Lista 04 - PGM2017

Renato Assunção - DCC, UFMG

Setembro 2017

1. Considerando o DAG da Figura 1, responda V ou F para as afirmações abaixo:

   - d-sep$_G(x_1, x_3 \mid \varnothing)$. Isto é, os nós 1 e 3 são d-separado sem condicionar em nenhuma outra variável? O símbolo $\varnothing$ representa o conjunto vazio.
   - d-sep$_G(x_1, x_3 \mid \{x_6, x_7\})$.
   - d-sep$_G(x_1, x_3 \mid \{x_2, x_6, x_7\})$.
   - d-sep$_G(x_6, x_7 \mid \varnothing)$.
   - d-sep$_G(x_6, x_7 \mid x_2)$.
   - d-sep$_G(x_6, x_7 \mid x_4)$.

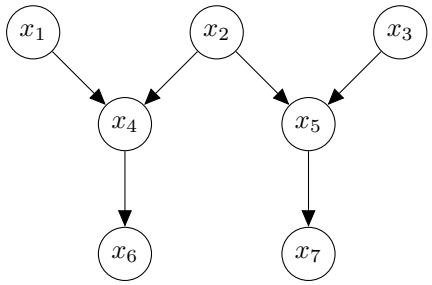

Figura 1: DAG with seven random variables.

2. Considerando o DAG da Figura 2, responda V ou F para as afirmações abaixo:

   - $(x_1 \perp\!\!\!\perp_p x_2 \mid \{x_4, x_6\})$
   - $(x_1 \perp\!\!\!\perp_p x_2 \mid \varnothing)$
   - $(x_1 \perp\!\!\!\perp_p x_2 \mid x_3)$
   - $(x_4 \perp\!\!\!\perp_p x_5 \mid x_3)$
   - $(x_4 \perp\!\!\!\perp_p x_5 \mid \varnothing)$
   - $(x_4 \perp\!\!\!\perp_p x_5 \mid \{x_1, x_2\})$
   - $(x_4 \perp\!\!\!\perp_p \{x_2, x_5\} \mid x_3)$
   - $(x_4 \perp\!\!\!\perp_p \{x_5, x_7\} \mid x_3)$

3. O DAG da Figura ?? ilustra dois modelos importantes que estudaremos mais a frente, o modelo de cadeia de Markov escondida e o modelo de filtro de Kalman. Considerando este DAG, responda V ou F para as afirmações abaixo:

   ??

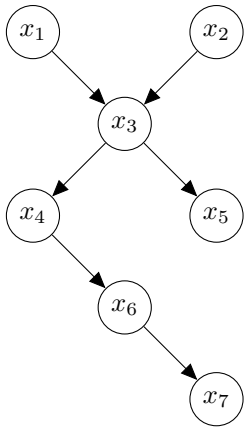

Figura 2: Another DAG with seven random variables.

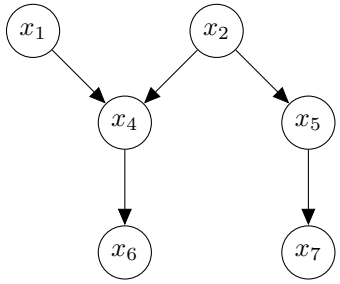

Figura 3: DAG with six random variables.

4. Obtenha o I-map do DAG na Figura 3.

5. Este eercício é adaptado de um homework de um curso de Modelos Gráficos dado por Andrew McCallum no departamento de ciência da computação na Universidade de Massachussetts. We will provide you with medical domain data: a joint probability distribution of symptons, conditions and diseases. Certain diseases are more likely than others given certain symptons, and a model such as the graphical model you are going to build is a simple version of one that could be used to help doctors making a diagnosis. All variables are binary ans they are as follows:

   - (0) Sum: IsSummer true if it is the summer season, false otherwise.
   - (1) Flu: HasFlu true if the patient has the flu.
   - (2) Poi: HasFoodPoisoning true if the patient has food poisoning.
   - (3) Hay: HasHayFever true if patient has hay fever (rinite alérigica, alergia a pólen).
   - (4) Pne: HasPneumonia true if the patient has pneumonia.
   - (5) Res: HasRespiratoryProblems true if the patient has problems in the respiratory system.
   - (6) Gas: HasGastricProblems true if the patient has problems in the gastro-intestinal system.
   - (7) Ras: HasRash true if the patient has a skin rash.
   - (8) Cou: Coughs true if the patient has a cough.
   - (9) Fat: IsFatigued true if the patient is tired and fatigued.
   - (10) Vom: Vomits true if the patient has vomited.
   - (11) Fev: HasFever true if the patient has a high fever.

The ground-truth joint probability distribution consists of twelve binary random variables and contains 2^12 possible configurations (numbered 0 to 4095), which is small enough that you can enumerate them exhaustively. The archive contains two files:

- **joint.dat:** The true joint probability distribution over the twelve binary variables. Since each variable is binary, we can represent a full variable assignment as a bitstring. This file lists all 2^12 assignments (one in each line) as pairs "Integer Probability" where "Integer" is an integer encoding of the bitstring. Specifically, assuming false=0 and true=1, an assignment to all variables results in a 12-bit binary number (with the index of the variables shown in parantheses above) which is converted to a decimal number.

  For example:

  - assignment 0 represents all variables are false (all are zero in the binary representation),
  - 1 represents only IsSummer is true (000 000 000 001 in the binary representation),
  - 2 represents only HasFlu is true (000 000 000 010 in the binary representation),
  - 3 represents IsSummer AND HasFlu are both true, and only these two variables (000 000 000 011 in the binary representation),
  - 4 represents only HasFoodPoisoning is true (000 000 000 100 in the binary representation),
  - 5 represents only HasFoodPoisoning AND IsSummer are true (000 000 000 101 in the binary representation),
  - 6 represents only HasFoodPoisoning AND HasFlu AND IsSummer are true (000 000 000 110 in the binary representation),

  and so on.
- **dataset.dat:** The dataset consists of samples from the above probability distribution. Each line of the file contains a complete assignment to all the variables, encoded as an integer (as described above).

I am suggesting an initial graphical model after talking to some medical doctors. This should not be seen as an unquestionable, indisputable model. Indeed, the medical doctors I consulted are not specialists in these diseases and they did not have much time. They themselves were reluctant about some of the arrows they suggested in and out of the DAG. Anyway, Figure 4 shows what this preliminary DAG is.

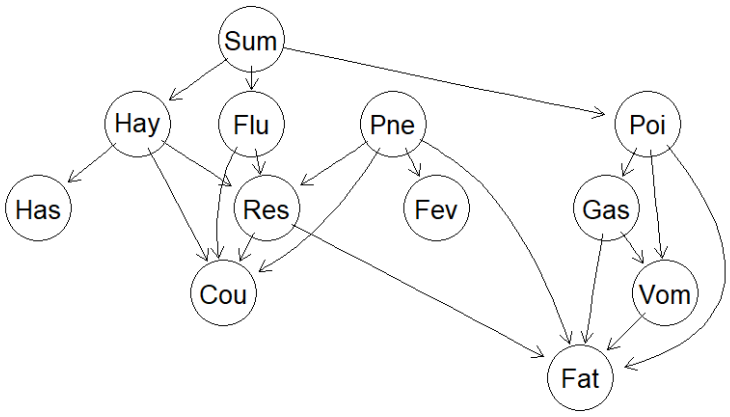

Figura 4: BN for the pneumonia-hay-flu diagnostic BN.

What you must turn in:

- Create a DAG with these variables and edges as in Figure 4.
- Estimating parameters: Using the mle method with the data in **dataset.dat:**, obtain the CPTs for the DAG in 4.

- Model Accuracy: Measure the similarity of your model to the true joint probability distribution (i.e., the probabilities in **joint.dat**). That is, for each assignment, how similar are the probabilities returned by your model to the true probability distribution. To keep things simple, you can compare the distributions based on their L1-distance. That is, for each assignment ai to all the variables, obtain p(ai) from the true joint distribution ((i+1)th row in joint.dat) and p(ai) using your model. The distance is defined as —p(a0)-p(a0)— + —p(a1)-p(a1)— + ... + —p(a4095)-p(a4095)—. An alternative distance measure more appropriate to probability distributions is KL-divergence. If you know what that is, and want to use it, you can evaluate using KL-divergence also.
- Querying: Use the graphical model above to answer some queries. A query consists of observed variables (for which we have an assignment), and query variables that over which we want the distribution. The remaining variables need to be marginalized (by summing them out). Since the domain is small you can implement this conditioning and marginalizing process by exhaustively enumerating all assignments (note that only assignments that are consistent with the observed values should be taken into account). Compare the results of these queries on your model to results obtained from using the true joint probability distribution. Try to think of some interesting queries that will demonstrate causal reasoning, evidential reasoning, and inter-causal reasoning. To get you started, here are some examples of queries to consider (but also create new ones of your own design):
  - What is the probability a patient has flu given they are coughing and have a high fever? (Observed Variables: HasFever=true, Coughs=true; Query Variable: HasFlu)
  - What is the probability distribution over the symptoms (HasRash, Coughs, IsFatigued, Vomits, and HasFever) given the patient has pneumonia?
    enditemize
- Varying the Structure: Experiment with multiple different graphical structures. Try adding or removing edges, or even changing the structure completely. In particular, the medical doctors were unsure about the edge HasRespiratoryProblems to Coughs. Compare these different structures to the true distribution table (using the L1-distance or KL metric). How close can you get? Do you think you can find the structure that we used to generate P and the data? As a baseline model, consider one that assumes all the variables are independent.
- Structure Learning: Instead of using intuition or the conditional independencies from the joint distribution, use search to discover the appropriate structure from data. The search technique can be greedy by starting with all variables being completely independent, and each edge added greedily if adding it results in an increase in the test set likelihood. Note that this will involve parameter estimation using train data for each edge added. To make search more efficient, assume that no variables has more than three parents.