Bayesian networks

Andrea Passerini
passerini@disi.unitn.it

Machine Learning

Graphical models

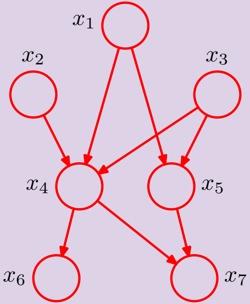
Why

- All probabilistic inference and learning amount at repeated applications of the sum and product rules
- *Probabilistic graphical models* are graphical representations of the *qualitative* aspects of probability distributions allowing to:
  - visualize the structure of a probabilistic model in a simple and intuitive way
  - discover properties of the model, such as conditional independencies, by inspecting the graph
  - express complex computations for inference and learning in terms of graphical manipulations
  - represent multiple probability distributions with the same graph, abstracting from their quantitative aspects (e.g. discrete vs continuous distributions)

Bayesian Networks (BN)

BN Semantics

- A BN structure ($\mathcal{G}$) is a *directed graphical model*
- Each node represents a random variable x_i
- Each edge represents a direct dependency between two variables



The structure encodes these independence assumptions:

$$\mathcal{I}_\ell(\mathcal{G}) = \{\forall i \; x_i \perp NonDescendants_{x_i} | Parents_{x_i}\}$$

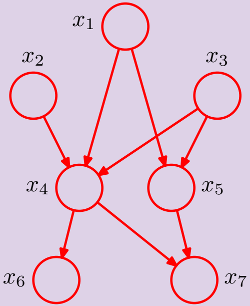
each variable is independent of its non-descendants given its parents

Bayesian Networks

Graphs and Distributions

- Let p be a joint distribution over variables $\mathcal{X}$
- Let $\mathcal{I}(p)$ be the set of independence assertions holding in p
- $\mathcal{G}$ in as *independency map* (I-map) for p if p satisfies the local independences in $\mathcal{G}$:

$$\mathcal{I}_\ell(\mathcal{G}) \subseteq \mathcal{I}(p)$$



Note

The reverse is not necessarily true: there can be independences in p that are not modelled by $\mathcal{G}$.

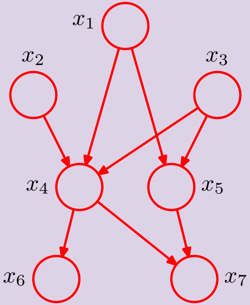
Bayesian Networks

Factorization

- We say that p *factorizes* according to $\mathcal{G}$ if:

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

- If $\mathcal{G}$ is an I-map for p, then p factorizes according to $\mathcal{G}$
- If p factorizes according to $\mathcal{G}$, then $\mathcal{G}$ is an I-map for p



Example

$$\begin{aligned} p(x_1, \ldots, x_7) = & p(x_1)p(x_2)p(x_3)p(x_4|x_1, x_2, x_3) \\ & p(x_5|x_1, x_3)p(x_6|x_4)p(x_7|x_4, x_5) \end{aligned}$$

Proof: I-map $\Rightarrow$ factorization

1. If $\mathcal{G}$ is an I-map for p, then p satisfies (at least) these (local) independences:

$$\{\forall i\ x_i \perp NonDescendants_{x_i} | Parents_{x_i}\}$$

2. Let us order variables in a *topological order* relative to $\mathcal{G}$, i.e.:

$$x_i \rightarrow x_j \Rightarrow i < j$$

3. Let us decompose the joint probability using the chain rule as:

$$p(x_1, \dots, x_m) = \prod_{i=1}^{m} p(x_i | x_1, \dots, x_{i-1})$$

4. Local independences imply that for each x_i:

$$p(x_i | x_1, \dots, x_{i-1}) = p(x_i | Pa_{x_i})$$

Proof: factorization $\Rightarrow$ I-map

1. If p factorizes according to $\mathcal{G}$, the joint probability can be written as:

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

2. Let us consider the last variable x_m (repeat steps for the other variables). By the product and sum rules:

$$p(x_m|x_1,\ldots,x_{m-1}) = \frac{p(x_1,\ldots,x_m)}{p(x_1,\ldots,x_{m-1})} = \frac{p(x_1,\ldots,x_m)}{\sum_{x_m} p(x_1,\ldots,x_m)}$$

3. Applying factorization and isolating the only term containing x_m we get:

$$= \frac{\prod_{i=1}^{m} p(x_i|Pa_{x_i})}{\sum_{x_m}\prod_{i=1}^{m} p(x_i|Pa_{x_i})} = \frac{p(x_m|Pa_{x_m})\cancel{\prod_{i=1}^{m-1} p(x_i|Pa_{x_i})}}{\cancel{\prod_{i=1}^{m-1} p(x_i|Pa_{x_i})}\cancelto{1}{\sum_{x_m} p(x_m|Pa_{x_m})}}$$

Bayesian Networks

Definition

A Bayesian Network is a pair $(\mathcal{G}, p)$ where p factorizes over $\mathcal{G}$ and it is represented as a set of conditional probability distributions (cpd) associated with the nodes of $\mathcal{G}$.

Factorized Probability

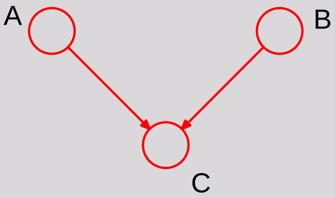
$$p(x_1, \ldots, x_m) = \prod_{i=1}^{m} p(x_i | Pa_{x_i})$$

Example: toy regulatory network

- Genes *A* and *B* have independent prior probabilities
- Gene *C* can be enhanced by both *A* and *B*

gene	value	P(value)
A	`active`	0.3
A	`inactive`	0.7

gene	value	P(value)
B	`active`	0.3
B	`inactive`	0.7



		A			
		`active`		`inactive`	
		B		B	
		`active`	`inactive`	`active`	`inactive`
C	`active`	0.9	0.6	0.7	0.1
C	`inactive`	0.1	0.4	0.3	0.9

Conditional independence

Introduction

- Two variables a, b are independent (written $a \perp b \,|\, \emptyset$) if:

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

- Two variables a, b are conditionally independent given c (written $a \perp b \,|\, c$) if:

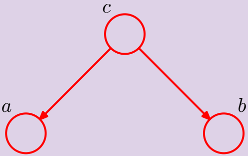
$$p(a, b|c) = p(a|c)p(b|c)$$

- Independence assumptions can be verified by repeated applications of sum and product rules
- Graphical models allow to directly verify them through the *d-separation* criterion

Tail-to-tail

- Joint distribution:

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

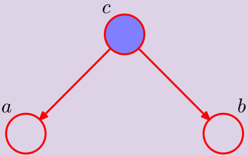


- a and b are **not independent** (written $a \not\perp\!\!\!\perp b \,|\, \emptyset$):

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

- a and b are **conditionally independent given** c:

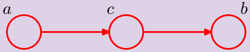
$$p(a, b|c) = \frac{p(a, b, c)}{p(c)} = p(a|c)p(b|c)$$



- c is *tail-to-tail* wrt to the path $a \to b$ as it is connected to the tails of the two arrows

d-separation

Head-to-tail

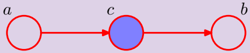


- Joint distribution:

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

- *a* and *b* are **not independent**:

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



- *a* and *b* are **conditionally independent given** *c*:

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

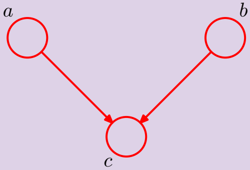
- *c* is *head-to-tail* wrt to the path $a \rightarrow b$ as it is connected to the head of an arrow and to the tail of the other one

d-separation

Head-to-head

- Joint distribution:

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

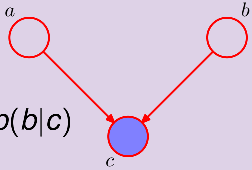


- a and b are **independent**:

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

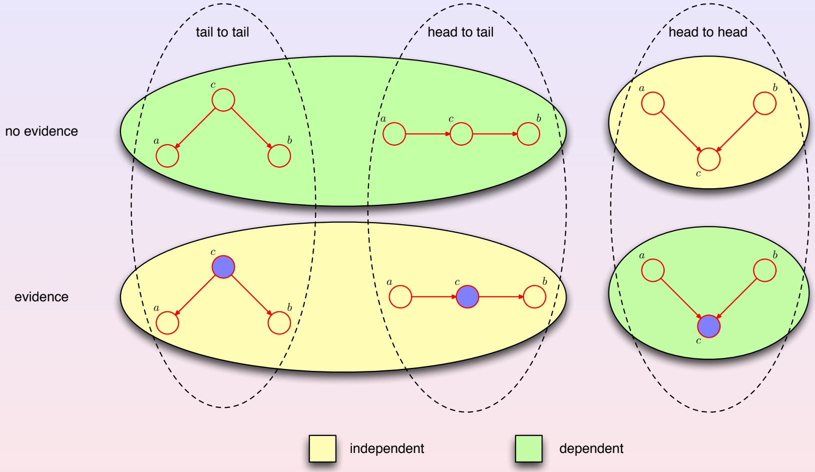
- a and b are **not conditionally independent given** c:

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



- c is *head-to-head* wrt to the path $a \rightarrow b$ as it is connected to the heads of the two arrows

d-separation: basic rules summary



Example of head-to-head connection

Setting

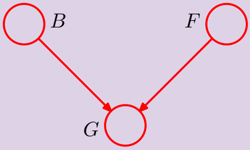
- A fuel system in a car:

  battery B, either charged ($B = 1$) or flat ($B = 0$)
  fuel tank F, either full ($F = 1$) or empty ($F = 0$)
  electric fuel gauge G, either full ($G = 1$) or empty ($G = 0$)

Conditional probability tables (CPT)

- Battery and tank have independent prior probabilities:

  $P(B = 1) = 0.9 \qquad P(F = 1) = 0.9$

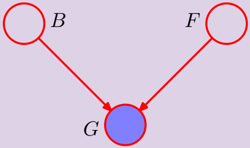


- The fuel gauge is conditioned on both (unreliable!):

$$P(G = 1|B = 1, F = 1) = 0.8 \quad P(G = 1|B = 1, F = 0) = 0.2$$
$$P(G = 1|B = 0, F = 1) = 0.2 \quad P(G = 1|B = 0, F = 0) = 0.1$$

Example of head-to-head connection

Probability of empty tank



- Prior:

$$P(F=0) = 1 - P(F=1) = 0.1$$

- Posterior after observing empty fuel gauge:

$$P(F=0|G=0) = \frac{P(G=0|F=0)P(F=0)}{P(G=0)} \simeq 0.257$$

Note

The probability that the tank is empty *increases* from observing that the fuel gauge reads empty (not as much as expected because of strong prior and unreliable gauge)

Example of head-to-head connection

Derivation

$$\begin{aligned} P(G=0|F=0) &= \sum_{B\in\{0,1\}} P(G=0,B|F=0) \\ &= \sum_{B\in\{0,1\}} P(G=0|B,F=0)P(B|F=0) \\ &= \sum_{B\in\{0,1\}} P(G=0|B,F=0)P(B) = 0.81 \end{aligned}$$

$$\begin{aligned} P(G=0) &= \sum_{B\in\{0,1\}} \sum_{F\in\{0,1\}} P(G=0,B,F) \\ &= \sum_{B\in\{0,1\}} \sum_{F\in\{0,1\}} P(G=0|B,F)P(B)P(F) \end{aligned}$$

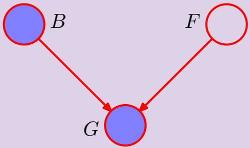
Example of head-to-head connection

Probability of empty tank

- Posterior after observing that the battery is also flat:

$$P(F=0|G=0,B=0)=$$

$$\frac{P(G=0|F=0,B=0)P(F=0|B=0)}{P(G=0|B=0)} \simeq 0.111$$



Note

- The probability that the tank is empty *decreases* after observing that the battery is also flat
- The battery condition *explains away* the observation that the fuel gauge reads empty
- The probability is still greater than the prior one, because the fuel gauge observation still gives some evidence in favour of an empty tank

d-separation

General Head-to-head

- Let a *descendant* of a node x be any node which can be reached from x with a path following the direction of the arrows
- A head-to-head node c unblocks the dependency path between its parents if either itself or *any of its descendants* receives evidence

General *d-separation* criterion

d-separation definition

- Given a generic Bayesian network
- Given A, B, C arbitrary nonintersecting sets of nodes
- The sets A and B are *d-separated* by C ($dsep(A; B|C)$) if:
  - All paths from any node in A to any node in B are *blocked*
- A path is blocked if it includes at least one node s.t. either:
  - the arrows on the path meet tail-to-tail or head-to-tail at the node and it is in C, or
  - the arrows on the path meet head-to-head at the node and neither it nor any of its descendants is in C

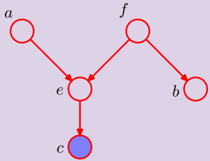
d-separation implies conditional independence

The sets A and B are independent given C ($A \perp B \mid C$) if they are d-separated by C.

Example of general d-separation

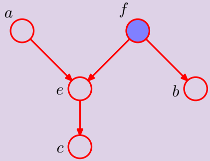
$a \not\perp\!\!\!\perp b | c$

- Nodes *a* and *b* are **not d-separated** by *c*:
  - Node *f* is tail-to-tail and not observed
  - Node *e* is head-to-head and its child *c* is observed



$a \perp b | f$

- Nodes *a* and *b* are **d-separated** by *f*:
  - Node *f* is tail-to-tail and observed



BN independences revisited

Independence assumptions

- A BN structure $\mathcal{G}$ encodes a set of *local* independence assumptions:

$$\mathcal{I}_\ell(\mathcal{G}) = \{\forall i \; x_i \perp NonDescendants_{x_i} | Parents_{x_i}\}$$

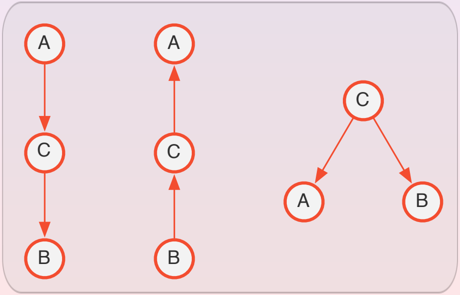
- A BN structure $\mathcal{G}$ encodes a set of *global* (Markov) independence assumptions:

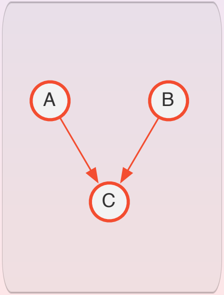
$$\mathcal{I}(\mathcal{G}) = \{(A \perp B|C) : dsep(A; B|C)\}$$

BN equivalence classes

I-equivalence

- Quite different BN structures can actually encode the exact same set of independence assumptions
- Two BN structures $\mathcal{G}$ and $\mathcal{G}'$ are *I-equivalent* if $\mathcal{I}(\mathcal{G}) = \mathcal{I}(\mathcal{G}')$
- The space of BN structures over $\mathcal{X}$ is partitioned into a set of mutually exclusive and exhaustive *I-equivalence classes*





I-maps vs Distributions

Minimal I-maps

- For a structure $\mathcal{G}$ to be an I-map for p, it does not need to encode all its independences (e.g. a fully connected graph is an I-map of any p defined over its variables)
- A *minimal I-map* for p is an I-map $\mathcal{G}$ which can't be "reduced" into a $\mathcal{G}' \subset \mathcal{G}$ (by removing edges) that is also an I-map for p.

Problem

A minimal I-map for p does not necessarily capture all the independences in p.

I-maps vs Distributions

Perfect Maps (P-maps)

- A structure $\mathcal{G}$ is a *perfect map* (P-map) for p if is captures all (and only) its independences:

$$\mathcal{I}(\mathcal{G}) = \mathcal{I}(p)$$

- There exists an algorithm for finding a P-map of a distribution which is exponential in the in-degree of the P-map.
- The algorithm returns an equivalence class rather than a single structure

Problem

Not all distributions have a P-map. Some cannot be modelled exactly by the BN formalism.

Building Bayesian Networks

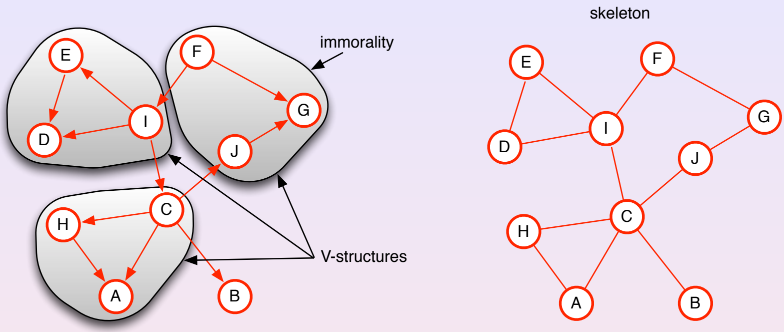
Practical Suggestions

- Get together with a domain expert
- Define variables for entities that can be *observed* or that you can be interested in *predicting* (latent variables can also be sometimes useful)
- Try following *causality* considerations in adding edges (more interpretable and sparser networks)
- In defining probabilities for configurations (almost) never assign zero probabilities
- If data are available, use them to help in *learning* parameters and structure (we'll see how)

Appendix

Additional reference material

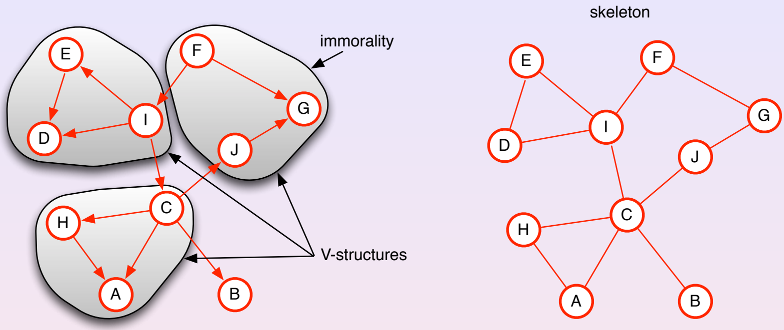
I-equivalence



Sufficient conditions

If two structures $\mathcal{G}$ and $\mathcal{G}'$ have the **same skeleton** *and the* **same set of v-structures** *then they are I-equivalent*

I-equivalence



Necessary and sufficient conditions

Two structures $\mathcal{G}$ and $\mathcal{G}'$ are I-equivalent if and only if they have the **same skeleton** *and the* **same set of immoralities**

Equivalence class

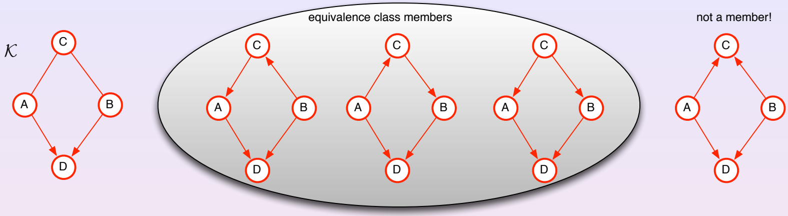
Partially directed acyclic graph (PDAG)

A PDAG is an acyclic graph with both directed and undirected edges

Representing an equivalence class

- An equivalence class for a structure $\mathcal{G}$ can be represented by a PDAG $\mathcal{K}$ such that:
  - If $x \to y \in \mathcal{K}$ then $x \to y$ should appear in all structures which are I-equivalent to $\mathcal{G}$
  - If $x - y \in \mathcal{K}$ then we can find a structure $\mathcal{G}'$ that is I-equivalent to $\mathcal{G}$ such that $x \to y \in \mathcal{G}'$

Equivalence class members



Generating members

- Representatives from $\mathcal{K}$ can be obtained by adding directions to undirected edges
- One needs to check that the resulting structure has the **same set of immoralities** as $\mathcal{K}$ (otherwise it's not in the equivalence class)

Markov blanket (or boundary)

Definition

- Given a directed graph with m nodes
- The *markov blanket* of node x_i is the minimal set of nodes making it x_i independent on the rest of the graph:

$$p(x_i|x_{j\neq i}) = \frac{p(x_1,\ldots,x_m)}{p(x_{j\neq i})} = \frac{p(x_1,\ldots,x_m)}{\int p(x_1,\ldots,x_m)dx_i}$$

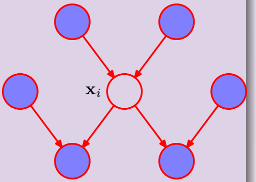
$$= \frac{\prod_{k=1}^m p(x_k|\mathrm{pa}_k)}{\int \prod_{k=1}^m p(x_k|\mathrm{pa}_k)dx_i}$$

- All components which do not include x_i will cancel between numerator and denominator
- The only remaining components are:
  - $p(x_i|pa_i)$ the probability of x_i given its parents
  - $p(x_j|\mathrm{pa}_j)$ where pa_j includes x_i $\Rightarrow$ the children of x_i with their *co-parents*

Markov blanket (or boundary)

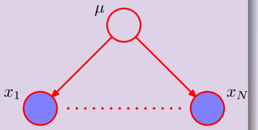
d-separation

- Each parent x_j of x_i will be head-to-tail or tail-to-tail in the path btw x_i and any of x_j other neighbours $\Rightarrow$ blocked
- Each child x_j of x_i will be head-to-tail in the path btw x_i and any of x_j children $\Rightarrow$ blocked
  - Each co-parent x_k of a child x_j of x_i be head-to-tail or tail-to-tail in the path btw x_j and any of x_k other neighbours $\Rightarrow$ blocked

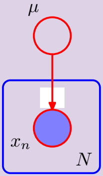


Example of i.i.d. samples

Maximum-likelihood



- We are given a set of instances $\mathcal{D} = \{x_1, \ldots, x_N\}$ drawn from an univariate Gaussian with unknown mean μ
- All paths between x_i and x_j are blocked if we condition on μ
- The examples are independent of each other given μ:



$$p(\mathcal{D}|\mu) = \prod_{i=1}^{N} p(x_i|\mu)$$

- A set of nodes with the same variable type and connections can be compactly represented using the *plate* notation