CS6730 : Assignment 1

Rohithram R | EE16B031

March 14, 2019

- 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.)

   Solution: n

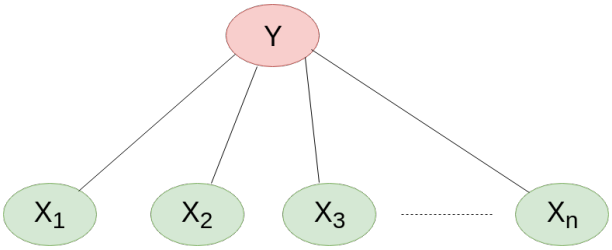
   (b) (2 points) What is the tree width of the cycle graph with n nodes? Give proof. (Hint: Answer does not depend on n.)

   Solution: 2, Tree width is always 2 for a cyclic graph. Given the set of chordal completed graphs over the given cycle graph, 3 is the largest possible no of vertices to form a clique i.e Maximum clique. Hence, Tree width can be calculated from the given definition above to be $3 - 1 = 2$.

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 $\{0, 1\}$.

   Solution:



Distribution for Naive Bayes model is given below :

$$\mathbf{P(X_1, X_2, X_3, ..., X_n|Y)} = \prod_{\mathbf{i=1}}^{\mathbf{i=n}} \mathbf{P(X_i|Y)}$$

Since there is no interdependency between the features as per Naive bayes model assumption the joint probability is given as above.

(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.

Solution: $P(X_i = 1|Y = j) =$

$$\frac{\phi(X_i = 1, Y = i) \sum_{x \in \mathbf{Val}(X_k)} \prod_{X_k \neq X_i} \phi(X_k = x, Y = i)}{\sum_{x \in \mathbf{Val}(X_k)} \prod_{X_k \neq X_i} \phi(X_k = x, Y = i) \; [\; \phi(X_i = 0, Y = i) + \phi(X_i = 1, Y = i) \;]}$$

$$P(X_i = 1|Y = j) = \frac{\phi(X_i = 1, Y = i)}{\phi(X_i = 1, Y = i) + \phi(X_i = 0, Y = i)}$$

As given above in the question,

$$P(X_i = 1|Y = j) = \left\{ \begin{array}{ll} 0.9, & \text{if } i = j \\ 0.1, & \text{if } i \neq j \end{array} \right\}$$

Substituting the values we get,

$$9\phi(X_i = 1, Y = i) = \phi(X_i = 0, Y = i)$$

and

$$\phi(X_i = 1, Y = i) = 9\phi(X_i = 0, Y = i)$$

These are the two distinct settings for factors in the given Markov Network.

(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.

Solution: After we eliminate the variable $\mathbf{Y}$, we get the complete graph over $\{\mathbf{X_1}, \mathbf{X_2}, .., \mathbf{X_n}\}$. Hence, all distribution P factorizes over this complete graph.

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"?

   Solution: False. We can infer this by a counter example. We consider the given student graph, observe the nodes D and I. Let $X = \{D\}$, $Y = \{I\}$, $Z = \emptyset$, from the structure it is d-sep$\{X, Y \mid Z\}$. But, since D and I have a common child G, after moralization D and I are connected by an edge forming a covered *v*-structure. Therefore, D and I aren't separated in H.

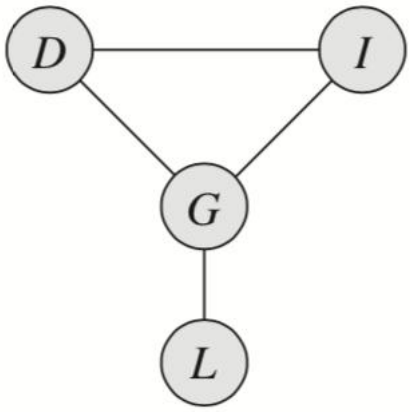
   (b) (2 points) Let $U = X \cup Y \cup Z$, and $\mathcal{G}' = \mathcal{G}[U \cup Ancs_U]$ be the induced sub-graph over U and its ancestors. Is "X and Y are d-separated given Z in $\mathcal{G}$ if and only if X and Y are d-separated given Z in $\mathcal{G}'$"?

   Solution: We observe that descendants of $U = X \cup Y \cup Z$ are barren nodes, so even if we remove them it won't affect the joint distribution of the rest of the variables $U \cup Ancs_U$. Hence, the independencies among the variables $U \cup Ancs_U$ in the graph $\mathcal{G}$ are preserved in the induced subgraph $\mathcal{G}'$. Thus, d-sep$\{X, Y \mid Z\} \in I(G) \iff$ d-sep$\{X, Y \mid Z\} \in I(G')$.

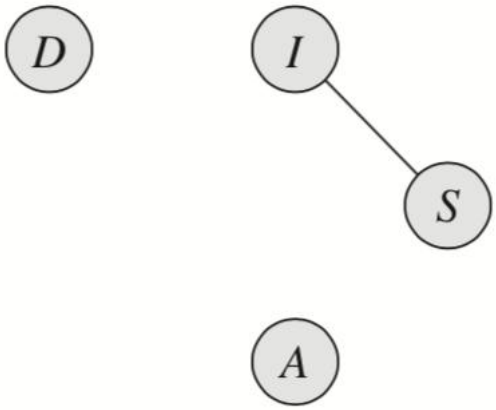
   (c) (4 points) Consider same definitions as in (b) and let $\mathcal{H}' = \mathcal{M}[\mathcal{G}']$. Is "X and Y are d-separated given Z in $\mathcal{G}'$ if and only if X and Y are separated given Z in $\mathcal{H}'$"?

   Solution:
   Yes. Let P_{U^*} be the Bayesian network where $U^* = U \cup Ancs_U$ which is defined over $\mathcal{G}'$ such that the CPD for any variable in U^* is same as in $\mathcal{G}$. All the variables used in the CPDs are in U^* as U^* is upwardly closed. We know that $(X \perp Y \mid Z) \in I(\mathcal{G}) \iff$ d-sep(X,Y | Z) and that P_{U^*} is a Gibbs distribution over H and hence P_{U^*} satisfies $(X \perp Y \mid Z)$. Note that if P factorizes over X and H is a Markov network over X and if P is a Gibbs distribution that factorizes over H, then H is an I-map for P. Hence, it's an sufficient and necessary condition.
   For example, Let's consider the student network. Let D and I be d-separated given L and X={D}, Y={I}, Z={L}. Let $U = \{D, I, L\}$, and $\mathcal{G}' = \mathcal{G}\ [U \cup Ancs_U]$. We introduce an undirected moralizing edge between D and I when moralizing the graph i.e $H = M[\mathcal{G}']$. But we see that d-sep(D , I |

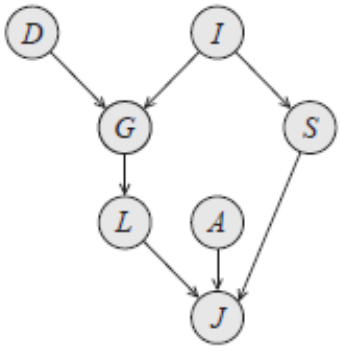
L)$\notin$ H′ . This dependence is also got from d-separation of $\mathcal{G}$ and also d-sep(D, I|L) $\notin \mathcal{G}'$.



Now X = D, Y = L, Z = S, A. Moralising the ancestral subgraph $\mathcal{G}'$ of U = X ∪ Y ∪ Z, we obtain,



It's clear that d -sep(D, I|S, A) ∈ moralised graph, $\mathcal{G}$, and hence in the subgraph $\mathcal{G}'$, Thus, we have d -sep(D, I|S, A).



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

   Solution: We derive the given formulas are Maximum Likelihood estimates of the data.

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

   $$P(V_i = 0|Y = y) = 1 - P(V_i = 1|Y = y)$$

   For Bayesian Network, additionally

   $$P(V_i = 1|V_{16} = v_{16}, Y = y) = \frac{\#(Y = y \cap V_i = 1 \cap V_{16} = v_{16}) + 1}{\#(instances Y = y \cap V_{16} = v_{16}) + d}$$

   $$P(V_i = 0|V_{16} = v_{16}, Y = y) = 1 - P(V_i = 1|V_{16} = v_{16} Y = y)$$

   where d is 2 since all the features are binary random variables.
   For evaluation when Naive Bayes used, $\mathcal{V} = [v_1, v_2, ..., v_{22}]$ be a sample in the given data for simplicity of notation.

   $$\log(P(Y = y|\mathcal{V}) = \log(P(Y = y)) + \sum_i \log(P(\mathcal{V}_i|Y = y)) - \log(P(\mathcal{V}))$$

   When Bayesian Network used,
   $\log(P(Y = y|\mathcal{V})) = \log(P(Y = y)) + \sum_{i \neq 8,9} \log(P(\mathcal{V}_i|Y = y)) + \sum_{i=8,9} \log(P(\mathcal{V}_i|V_{16} = \mathcal{V}_{16}, Y = y)) - \log(P(\mathcal{V})$

   (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.

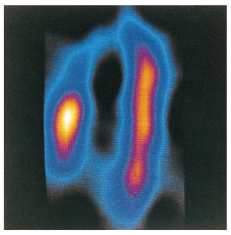
   Solution: Test Accuracy using Naive Bayes classifier is 80% . Test Accuracy using Bayesian Network is 80%
   Please find the jupyter notebook file for the code used to achieve the results given above and solution report here **https://github.com/Rohithram/CS6730_Assignments**. Not attaching here, as it doesn't fit inside box.

   (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.

   Solution: To learn the structure from the given data,

- We can find the correlation matrix of all the features.
- Take the top k correlated feature pairs, where k is hyperparameter can be found using cross validation by splitting train data.
- Since searching over the exponential no of possible BN structures blindly is not efficient, so we restrict this by using the correlation of features.
- But since we don't know the causation still, we have to randomly choose which one has to be parent among the selected pairs.
- By experimenting in this manner, we can achieve a better result than we got earlier. Due to time constraint, I couldn't implement it, but this is the idea I have in mind.

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.



SPECT image of a normal heart.

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

- 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.