Inference and Representation

Rachel Hodos

New York University

Lab 2, September 9, 2015

Outline

1. Review from lecture
2. Introduction to pymc3

Conditional Independence Triplets

- Bayesian networks represent conditional independencies
- Independence can be identified in any graph by understanding these three cases on triplets:
  - cascade (or chain)
  - common parent (or common cause)
  - common child (or v-structure)

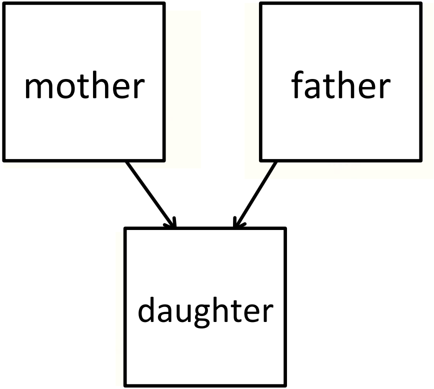
An aside..

- Alternative definition of independence:

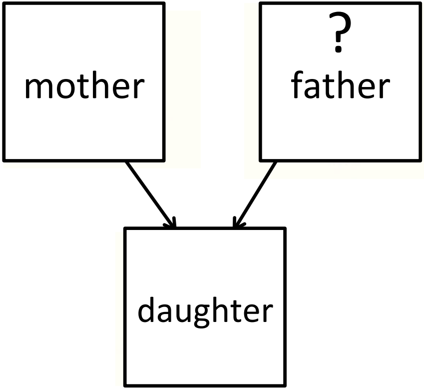
$$(A \perp B|C) \text{ iff } p(A|B, C) = P(A|C)$$

- You will prove this in your homework.

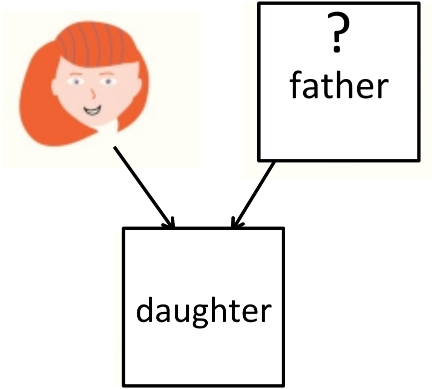
V-structures



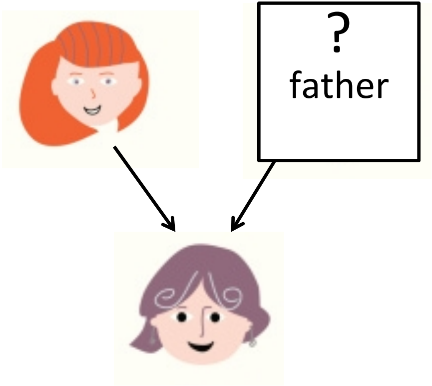
V-structures



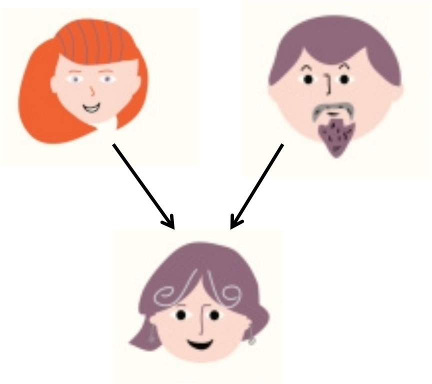
V-structures



V-structures



V-structures



Review of I-maps and P-maps

- Let $I(G)$ be the set of all conditional independencies implied by the directed acyclic graph (DAG) G
- Let $I(p)$ denote the set of all conditional independencies that hold for the joint distribution p.
- A DAG G is an **I-map** (independence map) of a distribution p if $I(G) \subseteq I(p)$
  - A fully connected DAG G is an I-map for *any* distribution, since $I(G) = \emptyset \subseteq I(p)$ for all p
- G is a **minimal I-map** for p if the removal of even a single edge makes it not an I-map
  - A distribution may have several minimal I-maps
  - Each corresponds to a specific node-ordering
- G is a **perfect map** (P-map) for distribution p if $I(G) = I(p)$

I-map, U-map, we all map...

- What's the point?
- Bayesian networks should *represent* or *visualize* our probability distribution
- If G is a perfect map, we can (to some degree) trust G to accurately represent p

When do the parameters matter?

- Could find state-dependent independencies
- May have some scientific significance (e.g. how strongly does molecule A regulate molecule B?)
- May care about what variable is most strongly associated with another variable

Introduction to python package pymc3

- Let's use pymc3 to define and learn a simple statistical model
- Example from pymc-devs.github.io/pymc3/getting_started
- Suppose we have three variables, X_1, X_2 and Y that follow this distribution:

$$Y \sim \mathcal{N}(\mu, \sigma^2)$$
$$\mu = \alpha + \beta_1 X_1 + \beta_2 X_2$$

- Parameters: $\alpha = 1, \sigma = 1, \beta_1 = 1, \beta_2 = 2.5$
- Y depends linearly on X_1 and X_2
- Let's do linear regression

Linear regression using pymc3

- We will assume we know the form of the model
- Let's define prior distributions over the model parameters:

$$\alpha \sim \mathcal{N}(0, 100)$$
$$\beta_i \sim \mathcal{N}(0, 100)$$
$$\sigma \sim |\mathcal{N}(0, 1)|$$

- pymc3 can find the MAP solution (*maximum a posteriori*, more about this later)