CS 5526 – Virginia Tech
Homework 1

Brennon Bortz
(PID: brennon, Campus: Blacksburg)

11 March 2015

Code for this assignment is available at `https://github.com/brennon/cs5526-hw1`. Python notebooks for Question 5 are available at `http://goo.gl/SeQn1b` (Naïve Bayes Classifier) and `http://goo.gl/BZdYD1` (Tree-Augmented Naïve Bayes Classifier).

Written Problems

1. (10 points) Give an example of a decomposition of a joint probability distribution that cannot be captured by a Bayesian network.

   Solution: The graph structure of Figure 1 with the following as the only independencies cannot be represented by a Bayesian network: $(A \perp D | B, C)$ and $(B \perp C | A, D)$.

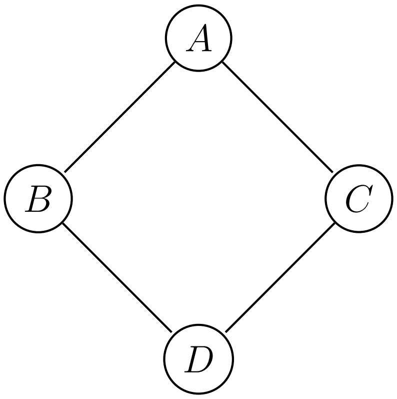

Figure 1: Impossible Bayesian network.

2. (10 points) How many possible Bayesian networks are there involving three random variables? Group these networks into equivalence classes where each class contains networks that encode the same (conditional) independence assumptions.

   Solution: There are 25 possible Bayesian networks involving three variables, comprising 8 different equivalence classes. Each row in Table 2 shows one of these possible networks. An arrow in one of the first three columns indicates the direction of the edge connecting the nodes given in the header row. A check in one of the last three columns indicates the conditional independency in the header row holds given the edges indicated in the first three columns. Where two rows share the same pattern of check in the last three columns, they are members of the same equivalence class.

X-Y	Y-Z	X-Z	$X \perp Y\|Z$	$X \perp Z\|Y$	$Y \perp Z\|X$
			✓	✓	✓
	$\leftarrow$		✓	✓	
	$\rightarrow$		✓	✓	
		$\leftarrow$	✓		✓
		$\rightarrow$	✓		✓
	$\leftarrow$	$\rightarrow$	✓		
	$\rightarrow$	$\leftarrow$	✓		
$\leftarrow$				✓	✓
$\rightarrow$				✓	✓
	$\leftarrow$	$\leftarrow$		✓	
$\leftarrow$	$\leftarrow$			✓	
$\leftarrow$	$\rightarrow$			✓	
$\rightarrow$	$\rightarrow$			✓	
$\leftarrow$		$\rightarrow$			✓
$\rightarrow$		$\leftarrow$			✓
$\rightarrow$		$\rightarrow$			✓
	$\rightarrow$	$\rightarrow$			
$\leftarrow$		$\leftarrow$			
$\leftarrow$	$\leftarrow$	$\leftarrow$			
$\leftarrow$	$\rightarrow$	$\leftarrow$			
$\leftarrow$	$\rightarrow$	$\rightarrow$			
$\rightarrow$	$\leftarrow$				
$\rightarrow$	$\leftarrow$	$\leftarrow$			
$\rightarrow$	$\leftarrow$	$\rightarrow$			
$\rightarrow$	$\rightarrow$	$\rightarrow$			

Table 1: Possible Bayesian networks in three variables.

3. (10 points) In the attached diagram (next page) assume that B and M are instantiated (i.e., evidence is introduced for these variables). List all the random variables that A is conditionally independent of.

   Solution: A is only independent of G given B and M.

4. (20 points) From Project Gutenberg, download the two files The Adventures of Sherlock Holmes by Arthur Conan Doyle (http://www.gutenberg.org/cache/epub/1661/pg1661.txt) and The Complete Works of Jane Austen (http://www.gutenberg.org/cache/epub/31100/pg31100.txt). Design a Markov sequence model that learns from each of these files (separately) and learns to write like Doyle or write like Austen. Note that your model need not be a HMM, just a probabilistic sequence model that predicts what the next word should be based on the current word (or current + past words, or some longer history). If you are adventurous, you can also explore the so-called skip gram models. How many bits of history would you need to use to

create a realistic model, for each author? What are the disadvantages of using more history to form your model? For full credit, give an explanation of the experiments you tried and one example of a pseudo-Doyle document and one example of a pseudo-Austen document (1 page max each).

Solution: In my opinion, using 3-grams creates the most realistic output text for both authors. If the n-grams are shorter, the text makes little sense. If the n-grams are longer, the frequencies of any given n-gram are much lower. This causes the model to copy large sections of text wholesale from the original text, which is not the intended result. In addition to this, using less history is much faster and requires far less memory. I attempted both models with 1-, 2-, 3-, and 4-grams. Samples of output text are provided with the source used to generate them in the repository linked at the beginning of this document.

5. (50 points) Consider the Mushroom dataset from the UCI machine learning repository. Develop a Naive Bayes classifier and a tree-augmented Naive Bayes classifier to classify this dataset. Separate the data into training and test (describe what percentages you used) and report quantitative results after k-fold cross-validation. (Choose k suitably.) Interpret your experimental results and analyze if the tree-augmented classifier provides an improvement. You are welcome to code up the classifiers from scratch or use some ready made software for parts of the assignment, but must document what you did (e.g., software, languages used) and provide a public URL where any code/scripts written are made available.

   Finally, I should note that I used the `Counter` class available from the UC Berkeley CS 188 course source code.[1]

   Solution: For each classifier (see above for links to source and iPython notebooks), I used 10-fold cross validation to acheive the following results:

Network Type	Accuracy
Naïve Bayes Classifier	99.4334%
Tree-Augmented Naïve Bayes Classifier	99.9507%

   Table 2: NBC and TA-NBC accuracies on UCI mushroom dataset.

   While the TA-NBC did provide an improvement over the NBC, it was slight. Furthermore, the additional effort involved in hand-coding the TA-NBC was hardly worth the effort given this minimal increase in accuracy.

   To implement these networks, I used the `libpgm` Python library[2]. For the TA-NBC in particular, however, much of the implementation was created by myself, as the library does not support TA-NBCs directly.

[1] `https://s3-us-west-2.amazonaws.com/cs188websitecontent/projects/release/search/v1/001/docs/util.html`

[2] `http://pythonhosted.org/libpgm/`