian Network below where it considers a person X who might suffer from a back injury as an event represented by the variable Back (denoted by B). Such an injury can cause a backache, an event represented by the variable Ache (denoted by A). The back injury might result from a wrong sport activity, represented by the variable Sport (denoted by S) or from new uncomfortable chairs installed at X 's office, represented by the variable Chair (denoted by C). In the latter case, it is reasonable to assume that a coworker will suffer and report a similar backache syndrome, an event represented by the variable Worker (denoted by W). Use the Bayesian Network below to answer the following questions, using inference by enumeration. Show your work.

- What is the probability that X has a back injury?
- What is the probability that a coworker complains of a back ache?
- What is the probability that X does not have a back ache?
- If a coworker complains about a backache syndrome then what is the probability that X has a back injury and a back ache?
- If a coworker complains about a backache, what is the probability that the chairs are new and uncomfortable?
- If X has a back ache, what is the probability that he played a wrong sport?
- If X has a back ache, what is the probability that he has a back injury?
- If X plays a wrong sport and X sits on new uncomfortable chairs what is the probability he has a back ache?
- If X plays a wrong sport and X sits on new uncomfortable chairs what is the probability a coworker has a back ache?

- (j) If X has a back injury, what is the probability that X sits on new uncomfortable chairs and plays a wrong sport?



Question 4: Language Identification with a Naïve Bayes Classifier

Naive Bayes is a simple, effective machine learning method that can be used to solve the problem of identifying the language of a document. You are to implement a Naive Bayes classifier that classifies a document as English, Spanish, or Japanese – all written with the 26 lower case letters and space.

The dataset for this assignment is `Q4.tar.gz` which is available at the class website. This dataset consists of training and test documents in English, Spanish and Japanese. Both the training dataset and the test dataset contain three subdirectories: `English/`, `Spanish/`, and `Japanese/`. These subdirectories in turn contain the documents as separate ASCII text files. The data is therefore organized as follows:

```
trainingdataset/English/
trainingdataset/Spanish/
trainingdataset/Japanese/
testdataset/English/
testdataset/Spanish/
testdataset/Japanese/
```

We will be using a character-based Naïve Bayes model. You need to view each document as a stream of characters, including space. We have made sure that there are only 27 different types of characters (*a* to *z*, and *space*).

You must compute and store the prior probabilities, $P(\text{English})$, $P(\text{Spanish})$ and $P(\text{Japanese})$, as well as the conditional probabilities, $P(c \mid \text{English})$, $P(c \mid \text{Spanish})$, and $P(c \mid \text{Japanese})$, where c is one of the 27 characters, from the training set. Store all probabilities as logs to avoid underflow. This also means you need to do arithmetic in log-space. That is, multiplications of probabilities become additions of log probabilities. Hints are given at the end of this question.

You are required to complete the following tasks:

1. Using all the characters in the training data, build a Naive Bayes classifier for the three languages. Implement your classifier using a log-likelihood formulation of Naïve Bayes.
2. Print $P(\text{English})$, $P(\text{Spanish})$ and $P(\text{Japanese})$, as well as the conditional probabilities $P(c \mid \text{English})$, $P(c \mid \text{Spanish})$, and $P(c \mid \text{Japanese})$ for all 27 characters c .
3. Evaluate the performance of your classifier on the *test set* using a confusion matrix. A confusion matrix summarizes the types of errors your classifier makes, as shown in Table 1. The columns are the true language a document is in, and the rows are the classified outcome of that document. The cells are the number of test documents in that situation. For example, the cell with row = English and column = Spanish contains the number of test documents that are really Spanish, but misclassified as English by your classifier.

Table 1. Confusion Matrix

	true English	true Spanish	true Japanese
Predicted English			
Predicted Spanish			
Predicted Japanese			

4. If someone prints out a test document, then shreds the paper, so that all characters in the document are still visible but their order is completely scrambled, will your classifier work on the scrambled text? Justify your answer.

You may implement your program any way you like, but you should write a Java class with the following calling convention:

```
java hw3 trainset_directory testset_directory
```

Your program should be able to complete the above tasks and output the required probability lists in task 2 and confusion matrix in task 3 to the standard output.

Deliverables:

1. Handin all relevant “.java” and “.class” files needed for your program.
2. Handin a Makefile to compile the code.

Hand in your code as specified by the course webpage.

Hints:**a. Computing prior probabilities**

Count the number of English documents in the training set:

$n_{English}$ = number of documents in the training set's `English` directory

Count the number of Spanish documents in the training set:

$n_{Spanish}$ = number of documents in the training set's `Spanish` directory

Count the number of Japanese documents in the training set:

$n_{Japanese}$ = number of documents in the training set's `Japanese` directory

Compute the total number of training documents:

$n_{Total} = n_{English} + n_{Spanish} + n_{Japanese}$

Compute the prior probability for English:

$P(English) = n_{English} / n_{Total}$

Compute the prior probability for Spanish:

$P(Spanish) = n_{Spanish} / n_{Total}$

Compute the prior probability for Japanese:

$P(Japanese) = n_{Japanese} / n_{Total}$

b. Computing conditional likelihoods

Let $n_{CharEnglish}$ be the total number of characters (including multiple occurrences of the same unique character, including spaces) contained in all English training documents, and let $n_{CharSpanish}$ and $n_{CharJapanese}$ be that for the Spanish and Japanese training documents, respectively.

For each of the 27 unique characters, c_i , compute three conditional probabilities:

$P(c_i | English) = countEnglish(c_i) / n_{CharEnglish}$, where $countEnglish(c_i)$ is the number of times character c_i occurs in all English documents in the training set. Similarly, compute $P(c_i | Spanish)$ and $P(c_i | Japanese)$.

c. From probabilities to log probabilities

Convert all probabilities to log probabilities to avoid underflow problems. Use the natural logarithm ($\log(x)$ in Java). Apply the log function to all probabilities.

d. Classifying a test document

Consider a new document, *doc*, from the test set. Suppose it contains the sequence of characters $c1, c2, \dots, ck$ (note: the same character may occur multiple times).

Compute the posterior probabilities (where a is the common denominator in Bayes's rule, which can be ignored for ranking the three languages):

$$\begin{aligned}
 P(\text{English} / \text{doc}) &= P(\text{English} \mid c1, c2, \dots, ck) \\
 &= \alpha P(c1, c2, \dots, ck \mid \text{English}) P(\text{English}) \\
 &= \alpha P(\text{English}) P(c1 \mid \text{English}) P(c2 \mid \text{English}) \dots P(ck \mid \text{English}) \\
 P(\text{Spanish} / \text{doc}) &= P(\text{Spanish} \mid c1, c2, \dots, ck) \\
 &= \alpha P(c1, c2, \dots, ck \mid \text{Spanish}) P(\text{Spanish}) \\
 &= \alpha P(\text{Spanish}) P(c1 \mid \text{Spanish}) P(c2 \mid \text{Spanish}) \dots P(ck \mid \text{Spanish}) \\
 P(\text{Japanese} / \text{doc}) &= P(\text{Japanese} \mid c1, c2, \dots, ck) \\
 &= \alpha P(c1, c2, \dots, ck \mid \text{Japanese}) P(\text{Japanese}) \\
 &= \alpha P(\text{Japanese}) P(c1 \mid \text{Japanese}) P(c2 \mid \text{Japanese}) \dots P(ck \mid \text{Japanese})
 \end{aligned}$$

Using log probabilities, the product becomes a sum:

$$\begin{aligned}
 \log P(\text{English} / \text{doc}) &= \log \alpha + \log P(\text{English}) + \log P(c1 \mid \text{English}) \\
 &\quad + \log P(c2 \mid \text{English}) + \dots + \log P(ck \mid \text{English})
 \end{aligned}$$

and similarly for Spanish and Japanese.

Classify the document as English if

$\log P(\text{English} / \text{doc}) > \log P(\text{Spanish} / \text{doc})$ and $\log P(\text{English} / \text{doc}) > \log P(\text{Japanese} / \text{doc})$. Similarly for the other two languages.