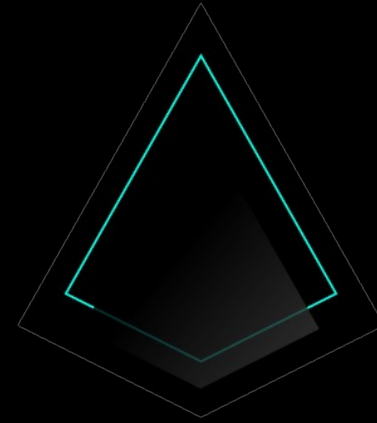


Hi! Setup for today that you'll need:

- A clone of github.com/oentaryorj/qb-causalnex-aiximpact
- Python 3.7/3.8
- Jupyter
pip install jupyterlab
- CausalNex
pip install causalnex
- PyGraphViz
pip install pygraphviz
 - Anaconda users: use **conda install pygraphviz** if pip install fails



CausalNex

CausalNex

Explainable Modelling and Causal Inference

Dr. Richard Oentaryo and Dr. Paul Beaumont



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



CausalNex

Agenda

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

Structure learning (10mins)

Inference & interventions (10mins)

CausalNex demo (10mins)

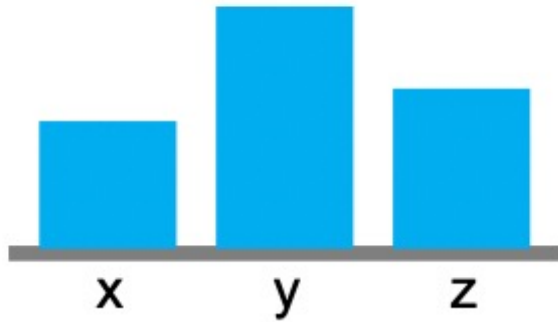
Epilogue (5mins)

Q&A (10mins)

The background features a solid black field on the left, transitioning into a series of parallel white diagonal lines on the right. A thin teal line starts from the top left and extends towards the center.

Introduction to causality & Bayesian networks

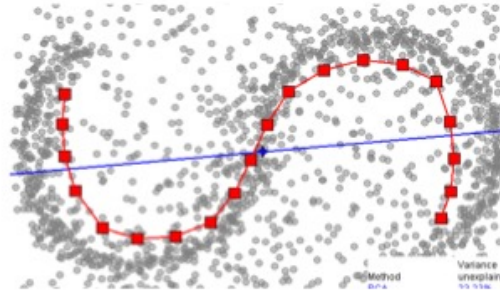
The world of machine learning – in a nutshell



Predictive

- Estimate the target for new observations

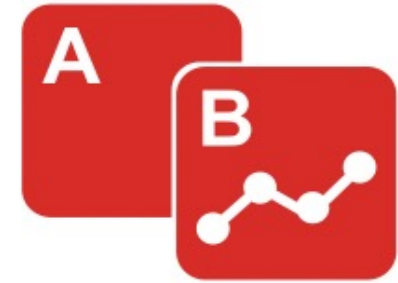
$$y = f(x)$$



Descriptive

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

$$y = f(x)$$

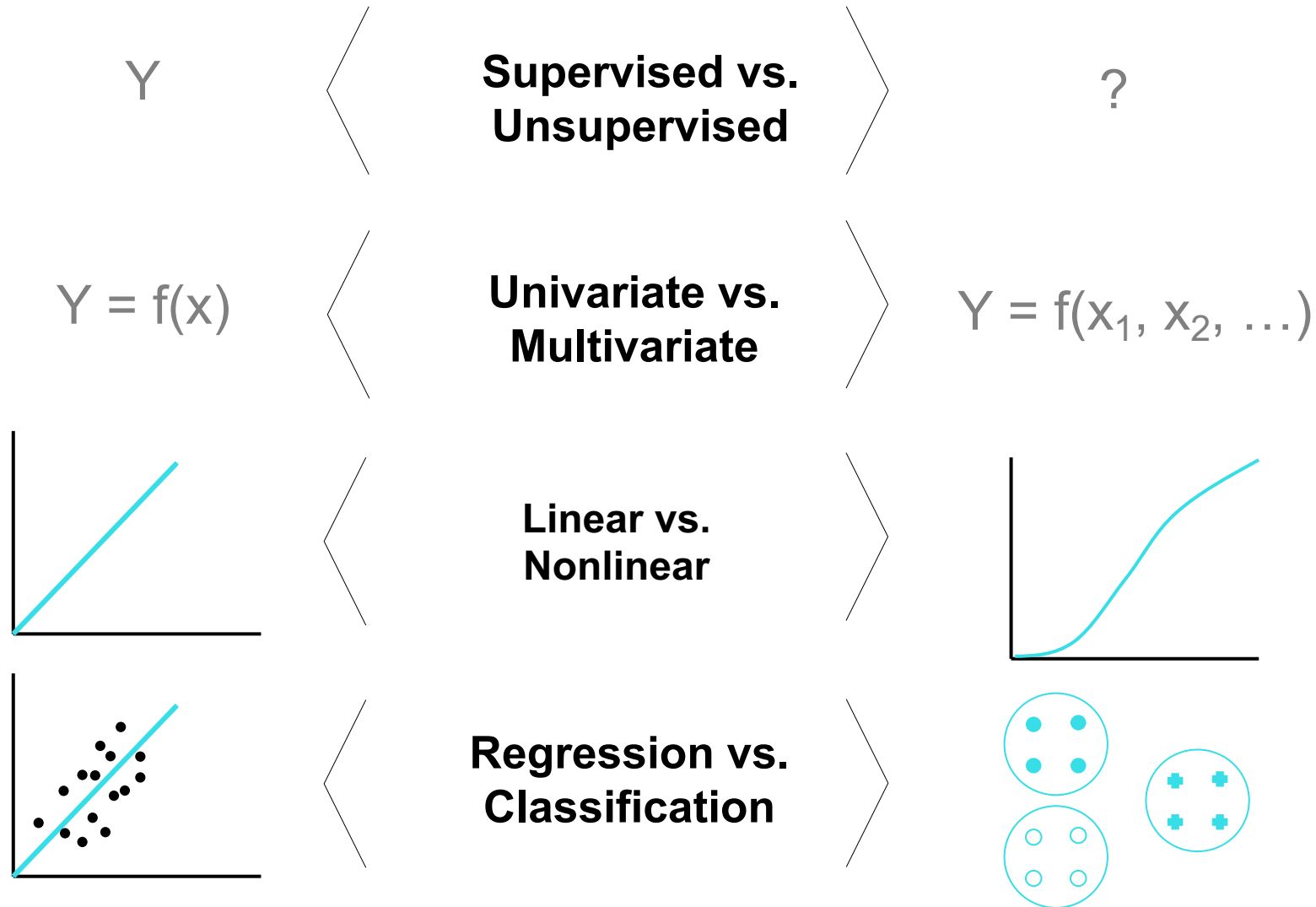


Prescriptive

- Find the inputs that give optimal performance
- f is known

$$y = f(x)$$

The truth is, there is no one way to explain different modelling techniques; they ‘split’ along multiple dimensions...



...the reality is complex and difficult to document

Classification

What group does this sample belong to?

Regression

What value would this sample take based on the data?

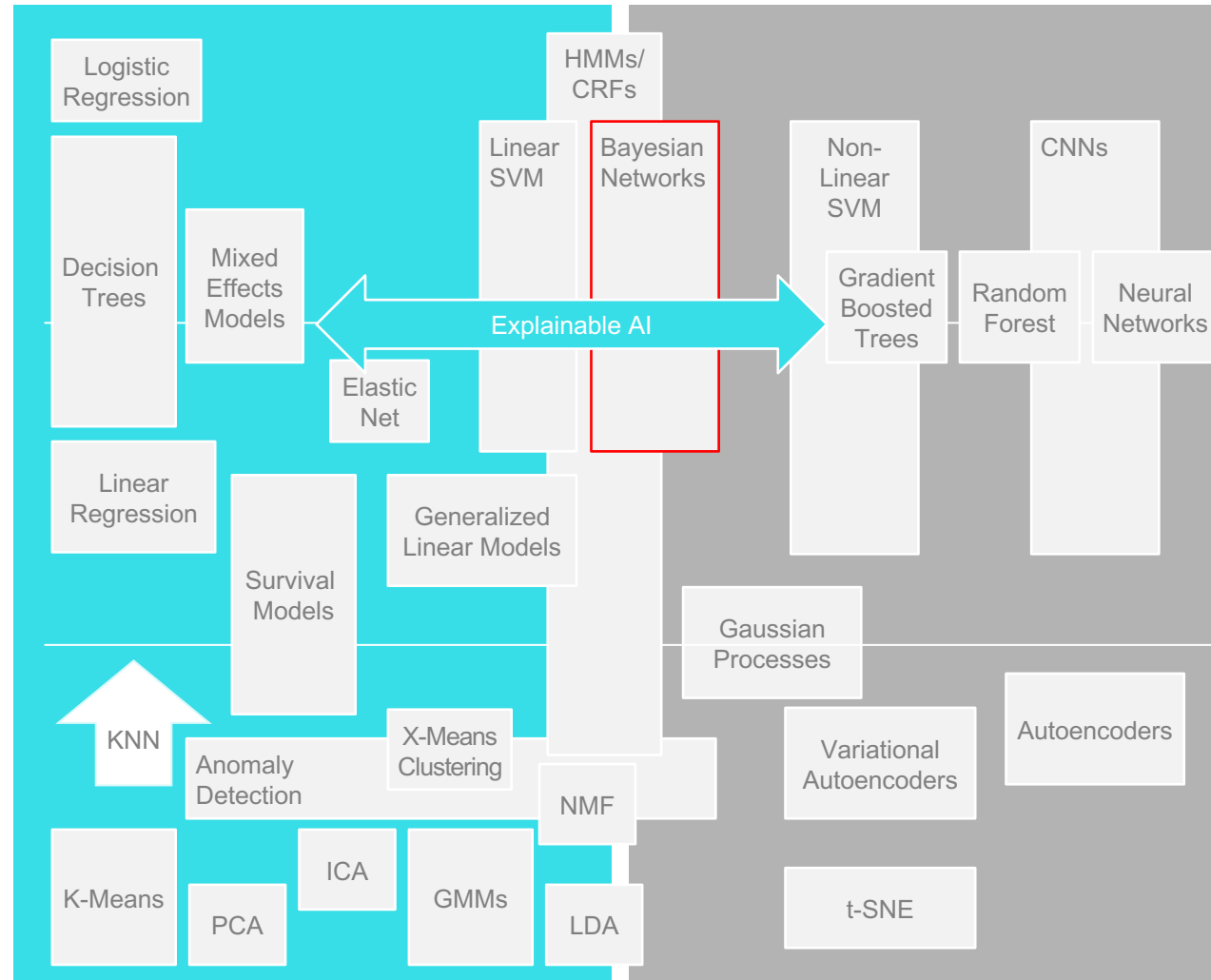
Unsupervised

What are the hidden / underlying patterns in my data?

Explanatory

Predictive

Prescriptive



Aside from the model choice, other important factors must be considered

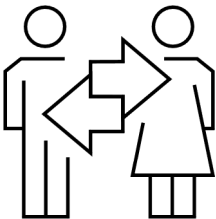
Deep dive NOT EXHAUSTIVE

QuantumBlack does active research in all of these areas (and more)... ..with many publications



Algorithm fairness/bias

Methods for detection and correction of bias in models, ensuring models make fair predictions devoid of discrimination



Explainability

Facilitation of high performing machine learning models to be interpreted by humans, i.e., provide explanations for black box models



Causal inference

Models able to encode causal relationships, not just correlations, and allowing inclusion of input from domain experts



Ongoing model performance

Tool to enable tracking how models perform over time, supporting in identifying when to take corrective actions to sustain performance



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?

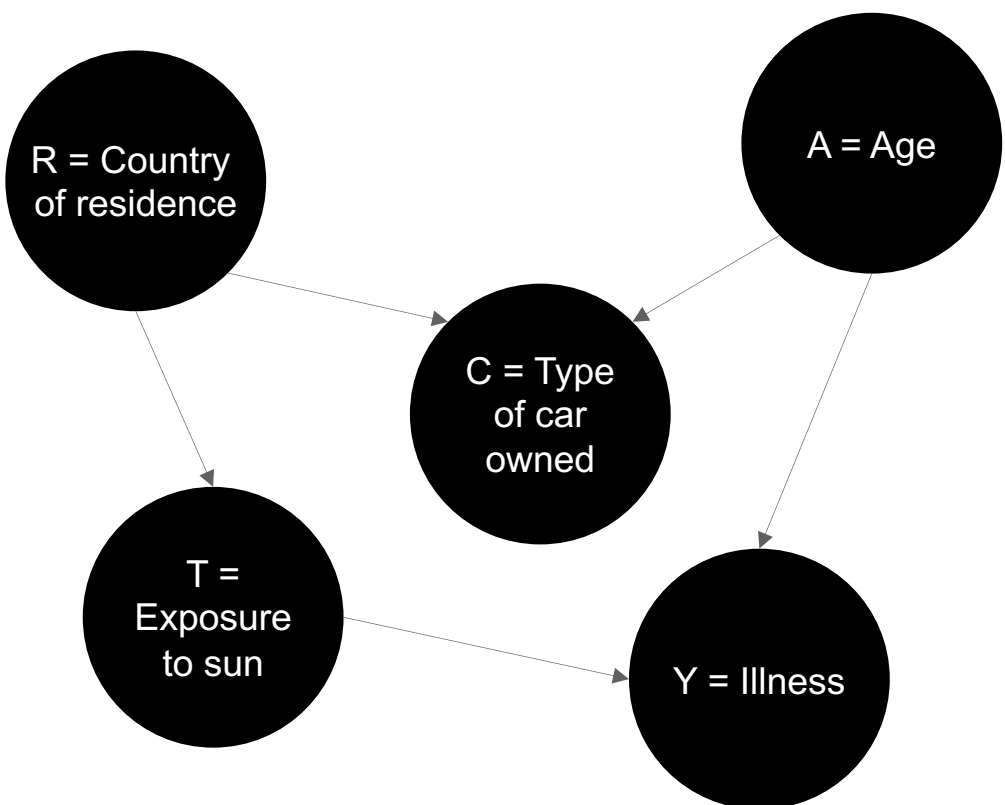


Correlation is not causation...



Tackling these problems with classical machine learning can be problematic

Goal:
Find the effect of sun exposure on the illness



$R = \varepsilon_1$
 $A = \varepsilon_2$
 $C = R + A + \varepsilon_3$
 $T = R + \varepsilon_4$
 $Y = T + 10A + \varepsilon_5$
 $\varepsilon_1, \dots, \varepsilon_5$ independent normal(0,1)

Q: Which is the better model?

```
model = sm.OLS(train.y, train[['t']])
results = model.fit()
print(results.summary())
```

Test RMSE = 10

OLS Regression Results

Dep. Variable:	y	R-squared:	0.019
Model:	OLS	Adj. R-squared:	0.019
Method:	Least Squares	F-statistic:	2.766e+04
Date:	Wed, 13 Feb 2019	Prob (F-statistic):	0.00
Time:	07:25:25	Log-Likelihood:	-5.2235e+06
No. Observations:	1401801	AIC:	1.045e+07
Df Residuals:	1401800	BIC:	1.045e+07
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
t	0.9984	0.006	166.317	0.000	0.987	1.010

```
model = sm.OLS(train.y, train[['t', 'car']])
results = model.fit()
print(results.summary())
```

Test RMSE = 7.7

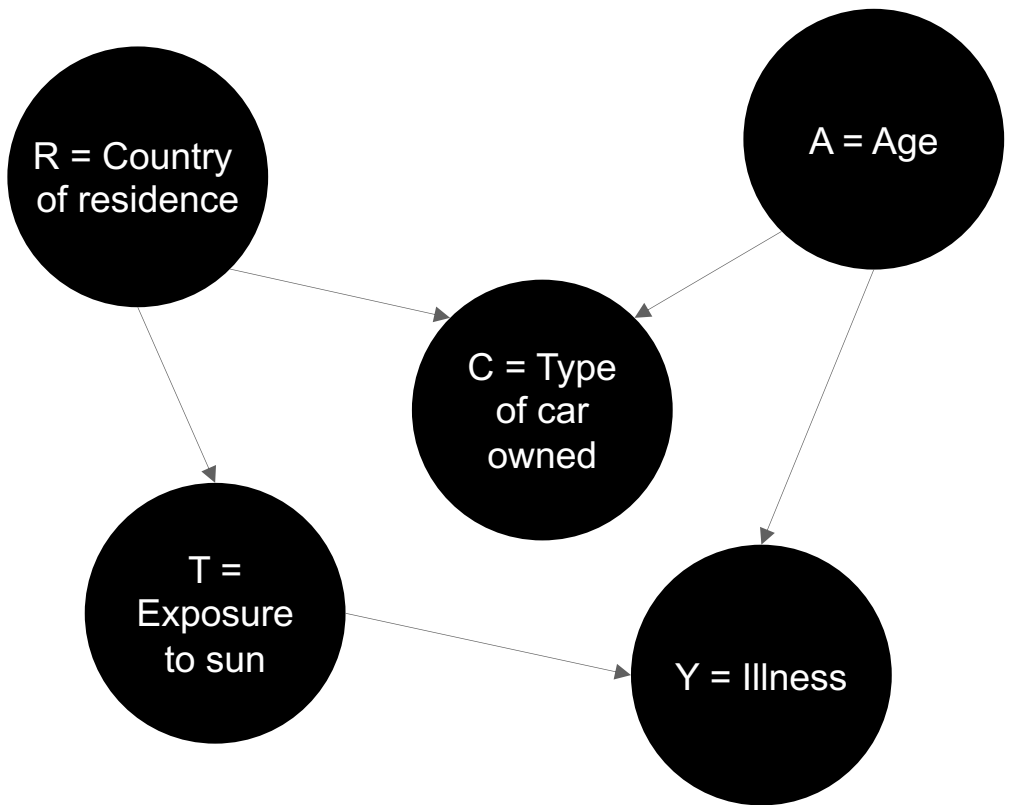
OLS Regression Results

Dep. Variable:	y	R-squared:	0.408
Model:	OLS	Adj. R-squared:	0.408
Method:	Least Squares	F-statistic:	4.835e+05
Date:	Wed, 13 Feb 2019	Prob (F-statistic):	0.00
Time:	07:27:29	Log-Likelihood:	-4.8695e+06
No. Observations:	1401801	AIC:	9.739e+06
Df Residuals:	1401799	BIC:	9.739e+06
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
t	-1.0036	0.005	-196.455	0.000	-1.014	-0.994
car	4.0024	0.004	959.740	0.000	3.994	4.011

Tackling these problems with classical machine learning can be problematic

Goal:
Find the effect of sun exposure on the illness



$R = \varepsilon_1$
 $A = \varepsilon_2$
 $C = R + A + \varepsilon_3$
 $T = R + \varepsilon_4$
 $Y = T + 10A + \varepsilon_5$
 $\varepsilon_1, \dots, \varepsilon_5$ independent normal(0,1)

Q: Which is the better model?

```
model = sm.OLS(train.y, train[['t']])
results = model.fit()
print(results.summary())
```

Test RMSE = 10

OLS Regression Results

Dep. Variable:	y	R-squared:	0.019
Model:	OLS	Adj. R-squared:	0.019
Method:	Least Squares	F-statistic:	2.766e+04
Date:	Wed, 13 Feb 2019	Prob (F-statistic):	0.00
Time:	07:25:25	Log-Likelihood:	-5.2235e+06
No. Observations:	1401801	AIC:	1.045e+07
Df Residuals:	1401800	BIC:	1.045e+07
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
t	0.9984	0.006	166.317	0.000	0.987	1.010

```
model = sm.OLS(train.y, train[['t', 'car']])
results = model.fit()
print(results.summary())
```

Test RMSE = 7.7

OLS Regression Results

Dep. Variable:	y	R-squared:	0.408
Model:	OLS	Adj. R-squared:	0.408
Method:	Least Squares	F-statistic:	1.235e+05
Date:	Wed, 13 Feb 2019	Prob (F-statistic):	0.00
Time:	07:27:29	Log-Likelihood:	-4.8695e+06
No. Observations:	1401801	AIC:	9.739e+06
Df Residuals:	1401799	BIC:	9.739e+06
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
t	-1.0036	0.005	-196.455	0.000	-1.014	-0.994
car	4.0024	0.004	959.740	0.000	3.994	4.011

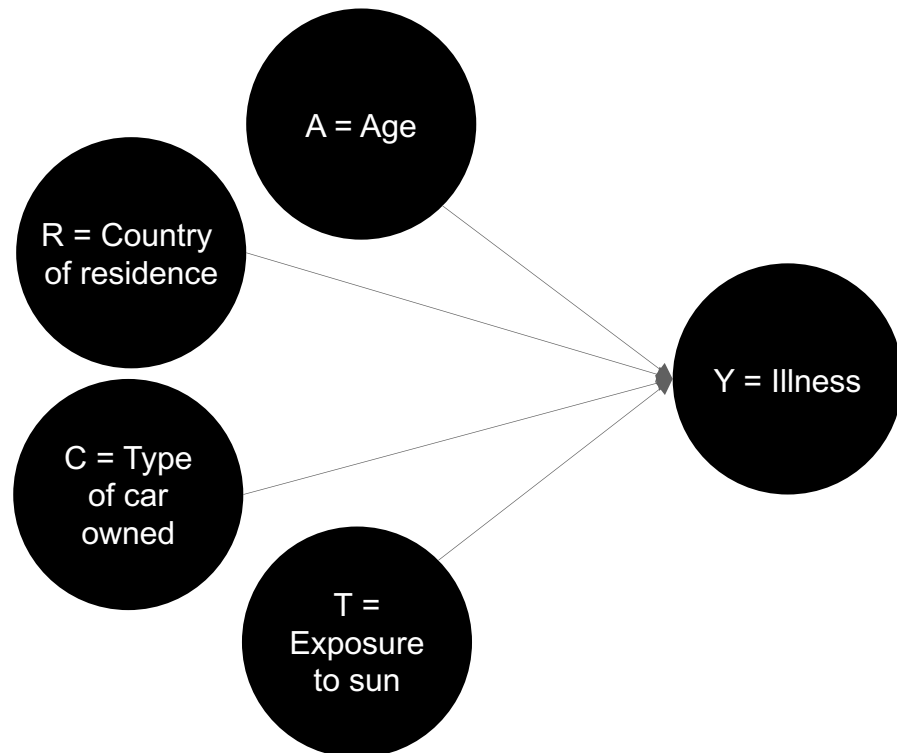
We can better identify the right intervention with graphical models

Establishing causal relationship is critical for us to recommend the right interventions

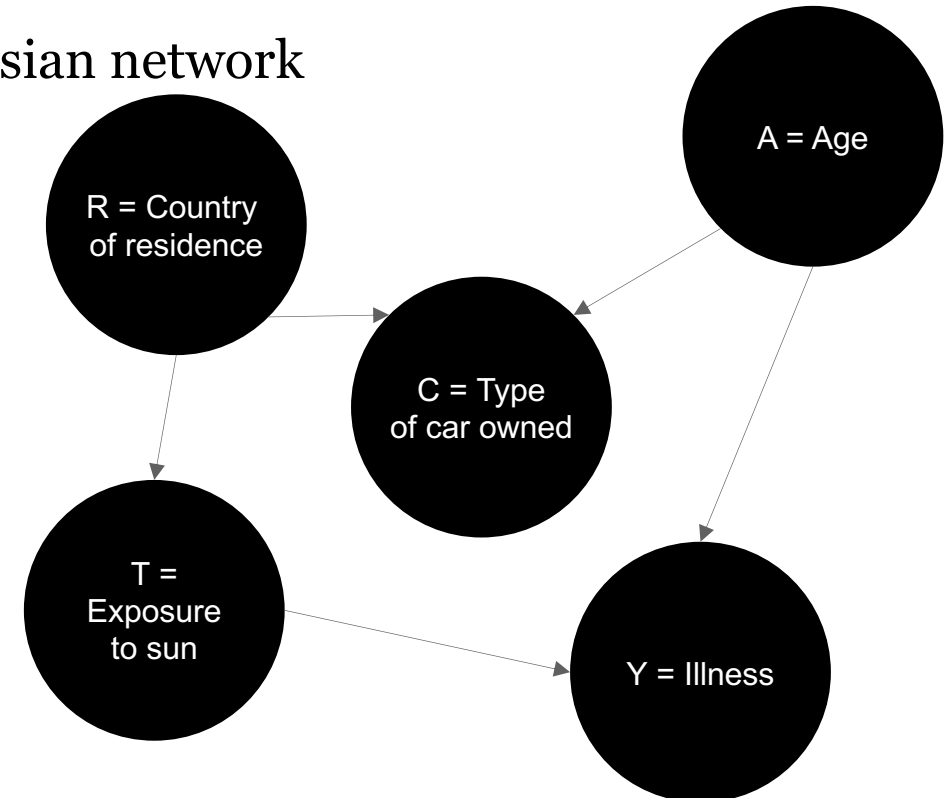
Traditional models, like linear regressions, have simplistic assumptions

We use graph models such as Bayesian Networks which can be more intuitive and allows domain expertise encoded with data to form a better understanding of relationships

Linear regression



Bayesian network



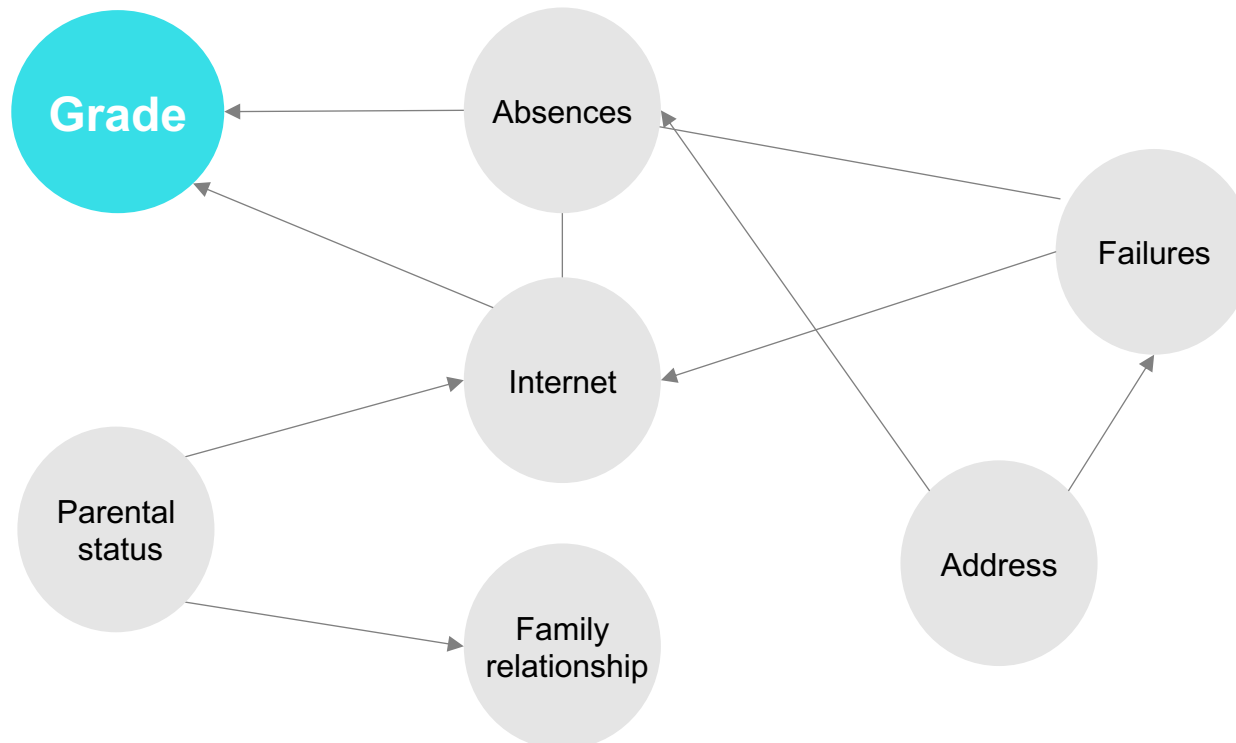
Bayesian networks

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.*”



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

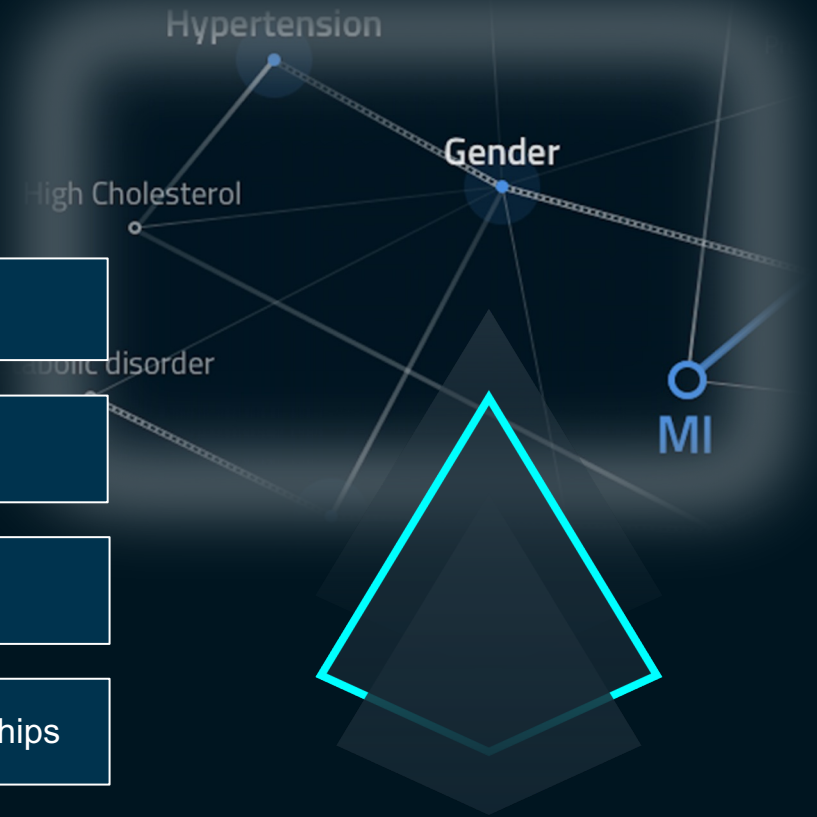
Structure learning – Learn relationships in data with NOTEARS, a state-of-the-art algorithm

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

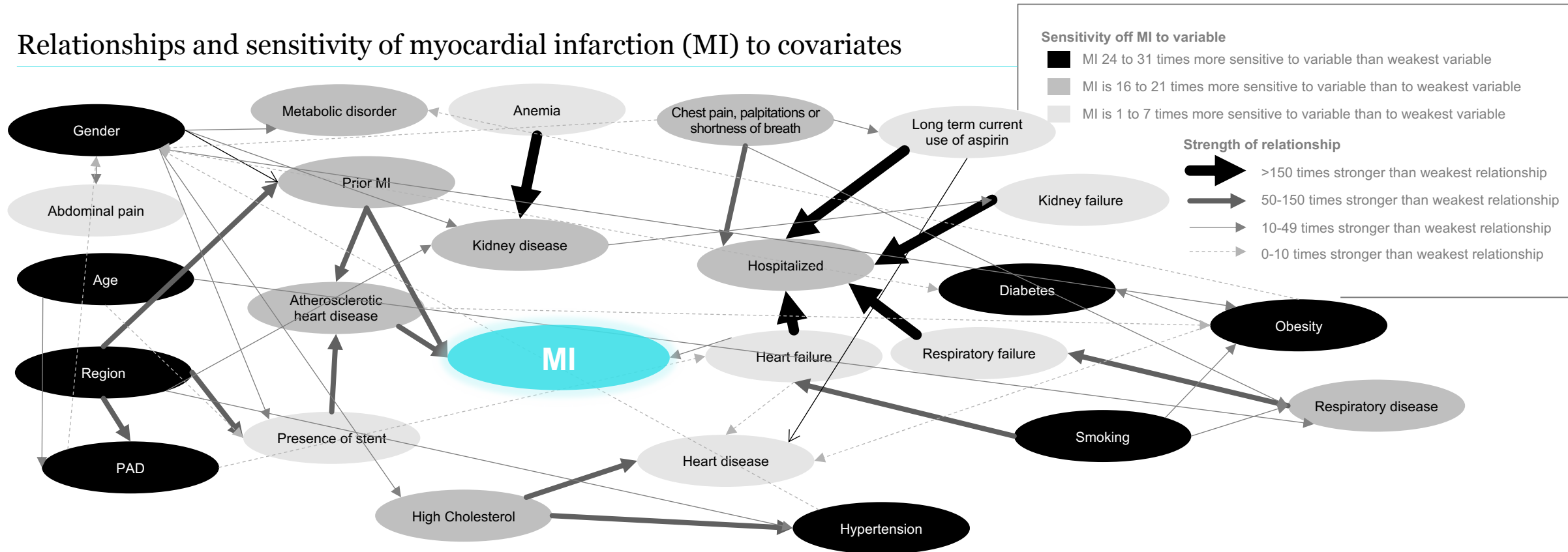


Read The Docs

```
`pip install causalnex`
```

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

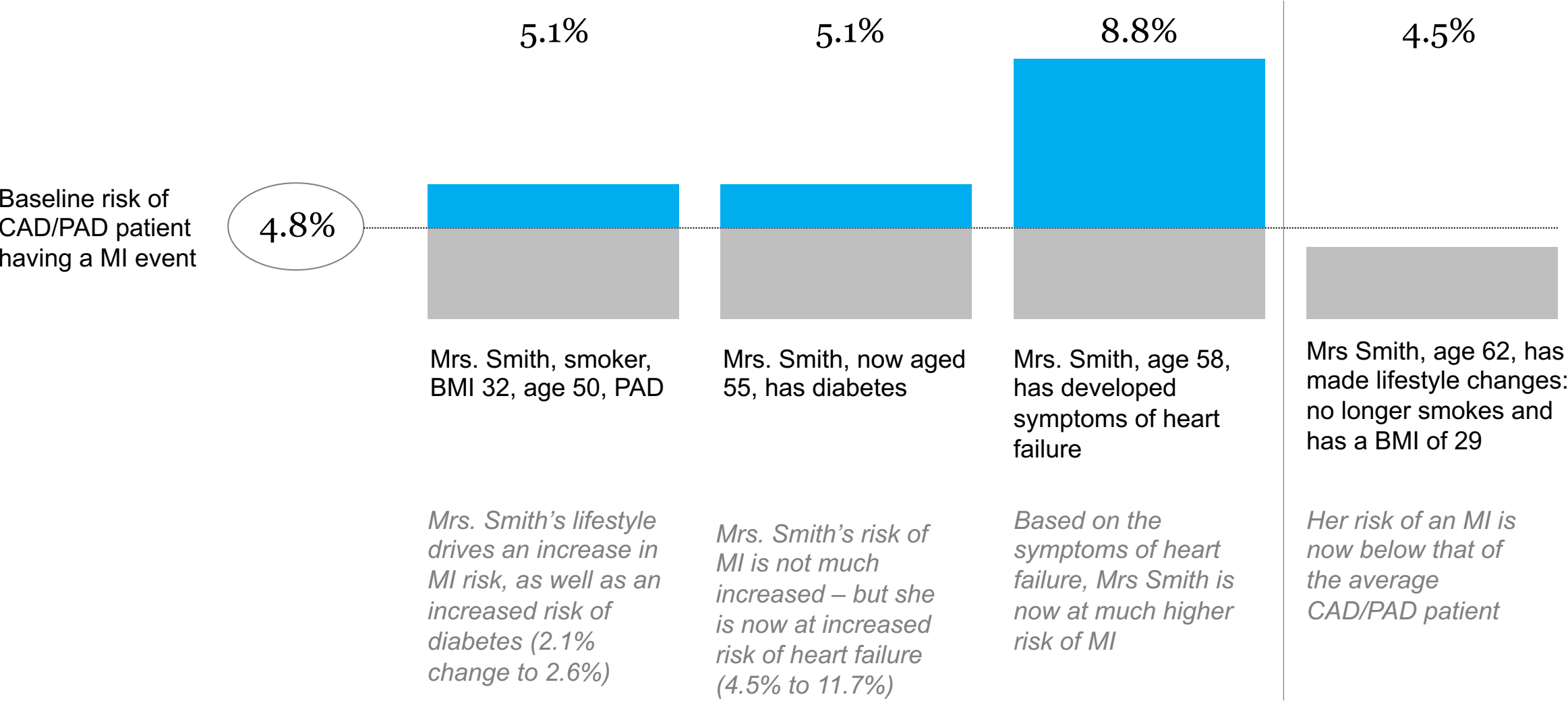
Relationships and sensitivity of myocardial infarction (MI) to covariates



- 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

With this approach we could better understand patient journeys

Risk of MI within 12 months¹



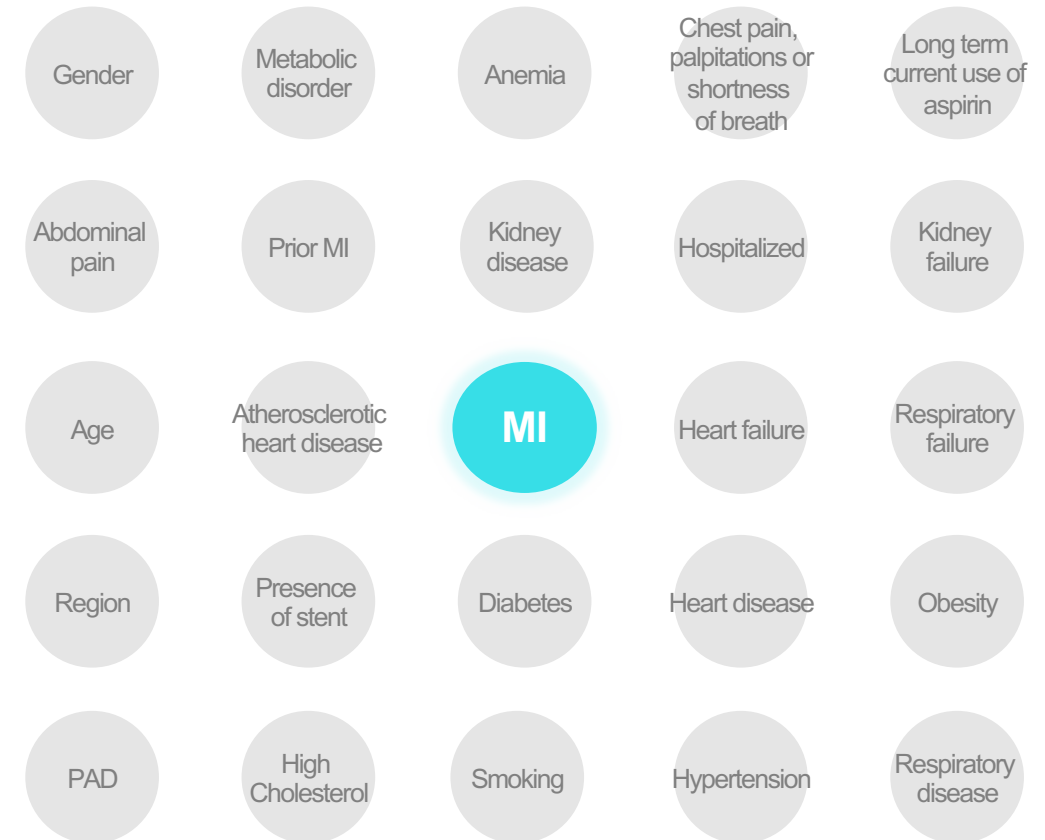
The background features a solid black field on the left and a large triangular area on the right filled with fine, parallel white diagonal lines. A single, thin cyan line extends from the top left corner towards the text.

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



Defining the structure of a Bayesian network

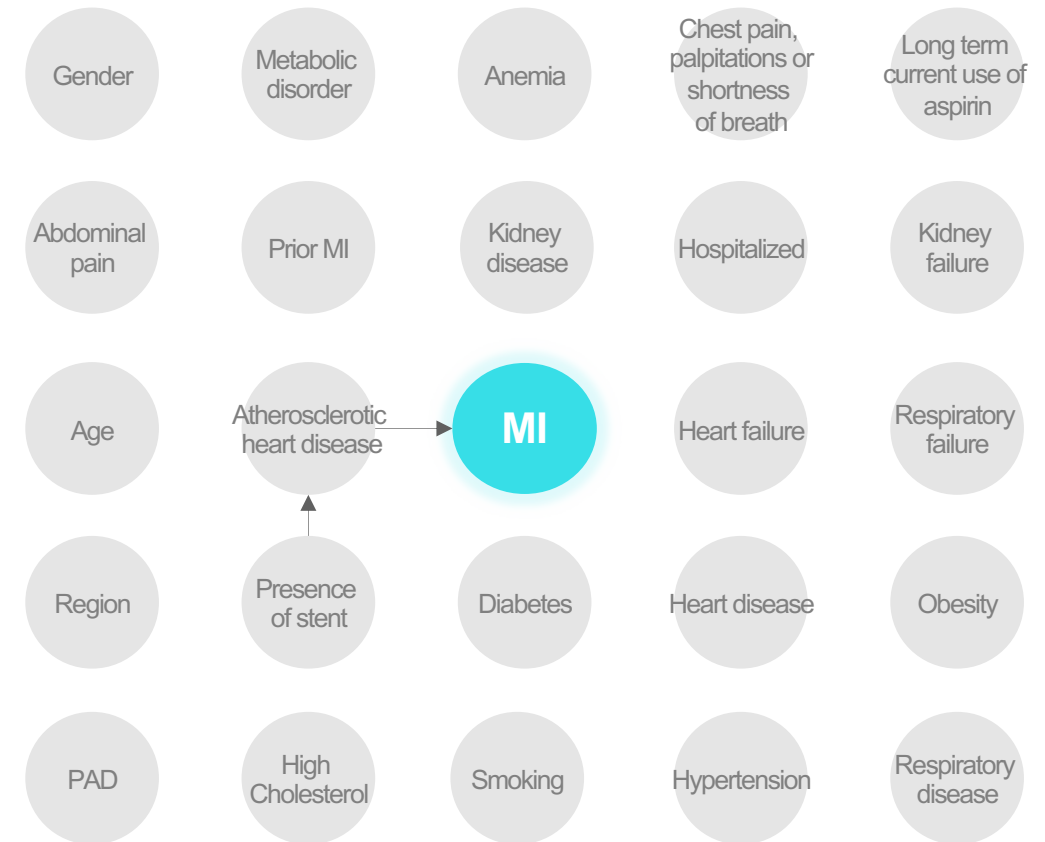
Domain Expertise

- 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	 Search methods	 Output type
Score-based <ul style="list-style-type: none">Find graph that maximizes a specified scoreCommon scores: BIC, MDL, BDe, BDeu	<ul style="list-style-type: none">Greedy local searchDynamic programmingInteger linear programmingGlobal continuous optimization	<ul style="list-style-type: none"><i>Directed acyclic graph (DAG)</i>
Constraint-based <ul style="list-style-type: none">Start from complete graph, delete edges between nodes that are conditionally independent (CI), orient edgesCan choose CI criterion: Chi-squared test, G test	<ul style="list-style-type: none">Full searchImproved by testing CI in the right order	<ul style="list-style-type: none">Equivalence class of DAGs called <i>completed partially directed acyclic graph (CPDAG)</i>

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.

- 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:

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

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

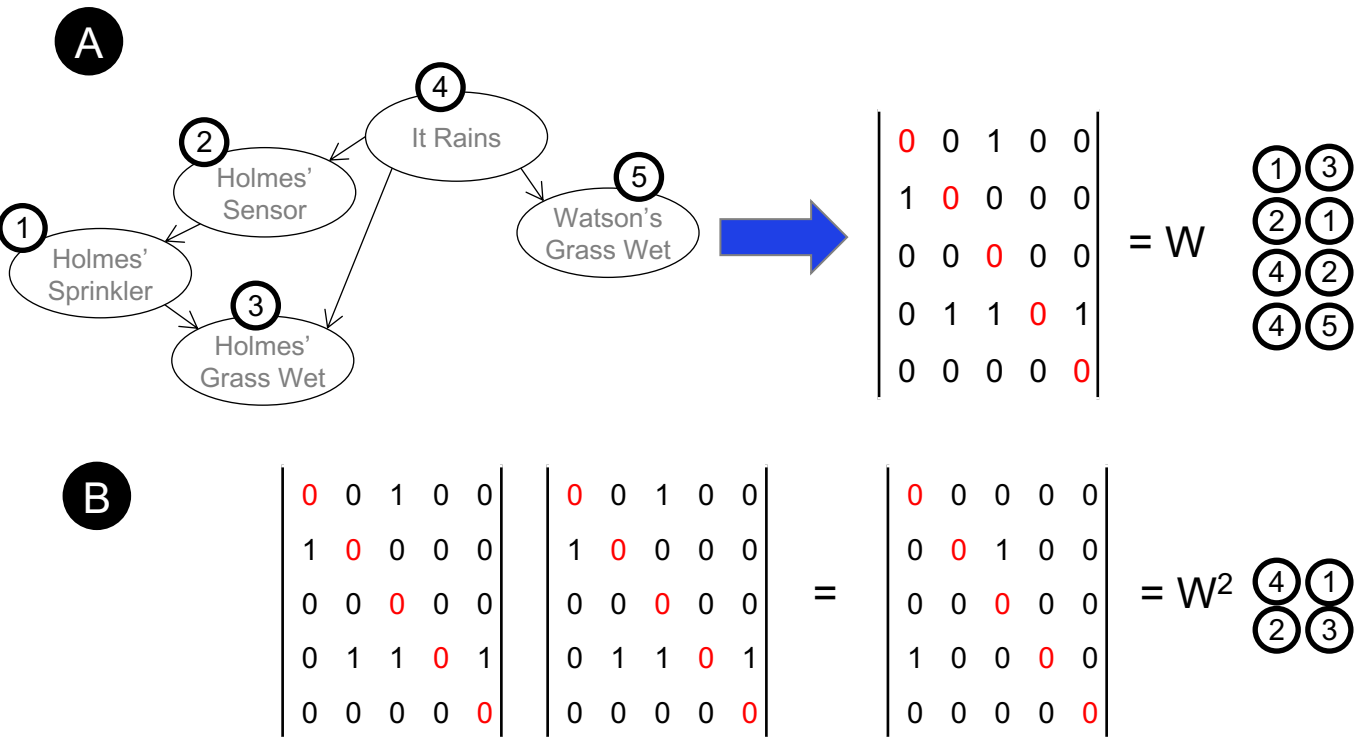
$$F(W) = \ell(W; \mathbf{X}) + \lambda \|W\|_1 = \frac{1}{2n} \|\mathbf{X} - \mathbf{X}W\|_F^2 + \lambda \|W\|_1.$$

¹ Non-combinatorial Optimization via Trace Exponential and Augmented lagRangian for Structure learning

Zheng, Xun, et al. "DAGs with NO TEARS: Continuous optimization for structure learning." Advances in Neural Information Processing Systems. 2018.

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:

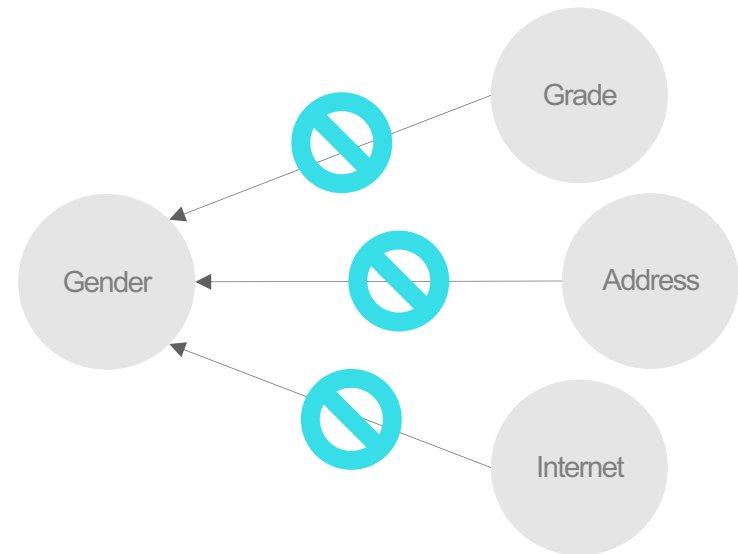
$$\sum_{k=1} \sum_i^d \frac{(W^{2k})_{ii}}{k!} = \text{trace}(e^{(W \odot W)}) - d = 0 (< \epsilon)$$

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

Prior to structure learning

Nothing should influence gender



After structure learning

This edge should be flipped



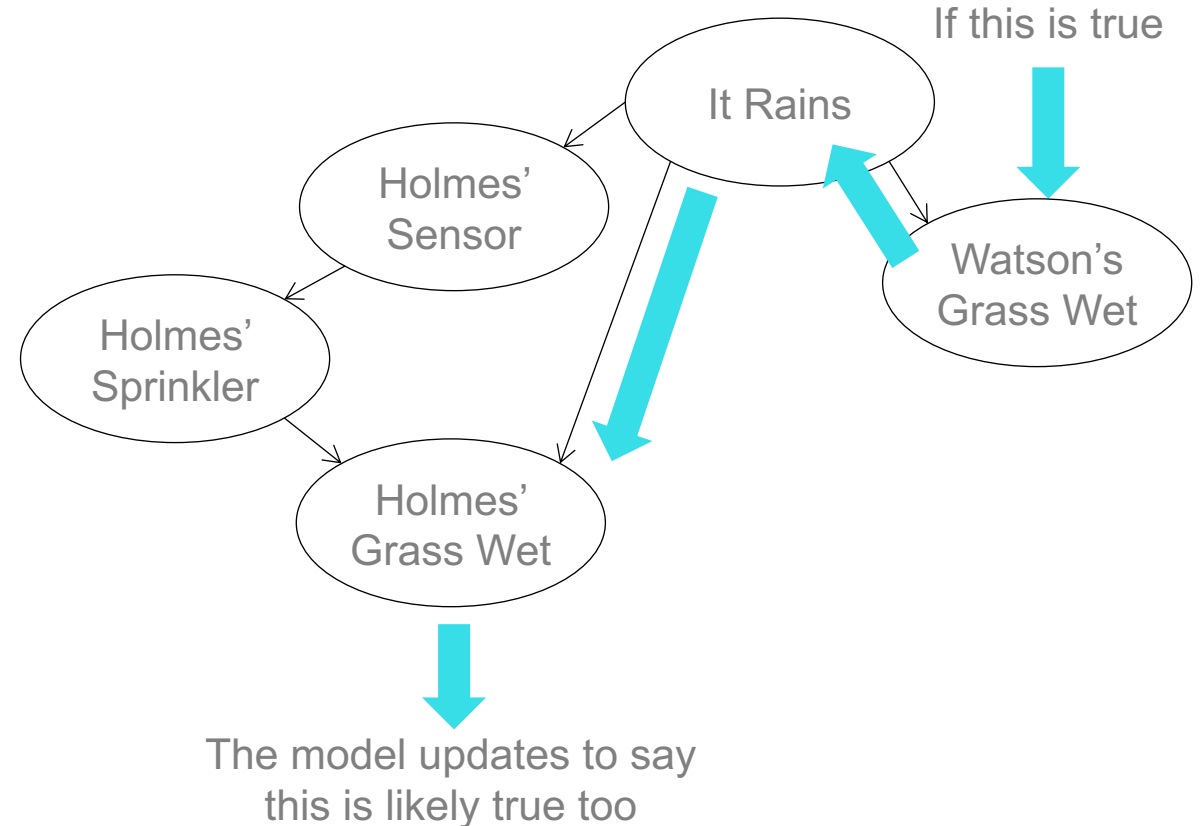
The background features a solid black area on the left and a white area on the right, separated by a diagonal line. The white area is filled with fine, parallel diagonal lines. A single teal line extends from the top left corner towards the center.

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?

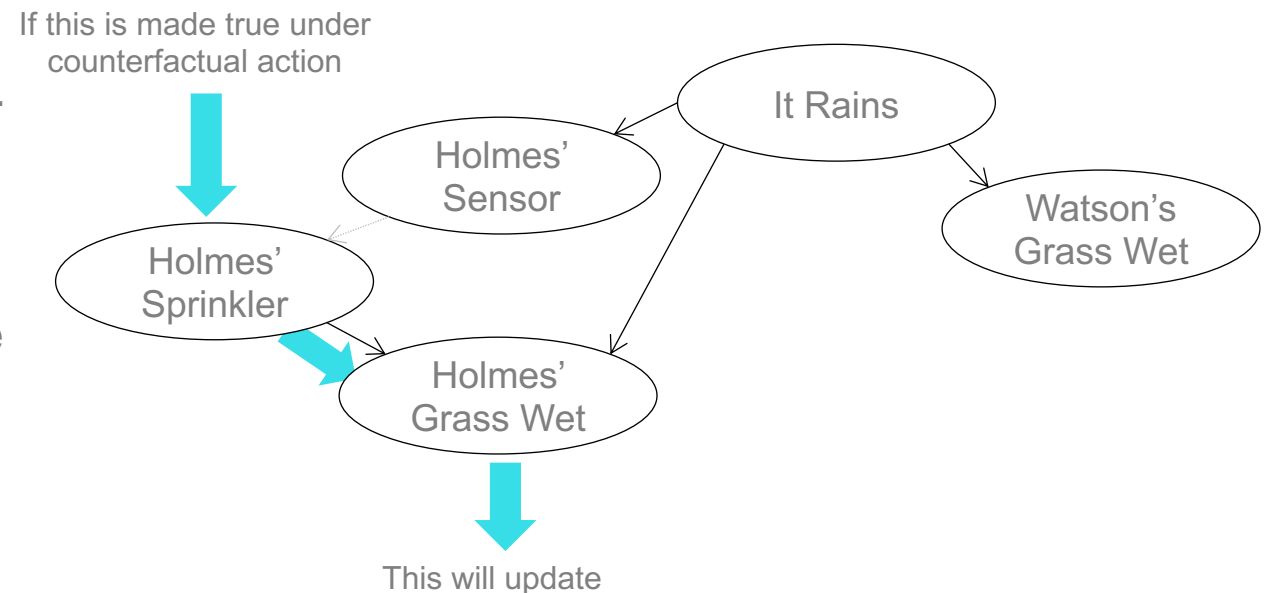
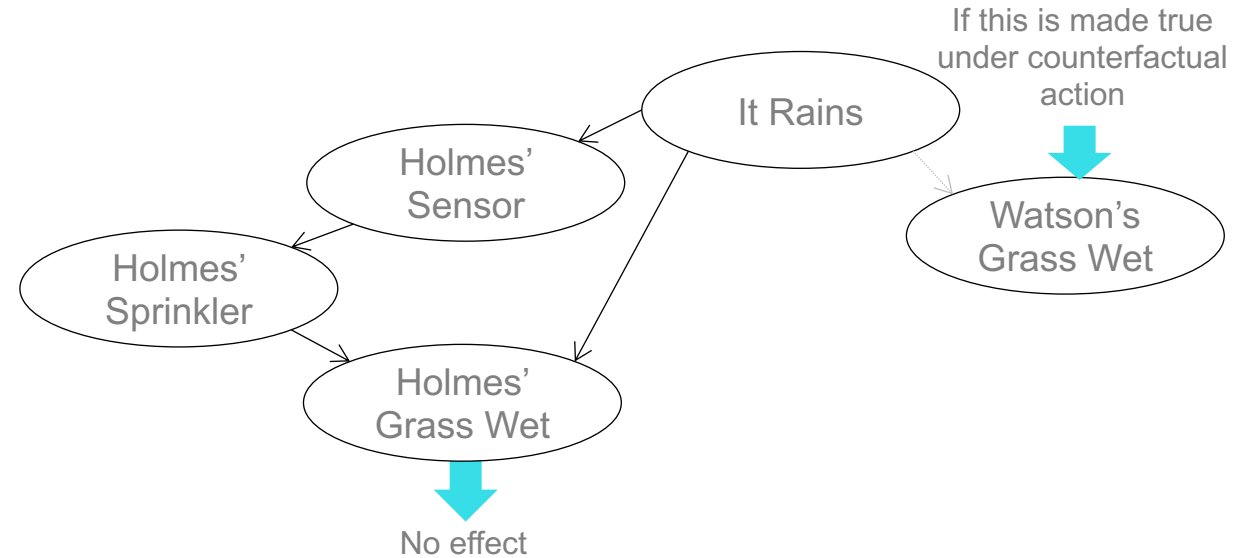
- The probabilities of variables in Bayesian Networks **update as observations are added to the model**. This is useful for inference, and for beginning predictive analytics
- Metrics can help us **understand the strength of relationships between variables** and identify key drivers of change. These 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.




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 definitely set 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

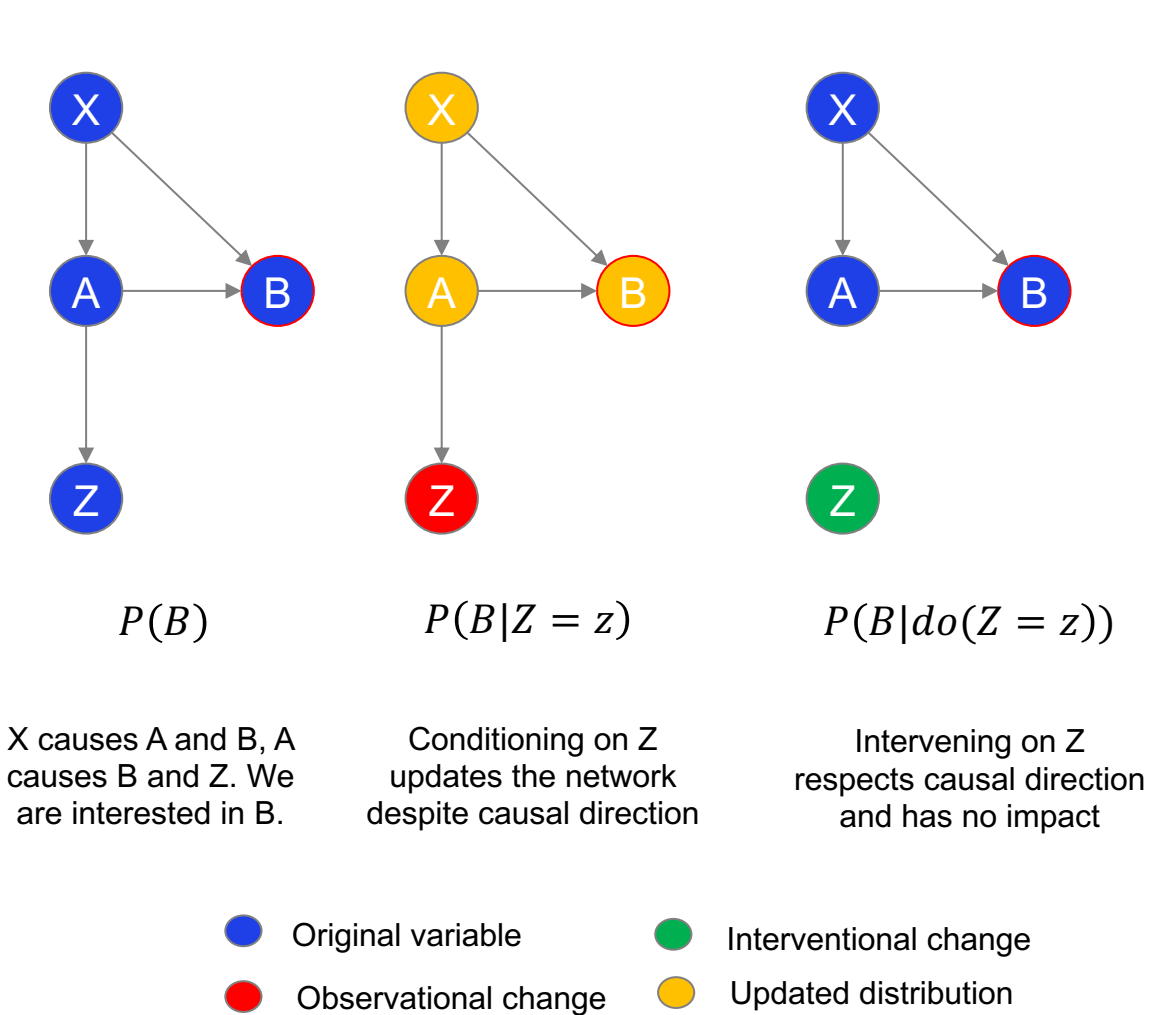


The differences between observational and interventional inference can be best viewed through SCM notation

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 (nor any other nodes) are 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.

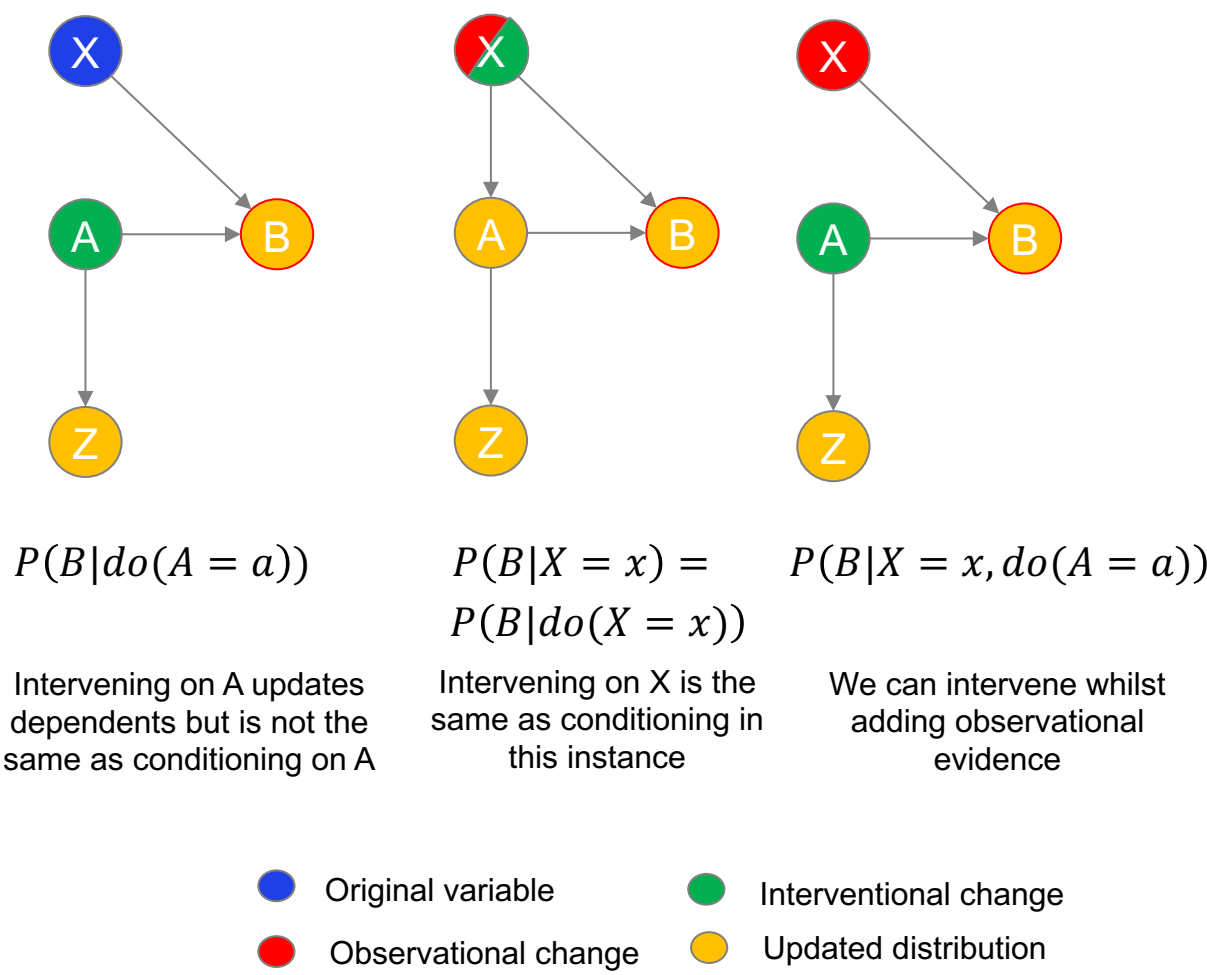


The differences between observational and interventional inference can be best viewed through SCM notation

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 descendents 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 .





Epilogue

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

Advantages

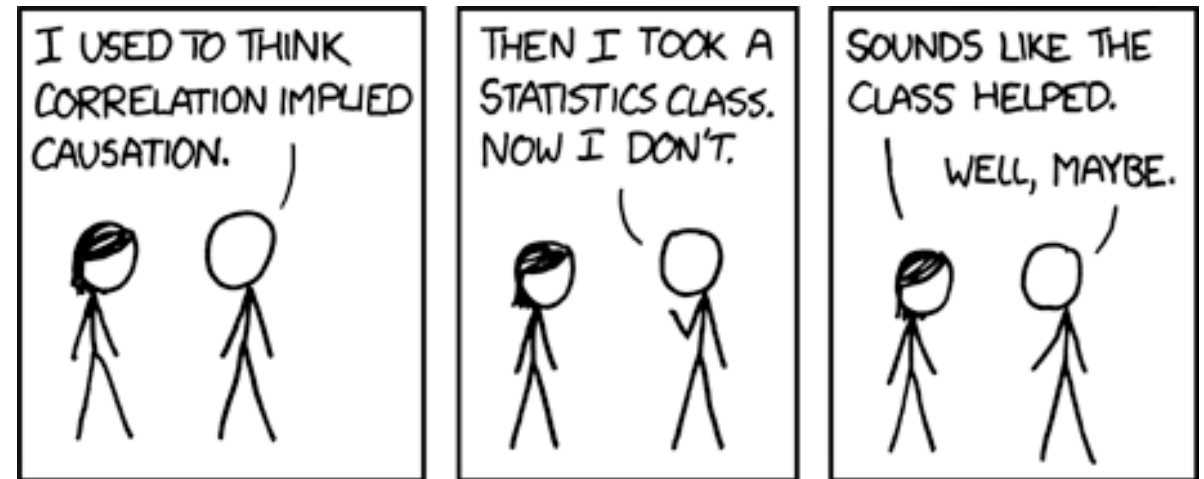
- Bayesian Networks offer a graphical representation that is **reasonably interpretable and easily explainable**
- Models can reflect both statistically significant information (learned from the data) and domain expertise simultaneously. Metrics can measure the significance of relationships and help 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
A MCKINSEY COMPANY