Integrated Causal Model for Software Cost Prediction & Control (SCOPE) Project

Mike Konrad, Robert Stoddard, Sarah Sheard,
Bill Nichols, Rhonda Brown, Dave Zubrow

March 12, 2019

DM19-0265

Project Overview

Problem: Cost estimation inaccuracy continues to be a dominant factor in DoD cost overruns.

Solution: Integrated, estimated causally-based structural equation models that provide a basis for calculating the impacts of project and organizational interventions under different scenarios, thereby helping determine the best course of action given project goals, status, and resource constraints.

Approach:

(1) Identify and collaborate with researchers having access to datasets relevant to the above problem and solution

(2) Apply causal discovery algorithms to determine which product, project, programmer, process, and other factors appear to be causal for cost, schedule, and quality

(3) Utilize structural equation modeling to quantify the relatively causal relationships

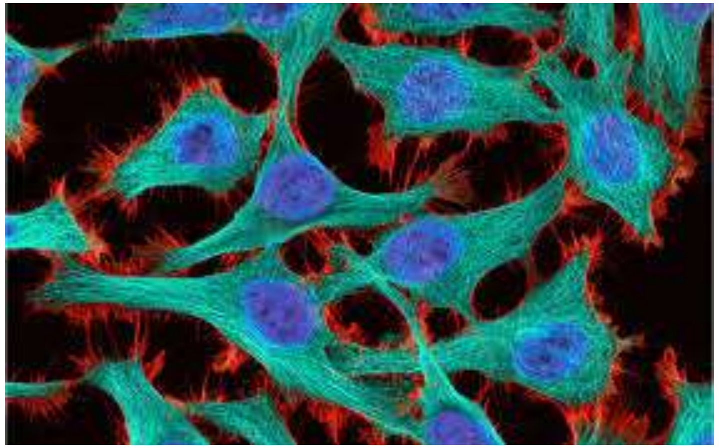
(4) Iterate, integrate, and validate

Attribution

A portion of the presentation that follows was **adapted from** "An Introduction to Causal Modeling and Discovery Using Graphical Models" **by David Danks**, Head of Philosophy Department at CMU: http://www.andrew.cmu.edu/user/ddanks/pubs.html.

Correlation doesn't inform us about causes

How do cancer cells differ from non-cancerous cells?



If we just want to predict which cells are cancerous, then correlations are sufficient.

If we want to change cancerous cells into non-cancerous ones (or at least, not dangerously cancerous), then we need causal knowledge.

Causation vs. correlation

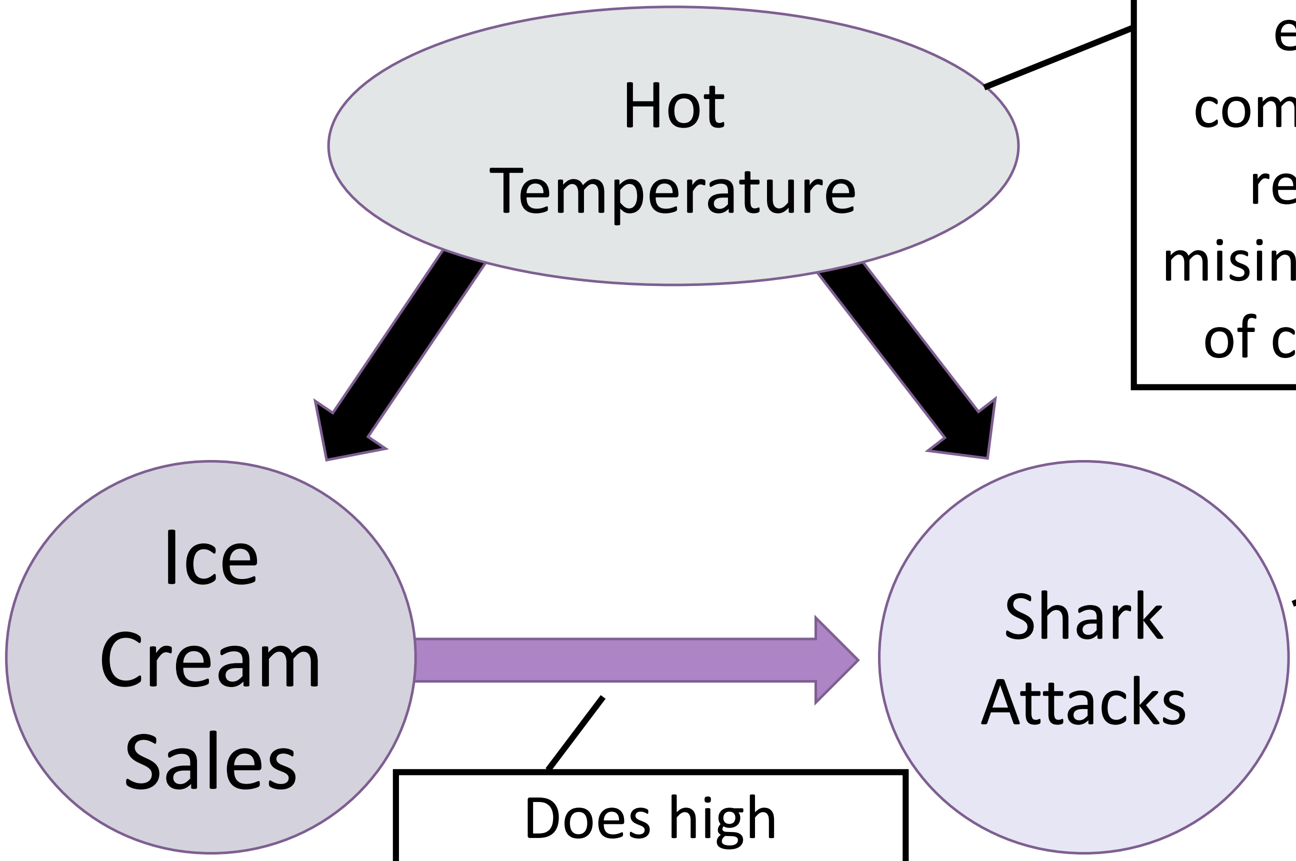
Correlation ➛ things tend to go together (or in opposite directions)

- Learning about one is informative about other

Causation ➛ changing one (from the outside) tends to change the other

- Manipulation of one leads (probabilistically) to variation in the other

More about Misinterpreting Correlation!



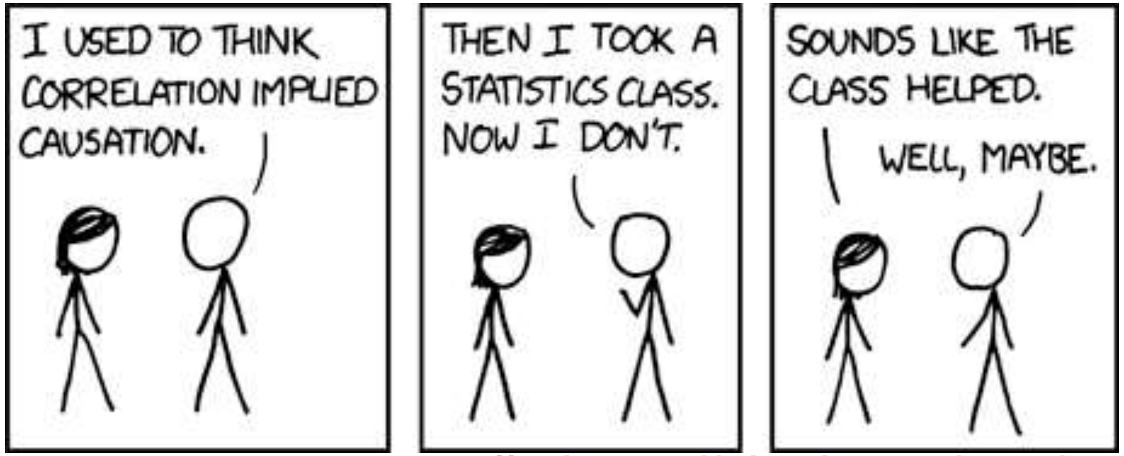
Often, an excluded common cause results in a misinterpretation of correlation!

Does high correlation imply causation?

So…to prevent shark attacks, we should limit the number of ice cream cones sold, right?

Causation vs. correlation

Statistics slogan: *Correlation ≠ Causation*



Credit: https://xkcd.com/552/

Better slogan: "Correlation doesn't *cause* causation, but is *correlated* with causation."

Prof. David Danks' summary: "Correlation is a noisy indicator of causation."

Causation vs. correlation

Different uses for each:

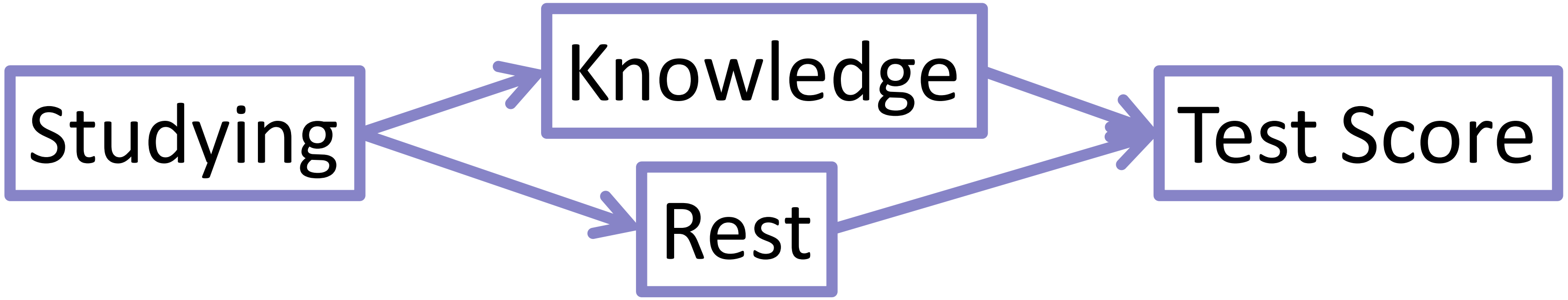
Correlation	Causation
Classifying & identifying	Influencing & acting
Informational value of different evidence	Using evidence to guide policy or actions
Prediction & reasoning given observations	Prediction & reasoning given interventions
Probable explanations for some event or issue	Ways to produce or prevent an event or problem

Causal learning: Framework

Causal graphical models

Graph ➛ qualitative (direct) causation

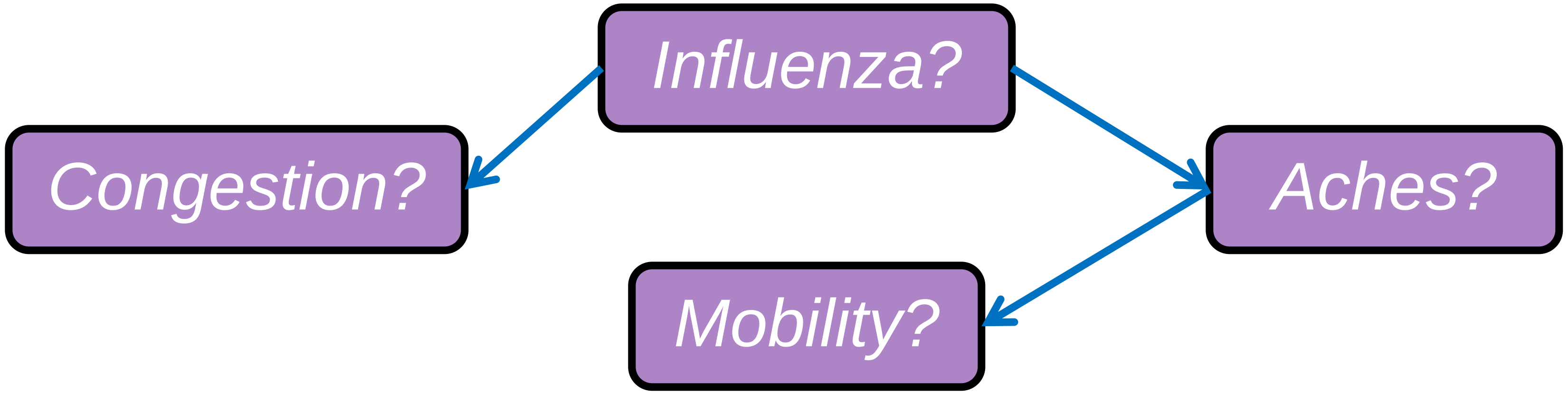
- Directed Acyclic Graph over variables
- Many variations (time-indexing, context variables, …)



Using causal knowledge

Basic idea of actions/interventions:

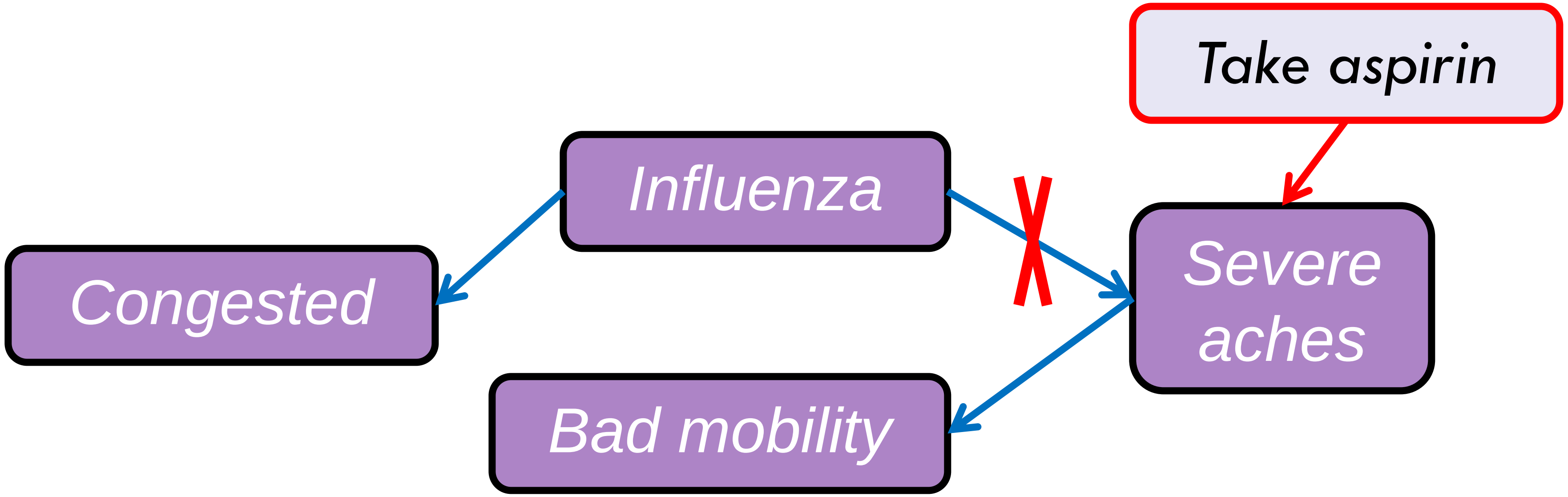
- Often, can “take control” of a node
- A *manipulation* that *changes* the causal system from “outside”
  - In contrast with merely observing the system



Using causal knowledge

Basic idea of actions/interventions:

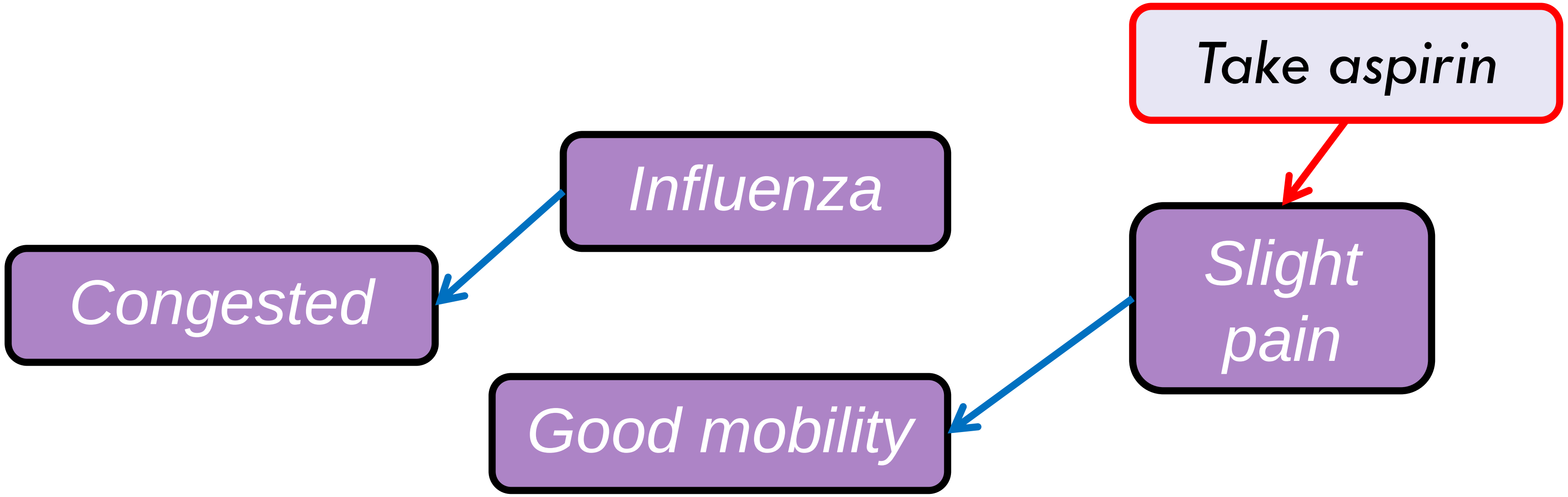
- Often, can “take control” of a node
- A *manipulation* that *changes* the causal system from “outside”
  - In contrast with merely observing the system



Using causal knowledge

Basic idea of actions/interventions:

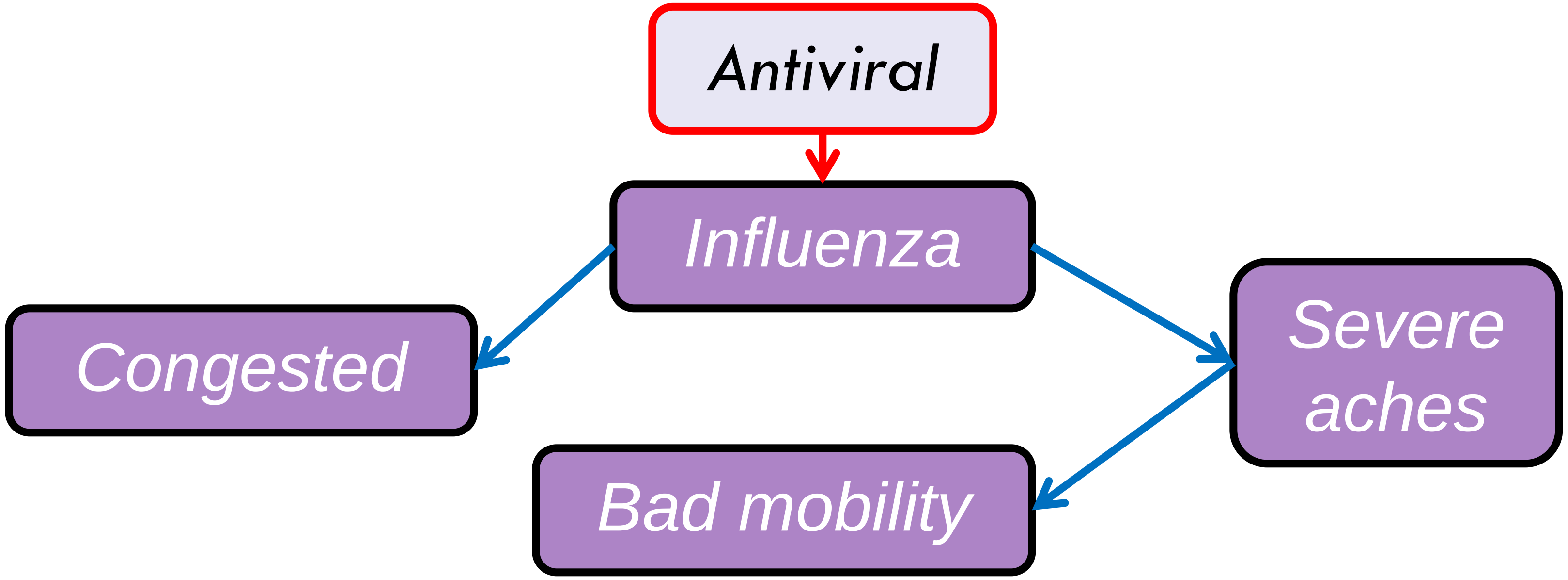
- Often, can "take control" of a node
- A *manipulation* that *changes* the causal system from "outside"
  - In contrast with merely observing the system



Using causal knowledge

Basic idea of actions/interventions:

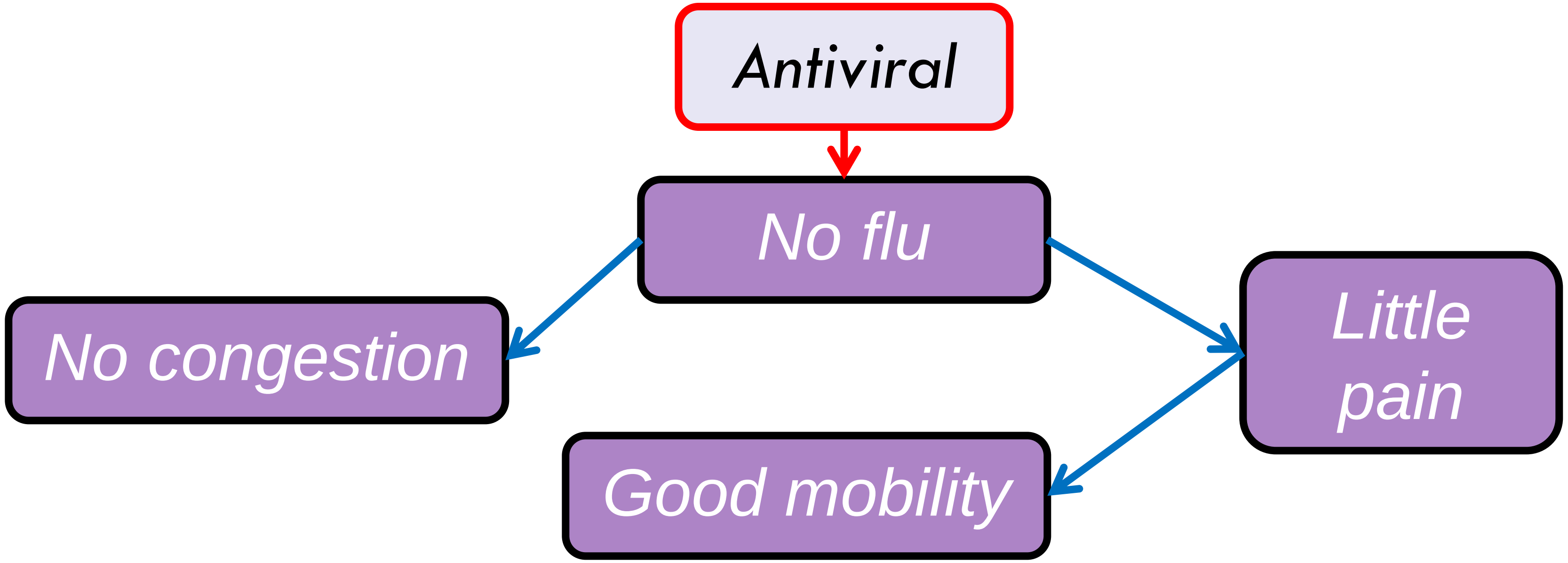
- Often, can "take control" of a node
- A *manipulation* that *changes* the causal system from "outside"
  - In contrast with merely observing the system



Using causal knowledge

Basic idea of actions/interventions:

- Often, can “take control” of a node
- A *manipulation* that *changes* the causal system from “outside”
  - In contrast with merely observing the system



Resurgence of Causal Learning in the Past 30 Years

Sewall Wright Path Models (1920's)

Structural Equation Models (1930's)

Social Science Path Models (1960's)

Bayesian Networks (1980's)

Glymour & Spirtes *et al* 1st ed. book on Causality (1988)

Pearl's Probabilistic Reasoning (1988)

Pearl's 1st ed. book on Causality (2000)

1930 1980 1985 1990 1995 2000 2005 2010

Glymour & Spirtes *et al* 2nd Edition Book on Causality (2001)

Pearl's 2nd Edition Book on Causality (2009)

Morgan Counterfactuals & Causality (2014)

Peters Elements of Causal Inference (2017)

Pearl The Book of Why (2018)

TETRAD – An Open Source Tool for Causal Learning

Carnegie Mellon University

http://www.phil.cmu.edu/tetrad/

University of Pittsburgh

http://www.ccd.pitt.edu/

For video tutorials from 2016 summer short course:

http://www.ccd.pitt.edu/training/presentation-videos/

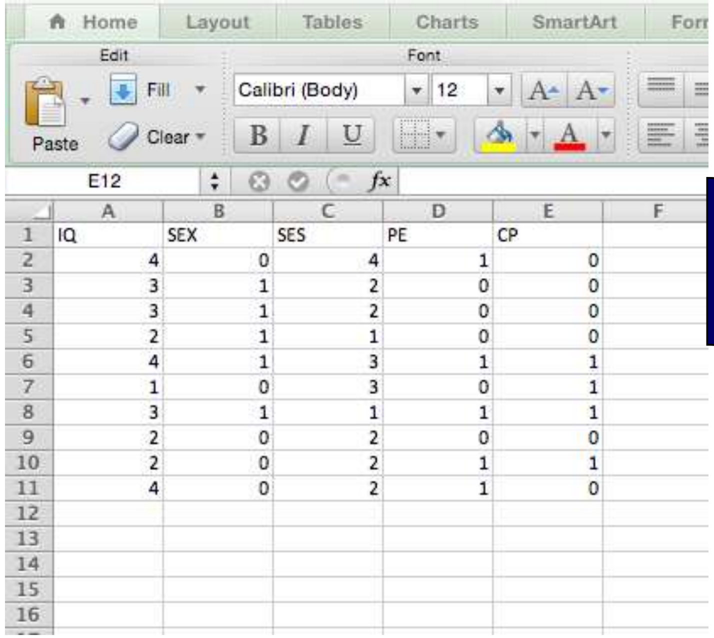
CMU OLI - Causal and Statistical Reasoning

http://oli.cmu.edu/courses/future/causal-statistical-reasoning/

Using a Causal Discovery Algorithm

Person XYZ

1. IQ: ______
2. Socio-Economic-Status: ______
3. Parental Encouragement: ______
4. College Plans: ______
5. Sex: ______



	A	B	C	D	E	F
1	IQ	SEX	SES	PE	CP	
2	4	0	4	1	0	
3	3	1	2	0	0	
4	3	1	2	0	0	
5	2	1	1	0	0	
6	4	1	3	1	1	
7	1	0	3	0	1	
8	3	1	1	1	1	
9	2	0	2	0	0	
10	2	0	2	1	1	
11	4	0	2	1	0	
12						
13						
14						
15						
16						

(Optional) Background Knowledge +

PC (or other) Algorithm

Pattern

Causal learning: Algorithms

Multiple types of methods for this idea:

1. **Constraint-based**: Calculate independences in the data and do “backwards inference”
2. **Score-based** (**Bayesian**): Calculate the likelihood of different DAGs given the data
3. **Hybrid**: Use constraint-based to get “close,” then Bayesian search around neighborhood

Example Constraint-Based and Score-Based Algorithms

PC Stable, constraint-based search algorithm

- Variant of PC, the most widely used algorithm (PC = **P**eter Spirtes and **C**lark Glymour)
- Resulting search graph does not depend on the order of the variables
- Parameters to tune (settings for running the algorithm):
  - **Independence Test type**: for example, Chi Square Test
  - **Alpha**: cutoff for p-values in independence testing; for small datasets, choose higher Alpha
  - **Collider discovery and conflicts**: Conservative (CPC) or Max-P; and Orient bidirected
  - **Maximum size of conditioning set**: when sample size is small, chose value in range 1..3

FGES (Fast Greedy Equivalent Search), score-based search algorithm

- Parameters to tune (settings for running the algorithm):
  - **Scoring method**: for example, BIC Score (BIC = Bayesian Information Criterion)
  - **Penalty Discount**: the default is 2; higher values lead to sparser graphs

Early Results

Practitioner	Software Size	Architecture	Complexity	Leadership
Challenge: Which factors affect a programmer's coding effort and quality?	**Challenge:** Which approaches to measuring code size most reflect factors affecting total effort?	**Challenge:** How might a project manager decide which areas of code to prioritize for maintenance?	**Challenge:** Which program and system complexity factors most affect cost, schedule, and performance?	**Challenge:** For action planning, which attributes of teaming and leadership improve team performance?
Approach: Apply Causal Discovery to data from students coding to the same ten requirements specifications.	**Approach:** Apply Causal Discovery to USC's Unified Code Count (UCC) project dataset.	**Approach:** Apply Causal Discovery to the results of a static code and design structure analysis to determine which type of architectural pattern violation most affects code quality.	**Approach:** Apply Causal Discovery to an existing project-survey dataset.	**Approach:** Apply Causal Discovery to results of 18 months of weekly surveys of software engineers from across a DoD organization to determine which factors most affect cost, schedule, and quality.
Results: To achieve precision, software estimation models should include both objective measures of requirements size as well as programmer-specific coding and defect factors.	**Results**: For IT-type systems, only COSMIC Function Points, Programmer Capability, and Documentation-Aligns-with-Lifecycle-Needs repeatedly recur as direct causes of total effort.	**Results:** Cyclic dependency was the single architecture pattern violation affecting code quality.	**Results:** The original analysis identified difficult requirements, stakeholder relationships, and cognitive fog; causal discovery confirmed only cognitive fog.	**Results:** Of the 20+ factors found to be highly correlated with cost, schedule, and quality, direct causal relationships were found for only two: Good Improvement Data and Stress From Overtime.

Summary

We're learning a lot about how to model subsets of the software development process with causal analytic techniques, giving us insight into how to intervene in a project to improve its outcomes.

- What we're learning will be captured through improved methodology and algorithms

Causal learning does provide useful insight over (and for improved) multiple regression, though smaller datasets remain challenging.

Further progress depends on new collaborations offering access or analysis of new datasets.

Where to Learn More

Pearl J, Glymour M, Jewell NP. Causal Inference in Statistics – A Primer (John Wiley & Sons, 2016).

Pearl J, Mackenzie D. The Book of Why: The New Science of Cause and Effect. (New York: Basic Books, 2018).

Spirtes Peter, “Introduction to causal inference.” Journal of Machine Learning Research 11 (2010) 1643-1662. http://jmlr.org/papers/volume11/spirtes10a/spirtes10a.pdf

The Tetrad Project. http://www.phil.cmu.edu/tetrad/

Jonas Peters, Dominik Janzing, Bernhard Schölkopf. Elements of Causal Inference: Foundations and Learning Algorithms. (Adaptive Computation and Machine Learning series, 2017).

Clark Glymour, Kun Zhang, and Peter Spirtes. A Brief Review of Causal Discovery Methods. (Frontiers, 2018).

Malinsky D, Danks D. Causal discovery algorithms: A practical guide. (Philosophy Compass, 2018). https://doi.org/10.1111/phc3.12470

Raghu VK, Poon A, Benos P. Evaluation of Causal Structure Learning Methods on Mixed Data Types. (JMLR 2018).

Selected Case Studies

Comparison of Parametric Cost Estimation methods -1

This study was our project's second collaboration, and was undertaken with Anandi Hira and Barry Boehm, late 2017-early 2018.

How is estimation typically done today?

- With parametric cost (effort) estimation models
- For example, COCOMO II:

$$Effort = 2.94 \times Size^{E} \times \prod_{i=1}^{17} EM_i$$

- Input: size, product and personnel attributes
- Effort in Person-Months (PM)
- Domain Experts
- Data calibration
- No causal analysis

Comparison of Parametric Cost Estimation methods -2

Dataset: Unified Code Count (UCC) code metrics tool maintained at USC

- UCC is released to users across world
- Primarily used in U.S. Aerospace industry
- Recommended for SLOC-based size input to Software Resources Data Report (SRDR)

Size estimators compared:

- Equivalent SLOC (ESLOC)
- IFPUG Function Points (FP)
- IFPUG Software Non-functional Assessment Process (SNAP)
- COSMIC Function Points (CFP)

Other variables analyzed: Applications Experience, Platform Experience, Use of Software Tools, Personnel Continuity, Documentation Match to Needs (DOCU), Analyst Capability (ACAP), Programmer Capability (PCAP), Product Complexity (CPLX)

Outcome of interest: Total Effort

Domain: mostly small-enterprise software of size 45 to 1425 logical LOC; four-month release cycles.

Comparison of Parametric Cost Estimation methods -3

	Direct cause of total effort							
Algorithm	**ESLOC**	**FP**	**SNAP**	**CFP**	**CPLX**	**ACAP**	**PCAP**	**DOCU**
Stepwise Regression (Adj R^2=.84)				Yes			Yes	
PC				Yes			Yes	
PC-Stable				Yes				
FGES				Yes			Yes	
FASK		Yes	Yes	Yes				Yes

Summary:

(1) Enterprise IT-type systems: these causes repeatedly recur:

- **Cosmic Function Points** (CFP)
- **Programmer Capability** (PCAP)
- Documentation-Aligns-with-Lifecycle-Needs

(2) Complemented results from multiple regression, but did not provide much additional insight.

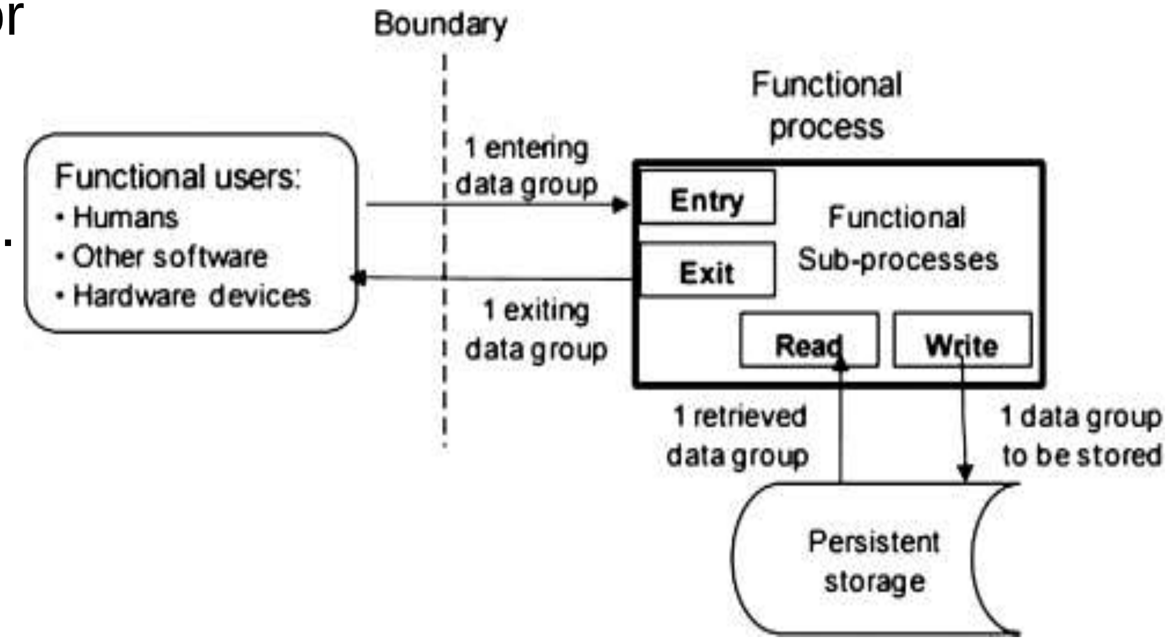
Threats to Validity

- Features of enterprise SW (multiple platforms, GUI, reports)
- Small Sample Size
  - Resulting in sparsely-connected graphs
- Dataset of convenience – UCC

Comparison of Parametric Cost Estimation methods -4

So what is it about **Cosmic Function Points** (CFP) that may make it a good predictor of effort for small-to-medium enterprise software?

- CFP is primarily intended for business applications dominated by functions that input data, store and retrieve data, and output **data**.
- CFP works by decomposing a specification for a system into functional processes into which data flows into or from. You then count the # of data group moves into or out of the system.



To the extent that developing such enterprise systems primarily involves identifying such functions and flows, effort can be predicted.

CFP was not evaluated for other applications, nor for predicting quality.

Case Study 1 (Complexity Drivers and Project Success) -1

Source: Sarah Sheard's Ph.D. dissertation, 2012

Research question: what complexity factors, determinable early in life of a program, impact project outcomes such as cost overrun, late delivery, performance shortfall?

Dataset: survey covering complexity factors and project success

- 41 items on a 3-point or larger ordinal scale
- 1 item (Delivered) on a binary scale (yes/no)
- 7 items representing project outcomes:
  - Delivered, EvolOp, GoodEst, Late, OverCost, PerfGap, Success
- 81 survey responses, 3/4 of them from aerospace

Case Study 1 (Complexity Drivers and Project Success) -2

Original result: Three of the complexity variables strongly predicted all outcomes:

Req-Diff	Difficult requirements are considered difficult to implement or engineer, are hard to trace to source, and have a high degree of overlap with other requirements. How many system requirements were there that were Difficult? (1) 1-10 (2) 10-100 (3) 100-1000 (4) 1000-10,000 (5) Over 10,000
CogFog	"The project frequently found itself in a fog of conflicting data and cognitive overload". Do you agree with this statement? (1) Strongly Agree (2) Agree (3) Neutral (4) Disagree (5) Strongly Disagree
StakeRelnship	Where did your project fit in the following eight attributes, on a scale of (1)Traditional, (2)Transitional, or (3)Messy Frontier? [Translating for] Stakeholder relationships: (1)Relationships stable (2)New relationships (3)Resistance to changing relationships

Case Study 1 (Complexity Drivers and Project Success) -3

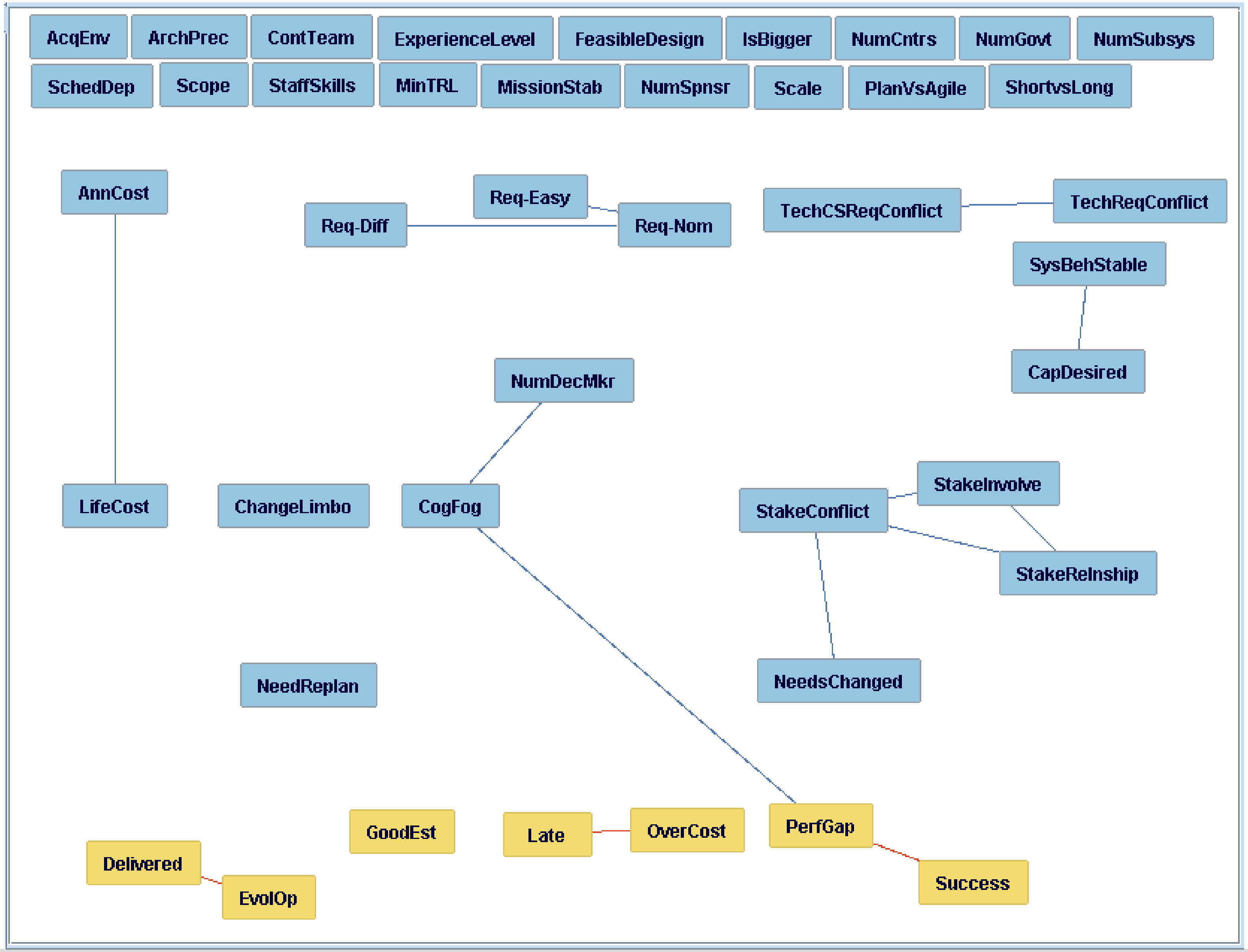
Both PC-Stable and FGES algorithms were applied.

Here is an example search result from applying PC-Stable (Alpha=.10) to the full dataset.

Outcome (Tier 5) variables are highlighted in yellow.

Note CogFog relationships.

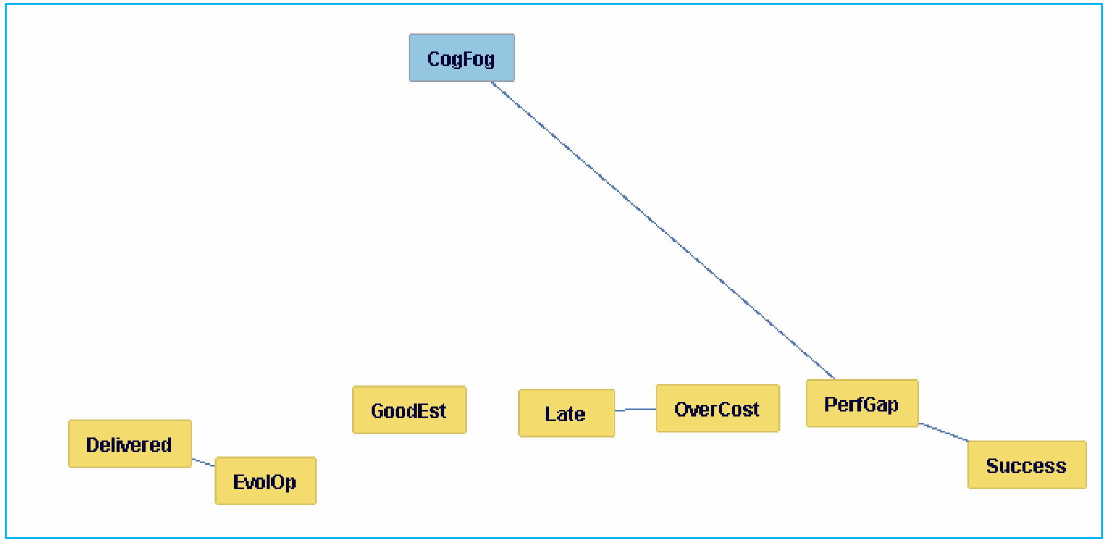
Variables without causal relationships were moved to the very top to help highlight for which variables direct causal relationships were found.

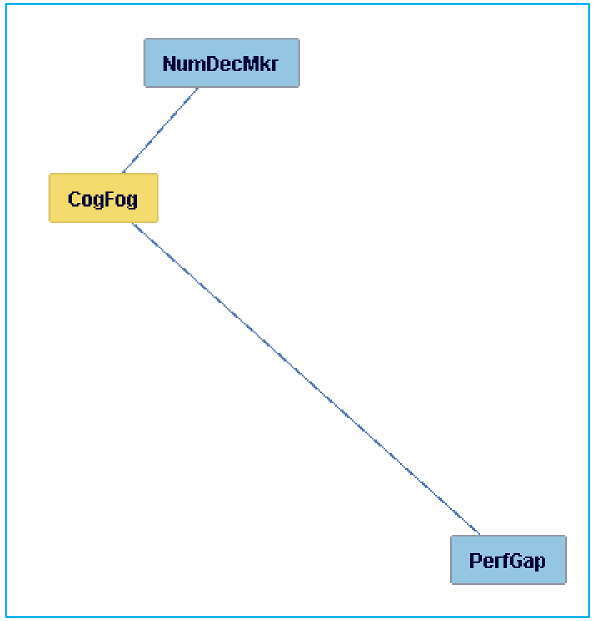


Case Study 1 (Complexity Drivers and Project Success) -4

On this slide we show two **Markov blankets**: (1) for all project outcomes; (2) for CogFog.

- A **Markov blanket** is a node, its parents, its children, and its children's parents. The Markov blanket of a node is the only knowledge needed to predict the behavior of that node. (Wikipedia)





In summary, what do these graphs tell us? How might we intervene in a project having a low likelihood of meeting project outcomes? We would **intervene** in a way that **reduces the number of decision makers**.

- This in turn should help **reduce** the amount of **cognitive fog**, which should help **reduce the performance gap** (specified mission-critical features vs. what was actually achieved).

Case Study 2 (Team Dynamics and Project Success) -1

Source: SEI Client, 2014

Research question: what team dynamics factors drive software project success?

Dataset: weekly surveys issued randomly to 30 software staff

- 33 items on a binary scale (Yes / No) representing independent team variables
  - The subset of the 120+ team factors identified by Watts Humphrey that reasonably could change on a weekly basis
- 3 items on a 4 point ordinal scale representing dependent project outcomes:
  - Project Quality, Schedule and Cost

Rationale for Binary Data:

- Staff were overworked; informal piloting indicated survey must not exceed 2-3 minutes of response time
- Staff wanted to point and click with minimal scrolling
- We achieved 90% response rates

Case Study 2 (Team Dynamics and Project Success) -2

Traditional correlation results:

Correlation measures used included Kendall tau-b, Kendall tau-c, Gamma and Spearman's. All were in agreement using the 0.05 cutoff for significance (blue highlighted cells).

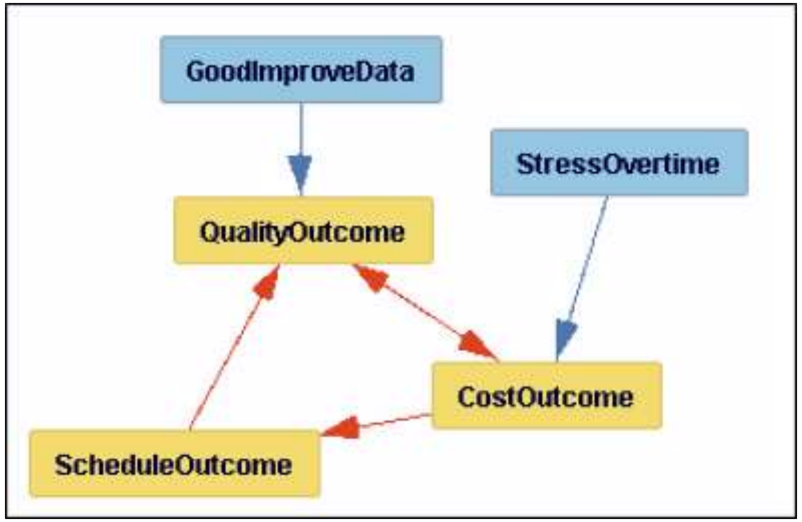
Ordinal logistic regression using 0.05 alpha for significance and McFadden pseudo Rsquare indicated significant factors (red borders).

	QualityOutcome	CostOutcome	ScheduleOutcome
ID			
WeekNumber			
Squadron			
IndivUnclearGoals			
IndivMotivateByLeader			
LeaderDealPerfProblems			
TeamConflictNotResolved			
PerfMeasured			
PrioritizedWork			
ChangeDirection			
QualitySuffer			
IndivUnhappyTasks			
MissedLateDecisions			
IndivSatisRole			
GoodMeetings			
ProcessNonCompliance			
TeamConsensus			
LackConsensusImpacts			
GoodProgressReviews			
GoodImproveData			
OpenClimateIdeas			
ExternalFeedback			
TeamLoadBalanced			
ReqtsNotAnalyzed			
NeedUnplannedHelp			
CustomerInvolved			
ProcessGuidanceUsed			
ProcessProbResolved			
IndivQualityData			
IndivTaskDisatisfaction			
GoodTeamCommunication			
StressOvertime			
OpenClimateIdeas2			
OpenTeamDiscussion			
InternalTeamCooperation			
F2FwithLeader			

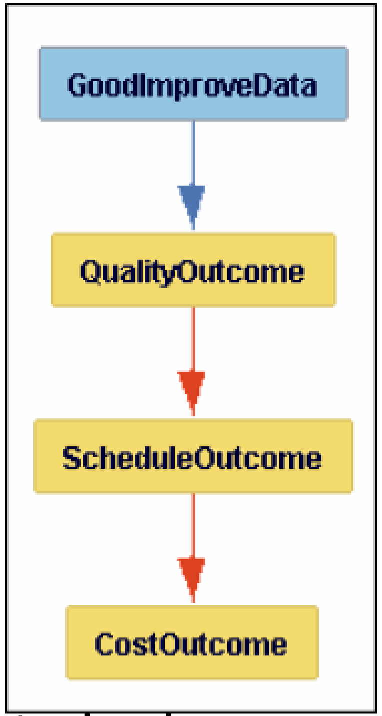
Case Study 2 (Team Dynamics and Project Success) -3

On this slide we show two Markov blankets for the **set of three outcomes**:

1) using PC-Stable, and



2) using FGES.



Although the two algorithms differ on the directed edges among the three outcomes, there is agreement on GoodImproveData causing QualityOutcome. PC-Stable adds StressOvertime as a cause of CostOutcome.

Case Study 2 (Team Dynamics and Project Success) -4

Although traditional statistical correlation depicted:

- 18 factors highly correlated with Quality [2 confirmed with Logistic Regression]
- 5 factors highly correlated with Cost, and
- 21 factors highly correlated with Schedule,

the causal search discovered:

- 1 factor (GoodImproveData) appears to cause Quality performance,
- 1 factor (StressOvertime) appears to cause Cost performance, and
- No independent factors appear to cause Schedule performance.

Mike Konrad
Email: mdk@sei.cmu.edu
Bob Stoddard
Email: rws@sei.cmu.edu
Sarah Sheard
Email: sheard@sei.cmu.edu

U.S. Mail
Software Engineering Institute
4500 Fifth Avenue
Pittsburgh, PA 15213-2612 USA

Website
www.sei.cmu.edu/contact.cfm