Bayesian Networks - Passerini

Mattia Carolo - @Carolino96

Bayesian Networks (BN)

Are probabilistic graphical models used in order to represent probability distributions across multiple related variables. This type of model permits to truly see what are the relations between variables from a qualitative aspect allowing to: - visualize the structure of the model in a intuitive way - discover propterties of the model by inspecting the graph (e.g. conditional independencies) - express complex computations for inference and learning in terms of graphical manipulations - represent multiple probability distribution without worrying about their quantitative part

Semantics

A BN structure (G) is a *directed graphical model* in which each node represents a random variable x_i. In this graph every edge represents a direct dependancy relationship between the two variables.

This structure encodes a set **local** indipendencies :

$$I_l(G) = \{\forall i x_i \perp \text{NonDescendatns}_{x_i} | \text{Parents}_{x_i}\}$$

where it states that x_i is independent ($\perp$) of it's non descendants given its parent. This separates all the parent nodes from the previous set of total nodes.

Graph and distributions

Let p be a joint distribution over variables X and in this let $I(p)$ the set of independencies held in p. With those two defined, G is an *indipendency map* (I-map) for p if p satisfies the local indipendencies in G:

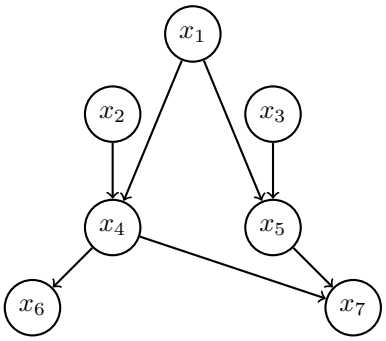
$$I_l(G) \subseteq I(p)$$

This means that if the graph has ceratain indipendences encoded than these should hold in the distribution we are working with.

With these relationhips we can decompose the probabilities associated to the variables according to the graph and this process is called **factorizing** where p factorizes according to G if:

$$p(x_1, ..., x_m) = \prod_{i=1}^{m} p(x_i | Pa_{x_i})$$

where if the first term (the probability itself) can be written as a product of the probability of a node given his parents.



for example the probability withput factorization of the p of the graph above can be written as $p(x_1, ..., x_7)$ to $p(x_1), p(x_2), p(x_3)p(x_4|x_1, x_2, x_3), p(x_5|x_1, x_3), p(x_6|x_4), p(x_7|x_4, x_5)$ where we can see the definitions of the node and his parents.

We say that the distribution p factorizes according to G if the joint probability can be decomposed this way.

> **NB**
>
> If G is an I-map for p, then p factorizes according to G (valid both ways)

Definition

> *A **Bayesian Network** is a pair (G, p) where p factorizes over G and it is represented as a set of conditional probability distributions (cpd) associated with the nodes of G*

Conditional independence

Two variables a, b are independent (written as $a \perp b|0$) if the joint prbability $p(a, b)$ can be written as $p(a)p(b)$ while they are conditionally independent given c (written as $a \perp b|c$) if

$$p(a, b|c) = p(a|c)p(b|c)$$

These assumptions can be verified by repeated applications of sum and product rules

add rules

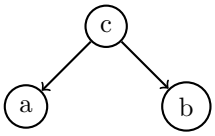
The vantage of having a craphical model is that we can verify directly those assumption through the use of the **d-separation** criterion

D-Separation

There are different structures that can describe the independency of the nodes. These are:

Tail-to-tail

In this type of structure the c node is connected to the other nodes only with tails (as we can see on the graph below)



This type of connection models the **joint distribution** where from a quantitative aspect it can be formalized as:

$$p(a, b, c) = p(a|c)p(b|c)p(c)$$

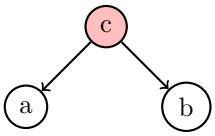
With this structure we can easily find that the pairs a, c and b, c are not independent since they are connected. For a, b if they are independent than $p(a, b)$ should be equal to $p(a)p(b)$. Still we know for the sum property that we can rewrite $p(a, b)$ as

$$p(a, b) = \sum_c p(a|c)p(b|c)p(c)$$

which is different from the formalization we want so a and b are not independent (written as $a \not\perp\!\!\!\perp b| \not\emptyset$)

> **N.B:** Important to notice that $p(a|c)p(b|c)p(c)$ which formalizes the joint distribution is the result used to infer the dependency

Another case we need to consider is "What if c is observed given that we already know that a and b are dependents?"



When a node is filled with a color we mean that the node is given as we can see on the graph above. Now we eant to formalize if a and b can be independent given c so now we want to reach this formalization

$$p(a, b|c) = p(a|c)p(b|c)$$

Through the chain rule method tho we know that $p(a, b, c)$ can be rewritten as $p(a, b|c)p(c)$ and since we are interested on a and b given c we can write that

$$p(a, b|c) = \frac{p(a, b, c)}{p(c)}$$

where applying the joint distribution formula to $p(a, b, c)$ we get that

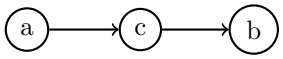
$$p(a, b|c) = \frac{p(a|c)p(b|c)\ \cancel{p(c)}}{\cancel{p(c)}} = p(a|c)p(b|c)$$

so a and b are conditionally independt so this means that if the causes are known than the different effects are independent. This is convenient even for case of studies where if certain sources are not present than we can cut some nodes which are not related to ours thanks to this property.

In the end with a Tail-to-tail connection a and b are independent only if c is given

Head to tail

Just to made it clear the local structures are named after the junctions on the considered node (c in our case). The **head-to-tail** strucure presents itself as



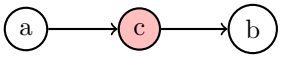
As we can see the interested node c is connected to a through the tail and with b with the head henceforth **head-to-tail**. The joint distibution will be still

$$p(a, b, c) = p(a|c)p(b|c)p(c)$$

Now we look if a and b are independent but already by looking at the graph we can infer that there is a direct dependability (a, c) and (b, c) so they are not independent and with a formalization

$$p(a, b) = p(a) \sum_c p(b|c)p(c|a) \neq p(a)p(b)$$

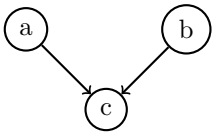
but what if c is given ?



In that case if we are given the direct causes we don't care whether they happened so we can cast out a part of the structure

$$p(a, b|c) = \frac{p(b|c)p(a|c)p(c)}{p(c)} = p(b|c)p(a|c)$$

Head-to-head



This is called **Head-to-head** structure because c is both child to a and b so now the joint distribution decomposes as

$$p(a, b, c) = p(c|a, b)p(a)p(b)$$

and if we formalize $p(a, b)$ we get that

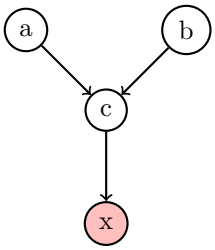
$$p(a, b) = \sum_c p(c|a, b)p(a)p(b) = p(a)p(b)$$

but looking over a and b only we can take them out and doing just the sum which will be corresponding to one (since we have all the known domain) which will leave us with just $p(a)p(b)$ so a and b **are independent**.

Instead if c is given we can treat it, for example, as the result state of some causes. As soon as we know one the related causes to our state we can infer that the others are not related to that or there is a low possibility of happening. This means that a and b are not related and here is the formalization

$$p(a, b|c) = \frac{p(c|a, b)p(a)p(b)}{p(c)} \neq p(b|c)p(a|c)$$

General head-to-head



If we consider a node x which is a descendant from c with any path. If we don't have c given still we can infer the dependency if the descendant node is given since if that particular node is active than we can infer that even the parent node could be activated

General *d-separation* criterion

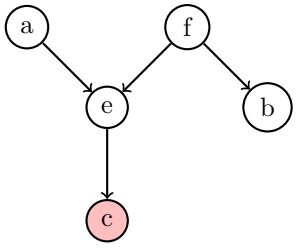
All the structures previously cited compone the general *d-separation* criterion which tells how to detect dependecies or independencies for any arbitrary sets of nodes. With these graphical rules we can manage the network and infer information about it. Given A, B, C arbitrary sets of nodes the sets A and B are *d-separated* by C (dsep$(A; B|C)$) if any path from any node in A to any node in B are blocked.

A path is blocked if it includes at least one node either:

- the arrows on the path meet tail-to-tail or head-to-tail at the node and it is in C
- the arrows on the path meet head-to-head at the node and neither it nor any of it's descendants is in C

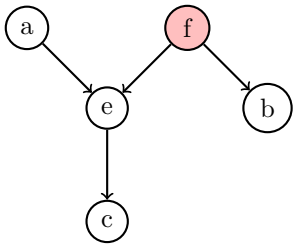
Example of general d-separation

a $\perp\!\!\!\perp$ **b**|**c**



if we take a and b we see that in the path there is node f which is connected tail-to-tail and it's not observed. Than we move to e where the connection is head-to-head and only it's child c is observed so there is anything b;ocking the path between the two so it's not **d-separated**

a $\perp$ **b**|**f**



in this case the first connection is made with f which is connected tail-to-tail in the path and it is **d-separated**

BN inedependencies revisited

As we stated at first we know that a graph G encodes a set of local indipendencies assumptions made like

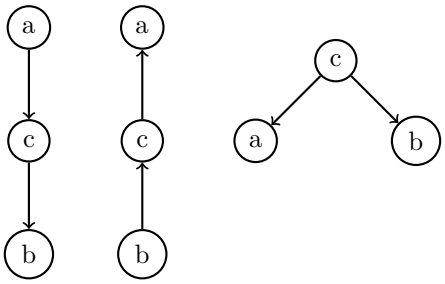
$$I_l(G) = \{\forall i x_i \perp \text{NonDescendatns}_{x_i} | \text{Parents}_{x_i}\}$$

but other than that thanks to the d-separation the same graph G can encode a set of *global*(Markov) independence assumptions

$$I(G) = \{(A \perp B|C) : \text{dsep(A;B|C)}\}$$

BN equivalence classes

From those perspectives different BN can end up encoding the same independencies and this is very important in terms of learning



if we look at the graphs above we can see that even if the have different directions all make the same assumptions about independence with c given or not. In those cases we say that two BN structures G and G' are *l-equivalent* if $I(G) = I(G')$. Moreover since all the graphs above infer the same independencies we call it an **l-equivalence class**.

I-maps vs Distibutions

When we try to model distributions with a BN we try to encode independencies in the distributions the graphical model has to be an I-map for p which means that if we encode a dependency it shoudld be on the distribution as well. Than the N of independencies we encode can make out graph more or less connected basing on how many independencies we introduced. To reduce this we introduce the concept of *minimal I-map* for p which is an I-map which can't be reduced further by removing edges inside the Graph that is also an I-map itself.

The problem is that a minimal I-map not always encodes all the possible independencies in p

Perfect Maps

A structure G is a *perfect map* for p if it captures all (and obly) it's dependencies (not all structures can ahave a perfect map)