Project 1 Discussion

Task 1:

- *Evaluate pairwise correlations and independences that exist in the data. Note that we can determine whether xi and xj are independent by testing if p(xi, xj) = p(xi)p(xj), where the joint probability between a pair of variables can be determined from the tables as p(xi, xj) = p(xi|xj)p(xj)*

Task 1:

$$\rho = \frac{\mathrm{P}(A \cap B) - \mathrm{P}(A)\,\mathrm{P}(B)}{\sqrt{\mathrm{P}(A)\,(1-\mathrm{P}(A))\,\mathrm{P}(B)\,(1-\mathrm{P}(B))}}$$

$$= \frac{(\mathrm{P}(A \mid B) - 1)\,\mathrm{P}(B)}{\sqrt{\mathrm{P}(A)\,(1-\mathrm{P}(A))\,\mathrm{P}(B)\,(1-\mathrm{P}(B))}}$$

For categorical variables with only two classes, we can directly get correlation from conditional probability with the above equation. This unfortunately, can not be applied to multi-class categorical variables

Task 1:

For the multi-categorical variables in our project, we can measure the closeness of P(x,y) and P(x)P(y) by:

(1) Calculate the cross entropy between P(x, y) and P(x)P(y)

(2) Or approximately, by calculating $\sum abs\left(\left(P(x,y) - P(x)P(y)\right)\right)$

(3) Or further approximately, by calculating $\sum abs\left(\left(P(x|y) - P(x)\right)\right)$ for some values of y with large P(y) (P(y) at least accumulate to 0.7), and then $\sum abs(P(y|x) - P(y))$ for some values of x with large P(x) (P(x) at least accumulate to 0.7)

Get the CPD table from project description, then write a few python function to do the work

Task 2:

- *Construct a Bayesian network with the fewest number of edges that maximizes the likelihood. One approach is to use the results of the first task and start drawing links between the most correlated pairs of variables. We can construct several Bayesian networks and obtain a score for each of them. One way of scoring is to determine the likelihood the network assigns to samples generated (using ancestral sampling). Note that the dataset changes for each model. Based on your best model, describe what a high probability th looks like (in words as well as in image form). Describe some low probability th as well.*

Task 2:

(1) Set a threshold on the previous calculated result to determine if two variables are independent or not (Note the range of correlation is between 0 and 1). We assume independence for pairs of variables not appearing in the CPD table.

(2) Construct several Directed Acyclic Graphs (DAG) based on (1) result, with a directed link between two correlated variables. If you happens to construct a cyclic graph instead of DAG, then you may want to try out the Junction tree functions and loop belief propagation functions in pgm.py.

Task 2 (cont.):

(3) We use likelihood to compare these different models. Calculate likelihood needs data. Because we don't have the concrete dataset for this project, we generate the data ourselves.

(4) Start from the parent, do ancestral sampling in the order of topological sort. Sample out the parent first, and then its children. When all six variables are samples out, then we got one data. Create a dataset with size at least 1,000 for each model (note different model generate different datasets).

Task 2 (cont.):

(5) Compare the different models either based on maximum likelihood or K2 score as you wish. Get the best model.

(6) Describe what a high probability th looks like. You can look into the dataset generated by this model and see what patterns appears most.

(7) Describe some low probability th as well. Similarly, you can look into the dataset generated by this model and see what patterns appears less.

Task 3:

- *Convert your best Bayesian network into a Markov network using moralization. Compare inferences using Bayesian network and the Markov network, in terms of computation time and accuracy.*

Task 3:

Do this task after you have completed task 4, as this task is largely open ended. Dive into the source code of **BayesianModel.to_markov()** for details and utilize the function on this task**.** You can try this for both the “th” task with the generated dataset and the “and” task with the original dataset AFTER the completion of Task 4 (which gives you the Bayesian model to convert)..

Task 4:

- *Use the "and" image dataset to construct a Bayesian network and evaluate the goodness score (likelihood of a dataset) of several Bayesian networks.*

  For this task, we have the dataset available. PGM.estimators class can be used to get the CPDs from the data. Follow materials from last lecture to search for the best Bayesian network.