



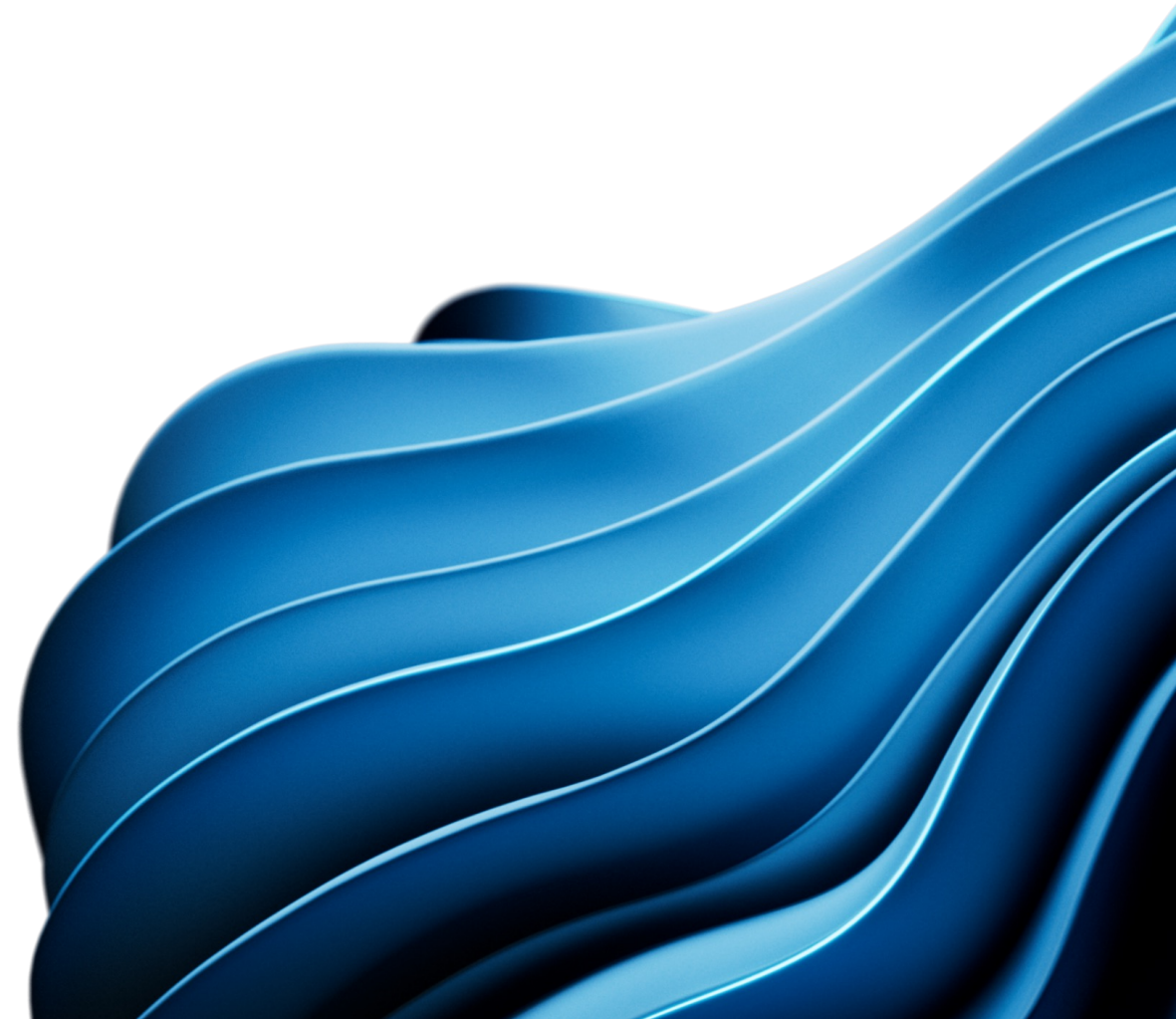
CausalNex

CausalNex

Explainable modelling and causal inference

Emily Jones, Senior Data Scientist

Confidential and proprietary: Any use of this material without specific permission of McKinsey & Company is strictly prohibited



Agenda

Introduction to ML, causality & Bayesian Networks (15mins)

Structure learning (10mins)

Inference & interventions (10mins)

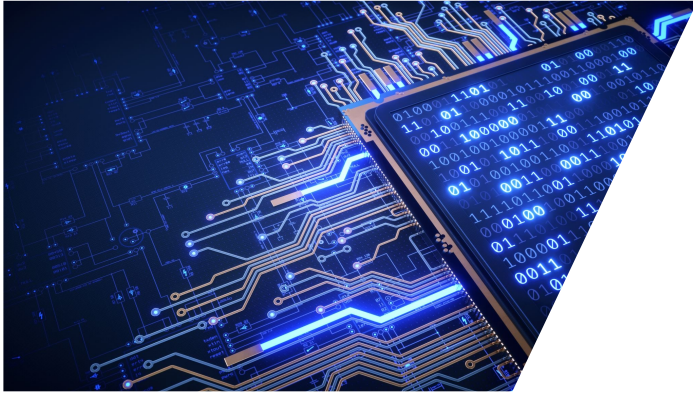
CausalNex demo (10mins)

Epilogue (5mins)

Questions (10 mins)

Introduction to causality & Bayesian Networks

The world of machine learning – in a nutshell



Predictive modelling

- Estimate the target for new observations

$$y = f(x)$$



Explanatory modelling

- Describe the effect that a change of certain inputs has on the target

$$y = f(x)$$



Optimisation

- Find the inputs that give optimal performance

$$y = f(x)$$

Key decisions require explanatory models

- Which medication will help a given patient?
- What marketing campaign will be most effective?
- How can a pharmaceutical company reduce non-conformities during their drug manufacturing process?
- What changes can a vehicle manufacturer make to their new product development process to reduce lead time?
- How can a company deploy resources to better serve customers?

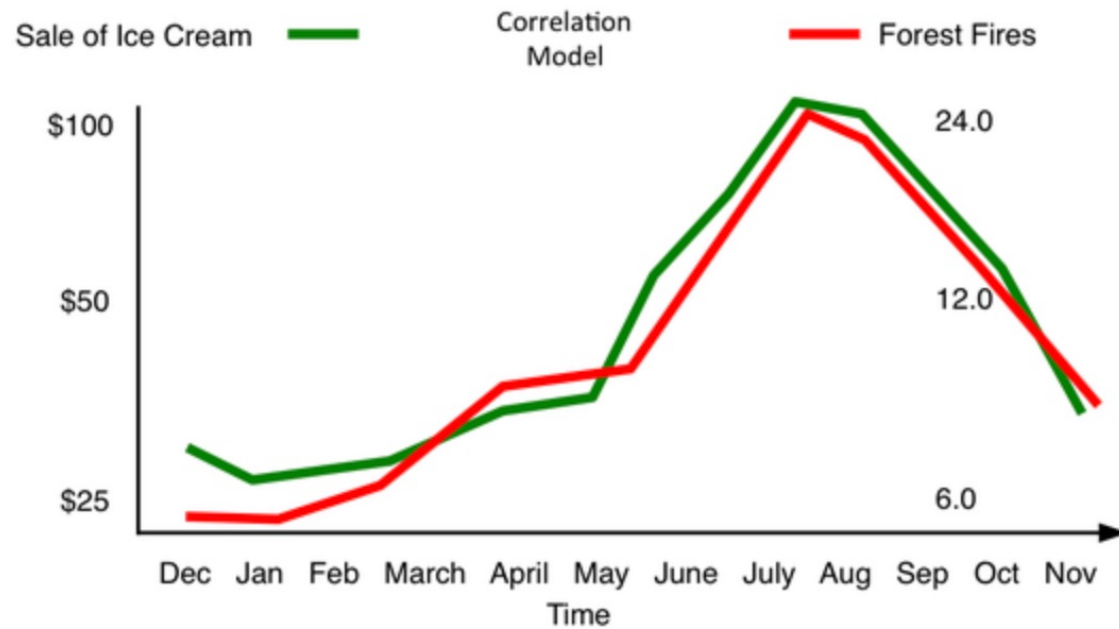


Correlation is not causation...

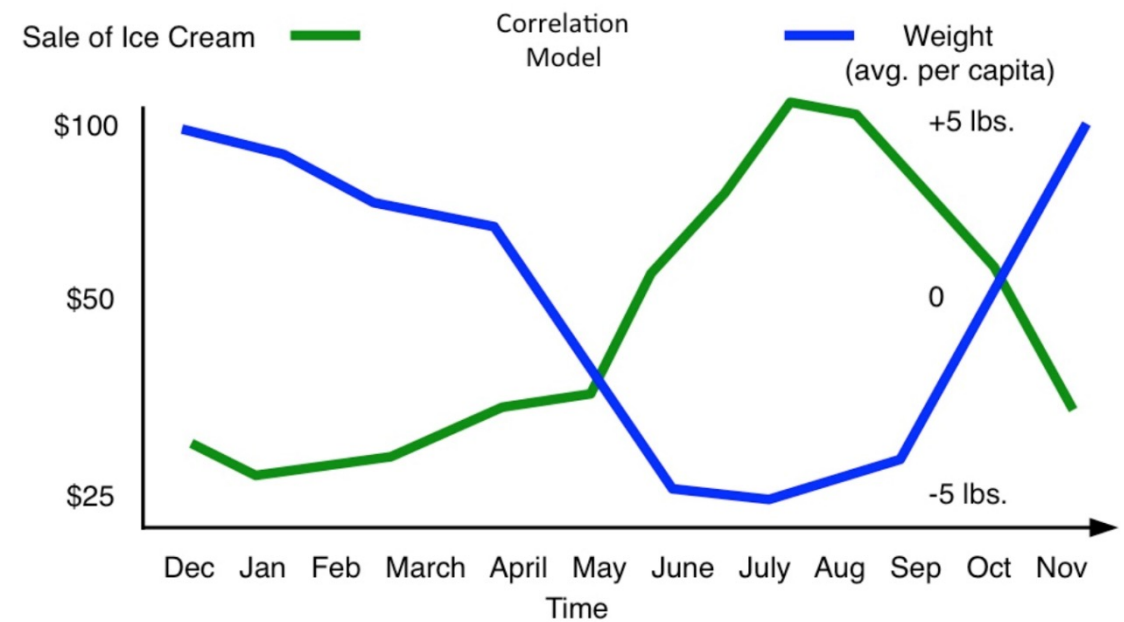


We'd expect explanatory models to make causal sense before acting on their recommendations

Does ice cream cause forest fires?



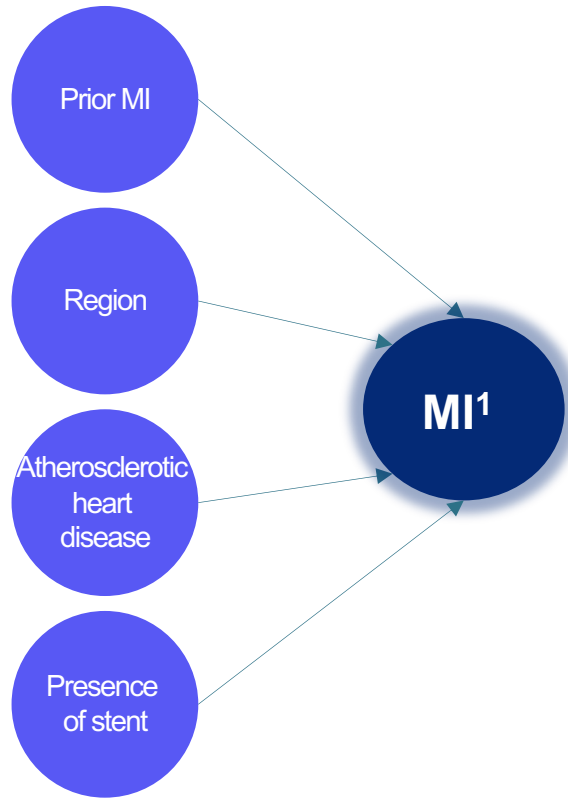
Is ice cream the new diet food?



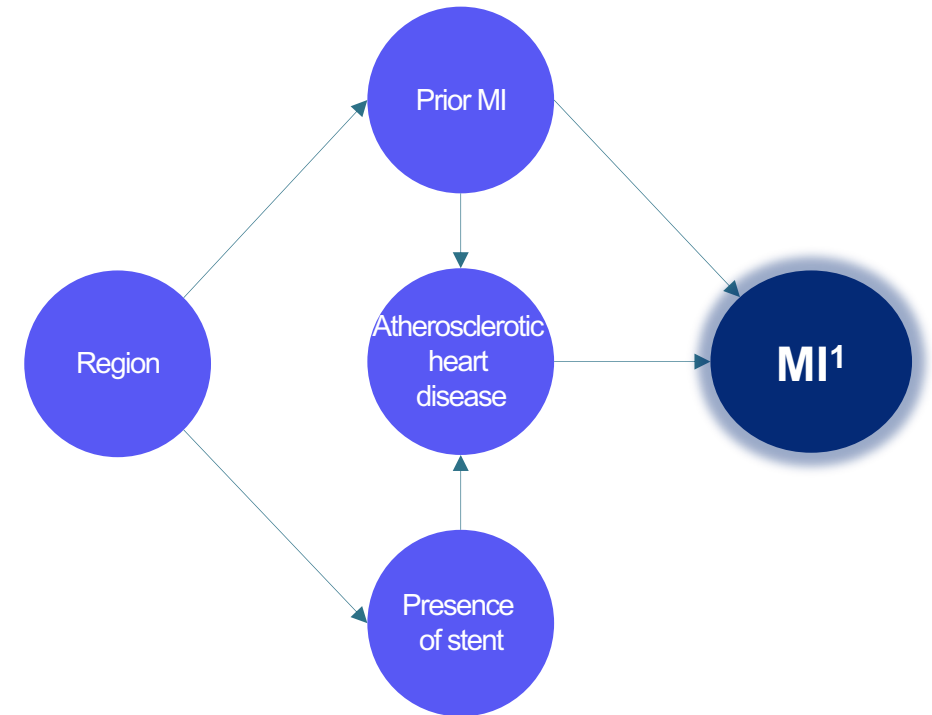
Comparison with linear regression

- Linear regression assumptions: all X are independent of each other; all X could have effect on y
- Regression could show that a patient's region has biggest impact on MI!

Linear regression



Bayesian network



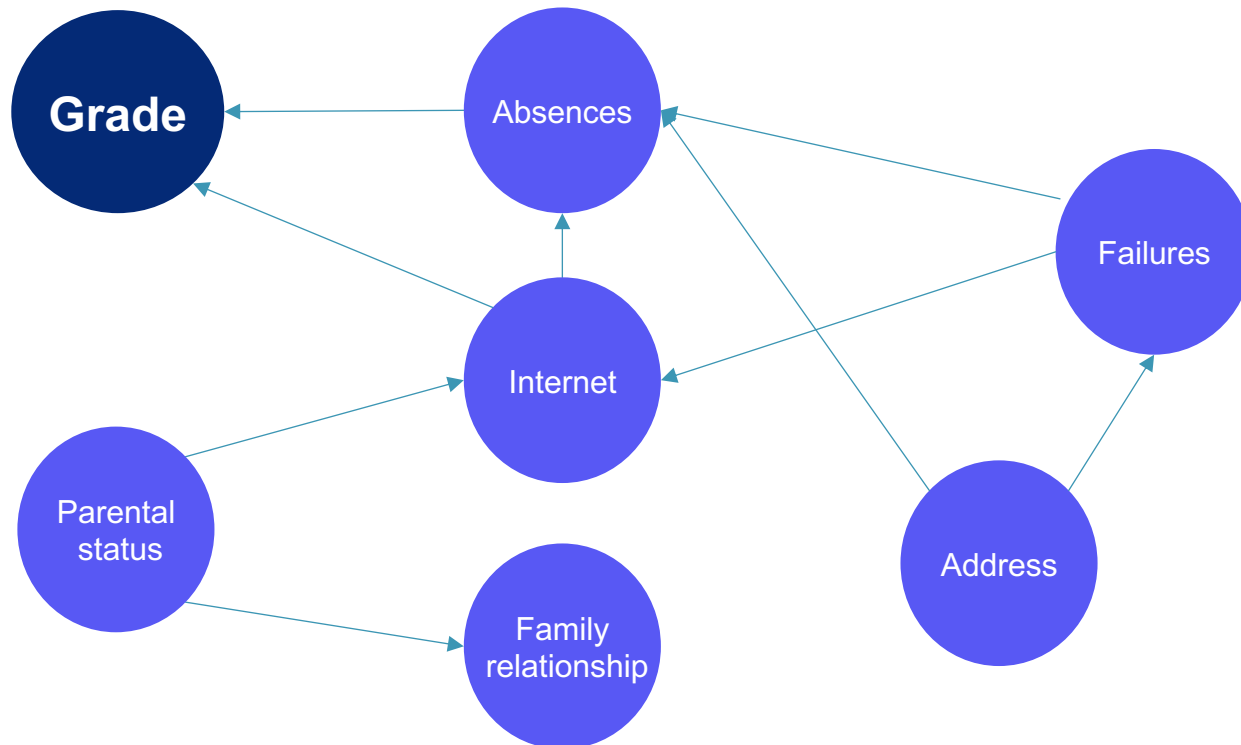
Bayesian networks

Use a **directed acyclic graph** to capture interdependencies between variables

Nodes represent variables

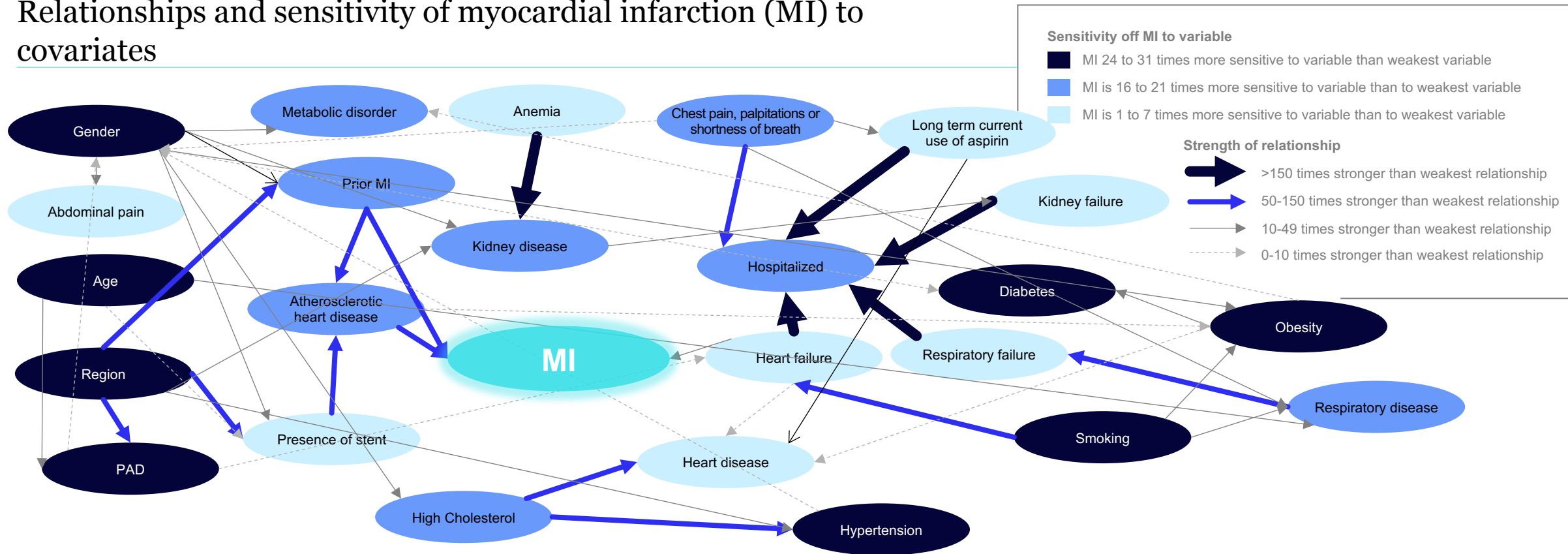
Edges represent relationships between variables

- $A \rightarrow B$ = “B depends on A”, or more precisely, “*The value of a node is independent of the rest of the variables in the graph given its parents.*”



Causal models can be used to support decision making in important domains such as healthcare

Relationships and sensitivity of myocardial infarction (MI) to covariates



- The network structure is generated from both data and domain knowledge.
- Incorporating domain expertise ensures the model represents a domain expert's view of causal relationships
- Quantifying the relationship between patient demographics, comorbidities, and cardiovascular events can be used to identify key drivers of patient risk

CausalNex is an **open-source** Python library that leverages **Bayesian Networks**

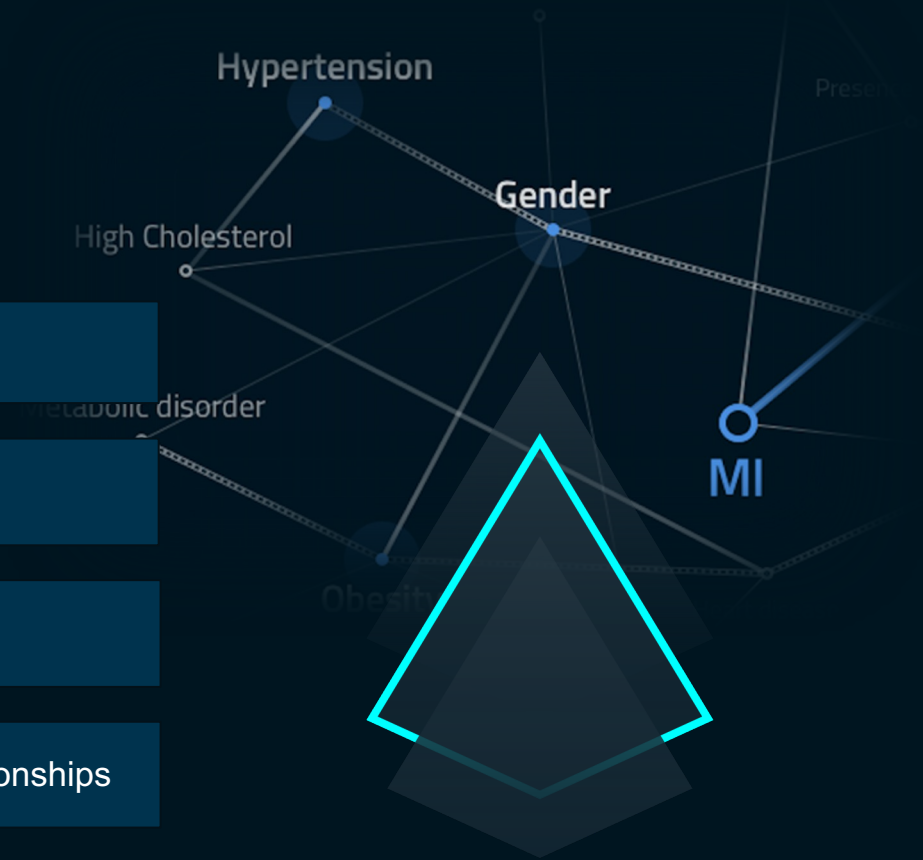
Structure learning – Learn relationships in data with NOTEARS, a state-of-the-art algorithm

Embed domain expertise – Enable experts to add and remove inaccurate relationships

Graph visualisation – Use extensions of NetworkX to help communicate results

Likelihood estimation – Estimate the probability distribution of variables based on their relationships

Counterfactual analysis – Infer what happens to *target Y* when *feature X* is changed



CausalNex

Powered by QuantumBlack

Status:



QuantumBlack Labs



PyPI

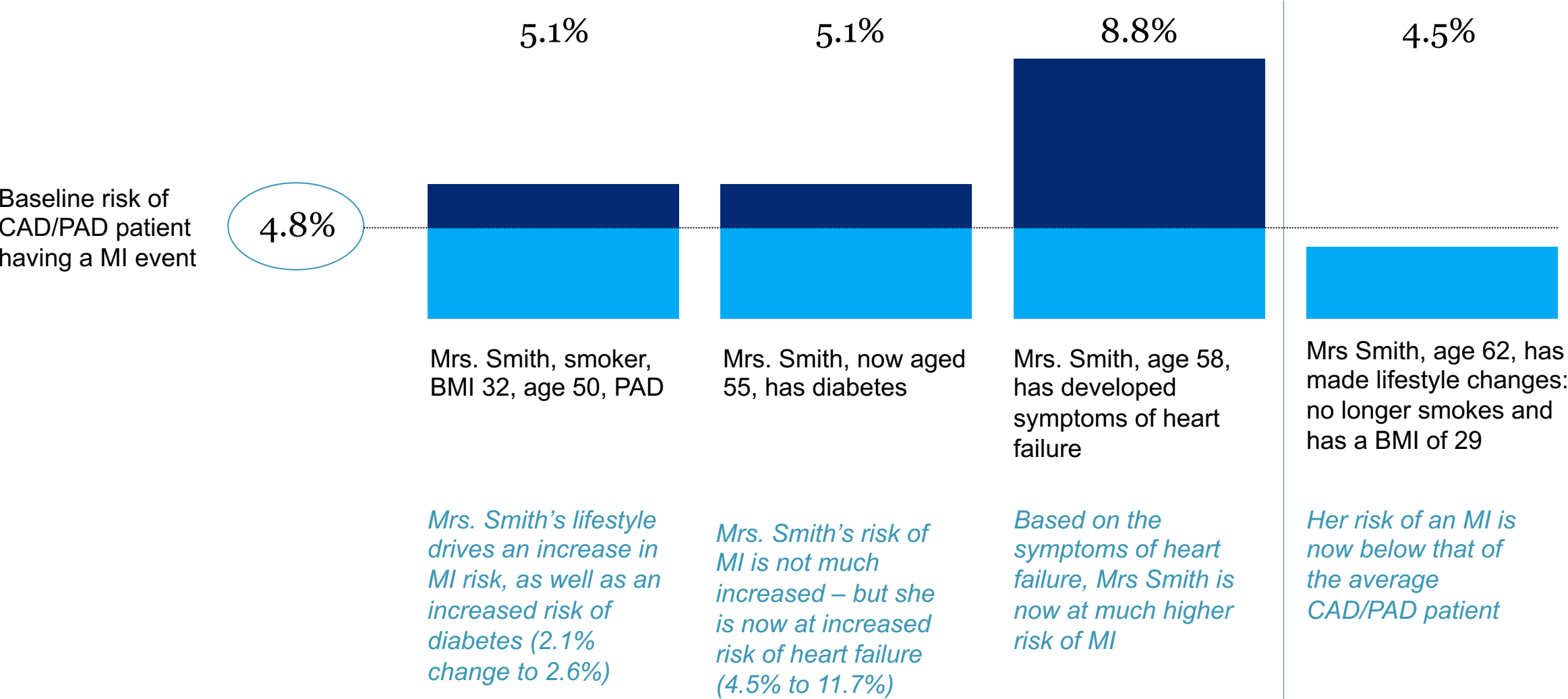


Read The Docs

```
`pip install causalnex`
```

With this approach we could better understand patient journeys

Risk of MI within 12 months¹



Structure Learning

Defining the structure of a Bayesian network

Domain Expertise

- Present all variables to an expert
- Ask them to tell us all relationships
- Experts should consider evidence hierarchy

Challenges

- How to deal with scale?
- What if we miss some relationships?
- How to deal with higher-order causal effects?

We can use Machine Learning to help experts!



Structure learning



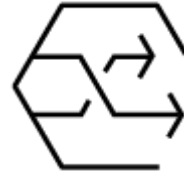
Evaluation

Score-based

- Find graph that maximizes a specified score
- Common scores: BIC, MDL, BDe, BDeu

Constraint-based

- Start from complete graph, delete edges between nodes that are conditionally independent (CI), orient edges
- Can choose CI criterion: Chi-squared test, G test



Search methods

- Greedy local search
- Dynamic programming
- Integer linear programming
- Global continuous optimization

- Full search
- Improved by testing CI in the right order



Output type

Directed acyclic graph (DAG)

Equivalence class of DAGs called completed partially directed acyclic graph (CPDAG)

Practical considerations for structure learning

- NP-hard problem due to **large search space** and **combinatorial acyclicity constraint**
- **Data type**: discrete, continuous, or mixed?
- **Time-varying data**: do we need a dynamic Bayesian network?
- **Model complexity**: linear model (needs less data) or nonlinear model (more flexible)?
- Do we need to account for **confounders** or **missing data**?

Practical considerations for structure learning

- NP-hard problem due to **large search space** and **combinatorial acyclicity constraint**
- **Data type**: discrete, continuous, or mixed?
- **Time-varying data**: do we need a dynamic Bayesian network?
- **Model complexity**: linear model (needs less data) or nonlinear model (more flexible)?
- Do we need to account for **confounders** or **missing data**?

CausalNex includes an implementation of DAGs with NO TEARS, a continuous optimization algorithm (Zheng et al.)

Formulation

- NOTEARS¹ is a score-based method. The objective is to optimize some score $F(W)$ subject to the weighted adjacency matrix W corresponding to a DAG.
- The authors propose an approach to convert the combinatorial optimization problem (left) into a **continuous** problem (right):

$$\begin{array}{ccc} \min_{W \in \mathbb{R}^{d \times d}} F(W) & & \min_{W \in \mathbb{R}^{d \times d}} F(W) \\ \text{subject to } G(W) \in \text{DAGs} & \iff & \text{subject to } h(W) = 0 \end{array}$$

- The loss function F incorporates a **smooth loss** and an **L1 regularization** term that encourages sparsity.
- Graph is a DAG if and only if $\text{trace}(W^k) = 0$ for all k :

- The key breakthrough is a **novel acyclicity constraint**.
- Graph is a DAG if and only if $\text{trace}(W^k) = 0$ for all k , or equivalently:

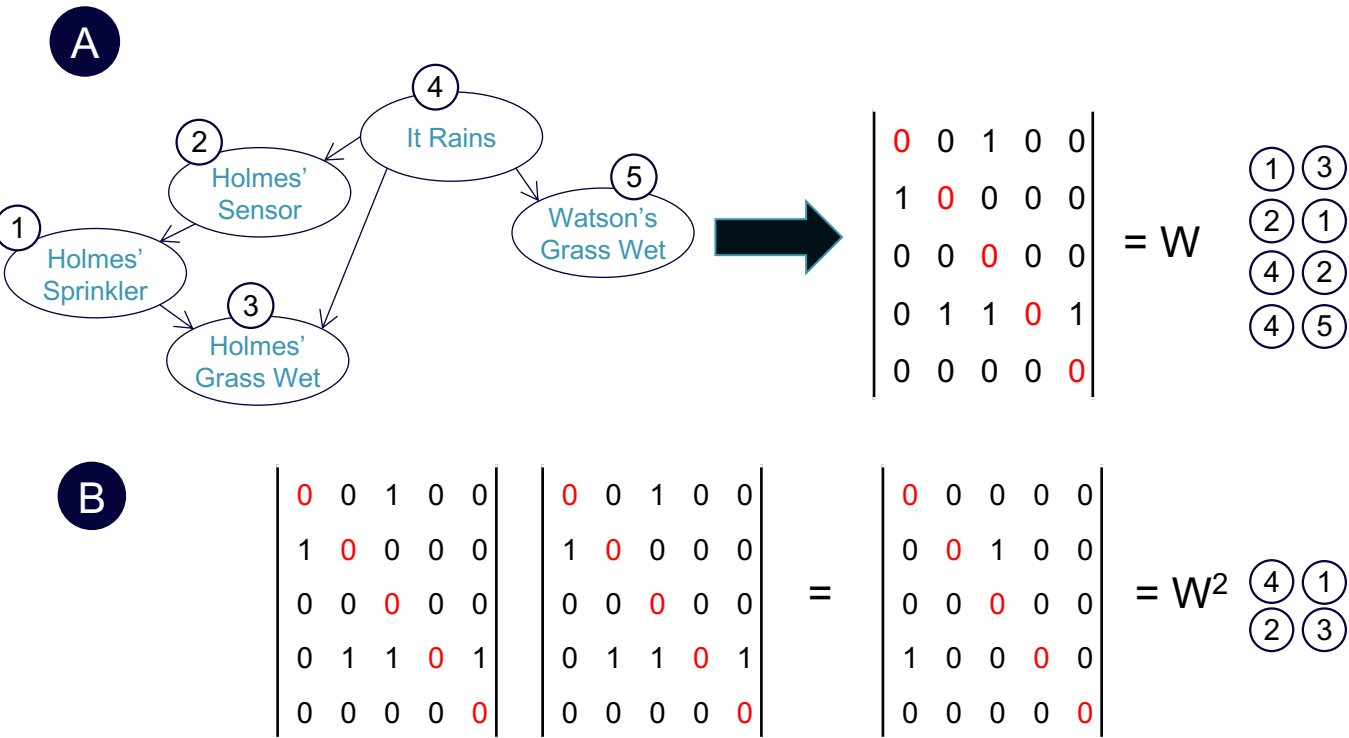
$$\sum_{k=1} \sum_i^d \frac{(W^{2k})_{ii}}{k!} = \text{Trace}(e^{(W \odot W)}) - d = 0$$

- The authors use a **least-squares loss** based on a linear model, but any smooth loss is compatible with the approach.

$$F(W) = \ell(W; \mathbf{X}) + \lambda \|W\|_1 = \frac{1}{2n} \|\mathbf{X} - \mathbf{X}W\|_F^2 + \lambda \|W\|_1.$$

CausalNex includes an implementation of DAGs with NO TEARS, a continuous optimization algorithm (Zheng et al.)

Previous techniques suffered because they needed to “check acyclicity holds” and this is a combinatorial optimization problem. The authors of *DAGs with NO TEARS* (Zheng et al.) convert this to a continuous test (that is faster and easier to incorporate into search algorithms), leveraging the properties of the adjacency matrix

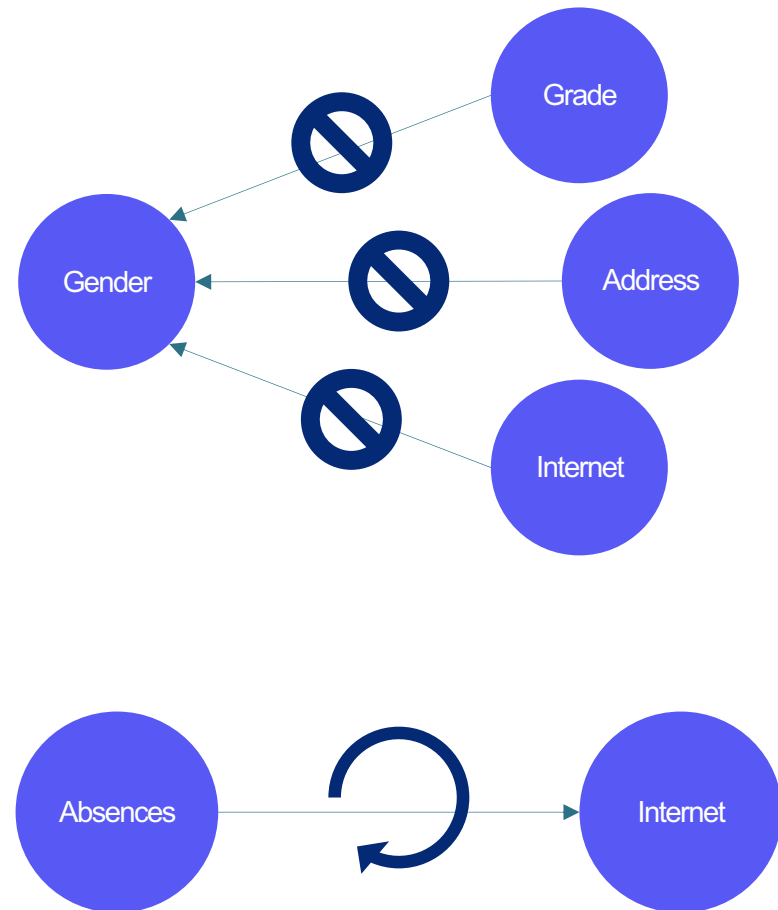


- A** The **leading diagonal** (or trace) of a DAG’s adjacency matrix, W , is all **zeros**.
- B** Raising W to a power, k will produce all possible paths k steps away. In a DAG, $\text{trace}(W^k) = 0$ for all k .
- $\text{trace}(W^k) = 0$ for all k is true iff:

$$\sum_{k=1} \sum_i^d \frac{(W^{2k})_{ii}}{k!} = \text{trace}(e^{(W \odot W)}) - d = 0 (< \epsilon)$$

Structure learning does not guarantee causality

- Structure learning algorithms make a best guess at direction – don't expect them to be correct
- **Always get experts to review the structure**
- Domain knowledge prior to structure learning
 - Constrain search space via tabu / required edges
- Domain knowledge after structure learning
 - Add / remove / flip edges



Each node has data that must be modelled by either a continuous or discrete distribution

Continuous

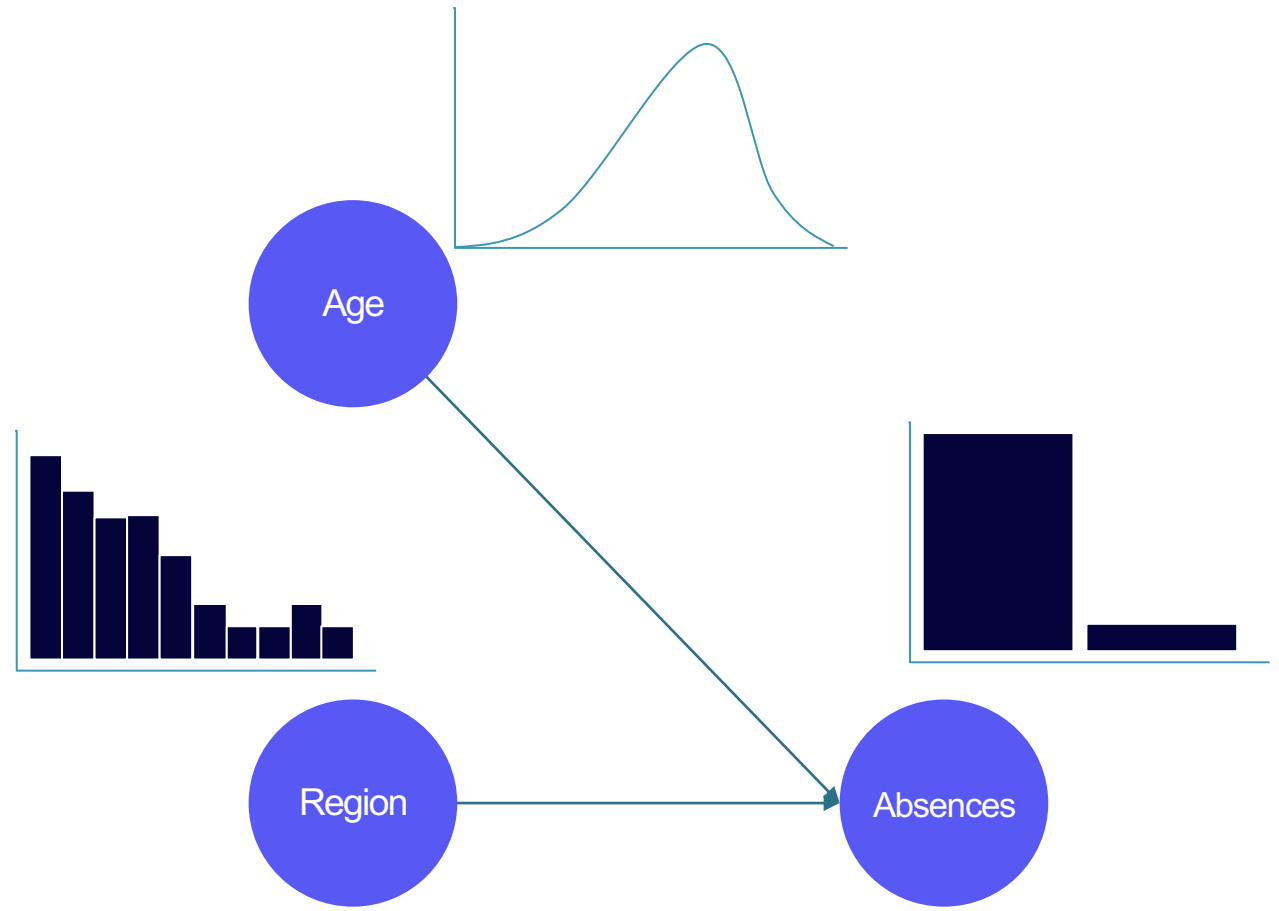
- Often assumes Gaussian distribution

Discretized

- Allows for data not normally distributed
- Flexible granularity, with continuous data bucketed as appropriate
- Discrete or categorical data can also be grouped if no significant difference between classes

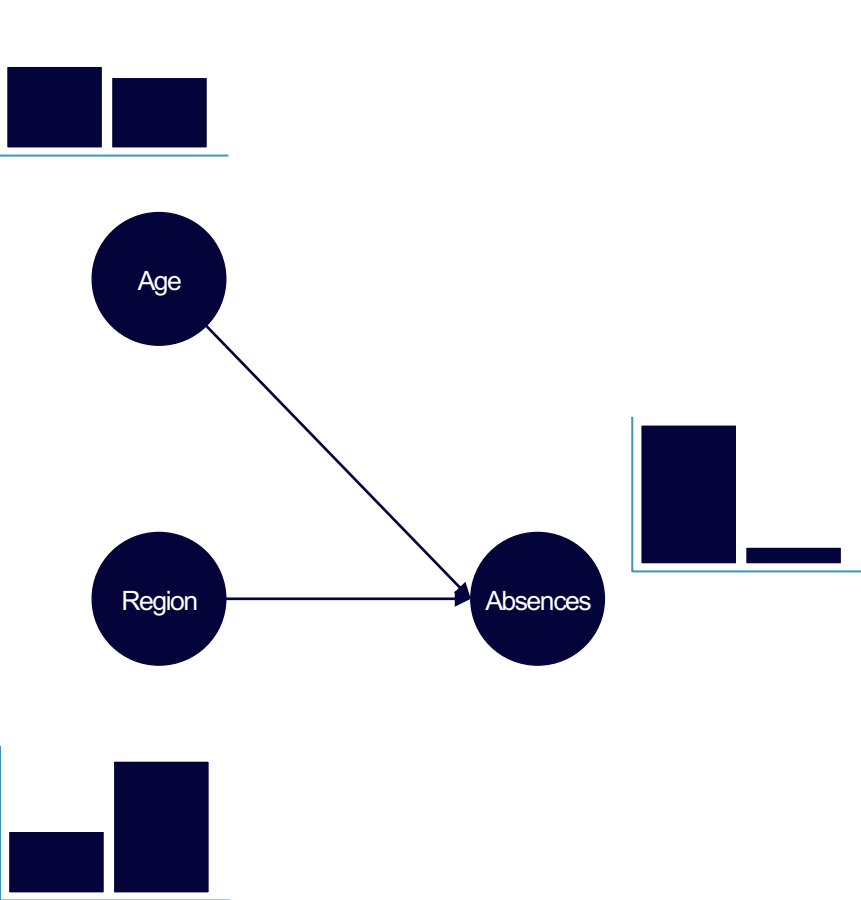
We use discretized

- Real world data is not Gaussian
- Flexibility and control over distributions

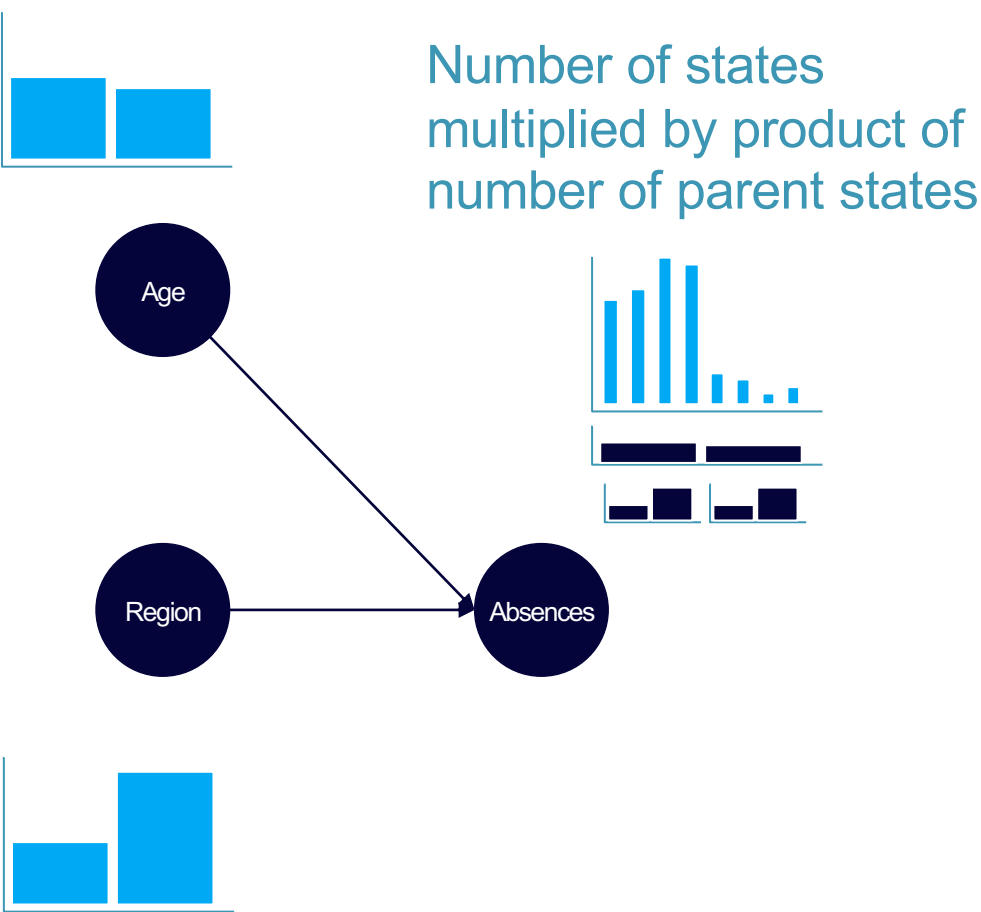


Behind every node in a Bayesian network is a conditional probability distribution (CPD)






Data distributions



Conditional probability distributions



Experts and data scientists should decide on an appropriate discretisation for each variable

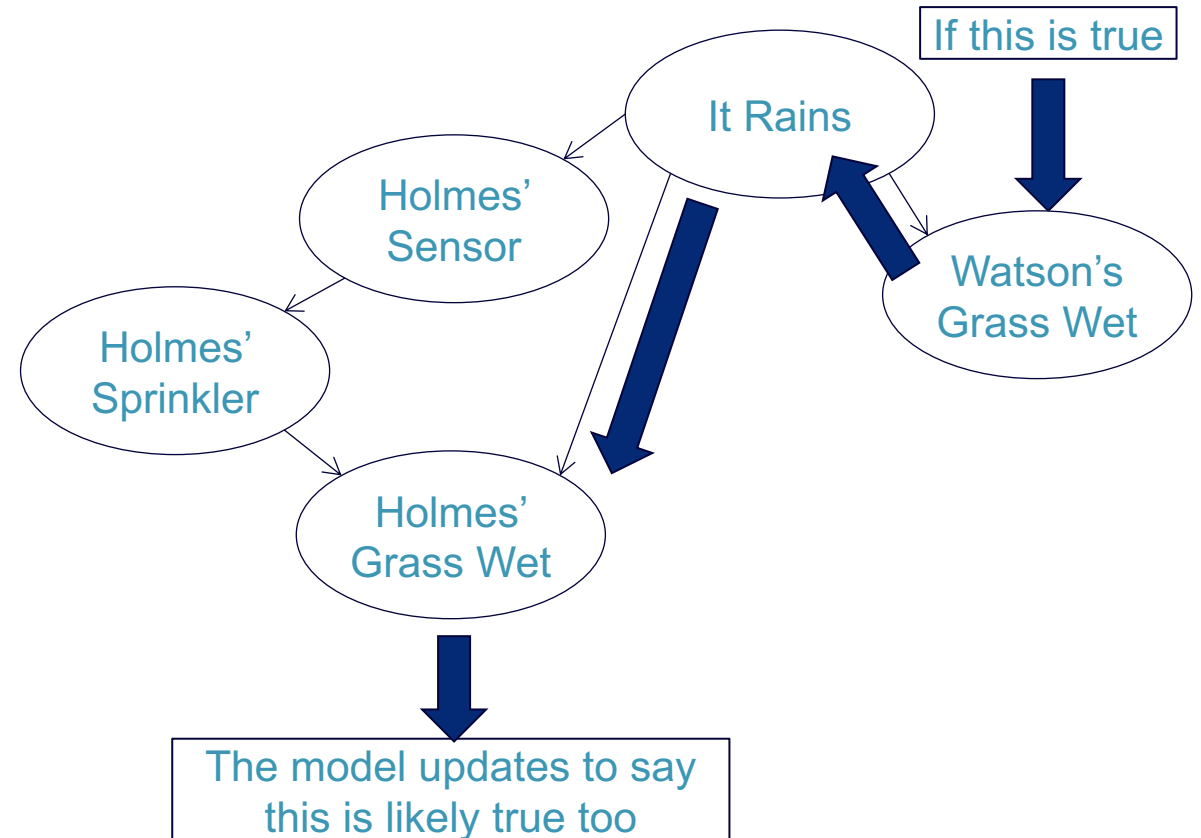
Method	Example	When To Use	
			
Equal Width		Known linear mappings	days -> years
Percentile		Desire to make comparisons	top X% of earners
Outliers		Interested in boundary cases	unusually tall
Fixed		Replicate industry-standard buckets	% -> Grade

Inference & interventions

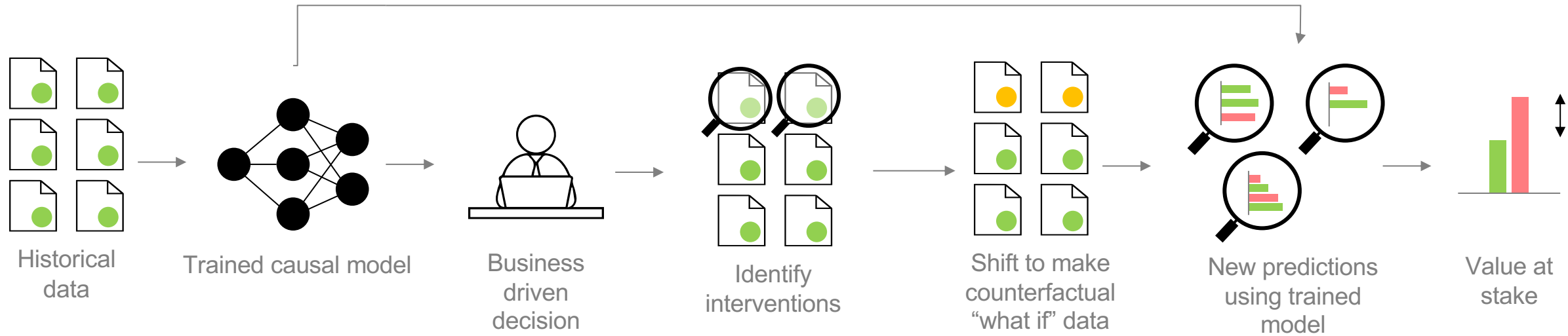
Learned models can be used to identify the ‘most important’ relationships between variables and update when given new evidence

How can we use Bayesian Networks?

- The probabilities of variables in Bayesian Networks **update as observations are added to the model**. This is useful for inference, and for beginning predictive analytics
- Metrics can help us **understand the strength of relationships between variables** and identify key drivers of change. These will be nodes that are most valuable to target in interventions
- We can leverage the fact that variables interact with each other to **run advanced value-at-stake (counterfactual) analysis**. This assesses the combined effect of actions without making the naive assumption that any two actions are independent.



More generally, if we model causally we can apply data science to business problems and perform counterfactual analysis to ask “what if?”

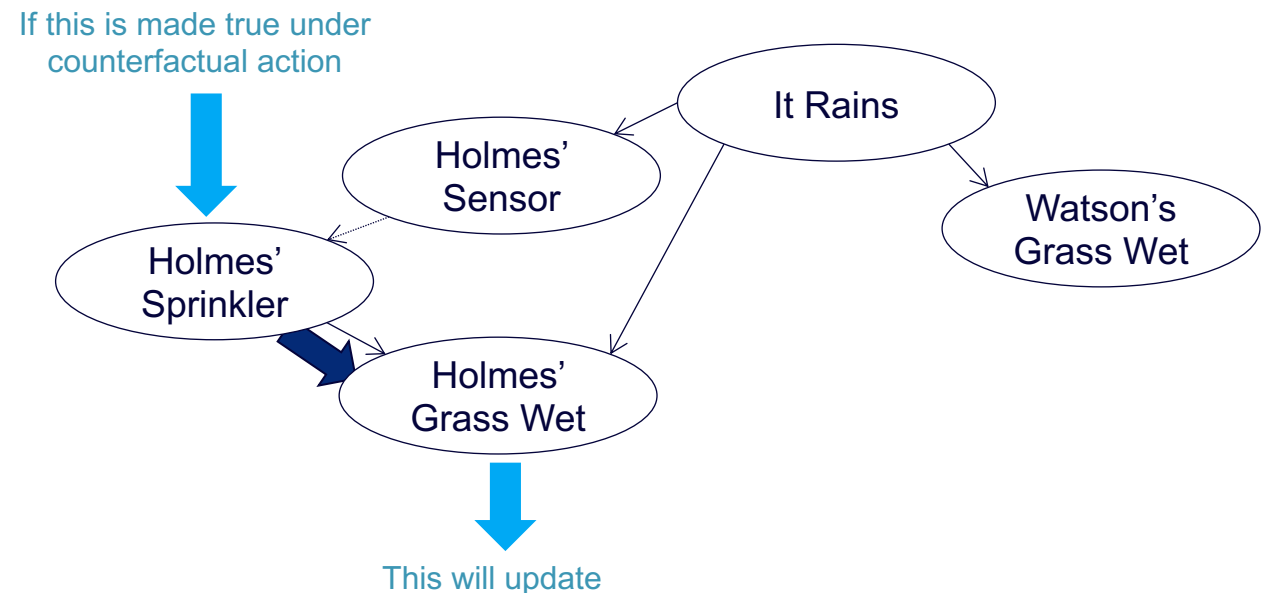
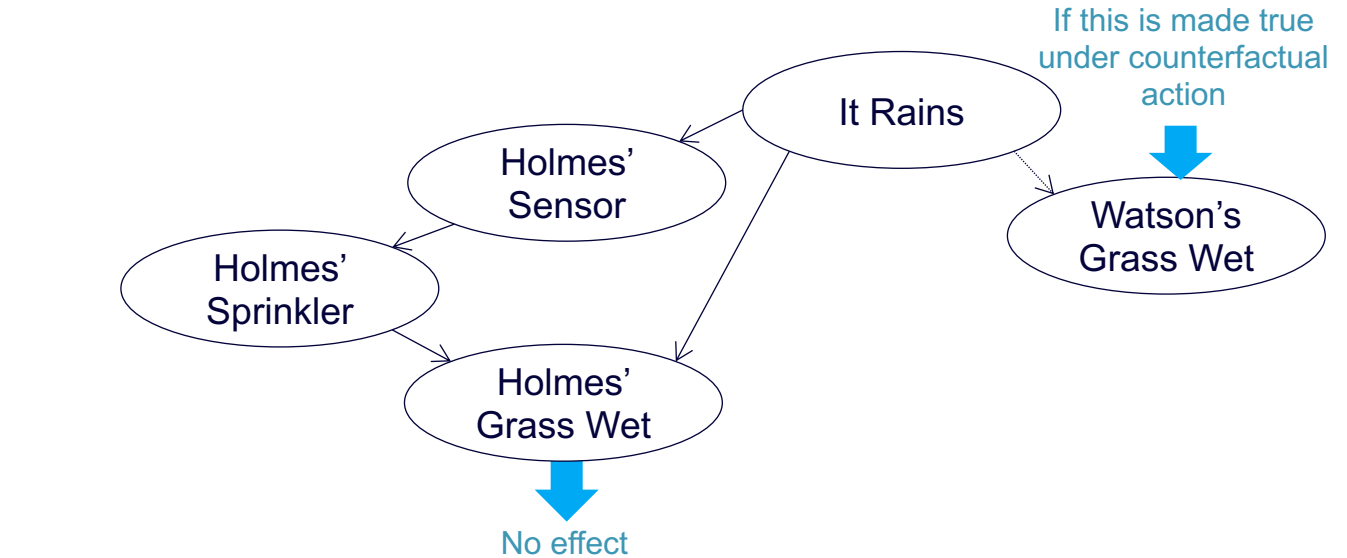


- Once we have trained a causal model, we identify counterfactuals that we would like to test and “intervene” on.
- These are generated by changing the historical data to reflect the actions of the intervention, and new predictions (of a target) are generated.
- Comparing these to the target from the “real” data allows us to calculate the value at stake of implementing the counterfactual change.
- **If our models aren’t causal, our “what if’s” could be very inaccurate**


Counterfactual actions are different to inference, and lead to different results

‘Conditioning’ versus ‘Doing’

- An external intervention that makes Watson’s grass wet (e.g. Watson’s watering can) has no effect on whether it rained, which is different to conditioning and the outcome of the previous slide.
- Interventions effect variables’ dependencies though: if Holmes’ sprinkler is set to on, Holmes’ Grass Wet marginal will change. In *this instance* counterfactual is same as conditioning.
- This is still not generally the same as conditioning, as probabilities don’t propagate to update parent nodes. If Holmes sprinkler and it rains shared a common parent, then counterfactual wouldn’t update it and thus would be different from conditioning

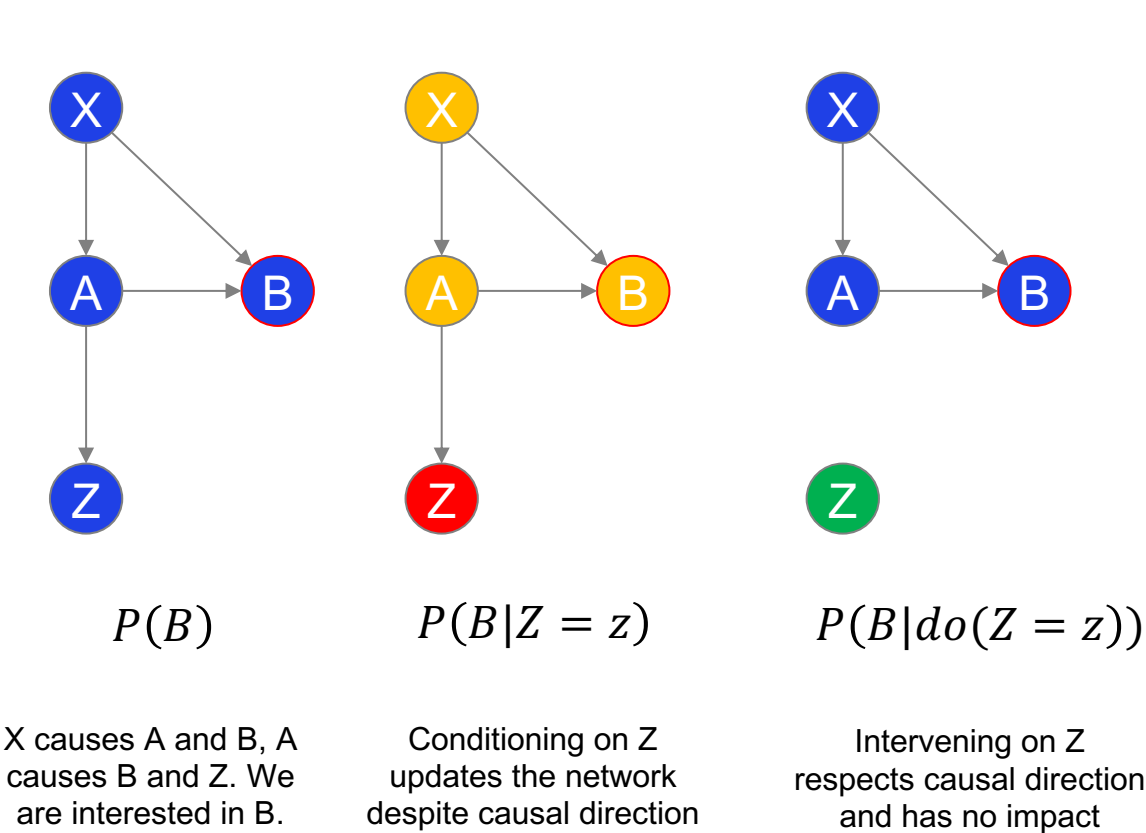


The differences between observational and interventional inference can be best viewed through SCM notation

For a causal model, where $X \rightarrow Y$ is denoted  **observing** evidence asks the model to update likelihoods of variables throughout the model based on an observation.

If Z is **observed** to have a specific value, what can we infer about the likely values A, B and X had at this point? The distributions of A, B and X all update given this observational information.

If we **intervene** on Z, the causal direction states that A or any other nodes are not impacted, and we “break” the link between A and Z. Distributions of A, B and X (in this instance) remain unchanged, because Z does not “cause” any of them, and so intervening on Z is folly.



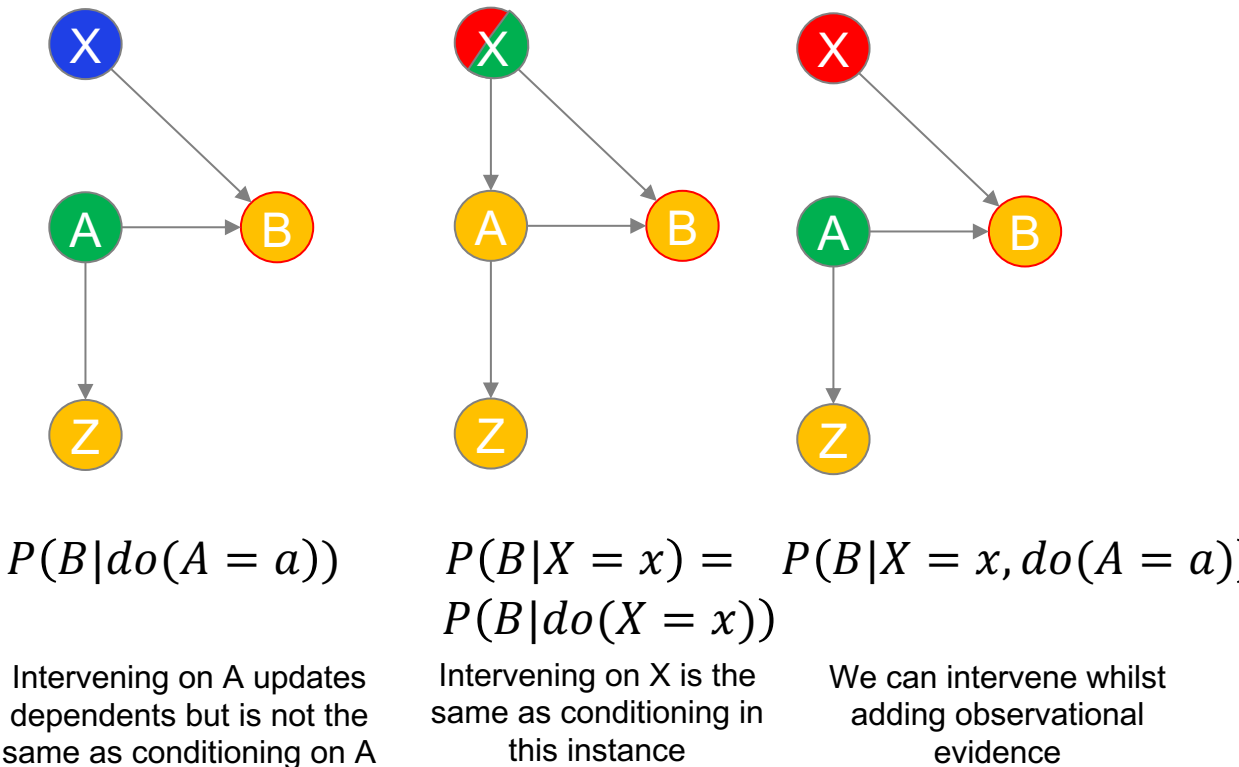
-  Original variable
-  Observational change
-  Interventional change
-  Updated distribution

The differences between observational and interventional inference can be best viewed through SCM notation

Intervening on A “breaks” its dependence from X (and doesn’t update the distribution of X , unlike conditioning on A). B and Z are both descendants of A , and their marginal probabilities would change due to the change in A .

For nodes with no parents, intervening on them is identical to receiving observational update.

We can combine interventions with observational data. For observation in X and intervention on A , B would update to reflect changes in both A and X . A still stops depending on X because of the intervention, and Z updates based only on the intervention change to A , and not due to updates to X .



- Original variable
- Observational change
- Interventional change
- Updated distribution

Epilogue

Bayesian Networks complement conventional modelling techniques and perhaps supersede their capabilities in some areas

Advantages

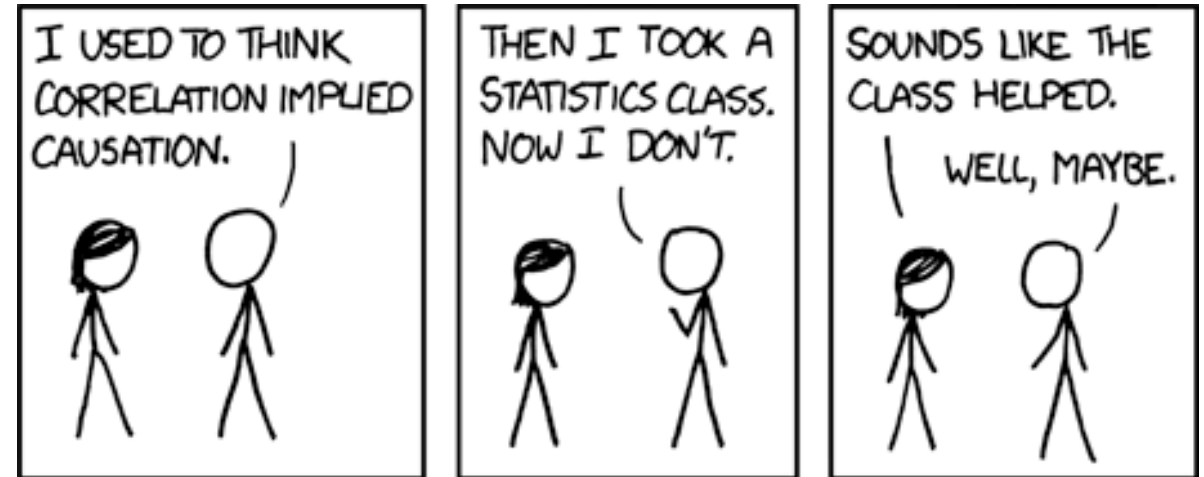
- Bayesian Networks offer a graphical representation that is **reasonably interpretable and easily explainable**
- Models can reflect both statistically significant information (learned from the data) and domain expertise simultaneously. Metrics can measure the significance of relationships and help identify the effect of specific actions
- Relationships captured between variables in a Bayesian Network are more complex yet hopefully **more informative than a conventional model**
- **Counterfactual actions** combine without naive independence assumptions

Considerations

- This is **not a way of automatically perfectly identifying causal relationships**, but it can help a human explore this
- Computational considerations limit the number of variables in a BN (max ~30)

Takeaways

- If we want to trust models for decisions, then we should expect them to make **causal sense**
- Training on observational data is common, and the causal direction of relationships is not always clear
- Methods exist to **help us identify possible causal relationships**, but domain experts can also help
- **Models that respect causality** also exist and thanks to recent advances are now easier to learn and deploy



Any Questions?



QuantumBlack
AI by McKinsey