Chiranjeev (P21CS007)
Vibhuti Sharma (P21AI002)
Preeti Dwivedi (M19DCS004)

Assignment - 2
Bayesian Inference

15-11-2021

1. Bayesian Network Introduction:

We did Bayesian Inference on the given data using Expectation Maximization (EM) to fill the samples having "?" with some probability. Hence, probability distributions or parameters of the given Bayesian Networks (BN) are estimated.

Bayesian network is a directed acyclic graphical representation of a set of variables and their conditional dependencies. Each variable is represented as a node in the graph and a directed edge between the nodes represents the parent-child relationship between the considered nodes.

To make inferences, we have used the given datasets containing samples of values observed for different variables.

2. Constraints, Inputs and Outputs:

2.1. Taken Inputs:

- **Samples:** Sample data can have missing values for some nodes in samples represented with "?".
- **Conditional Dependencies:** It is the array that tells which node depends on other nodes.
- **Other inputs are number of variables and values the variables can take.**
- **0 < No. of nodes < 1000** and **0 < No. of Samples < 1M**

2.2. Output:

- Output produced is the probability distribution of each variable.

3. Approach Discussion:

3.1. Code Walkthrough:

Function BAYESIAN_NETWOK-FN(Input) returns the probability distribution of each node

i. **Find samples with missing values**: Using MISSING_VALUES-FN, we find the samples of dataset which contain missing values '?' and the exact location of '?' in the given dataset.
ii. **Add samples to dataset:** Based on all the possible values that a '?' can take, we replace the sample having '?' with all possible combinations. This has been done using MAKE_COMPLETE_SAMPLE_LIST-FN.
iii. **Initialize CPTS of each node randomly**: We make a dictionary to store the CPTs of each node. We also make **initial estimations (**$\mathbf{theta_0}$**)** by randomly assign probabilities. This is done by calling MAKE_DICT_FOR_EACH_NODE_CPT-FN.
iv. **Repeat**

a. **Expectation Step of EM algorithm:** At each iteration t, complete the dataset (update the weights in dataset) based on $theta_t$.
Illustration: Let an incomplete sample, S: (1, ?, 0). If x_2 can either be 0 or 1 then to complete the data EM computes $P(x_2 \mid x1=1,x3=0 / theta_t)$ as follows:
$P(x_2=0 \mid x1=1, x3=0 / theta_t) = 0.3$
$P(x_2=1 \mid x1=1, x3=0 / theta_t) = 0.7$
EM splits 'S' into the following two **partial** data cases:

(1, ?, 0) =>(1, 0, 0) [0.3], (1 , 1, 0) [0.7]
In Maximization step, each of them is counted as (1/2) of a data case.
This has been done using E_STEP-FN.

b. Maximization Step of EM Algorithm: Obtain $theta_{t+1}$ by re-estimating parameters using the completed dataset we receive after the expectation step. Some of the cases are partial data cases. While re-estimating, partial data cases are counted according to their associated weights. The maximization step has been implemented using the following two functions:
GET_M_STEP_PROB_DEP_NODES-FN

GET_M_STEP_PROB_INDEP_NODES-FN

until EM converges (CHECK_CONVERGENCE-FN)
return the probability distribution of each node

3.2. Function Description:

All the functions that we have used in our code are described in Table 1.

Table 1

Function	Synopsis	Returns
MISSING_VALUES-FN	The function iterates over the given dataset and finds the samples which contain a missing value ('?') It also finds the exact location of the '?' in dataset.	• List: A list having '0' at the index of row with missing value in dataset and '1' otherwise. • List: A 2D list with size equal to that of the given dataset. It has '0' at the location of missing value, otherwise 1.
NUMBER_OF_ENTRIES_IN_COMPLETE_SAMPLE_LIST-FN	The function calculates the number of samples in the dataset if we add all the possible combinations of the '?' to the original dataset.	• Int: Number of samples
ADD_WEIGHT_COL_IN_SAMPLE_LIST-FN	The function attaches the weights and identifiers to every sample in the given dataset based on whether the sample has a missing value or not.	• List: Updated dataset (with weights)
MAKE_COMPLETE_SAMPLE_LIST-FN	The function replaces the '?'s with all their possible values based on the list of values that the variable, whose value is missing, can take. After replacement, it adds all the newly generated samples to the given dataset and removes the rows with '?'.	• List Updated dataset (with all possible samples after replacing '?')
FIND_PARENT_NODES-FN	Using the given dependency matrix, the function stores the parents of every node in a list (if any). It also stores the number of parents a node has in a separate list.	• List: A nested list which contains the list of parents of every node. • List: List containing the number of parents for each node

GENERATE_RANDOM_PROBAB-FN	This is a utility function for the function described in next row. The function generates a random probability distribution for each variable.	• List: Random probability distribution for all the values a variable/node can tak
CAL_NODE_PROB_DITTRI-FN	This is a utility function for the function described next. It returns the random probability for each node based on their dependencies.	• Dictionary: Random probability distribution for each node based on their dependencies.
MAKE_DICT_FOR_EACH_NODE_CPT-FN	The function makes a nested dictionary for each node/variable in the given BN. Using the functions mentioned above, it initializes the CPTs for each node randomly (initial estimations, $theta_0$).	• Dictionary: CPT for each node
GET_PROB-FN	This is a utility function for Expectation Step of EM Algorithm. It computes the probabilities based on $theta_x$.	• Float: Probability of a particular sample.
ROW_WITH_MISSING-FN	This is a utility function for Expectation Step of EM Algorithm. It takes a sample and tells the variable/node for which the data was missing in that sample.	• Int: The index of missing data in a sample
E_STEP-FN	The function implements the Expectation step i.e., at every iteration, t , it completes the dataset based on $theta_t$.	• List: Updated dataset (based on CPTs estimated in previous iteration)
GET_M_STEP_PROB_DEP_NODES-FN	The function implements the Maximization step of EM Algorithm for dependent nodes/variables (having parents).	• Float: Probability calculated based completed dataset (dataset completed using re-estimation in E-Step)
GET_M_STEP_PROB_INDEP_NODES-FN	The function implements the Maximization step of EM Algorithm for independent nodes/variables (having no parents).	• Float: Probability calculated based completed dataset (dataset completed using re-estimation in E-Step)
CHECK_CONVERGENCE-FN	The functions tells when to terminate the EM Algorithm by comparing the CPTs of current step with the previous step.	• Int: If no change in CPTs then returns 0 (therefore, an indication to terminate EM) otherwise it returns 1.
BAYESIAN_NETWOK-FN	The function puts it all together. By using the above-mentioned functions, it first makes random estimations. Then, implements EM algorithm and finally, after the EM converges, returns the CPTs for each node.	• Dictionary: Final CPT for each node/variable in the given BN.

Some points about code:

- The code is represented in the way as we write probabilities in visual manner so as to look human interpretable format for calculating probabilities.