MAC6916 PROBABILISTIC GRAPHICAL MODELS
LECTURE 11: CONSTRAINT-BASED STRUCTURE LEARNING

DENIS D. MAUÁ

1. Introduction

We now tackle the problem of inferring the structure of a Bayesian network from data. We assume categorical variables (although the apporach presented here can easily be extended to continuous variables). There are mainly two approaches to this problem. The first approach, which we study in this lecture, is connected to the independence representation view of Bayesian networks. In this approach, termed *constrained-based structure learning*, a independence oracle is assumed available (which tests for conditional independence of variables), and a Markov equivalent structure is recovered making a number of calls to that oracle. The second approach, termed *score-based structure learning*, takes the distribution view of Bayesian networks, and attempts at finding a structure which fits the empirical distribution in the data. This is usually posed and solved as a combinatorial problem over the space of DAGs guided by a polynomial-time computable scoring rule.

constrained-based structure learning

score-based structure learning

The number of DAG structures over n nodes satisfies Robinson's formula [4]:

$$\#\mathrm{DAG}(n) = \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} \#\mathrm{DAG}(n-i)\,.$$

This number grows extremely quickly (superexponential). For instance:

n	$\#\mathrm{DAG}(n)$
1	1
2	3
3	25
⋮	⋮
8	783 702 329 343
9	1 213 442 454 842 881
10	4 175 098 976 430 598 143

This number can be reduced by noting that several structures are Markov equivalent and hence indistinguishable from data (alone) by conditional independence testing. While there is not a known formula for the number of equivalence classes, it is lower bounded by the number of equivalence classess of size 1 (i.e., which have

Date: May 10, 2016.

a single DAG), which is given by the following formula [1].

$$\#\text{EQ1}(n) = \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} (2^{n-i} - (n-i))^i \#\text{EQ1}(n-i)\,.$$

Note that similarity to the counting of DAGs, and the fact that $(2^{n-i} - (n-i))^i$ approaches $2^{i(n-i)}$ as $n \to \infty$. This implies that the number of equivalence classes is still intractably large. Gillispse and Erlman [2] have computed the number of equivalence classes and their relative size for up to $n = 10$ and found that:

n	#EQUIV(n)	#EQUIV(n)/#DAG(n)
1	1	1.0
2	2	0.667
3	11	0.440
⋮	⋮	
8	212 133 402 500	0.271
9	326 266 056 291 213	0.269
10	1 118 902 054 495 975 141	0.268

This number should suffice to convince anyone that any simplistic approach (such as listing DAGs or equivalence classes) to structure learning is infeasible even for small domains.

2. The PC Algorithm

By definition, any two Markov equivalent structures induce the same answers from an independence oracle, so that no algorithm that is based purely on this oracle can prefer one structure over the other. It seems reasonable then to learn an *equivalence class* instead of a single structure. Recall that an equivalence class contains all Markov equivalent structures. And a *partially directed acyclic graph* of an equivalence class is a graph containing both directed and undirected graphs, and such that an edge appears directed if and only if the corresponding arc appears in all the structures in the equivalence class. The PC (Peter & Clark) algorithm learns a partially directed acyclic graph representing the (in)dependences in the data using only *polynomially many calls* to the oracle, each call involving only a *bounded number of variables* [3]. To meet these criteria, the following assumptions are made:

equivalence class

partially directed acyclic graph

(1) The data has been generated by a Bayesian network whose structure has bounded in-degree d.
(2) The independence oracle is an exact test of the conditional independences in the generating network
(3) D-separation is complete in the originating Bayesian network (i.e., two sets of variables are d-separated by a third set iff they are conditionally independent).

All of these assumptions are unrealistic: data is seldom generated by a stationary Bayesian network distribution *faithful* to its structure (meaning all graphical dependences are verified in the distribution), and conditional independence tests are imperfect.

faithful

The algorithm has the following steps:

(1) Test the conditional independence between each pair of variables in order to derive the conditional dependences and independences.
(2) Identify the graph skeleton (= undirected graph) induced by those relations.
(3) Identify convergent connections (orient $X \to Z \leftarrow Y$ if there is $X - Z - Y$ in the skeleton and no $X - Y$).
(4) Identify derived directions.

The output of the algorithm is a PDAG representing the class of Markov equivalent networks. Recall the following properties of independence in Bayesian networks (given as an exercise in Lecture 4):

Lemma 1. *Prove that the following statements are equivalent for two nodes X and Y in an acyclic directed graph G:*

(i) X *and* Y *are adjacent.*
(ii) there is no set $\mathcal{Z} \subseteq \mathcal{V} - \{X, Y\}$ *that d-separates* X *and* Y*.*
(iii) X *and* Y *are* not *d-separated by* $an(X) \cup an(Y)$*.*
(iv) X *and* Y *are* not *d-separated by* $pa(X) \cup pa(Y)$*.*

According to Properties (i), (ii) and (iv), if X and Y are *not* adjacent in the originating network, then either $X \perp\!\!\!\perp Y \mid \mathrm{pa}(X)$ or $X \perp\!\!\!\perp Y \mid \mathrm{pa}(Y)$ will be answered affirmatively by the oracle. Conversely, if the variables *are* adjacent then any query $X \perp\!\!\!\perp Y \mid \mathcal{Z}$ with $X, Y \notin \mathcal{Z}$ will be responded negatively. Since we assumed $|\mathrm{pa}(X)| \leq d$ for any X, then X and Y are adjacent if and only if we cannot find a set $\mathcal{Z}$ (not containing X and Y) of size at most d such that $X \perp\!\!\!\perp Y \mid \mathcal{Z}$. Call any such set $\mathcal{Z}$ a *witness set* (for X and Y). Note that there are $O(2^d)$ witness sets for a pair of variables, and this is considered constant w.r.t. to the input size (as d is considered constant). Moreover, since a witness set is a parent set of one of the endpoints, we need only to consider subsets of neighbors of these variables (i.e., we can ignore variables that have been deemed independent of both endpoints by the oracle). Building on these observations, the algorithm to identify the skeleton of the PDAG proceeds as follows.

witness set

(1) Start with a complete graph H over the variables.
(2) For each edge $X - Y$, consider all possible witness sets $\mathcal{Z}$ (from smaller to larger)
(3) For each such set $\mathcal{Z}$, query $X \perp\!\!\!\perp Y \mid \mathcal{Z}$; if the answer is affirmative, remove $X - Y$ from H and proceed to the next edge (mark $\mathcal{Z}$ as the witness of that edge).

Note that in Step 3, we only need to consider subsets of either $\mathrm{ne}(X)$ or $\mathrm{ne}(Y)$ in the current H.

The skeleton identification procedure examines all the n^2 pair of variables, and for each performs $O(n^d)$ calls to the oracle; hence, it consumes $O(n^{d+2})$ time. If each oracle call takes polynomial time in the number of variables and in the data set size, the overall time complexity is also polynomial (since d is assumed constant). Recall that an equivalence class is characterized by a skeleton and a set of immoralities. Thus, it remains to devise a procedure that finds the immoralities. So consider a skeleton S and a list of witness sets for each absent edge in S. A candidate immorality is a triple $X - Z - Y$ such that X and Y are not adjacent. Consider any such triple. If it is *not* and immorality then in the originating graph we will have either one of the serial connections

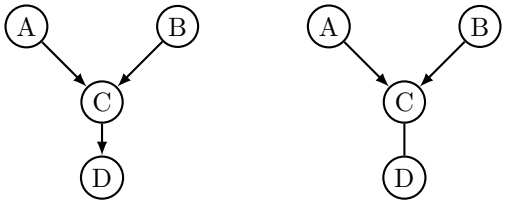
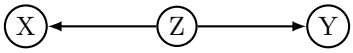


FIGURE 1. A Bayesian network structure which is also the partially directed graph of its equivalence class (left), and a possible output of the procedure described (right).

or the divergent connection



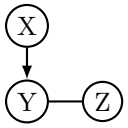
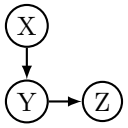
In any of three cases, Z d-separates X and Y and X and Y are d-connected if Z is not given. Since the composition property applies (by the faithfulness assumption), this implies that $X \perp\!\!\!\perp Y \mid \mathcal{Z}$ if and only if $\mathcal{Z}$ contains Z. On the other hand, if the triple is indeed an immorality (in the originating graph) then $X \not\perp\!\!\!\perp Y \mid \mathcal{Z}$ for any set $\mathcal{Z}$ containing Z. So if $X - Z - Y$ is an immorality, then Z appears in all witness sets of $X - Y$; and if $X - Z - Y$ is not an immorality then Z does not appear in any witness set of $X - Y$. This leads to a simple procedure for determining immoralities:

(1) For each potential immorality $X - Z - Y$, verify if Z is in the witness set associated with $X - Y$
(2) If Z is *not* in the witness set, then orient $X \to Z \leftarrow Y$ in S; otherwise move on to the next potential immorality.

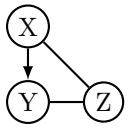
The outcome of the procedure above is not yet the partially directed acyclic graph representing the Markov equivalence class defined by the oracle, because there might be arcs $X \to Y$ which appear in every structure in the class and yet are undirected in S. This occurs when a different orientation of that edge in S would either create a cycle or an immorality. For example, the structure in Figure 1 has a single immorality. This entails a unique orientation for the edge $C - D$, and excludes the graph on the right from representing the Markov equivalence class of that graph.

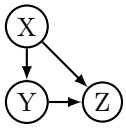
The final step of the PC algorithm is thus the orientation of arcs from the partially directed skeleton S (obtained from the previous step). This is achieved by recursively applying any of the following rules (in any order) to S until convergence. Since these rules only orient edges, this step necessarily terminates in $O(n^2)$ steps.

Prohibited immorality:

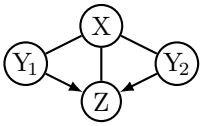
If  then 

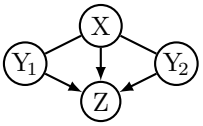
Acyclicity:

If 

then 

Quartet:

If 

then 

The correctness of the first two rules is straightforward. For the third rule, we can show correctness by contradiction: if we assume that $X - Y_1$ and $X - Y_2$ are undirected in the equivalence class, then any other orientation of $X - Z$ would either create a cycle or an immorality. These rules take at most polynomial time, and hence the overall algorithm takes polynomial time in the number of variables.

The overall procedure can be shown to be sound and complete. That is, the PC algorithm returns the partially directed acyclic graph of the Markov equivalence class represented by the oracle. Moreover it thus so in polynomial time (considering the boundedness assumption on the in-degree).

3. Statistical Independence Testing

To complete the PC algorithm, we assumed need to discuss how to obtain an independence oracle $X \perp\!\!\!\perp Y \mid \mathcal{Z}$, which depends on data set $\mathcal{D}$. Fix $X, Y, \mathcal{Z}$, so that the oracle is a function of only $\mathcal{D}$; call $I(\mathcal{D})$ this function, that is, I is a function that answers yes or no given a data set $\mathcal{D}$. Assume that $\mathcal{D}$ was generated by a distribution $p(X, Y)$. Ideally, we would like to have

$$I(\mathcal{D}) = \text{yes} \Leftrightarrow p(X, Y|\mathcal{Z}) = p(X|\mathcal{Z})p(Y|\mathcal{Z})\,.$$

In practice, however, we do not have access to $p(X, Y)$, and the oracle will not be perfectly reliable. The mistakes made by the oracle can be classifier into two types

Type I: or false positive, when the oracle *rejects* a true independence statement.

Type II: or false negative, when the oracle *accepts* a false independence statement.

The type I error rate is formalized as:

$$\alpha = \mathbb{P}(\{\mathcal{D} : I(\mathcal{D}) = \text{no}\}|X \perp\!\!\!\perp Y \mid \mathcal{Z})\,.$$

This probability value is known as the *significance level*. Note that by *controlling* the type I error we, in principle, make no commitment to type II errors (as they are probabilities conditioned on a disjoint event). In terms of an independence oracle used to learn the structure of a Bayesian network, the smaller the value of α the sparser is the induced graph. significance level

Data-based oracles are often obtained by *thresholding* a data statistic, that is, by returning yes if and only if

$$f(\mathcal{D}) \leq t\,,$$

where f is a real-valued function and t is a real-valued. The probability of observing a value of $f(\mathcal{D}) > t$ under the (null) hypothesis of independence is known as the *p-value* of the test: p-value

$$\text{p-value}(t) = \mathbb{P}(\{\mathcal{D} : f(\mathcal{D}) > t\} | X \perp\!\!\!\perp Y \mid \mathcal{Z}) .$$

A common statistic for testing independence of categorical variables is the Chi-Square statistic:

$$\chi^2(\mathcal{D}) = \sum_{\nu \sim \mathcal{Z}} \sum_{x,y} \frac{(N[X=x, Y=y, \nu] - N[X=x, \nu] N[Y=y, \nu]/N[\nu])^2}{N[X=x, \nu] N[Y=y, \nu]/N[\nu]} .$$

It is the mean squared error between the *empirical distribution* of the joint distribution of the two variables, and the joint distribution of two independent variables (i.e., $p(X, Y, \mathcal{Z}) = p(X|\mathcal{Z})p(Y|\mathcal{Z})p(\mathcal{Z})$). When X and Y are (conditionally) independent under the empirical distribution, the χ-statistic returns 0; otherwise it returns a positive value proportional to the discrepancy between the observed and the hypothesized distributions. The Chi-Square statistic follows *approximately* a Chi-Square distribution with $(|\Omega_X| - 1)(|\Omega_Y| - 1) \prod_{Z \in \mathcal{Z}} |\Omega_Z|$ degrees of freedom under the (null) hypothesis of independence of variables, and of i.i.d. data. The approximation is fairly accurate when the data set is moderately large (say, greater than 30 records).

Typically, the significance level is set to 0.05, so that an idealized procedure would make one false rejection in every 20 tests. Of course, the several approximations made usually move this number away from the ideal. A more serious issue is the large number of tests, a phenomenon called *multiple hypothesis testing*: the type I error probability increases as the number of independence tests increases. Controlling for this type of error by a decrease of the significance level often leads to an increase of type II errors. This is particularly troublesome when the data set size is small compared to the maximum (assumed) number of parents d. In practice, constraint-based structure learning is most often used to provide a first rough estimate of the structure, that is then refined by a score-based structure learning procedure (which is the subject of our next lecture).

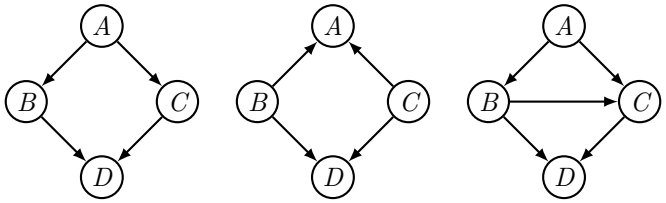
multiple hypothesis testing

4. Reading

No recommended reading this week.

5. Exercises

Exercise 1. *Learn a PDAG using d-separation as oracle in each of the following structures.*



Exercise 2. *Answer the following questions:*

(i) Give an example where the PC algorithm reconstructs the wrong structure due to the presence of a single wrong answer of the oracle.

(ii) Give an example where the algorithm reconstructs the correct skeleton but makes a single mistake when extracting the immoralities (and hence learns the wrong structure).

6. Assignment

No assignment this week.

References

[1] B. Steinsky. Enumeration of labeled chain graphs and labeled essential directed acyclic graphs. Discrete Mathematics, volume 270, pp. 267–278.

[2] S. Gillispie and M. Perlman. Enumerating markov equivalence classes of acyclic digraph models. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, pp. 171–177, 2001.

[3] P. Spirtes and C. Meek. Learning Bayesian Networks with Discrete Variables from Data. In *Proceedings of the 1st International Conference on Knowledge Discovering and Data Mining*, pp. 294–300, 1995.

[4] R.W. Robinson. Counting labeled acyclic digraphs. New Directions in the Theory of Graphs, Academic Press, pp. 239–273, 1973.

[5] S. Gillispie and M.D. Perlman. The size distribution for Markov equivalence classes of acyclic digraph models. Artificial Intelligence, volume 141, pp. 137–155.