Bayesian Networks in Healthcare

Artificial Intelligence for Digital Health (AID)

M.Sc. in Digital Health – University of Pisa

Davide Bacciu (davide.bacciu@unipi.it)

IN SUPREMÆ DIGNITATIS · 1343 ·

Lecture(s) Outline

- Introduction to Bayesian networks
  - Graphical formalism
- Structure and components of Bayesian networks
  - Random variables and conditional independence
  - Factorized distributions
  - Relevant graphical substructures
  - Reasoning graphically on conditional independence
- Learning in Bayesian Networks
- Applications in healthcare for diagnosis, prognosis, and decision support systems

Probabilistic models

- ML models that represent knowledge inferred from data under the form of probabilities
  - Probabilities can be sampled: new data can be generated
  - Supervised, unsupervised, weakly supervised learning tasks
  - Incorporate prior knowledge on data and tasks
  - Interpretable knowledge (how data is generated)
- The majority of the modern task comprises large numbers of variables
  - Modeling the joint distribution of all variables can become impractical
  - Exponential size of the parameter space
  - Computationally impractical to train and predict

Bayesian Networks - A Graphical Framework

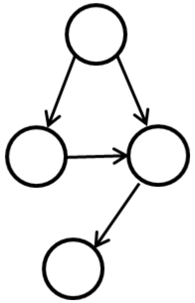
- Representation
  - Bayesian Networks are a compact way to represent exponentially large probability distributions
  - Encode conditional independence assumptions
- Inference
  - How to query (predict with) a Bayesian Network?
  - Probability of unknown random variable X given observed ones $\boldsymbol{d}$, $P(X|\boldsymbol{d})$
- Learning
  - Fitting the parameters associated with the model probability distribution
  - An inference problem after all

Graphical Representation

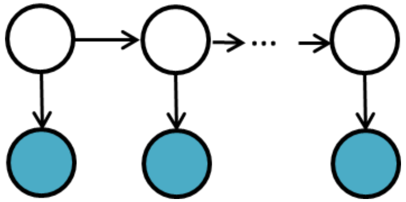
A graph whose **nodes** (vertices) are **random variables** whose **edges** (links) represent **probabilistic relationships** between the variables

Bayesian Network (BN)

Directed edges express dependence relationships



Dynamic BNs



Allow the BN structure to change to reflect dynamic processes

Probability factorization in probabilistic ML

Representing Joint Distributions

- The main goal of **probabilistic modeling** is to define models able to represent the **joint distribution** of a set of variables.
- Probabilistic models enable
  - **Sampling** new instances
  - Inferencing values of **hidden** variables
  - Estimating the **likelihood** of a configuration
  - …

Representing Joint Distributions

- Assume N discrete random variables with k distinct values.
- How many parameters in the **joint probability distribution**?

Y_1	Y_2	Y_3	$P(Y_1, Y_2, Y_3)$
0	0	0	0.03
0	0	1	0.12
0	1	0	0.31
⋮	⋮	⋮	⋮
1	1	1	0.04

$$k^N - 1$$

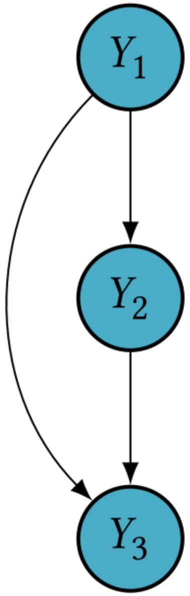
Representing Joint Distributions

- What if we compute the probability **one variable** at the time?
- We can exploit the **chain rule** to decompose the joint.

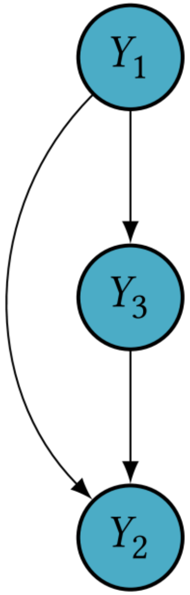
$$
\begin{aligned}
P(Y_1, Y_2, Y_3) &= P(Y_1)P(Y_2 \mid Y_1)P(Y_3 \mid Y_1, Y_2) \\
&= P(Y_2)P(Y_1 \mid Y_2)P(Y_3 \mid Y_1, Y_2) \\
&= \ldots \\
&= P(Y_3)P(Y_2 \mid Y_3)P(Y_1 \mid Y_2, Y_3).
\end{aligned}
$$

Representing Joint Distributions

- The **order** of the variables can be represented by **directed graphs**.

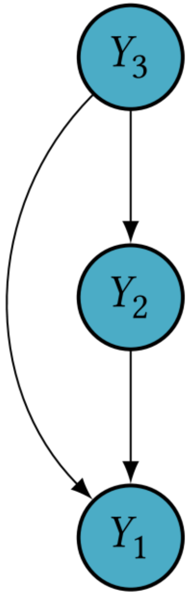


$P(Y_1)P(Y_2 \mid Y_1)P(Y_3 \mid Y_1, Y_2)$



$P(Y_1)P(Y_3 \mid Y_1)P(Y_2 \mid Y_1, Y_3)$

...



$P(Y_3)P(Y_2 \mid Y_3)P(Y_1 \mid Y_2, Y_3)$

Representing Joint Distributions

- Decomposing the joint with the **chain rule** reduces the **number of parameters**?
- No! 🥹

$$P(Y_1, Y_2, Y_3) = \underbrace{P(Y_1)}_{1}\underbrace{P(Y_2 \mid Y_1)}_{2}\underbrace{P(Y_3 \mid Y_1, Y_2)}_{4}$$

$$\sum_{i=0}^{N-1}(k-1)k^i = k^N - 1$$

Marginal and Conditional Independence

- Two random variables X and Y are **independent** if knowledge about X does not change the uncertainty about Y and vice versa

$$I(X,Y) \iff X \perp Y \iff P(X,Y) = P(X \mid Y)P(Y)$$
$$= P(Y \mid X)P(X) = P(X)P(Y).$$

Representing Joint Distributions

- When variables are **independent**, we only need Nk parameters.

$$P(Y_1, Y_2, Y_3) = P(Y_1)P(Y_2 \mid Y_1)P(Y_3 \mid Y_1, Y_2)$$
$$= \underbrace{P(Y_1)}_{\mathbf{1}}\underbrace{P(Y_2)}_{\mathbf{1}}\underbrace{P(Y_3)}_{\mathbf{1}}$$

Marginal and Conditional Independence

- Two random variables X and Y are **conditionally independent** given Z if knowledge about X does not change the uncertainty about Y and vice versa on the conditional distribution

$$
\begin{aligned}
I(X,Y \mid Z) \iff X \perp Y \mid Z \iff P(X,Y \mid Z) &= P(X \mid Y, Z)P(Y, Z) \\
&= P(Y \mid X, Z)P(X, Z) \\
&= P(X \mid Z)P(Y \mid Z).
\end{aligned}
$$

Representing Joint Distributions

- Conditional independences reduce the **number of parameters**
- Yes! 🥳

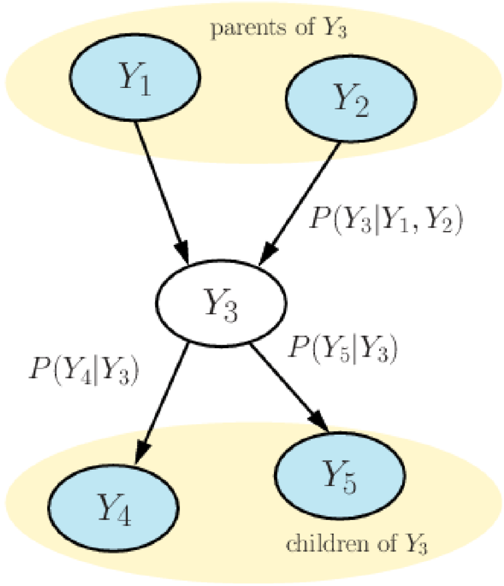
$$Y_1 \perp Y_3 \mid Y_2$$

$$\Longrightarrow \quad P(Y_1, Y_2, Y_3) = P(Y_1)P(Y_2 \mid Y_1)P(Y_3 \mid Y_1, Y_2)$$

$$= \underbrace{P(Y_1)}_{1}\underbrace{P(Y_2 \mid Y_1)}_{2}\underbrace{P(Y_3 \mid Y_2)}_{2}$$

Bayesian Networks

Bayesian Network



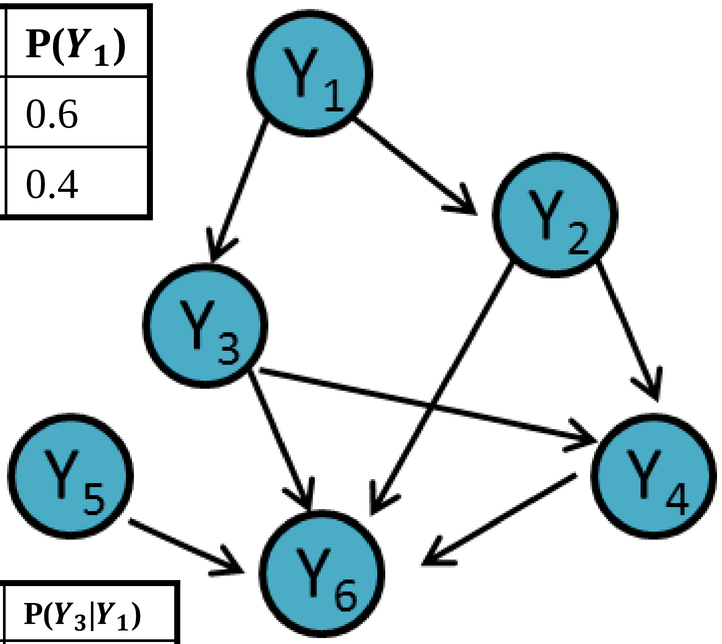
- Directed Acyclic Graph (DAG) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
- Nodes $v \in \mathcal{V}$ represent random variables
  - Shaded $\Rightarrow$ observed
  - Empty $\Rightarrow$ un-observed
- Edges $e \in \mathcal{E}$ describe the conditional independence relationships

Conditional Probability Tables (CPT) local to each node describe the probability distribution given its parents

$$P(Y_1, \dots, Y_N) = \prod_{i=1}^{N} P(Y_i \,|pa(Y_i))$$

Joint probability factorization

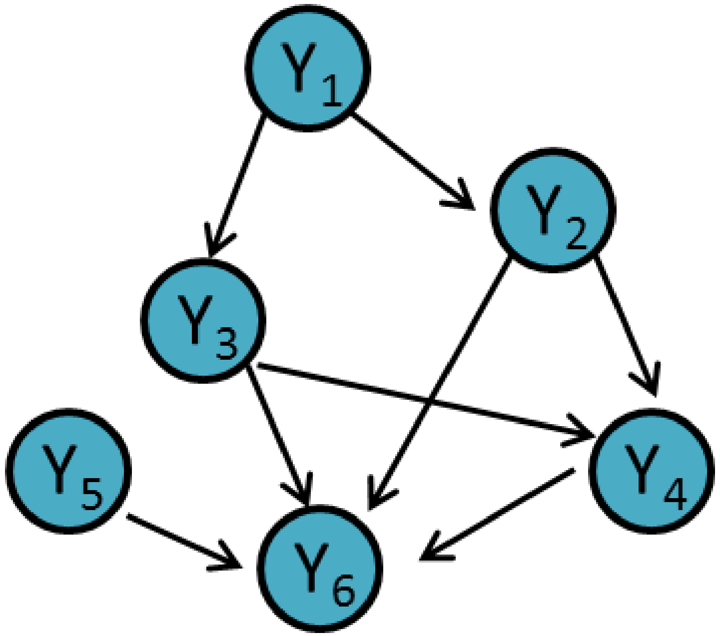
Y_1	$P(Y_1)$
false	0.6
true	0.4



Y_1	Y_3	$P(Y_3 \mid Y_1)$
false	false	0.4
false	true	0.6
true	false	0.9
true	true	0.1

- Let L be the **maximum number of ingoing edges** in a Bayes Net.
- Then, the number of parameters is **at most** $N \cdot (k-1)^L$
- ⇒ The **sparser** the network, the less "complex" the parameters.

Causality or Dependence?



- Are these relations **causal**?
- In general **no**, a Bayesian Network represent **statistical dependence** relations.
- However, they **might** coincide with causal dependence under further **assumptions**.

Local Markov Property

Definition (Local Markov property)

Each node / random variable is conditionally independent of all its non-descendants given a joint state of its parents

$$Y_v \perp Y_{V \setminus \mathrm{ch}(v)} \mid Y_{pa(v)} \text{ for all } v \in V$$

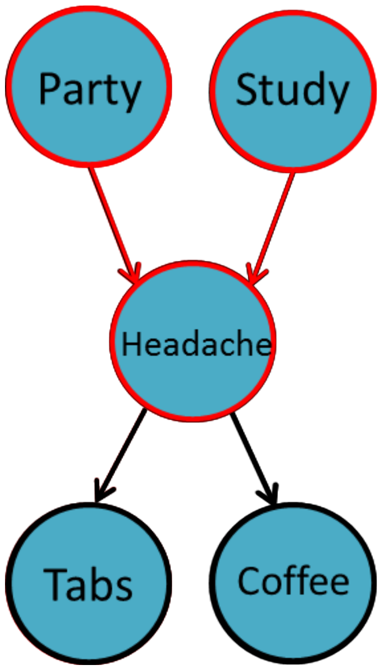
$Party$ and $Study$ are marginally independent

- $Party \perp Study$

However, local Markov property does not support

- $Party \perp Study \mid Headache$
- $Tabs \perp Party$

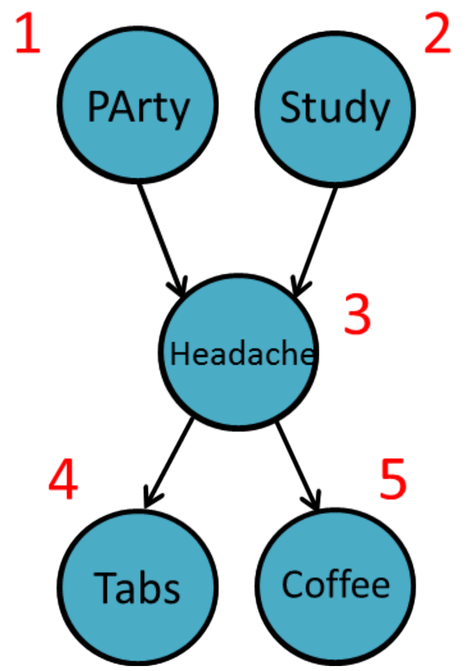
But $Party$ and $Tabs$ are independent given $Headache$



Joint Probability Factorization

An application of Chain rule and Local Markov Property

1. Pick a topological ordering of nodes
2. Apply chain rule following the order
3. Use the conditional independence assumptions



$$P(PA,S,H,T,C) = P(PA) \cdot P(S|PA) \cdot P(H|S,PA) \cdot P(T|H,S,PA) \cdot P(C|T,H,S,PA)$$
$$= P(PA) \cdot P(S) \cdot P(H|S,PA) \cdot P(T|H) \cdot P(C|H)$$

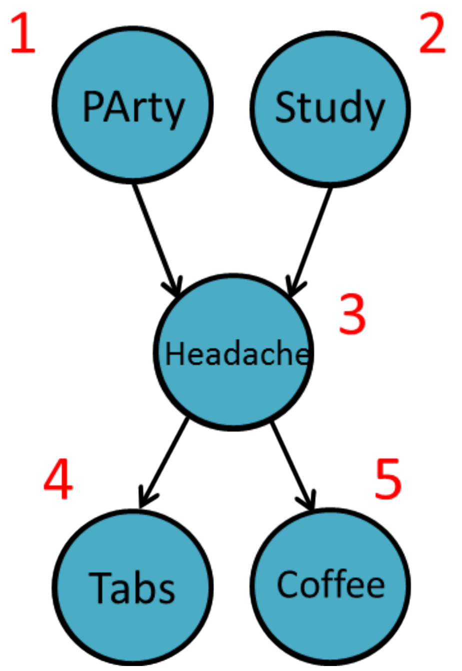
(Ancestral) Sampling of a BN

A BN describes a generative process for observations

1. Pick a topological ordering of nodes
2. Generate data by sampling from the local conditional probabilities following this order

Generate i-th sample for each variable PA, S, H, T, C

1. $pa_i \sim P(PA)$
2. $s_i \sim P(S)$
3. $h_i \sim P(H|S = s_i, PA = pa_i)$
4. $t_i \sim P(T|H = h_i)$
5. $c_i \sim P(C|H = h_i)$

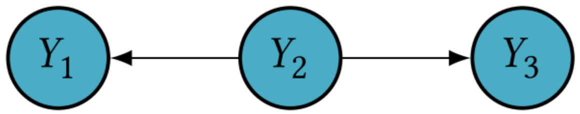


Conditional Independence in Bayesian Networks

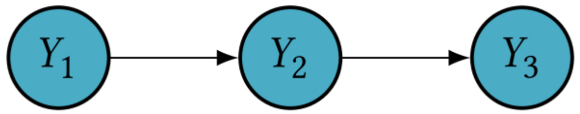
Fundamental BN structures

There exist **three fundamental substructures** that determine the conditional independence relationships in a Bayesian Network.

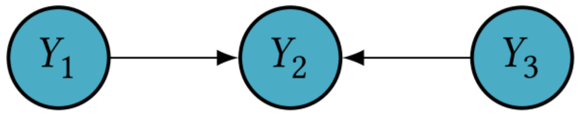
- **Tail-to-Tail** (Fork, “Common Cause”)



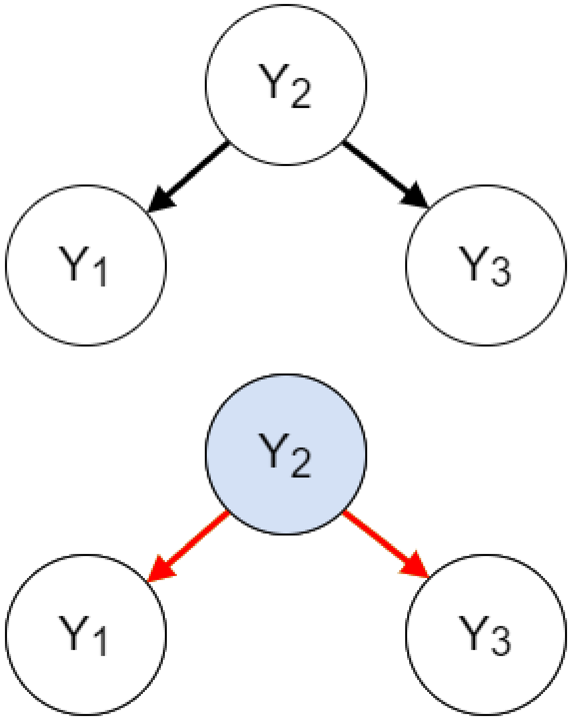
- **Head-to-Tail** (Chain, “Causal Effect”)



- **Head-to-Head** (Collider, “Common Effect”)



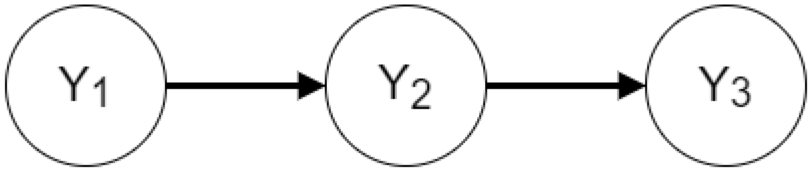
Tail-to-Tail Connections

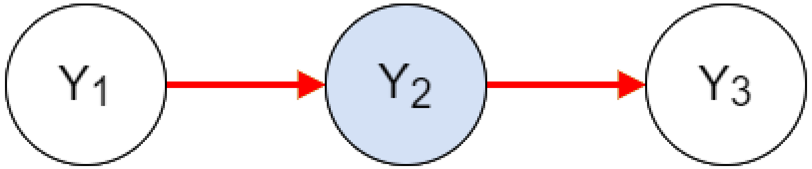


- Corresponds to
  $$P(Y_1, Y_3|Y_2)P(Y_2) = P(Y_1|Y_2)P(Y_3|Y_2)P(Y_2)$$
- If Y_2 is unobserved then Y_1 and Y_3 are marginally dependent
  $$Y_1 \not\perp Y_3$$
- If Y_2 is observed then Y_1 and Y_3 are conditionally independent
  $$Y_1 \perp Y_3|Y_2$$

When Y_2 in observed is said to **block the path** from Y_1 to Y_3

Head-to-Tail Connections





Observed Y_2 **blocks the path** from Y_1 to Y_3

- Corresponds to

$$P(Y_1, Y_2, Y_3) = P(Y_1)P(Y_2|Y_1)P(Y_3|Y_2)$$
$$= P(Y_1|Y_2)P(Y_3|Y_2)P(Y_2)$$

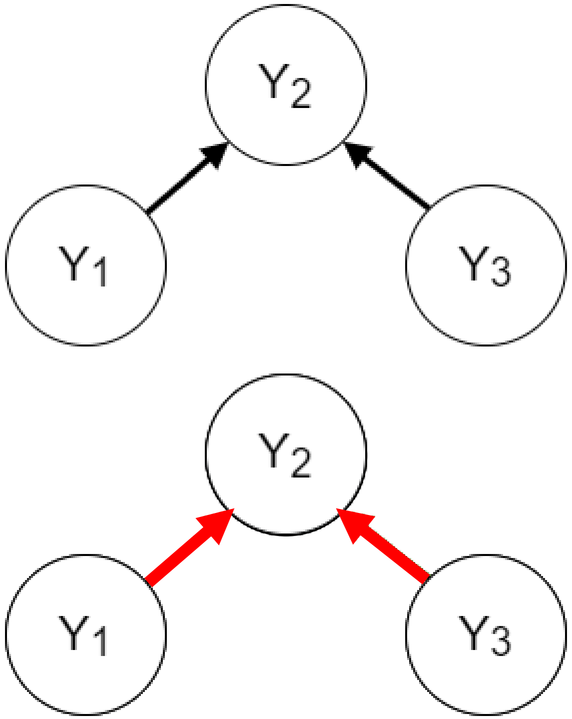
- If Y_2 is unobserved then Y_1 and Y_3 are marginally dependent Type equation here.

$$Y_1 \not\perp Y_3$$

- If Y_2 is observed then Y_1 and Y_3 are conditionally independent

$$Y_1 \perp Y_3 | Y_2$$

Head-to-Head Connections



- Corresponds to
$$P(Y_1, Y_2, Y_3) = P(Y_1)P(Y_3)P(Y_2|Y_1, Y_3)$$
- If Y_2 is observed then Y_1 and Y_3 are conditionally dependent
$$Y_1 \not\perp Y_3 | Y_2$$
- If Y_2 is unobserved then Y_1 and Y_3 are marginally independent
$$Y_1 \perp Y_3$$

If any Y_2 **descendants** is observed it **unlocks the path**

Blocked Path

Let $r = (Y_1 \leftrightarrow \cdots \leftrightarrow Y_2)$ be an **undirected path** between Y_1 and Y_2.

The path r is **blocked** by a set Z if one of the following holds:

- r contains a **fork** (tail-to-tail) $Y_i \leftarrow Y_c \rightarrow Y_j$ such that $Y_c \in Z$, or
- r contains a **chain** (head-to-tail) $Y_i \rightarrow Y_c \rightarrow Y_j$ such that $Y_c \in Z$, or
- r contains a **collider** (head-to-head) $Y_i \rightarrow Y_c \leftarrow Y_j$ such that **neither Y_c nor its descendants are in Z**.

d-Separation

Definition (d-separated path)

Let $r = Y_1 \leftrightarrow \cdots \leftrightarrow Y_2$ be an undirected path between Y_1 and Y_2, then r is d-separated by Z if there exist at least one node $Y_c \in Z$ for which path r is blocked.

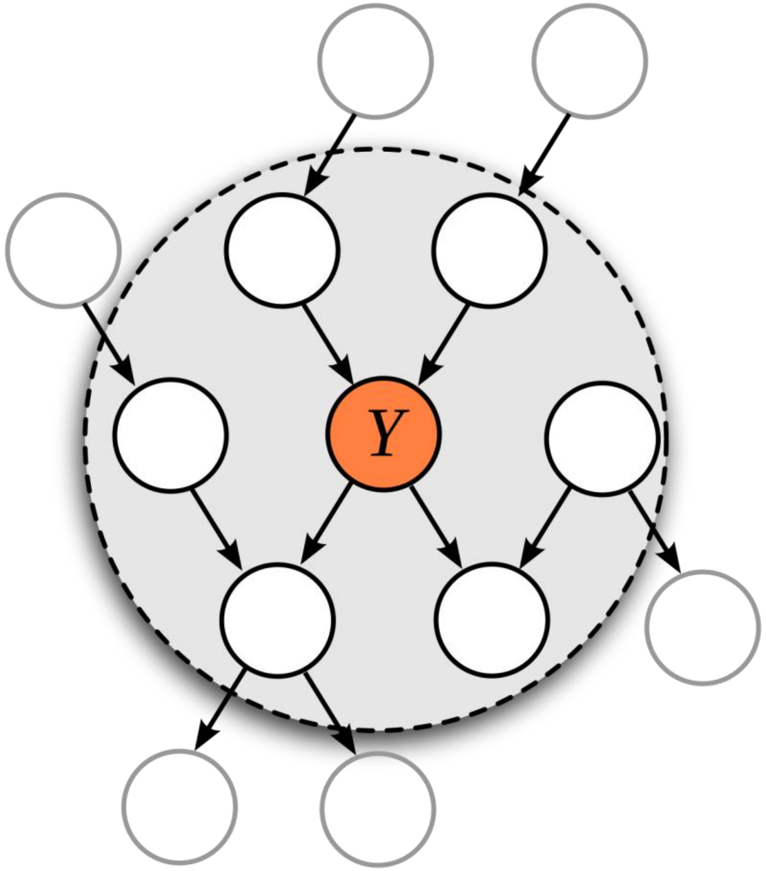
d-Separation

Definition (d-separation)

Two nodes Y_i and Y_j in a BN $\mathcal{G}$ are said to be d-separated by $Z \subset \mathcal{V}$ (denoted by $Dsep_{\mathcal{G}}(Y_i, Y_j | Z)$) if and only if all undirected paths between Y_i and Y_j are d-separated by Z

$$Y_1 \perp_{\mathcal{G}} Y_2 \mid Z$$

Markov Blanket

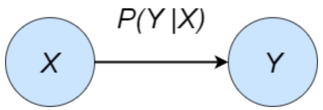
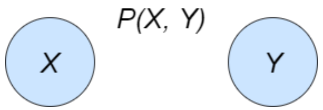


- The **Markov Blanket** $\mathrm{Mb}(Y)$ of a node Y is the minimal set of vertices that **shield the node** from the rest of the Bayesian Network.
- In a DAG, the Markov Blanket of Y contains
  - Its parents Pa(Y)
  - Its children Ch(Y)
  - Its children's parents Pa(Ch(Y))
- The behavior of a node can be **completely determined and predicted** from the knowledge of its Markov Blanket.

$$P(Y \mid \mathrm{Mb}(Y), Z) = P(Y \mid \mathrm{Mb}(Y)) \; \forall Z \notin \mathrm{Mb}(Y)$$

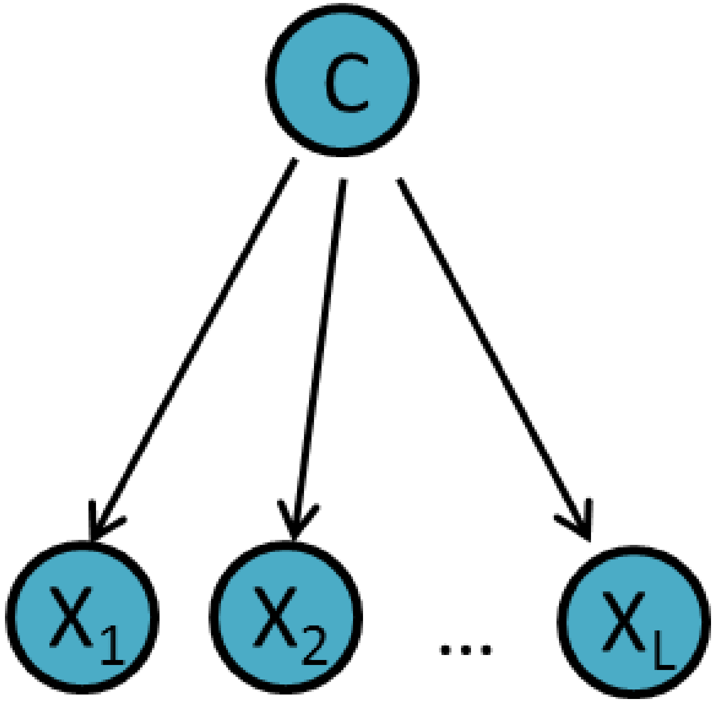
Learning in Bayesian Networks

Learning with Bayesian Networks

		Structure	
		Fixed Structure 	Fixed Variables 
Data	Complete	Naive Bayes Calculate Frequencies (ML)	Discover dependencies from the data Structure Search Independence tests
	Incomplete	Latent variables EM Algorithm (ML) MCMC, VBEM (Bayesian)	Difficult Problem Structural EM
		Parameter Learning	**Structure Learning**

Learning Parameters on a Simple Bayesian Network

The Naive Bayes Classifier



The naïve independence assumption

- Input features Y_i are independent given the class

$$P(C, \mathrm{X}_1, \dots, \mathrm{X}_\mathrm{L}) = P(C) \prod_{i=1}^{\mathrm{L}} P(\mathrm{X}_i | C)$$

Learning entails finding the values of $P(C)$ and $P(\mathrm{X}_i|C)$ (for all i)

Naive Bayes – Maximum Likelihood Learning

- Consider N observed training pairs $\boldsymbol{d} = \{(\boldsymbol{x}_n, c_n)\}_{n=1:N}$ s.t. $\boldsymbol{x}_n =< x_{1n}, \ldots, x_{Ln} >$
- The model likelihood is the probability of the data $\boldsymbol{d}$ given the model parameters $\theta = \{P(C), P(X_1|C), \ldots, P(X_L|C)\}$ (for Naïve Bayes on discrete data)

$$P(\boldsymbol{d}|\theta) = \prod_{n=1}^{N} P(c_n) \prod_{i=1}^{L} P(x_{in}|c_n)$$

- Learning equations for the model are derived by maximization of the logarithm of the likelihood

$$\theta^* = \max_{\theta} \log P(\boldsymbol{d}|\theta)$$

- For a model as simple as the Naïve Bayes this optimization can be easily computed and closed form update equations are obtained

Example of Naive Bayes Learning Rules

It is all about counting frequencies of events occurring (this is true in general for maximum-likelihood learning with discrete variables)

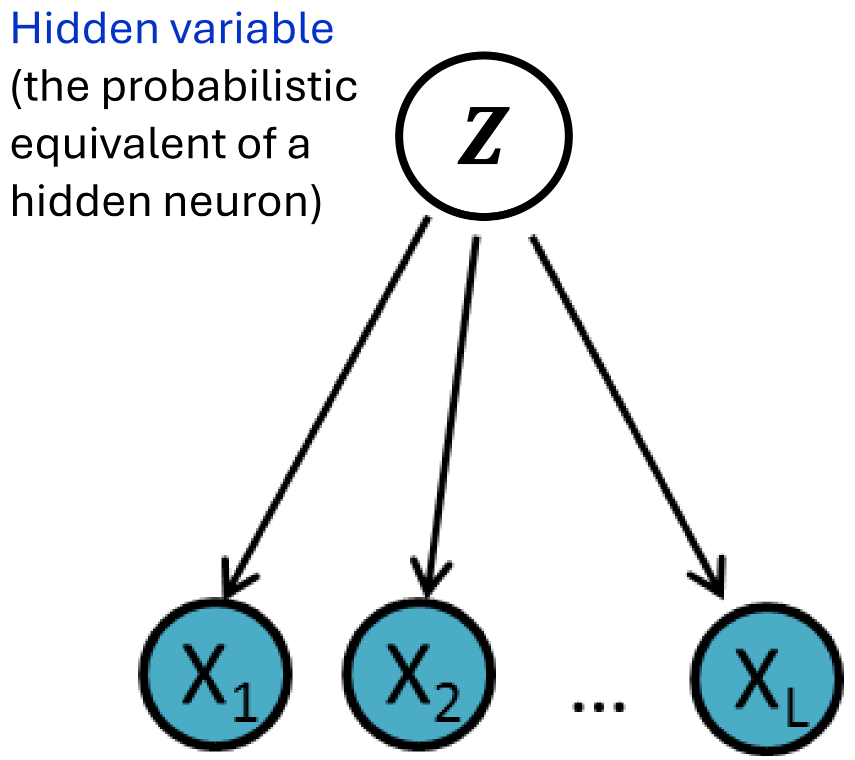
- $N(k) \rightarrow$ Number of samples in class k
- $N_{is}(k) \rightarrow$ Number of samples in class k where the i-th attribute has value s

$$P(C = k) = \frac{N(k)}{N}$$

$$P(X_i = s|C = k) = \frac{N_{is}(k)}{\sum_{s=1}^{S_l} N_{is}(k)}$$

In general, everything works this smoothly whenever your Bayesian Network does not contain non-observable variables

Bayesian Networks and Hidden Variables



- Hidden variables are introduced to explain complex relationships between observed data in simple ways
- Allow to apply conditional independence simplifications

$$P(X_1, \dots, X_L) \approx \sum_{z} P(Z) \prod_{i=1}^{L} P(X_i | Z)$$

- Learning becomes more complex because we do not have ground truth observations for Z
  - We need to make probabilistic hypotheses on Z to learn the model parameters θ

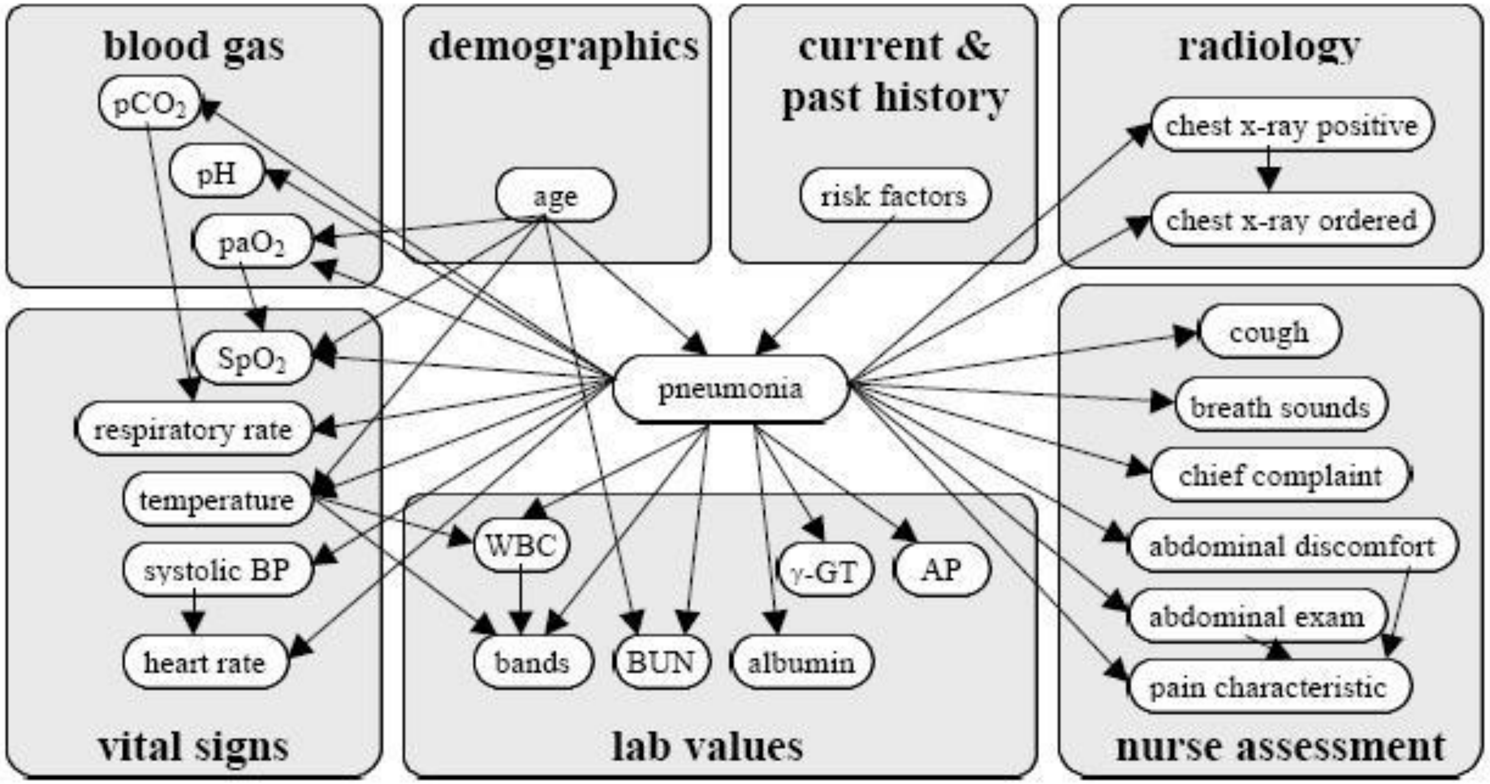
Bayesian Networks in Healthcare

Why Bayesian Networks in Healthcare

You would like to determine how likely the patient has pneumonia given that the patient has a cough, a fever, and difficulty breathing

- We are not 100% certain that the patient has pneumonia ⇒ Reasoning with uncertainty (a probabilistic approach)
- You know that some symptoms connect with diagnosis ⇒ Fitting prior knowledge into the model
- X given that Y occurs ⇒ Conditional probabilities and independence
- How did you come up with the diagnosis? ⇒ Interpretability requirements

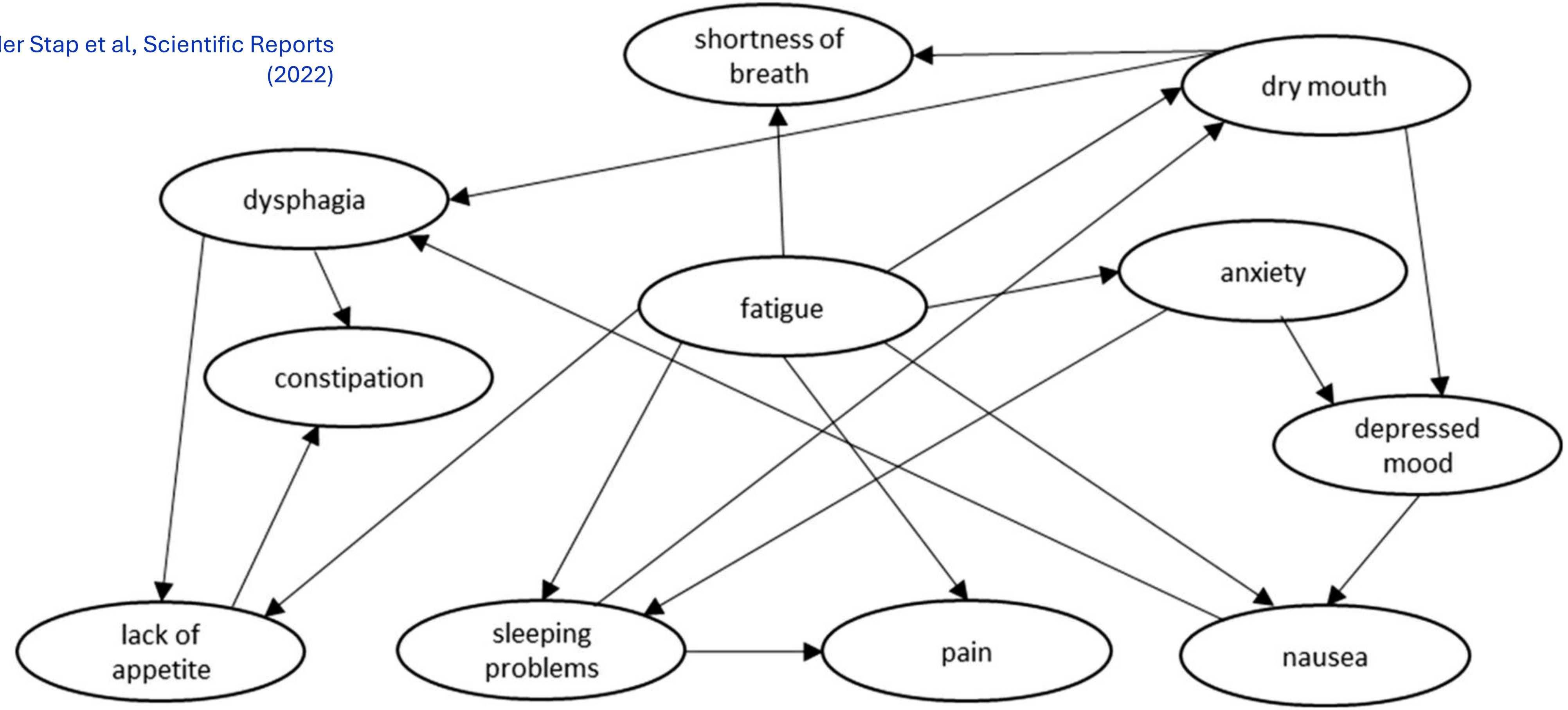
A Bayesian Network for Pneumonia



Aronsky, D. and Haug, P.J., Diagnosing community-acquired pneumonia with a Bayesian network, In: *Proceedings of the Fall Symposium of the American Medical Informatics Association*, (1998) 632-636.

Studying simultaneous symptoms in patients with advanced cancer

van der Stap et al, Scientific Reports (2022)



From an Inferential Perspective

Main symptom[a]	Main symptom[a]	Predicted simultaneous symptom	Conditional probability of experiencing simultaneous symptom (%)
Fatigue +	Sleeping problems +	Pain	54.4
Fatigue +	Sleeping problems –		37.6
Fatigue –	Sleeping problems +		40.0
Fatigue –	Sleeping problems –		13.8
Fatigue +	Anxiety +	Sleeping problems	63.5
Fatigue +	Anxiety –		41.4
Fatigue –	Anxiety +		56.3
Fatigue –	Anxiety –		18.3
Fatigue +	Sleeping problems +	Dry mouth	62.7
Fatigue +	Sleeping problems –		47.8
Fatigue –	Sleeping problems +		45.0
Fatigue –	Sleeping problems –		22.8
Dry mouth +	Nausea +	Dysphagia	54.2
Dry mouth +	Nausea –		33.0
Dry mouth –	Nausea +		31.3
Dry mouth –	Nausea –		5.8
Fatigue +	Dysphagia +	Lack of appetite	80.0
Fatigue +	Dysphagia –		56.4
Fatigue –	Dysphagia +		81.0
Fatigue –	Dysphagia –		24.4

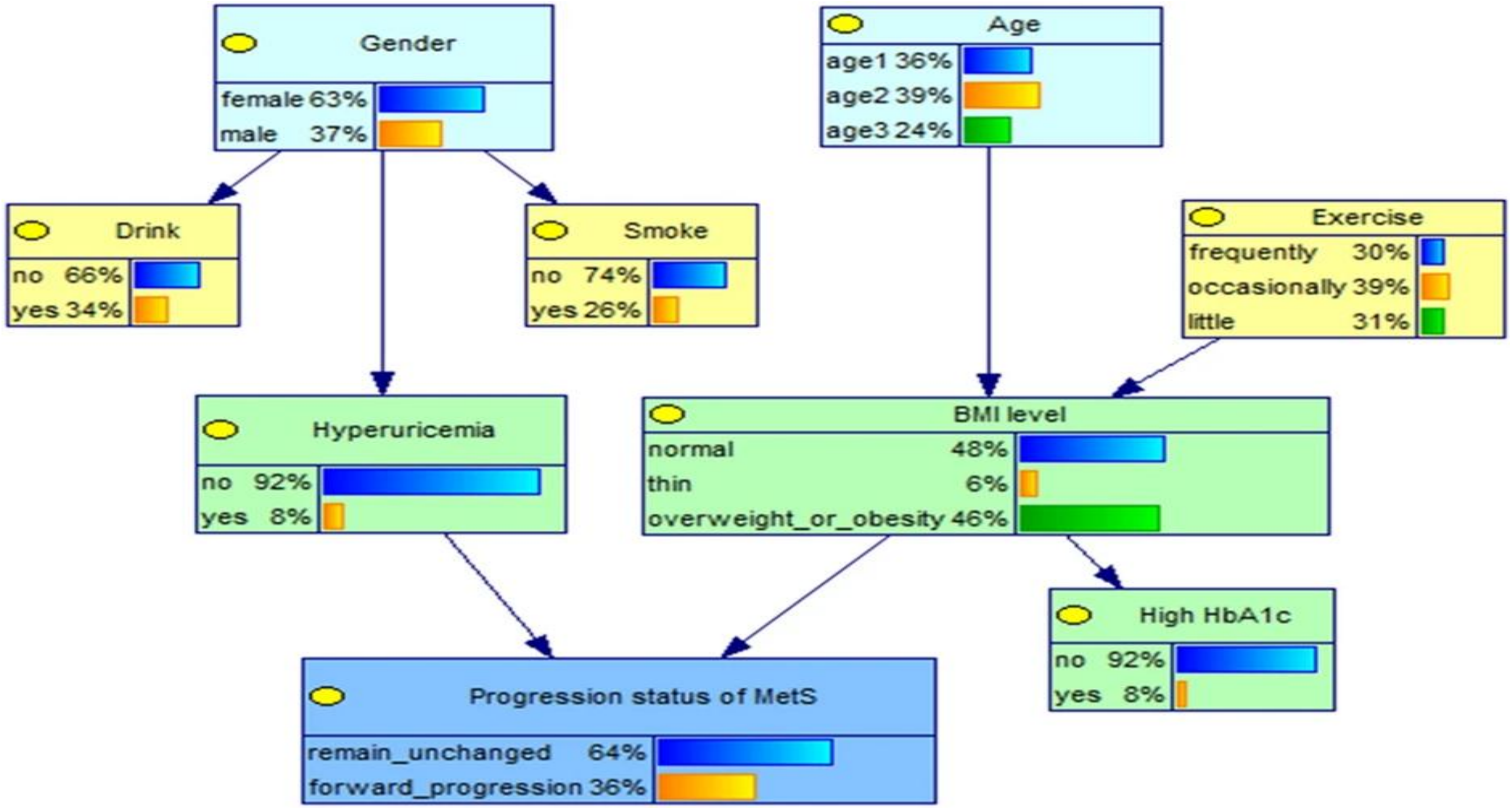
Fixed evidential data

→ Inferred non-observed simultaneous symptom

van der Stap et al, Scientific Reports (2022)

A View on Data/Phenomena Interpretation

Understanding factors contributing to progression of metabolic syndrome (MetS)

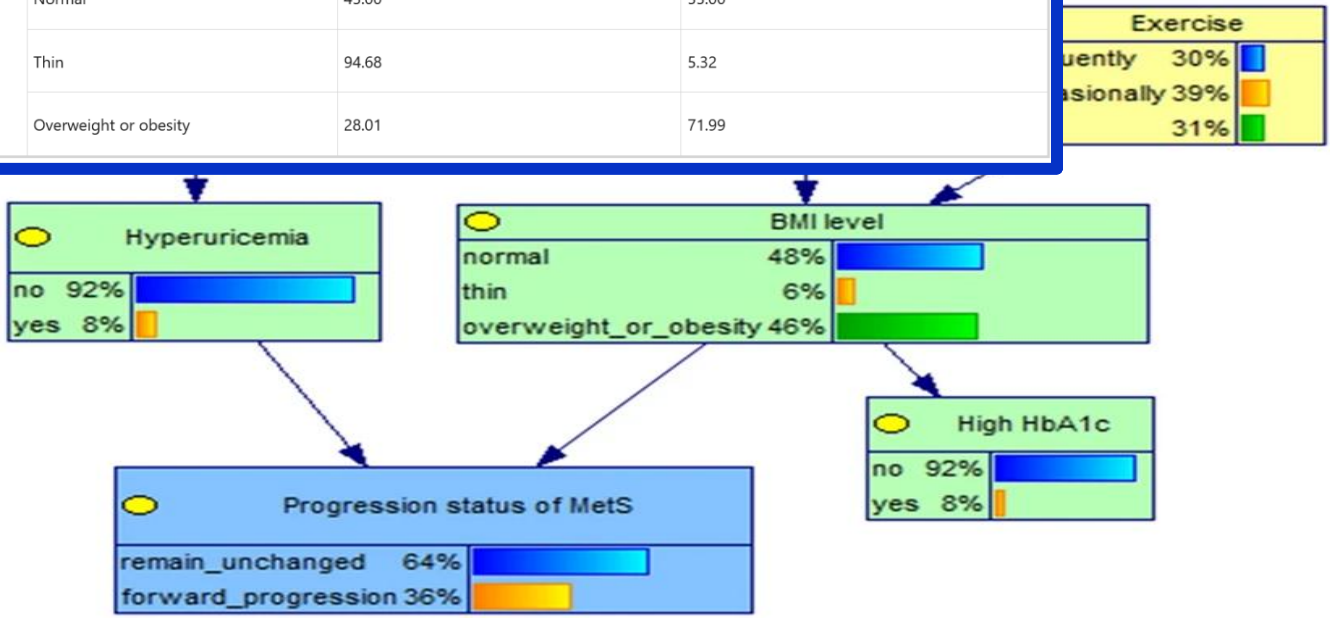


Razbek et al, Nature (2024)

A View on Data/Phenomena Interpretation

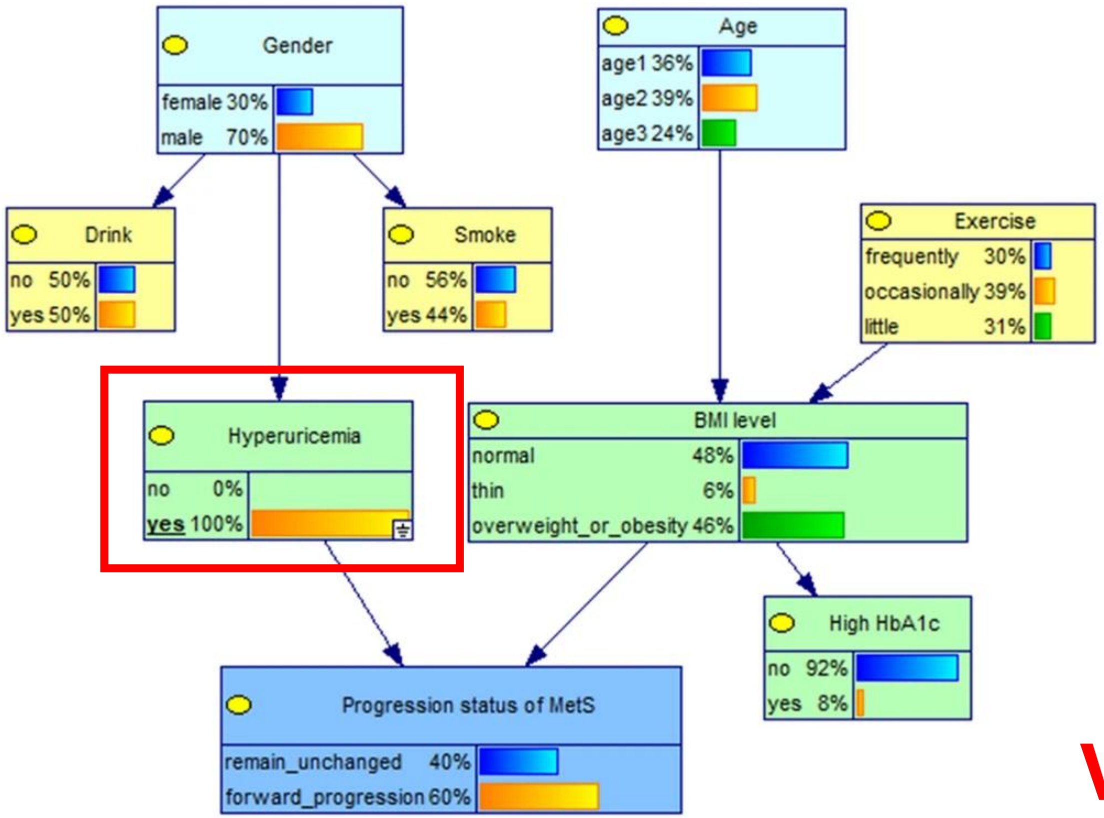
Conditional probability tables learned by maximum likelihood

Hyperuricemia	BMI level	Remain unchanged (%)	Forward progression (%)
No	Normal	70.61	29.39
	Thin	65.73	34.27
	Overweight or obesity	62.00	38.00
Yes	Normal	45.00	55.00
	Thin	94.68	5.32
	Overweight or obesity	28.01	71.99

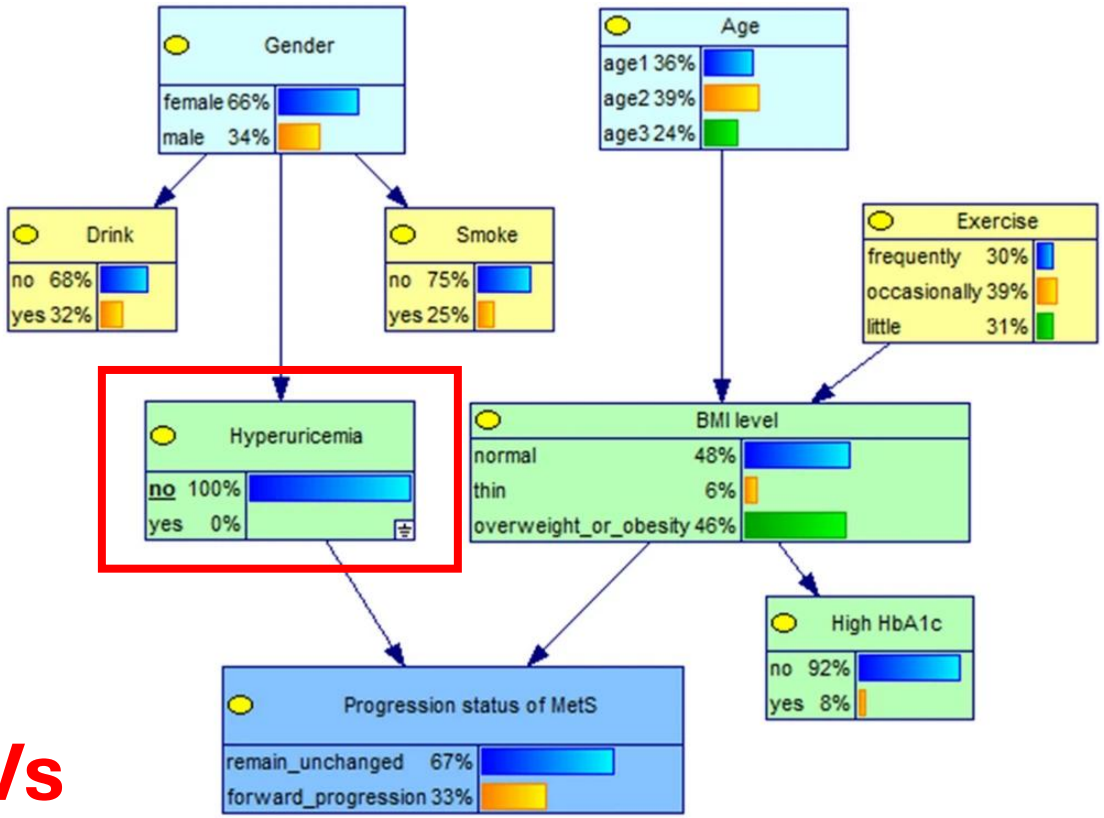


Razbek et al, Nature (2024)

Visually Comparing Differences Based on Changing Risk Factors

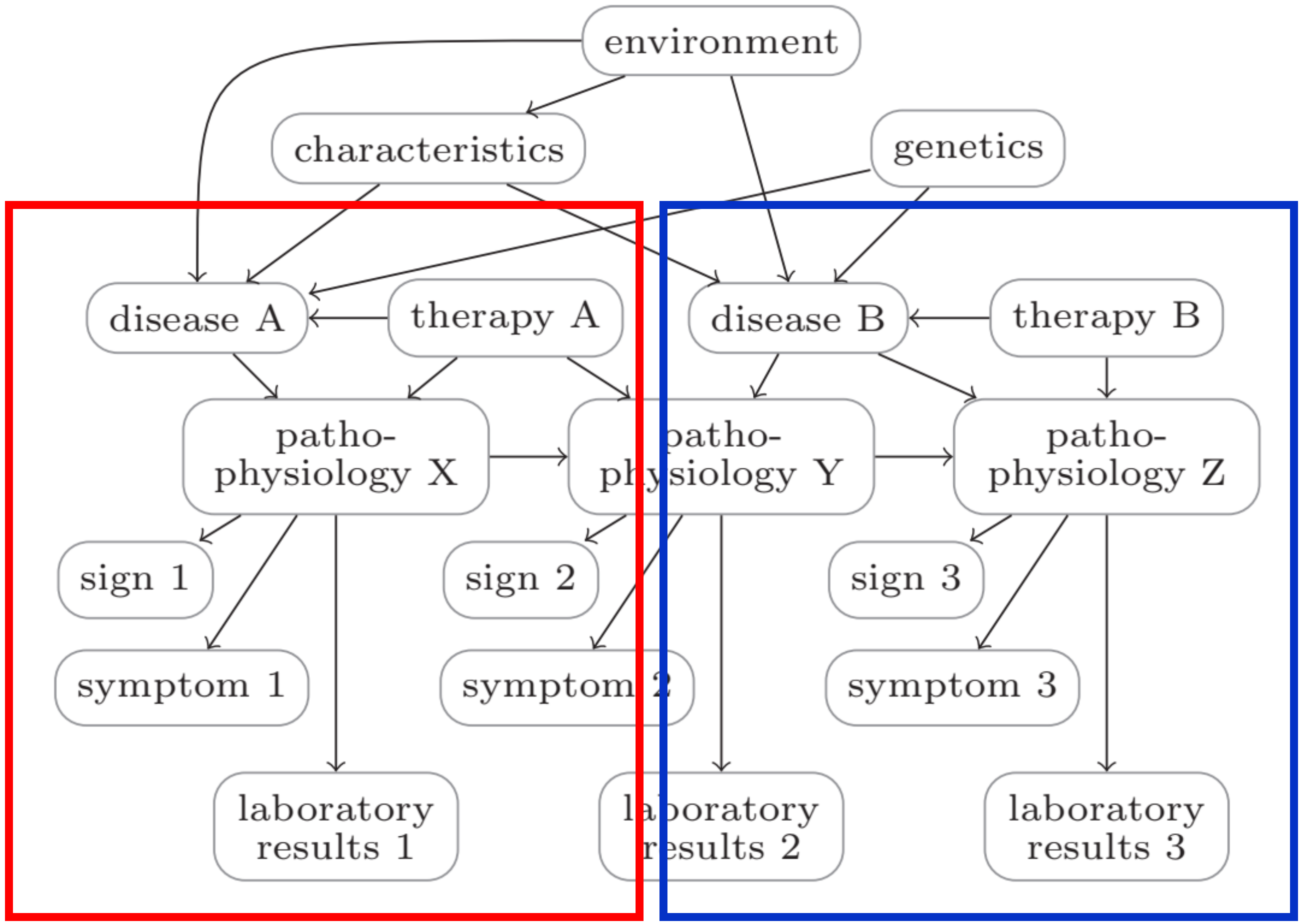


Vs



Razbek et al, Nature (2024)

Subpopulations in Bayesian Networks



Lappenschaar et al, Artificial Intelligence in Medicine (2013)

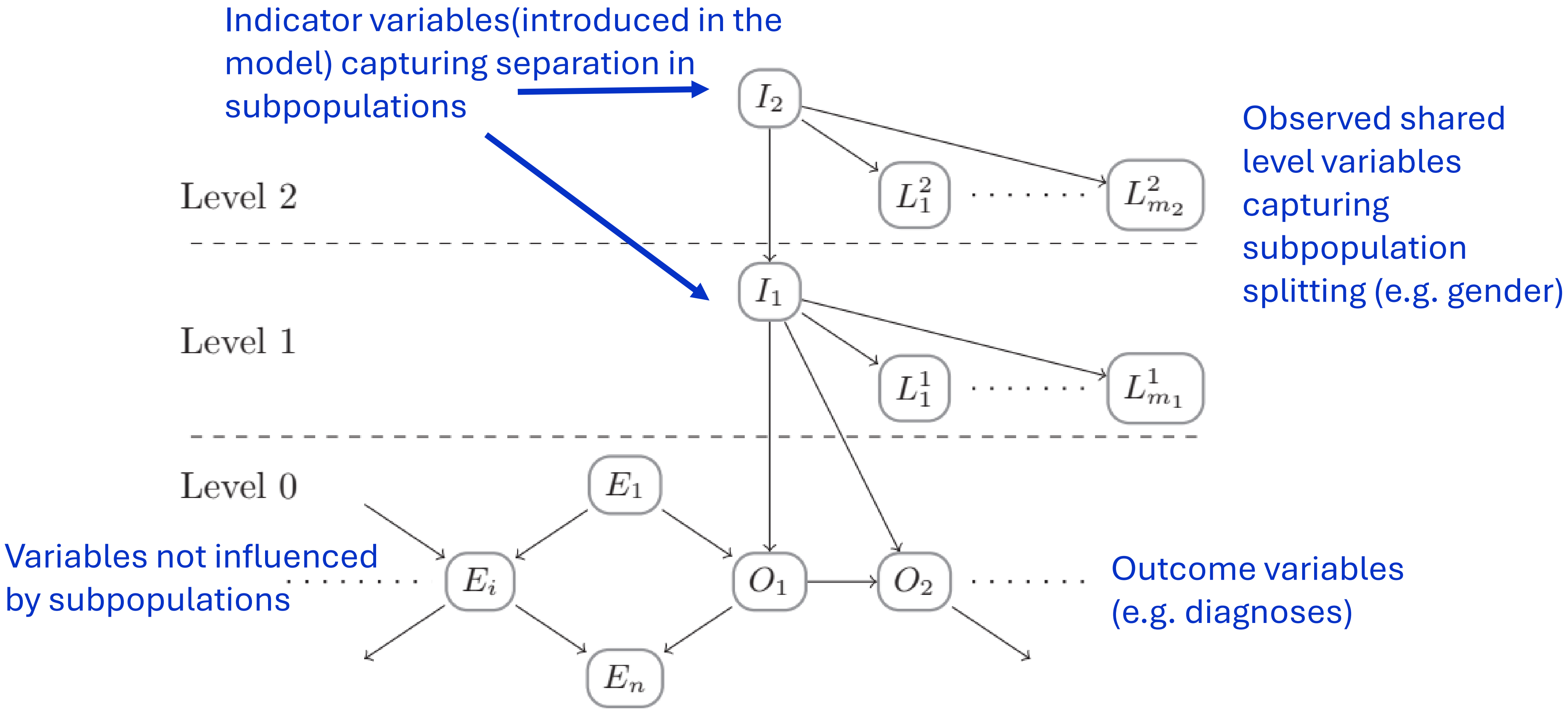
In multimorbidity problems datasets are typically collected from different sources

- family practices
- sub-populations (social, geographic, demographic)

We need to correct for this or we will have spurious interactions between disease variables

The gender influenced estimate of height in linear regression!

Multilevel Bayesian Networks



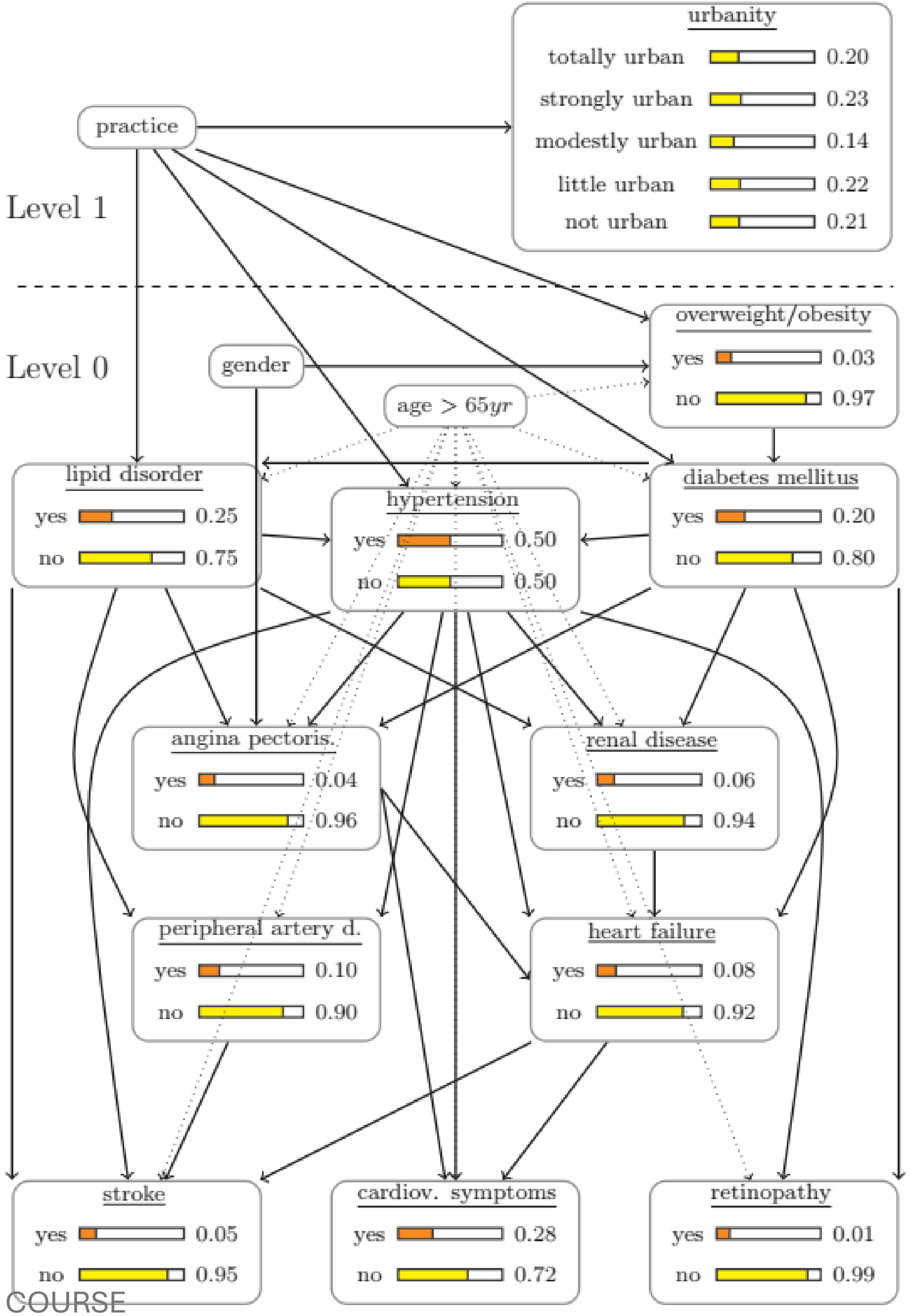
Lappenschaar et al, Artificial Intelligence in Medicine (2013)

Multilevel BNs for multi-disease prediction

Different subpopulation induced by the different practices (indicator) collecting data

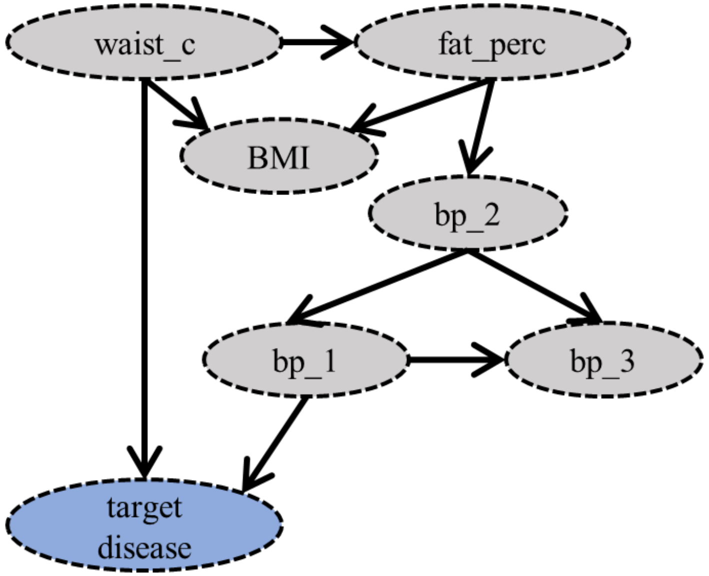
Different practices observable in their urbanity (level variable)

Lappenschaar et al, Artificial Intelligence in Medicine (2013)

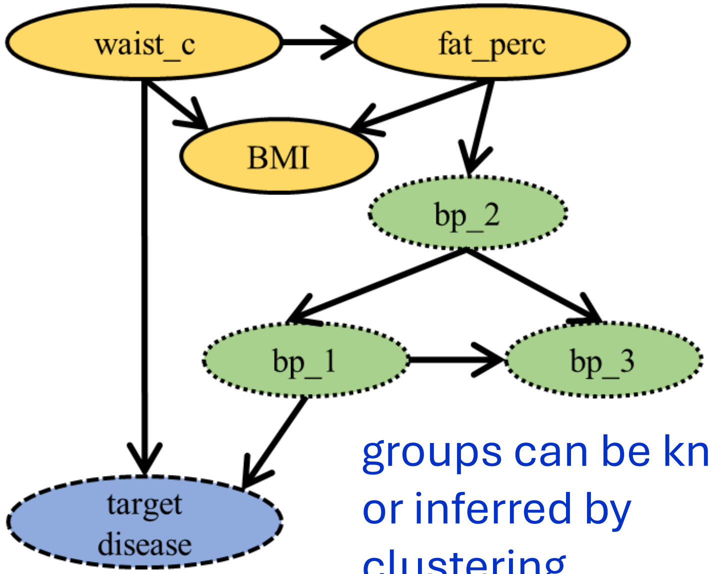


Modular Bayesian Networks

Define Bayesian networks over groups of features to improve interpretability

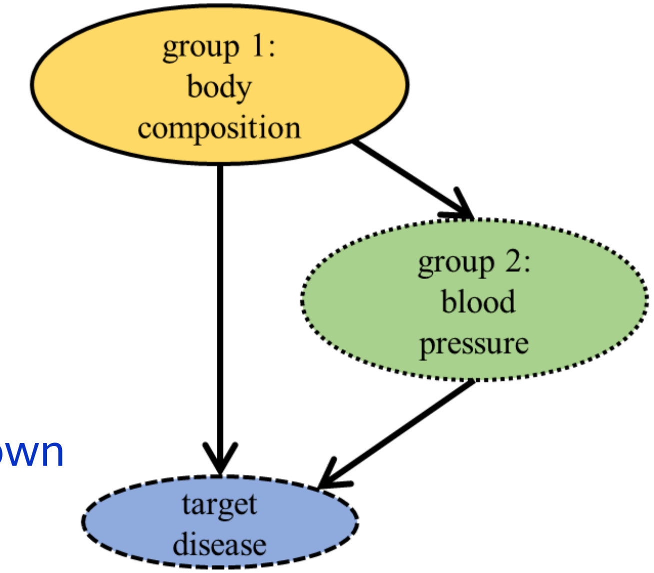


Bayesian network



groups can be known or inferred by clustering

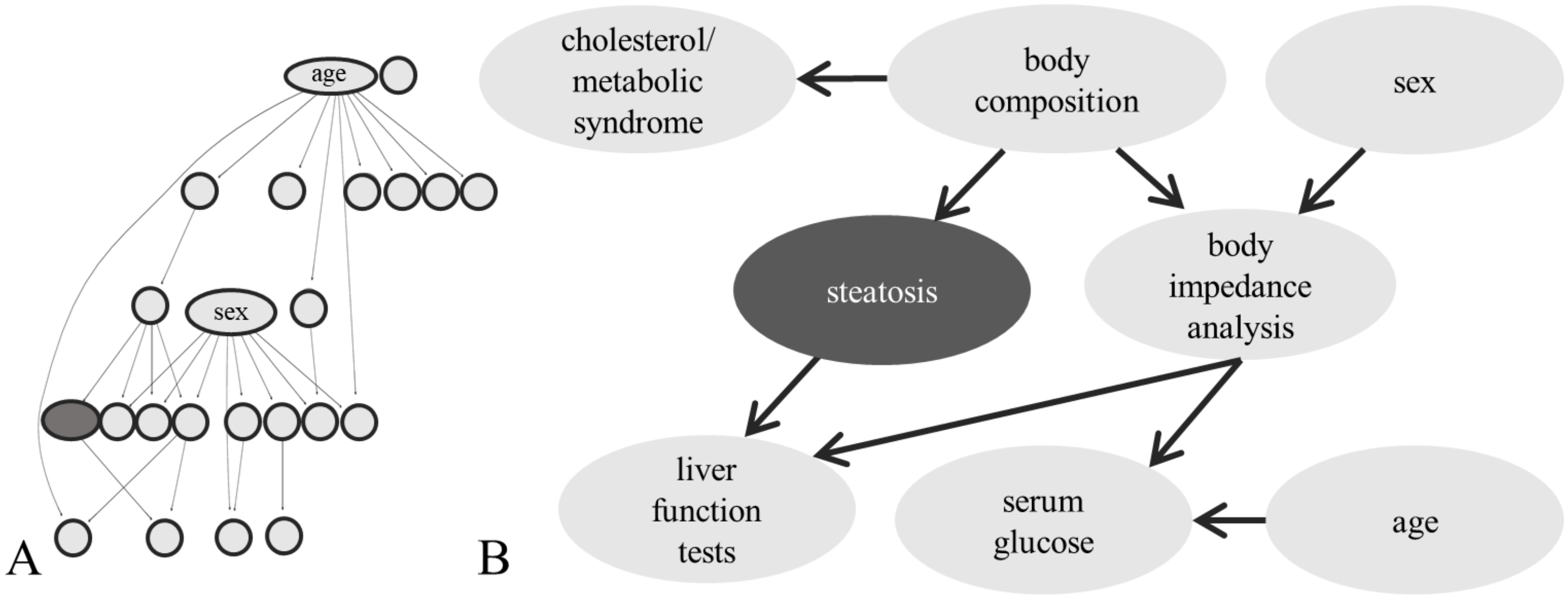
Bayesian network with variable grouping



Group Bayesian network

Becker et al, Plos Computational Biology (2021)

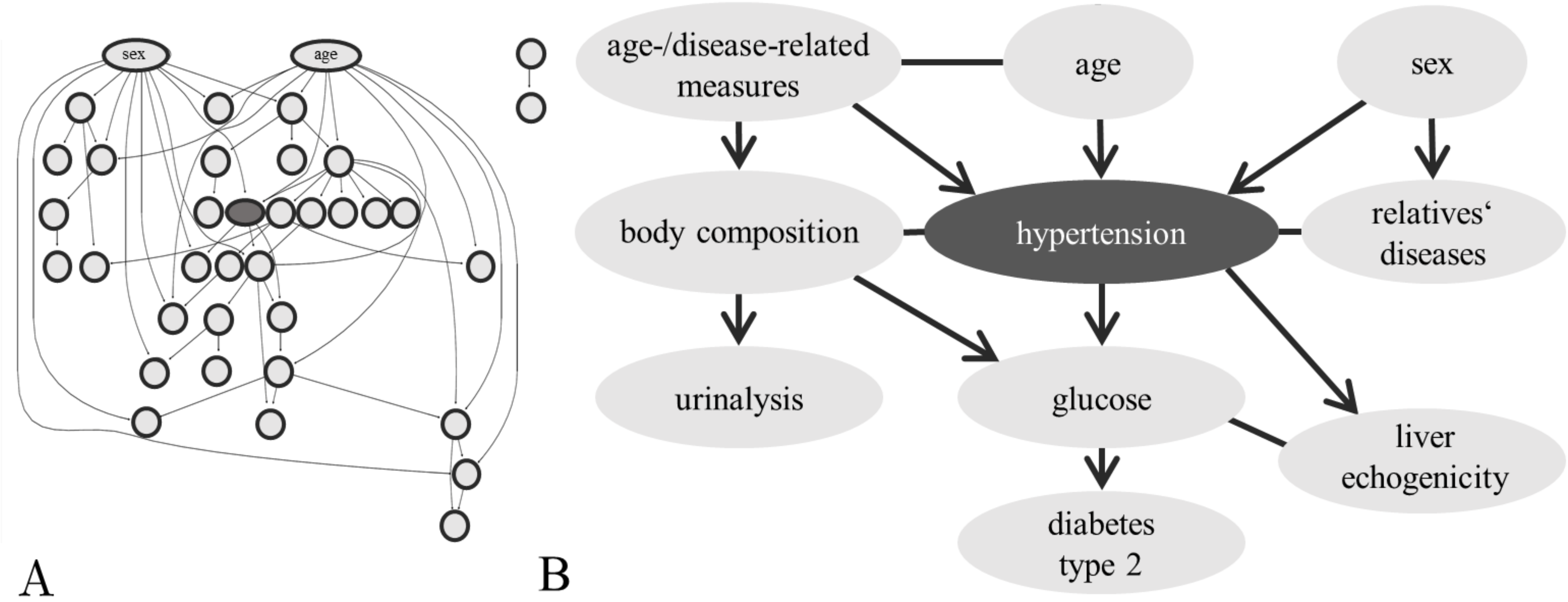
Modular BNs - Steatosis



Model	AUROC	± sd	AUPRC	± sd
logistic regression	0.82	±0.02	0.78	±0.03
detailed Bayesian network	0.55	±0.04	0.57	±0.06
group Bayesian network	0.80	±0.02	0.76	±0.04
refined group Bayesian network	0.84	±0.03	0.81	±0.02

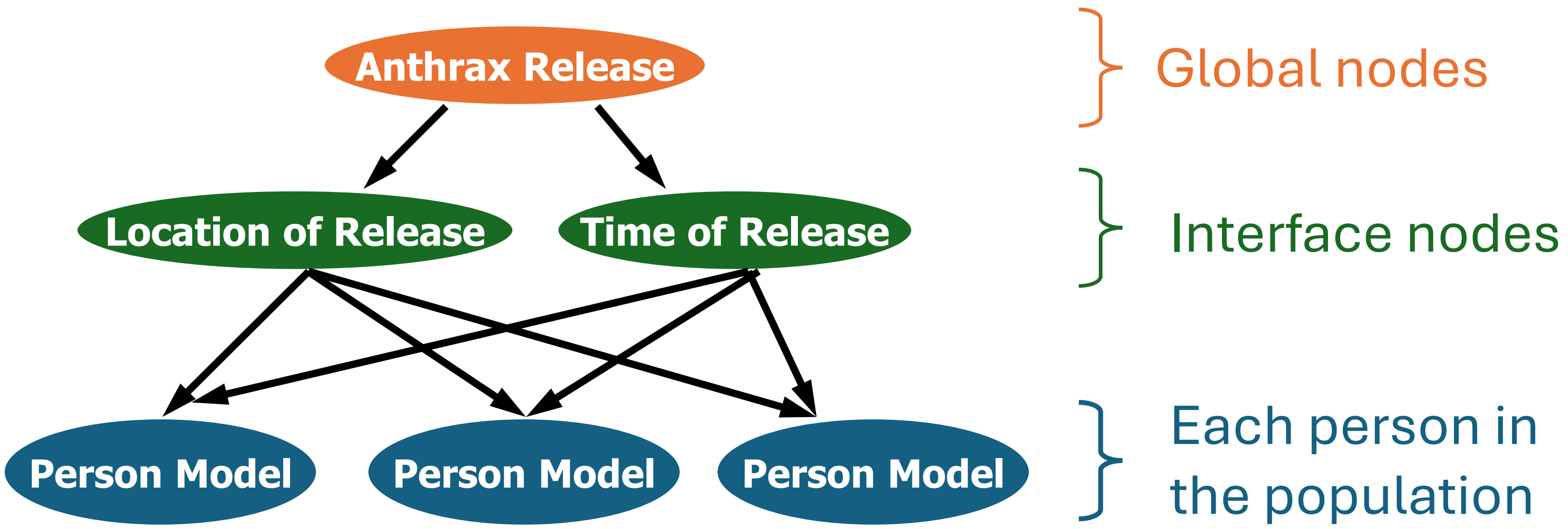
Becker et al, Plos Computational Biology (2021)

Modular BNs - Hypertension



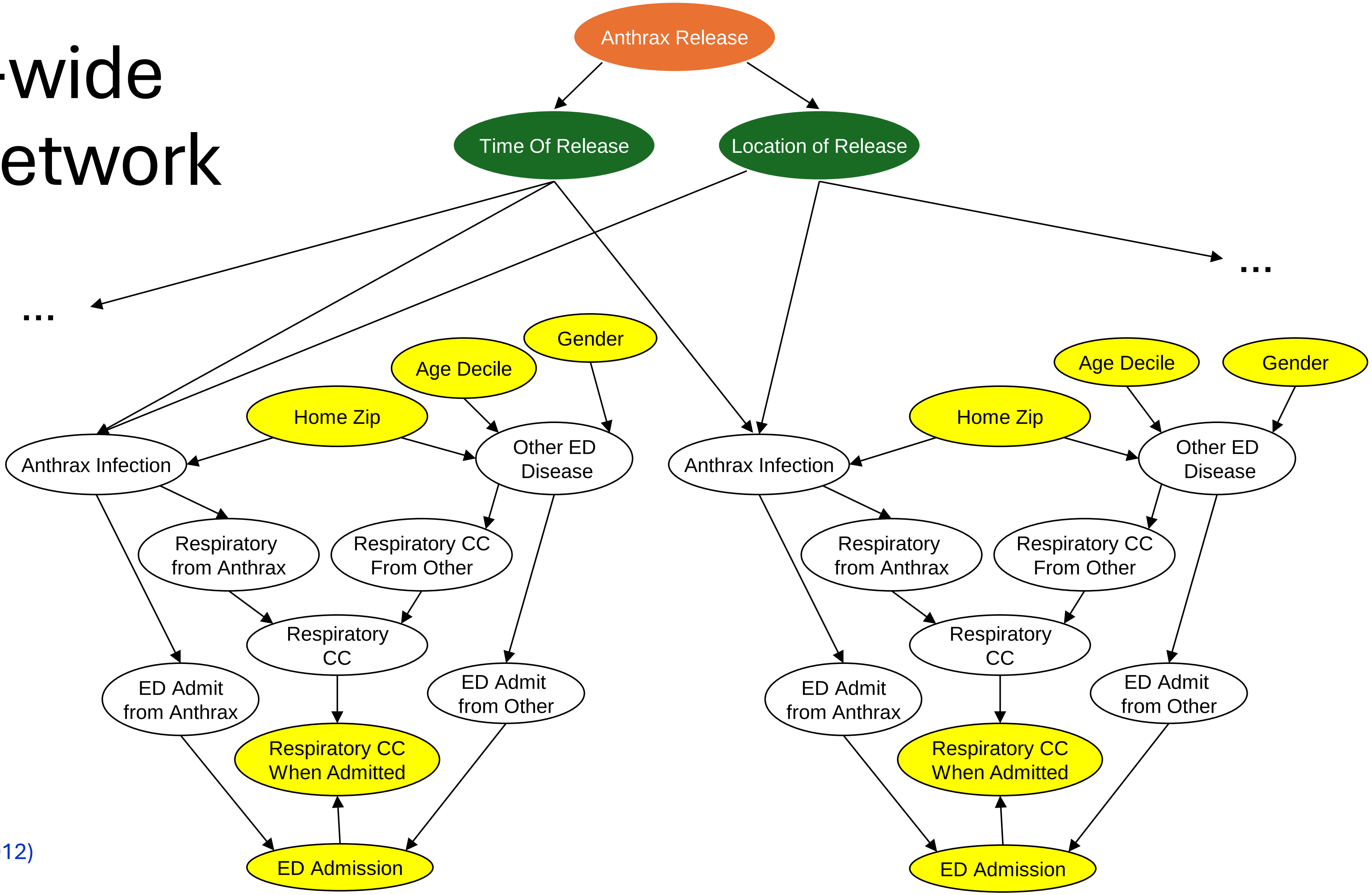
Becker et al, Plos Computational Biology (2021)

Population-wide Bayesian Networks



Cooper et al, Uncertainty in AI (2012)

Example of population-wide Bayesian Network



Cooper et al, Uncertainty in AI (2012)

Steps to Use Bayesian Networks

- Design the structure of the network by identifying variable (nodes) associations (edges)
- Fit the parameters of the Bayesian Network by maximum likelihood
- Make predictions (e.g. diagnose a disease)
- Sample observations (e.g. complete missing variables)
- Reason on associations

Next lecture

- ~~Design~~ Learn the structure of the network by identifying variable (nodes) associations (edges)
- Fit the parameters of the Bayesian Network by maximum likelihood
- Make predictions (e.g. diagnose a disease)
- Sample observations (e.g. complete missing variables)
- Reason on ~~associations~~ causal relationships

Wrap-up

Take home lessons

- Bayesian network represent asymmetric relationships between RV and conditional probabilities in compact way
- Allow to reason graphically on probabilistic concepts: we can easily map inference and conditional independence tests into graph-based algorithms
- Learning is easily achieved by maximum likelihood when all RV are observed
- Useful features for healthcare applications
  - Reasoning under uncertainty
  - Integration of prior knowledge
  - Interpretability
- Very parametric: only as good as your ability to take design choices (distribution, independence,...) that are close to the underlying data/task process

Next lecture

- ~~Design~~ Learn the structure of the network by identifying variable (nodes) associations (edges)
- Fit the parameters of the Bayesian Network by maximum likelihood
- Make predictions (e.g. diagnose a disease)
- Sample observations (e.g. complete missing variables)
- Reason on ~~associations~~ causal relationships