CS6730 : Assignment 1

Instructor and TAs

Release: 26th Feb 2019; **Due: Mar 7th, 11.59pm**

- Submit to **GradeScope a single LaTeX-generated pdf file** containing your solutions. Please type your answers in the solutions blocks in the source LaTeX file of this assignment.
- The final question worth half overall points is a programming assignment that asks for (a) the formulas you used, (b) a well-documented code you wrote, and (c) submission of predictions from your code to a kaggle competition.
- You are encouraged to collaborate/discuss with other students on this assignment, but write your solutions/code in your own words.

---

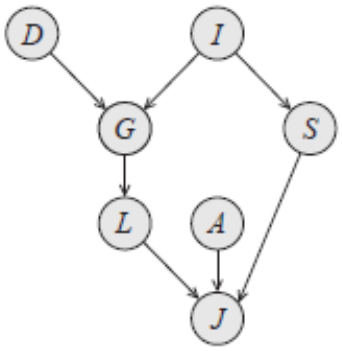
1. (6 points) [HOW TREE-LIKE ARE YOU?] Let H be an undirected graph with n nodes. Let $T(H)$ be the set of all chordal graphs with n-nodes that contain all edges in H. The tree width of H is defined as $\min\{\text{Max-Clique}(H') - 1 : H' \in T(H)\}$.
   (a) (4 points) What is the tree-width of the $n \times n$ grid graph containing n^2 nodes? Give proof. (Hint: Answer scales linearly with n.)
   (b) (2 points) What is the tree width of the cycle graph with n nodes? Give proof. (Hint: Answer does not depend on n.)
2. (7 points) [NAIVE GETS DISORIENTED]
   (a) (1 point) Give the MN structure and distribution for the Naive Bayes model, with the class label Y taking values in the set $\{1, 2, \ldots, n\}$ and feature values $X_1, \ldots, X_n$ taking values in $\{0, 1\}$.
   (b) (4 points) Give two distinct settings of the factors in the Markov network, so that $P(X_i = 1|Y = j) =$ 0.9 if i = j and 0.1 otherwise.
   (c) (2 points) One operation on MNs that arises in many settings (including variable elimination) is the marginalization of some node in the network. Give the minimal MN I-map for just the set of feature random variables $X_1, \ldots, X_n$ and also the form of any distribution P that factorizes over such a network.
3. (7 points) [MORAL SOUND OF (D)SEP] Let $\mathcal{G}$ be a Bayesian Network DAG over $\mathcal{X}$, and let $\mathcal{H} = \mathcal{M}[\mathcal{G}]$ be the moralized version of G. Let X, Y, Z be **any subsets** of $\mathcal{X}$. Then, state whether these statements are true or false, and briefly justify why. You can use the "Student" BN shown in figure below to obtain counter-examples or proof intuitions.
   (Note: This question provides tools for thinking about d-sep criteria in terms of the simpler sep criterion by moralization of the appropriate graph. This helps prove soundness of d-sep (which we only argued

intuitively in class using all three-node DAGs) using soundness of sep (which is much easier to prove, as shown in class)]].

(a) (1 point) Is "X and Y are d-separated given Z in G if and only if X and Y are separated given Z in H"?

(b) (2 points) Let $\mathbf{U} = \mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}$, and $\mathcal{G}' = \mathcal{G}[\mathbf{U} \cup \mathrm{Ancs}_{\mathbf{U}}]$ be the induced sub-graph over $\mathbf{U}$ and its ancestors. Is "X and Y are d-separated given Z in $\mathcal{G}$ if and only if X and Y are d-separated given Z in $\mathcal{G}'$"?

(c) (4 points) Consider same definitions as in (b) and let $\mathcal{H}' = \mathcal{M}[\mathcal{G}']$. Is "X and Y are d-separated given Z in $\mathcal{G}'$ if and only if X and Y are separated given Z in $\mathcal{H}'$"?



4. (20 points) [NAIVE REORIENTATION AND UPGRADE] This programming assignment involves building Naive Bayes (NB) vs. Bayesian Network (BN) classifiers for detecting heart attack using cardiac images. Specifically, for the NB classifer and the BN classifier (whose structure is specified below), we ask for:

   (a) (5 points) formulas you used for (i) estimating conditional probability tables (CPTs' parameters) in the training step; and (ii) the log conditional probability of a class given all features in the testing step; and

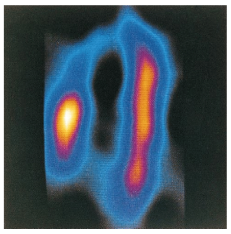
   (b) (15 points) (i) accuracy on the test set, obtained by submitting your code implementing each of the two classifiers to this kaggle competition link, and (ii) source-code listing of your well-documented code implementing the two classifiers in a language of your choice. The final submission to kaggle should be your best-performing classifier code.

   (c) (10 points) (Extra Credit worth 2.5% of overall course marks) If you can modify the BN structure to get a better-performing classifier, please submit that BN classifier's predictions to kaggle, and mention here how you arrived at its structure to get extra credit.

   Detailed Instructions:
   A SPECT (Single Proton Emission Computed Tomography) scan of the heart is a noninvasive nuclear imaging test. Doctors use SPECT images to diagnose coronary artery disease and to detect if a heart attack occurred.

   You will classify patients based on their cardiac SPECT images. Each patient will be classified into one of two categories: normal (zero) and abnormal (one). Each SPECT image was pre-processed to extract multiple features. As a result, 22 binary features were created for each patient from their SPECT image. Your task is to build classifiers based on this data, and then use it to predict if a patient is normal or not.

   Naive Bayes Classifier



SPECT image of a normal heart.

- Binary Classification: Your program is intended for binary classification (i.e., classify patients as zero or one).
- Assume both the classes have same prior.
- Add-one Smoothing: To avoid any zero-count issues, use Laplace estimates (pseudocounts of 1) when estimating all probabilities. That is,

$$P(Y = y) = \frac{\#(y) + 1}{\#(instances) + d}$$

  where #(y) denotes the number of instances having Y = y and d denotes the number of distinct values in Y [ref].
- Logs: Convert all probabilities to log probabilities to avoid underflow problems, using the natural logarithm.
- To break ties, classify as one (abnormal).

Bayesian Network Classifier

- All the points same as the Naive Bayes Classifier.
- Structure: All features depend on class variable (same as naive bayes). Along with that, both feature 8 (V8) and feature 9 (V9) are dependent on feature 16 (V16).

Skeleton Code
The code should have these functions:

- NBfit
  - Load the training data (train.csv).
  - Separate the classes and calculate the individual conditional probabilities for each feature given class.
- NBeval
  - Load the testing data (test.csv).
  - Separate the actual class labels.
  - For each sample point in test data, calculate the log conditional probability of class given the sample point.
  - Assign each sample point to the class having high probability.
  - Use the Accuracy function to evaluate the NB classifier.
- BNfit
  - Same as NBfit, but the individual conditional probabilities will change according to the BN.
- BNeval
  - Same as NBeval.