Bayesian Network Research at Monash

Kevin B. Korb
Clayton School of IT
Monash University
Melbourne, VIC Australia

Bayesian Networks

What are Bayesian Networks?

Uncertainty is all around us, and usefully represented with probabilities.

Bayesian nets are computer representations that simplify probabilistic reasoning.

Uncertainty

Uncertainty

Uncertainty

Bayesian Networks

Definition (Bayesian Network)

A graph where:

1. The nodes are random variables.
2. Directed arcs represent direct dependencies between nodes.
3. Each node has a conditional probability function that *quantifies* the effects of its parents.
4. It is a directed acyclic graph (DAG), i.e. no directed cycles.

Pearl's Alarm Example

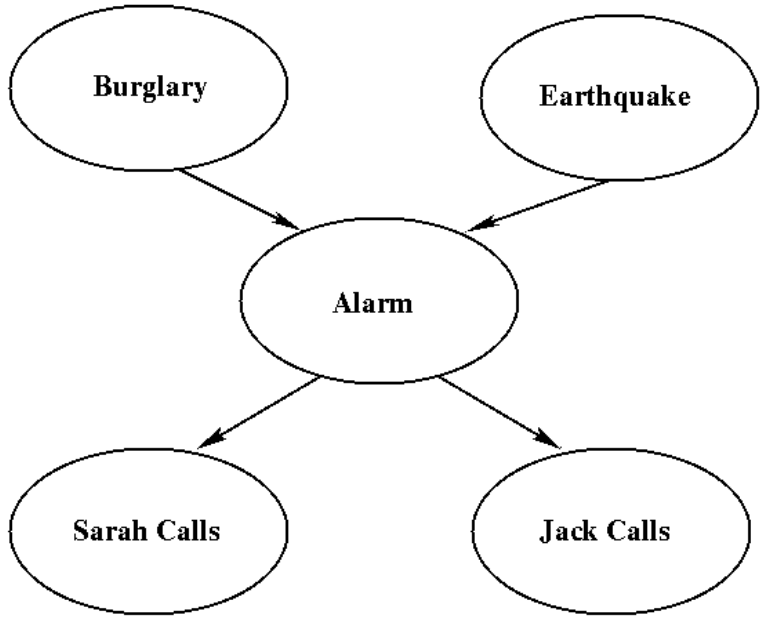
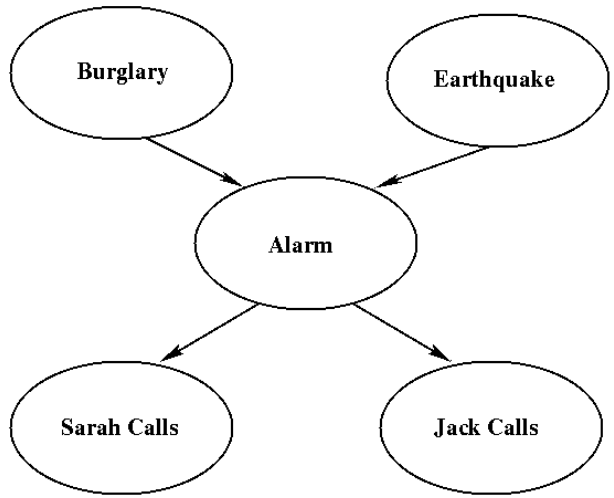


Figure: Pearl's Alarm Example

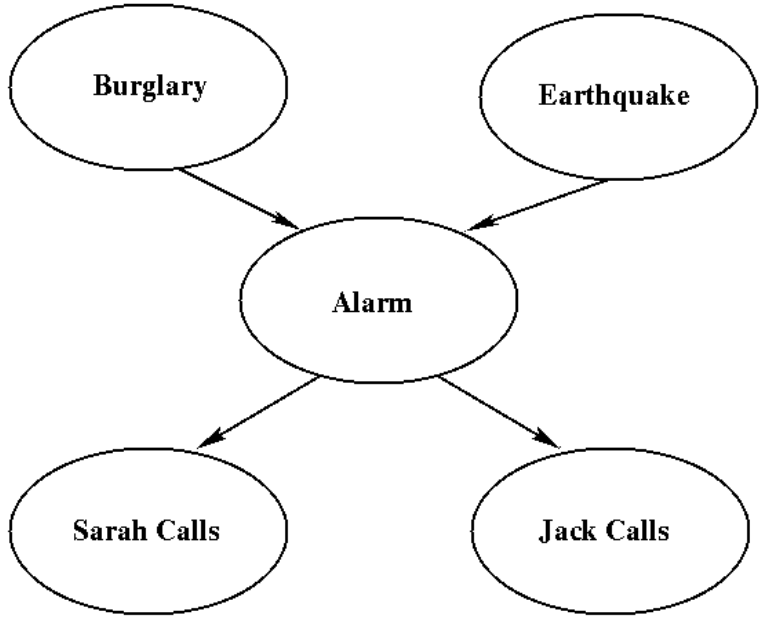
Factorization

The advantage of graphical models is that we have a grahical criterion for systematically simplifying this computation, yielding:

$$
\begin{aligned}
P(E,B,A,J,S) &= \frac{P(B,E,A,J,S)}{P(E)}P(E) \\
&= P(B,E,A,J|E)P(E) \\
&= \ldots \\
&= P(S|A)P(J|A)P(A|B,E)P(B)P(E)
\end{aligned}
$$



Causality & Probability



Three important relationships:

- Causal chains: $P(C|A \wedge B) = P(C|B) \equiv A \perp\!\!\!\perp C|B$
- Common causes: $P(C|A \wedge B) = P(C|B) \equiv A \perp\!\!\!\perp C|B$
- Common effects (collisions):
  $P(A|C \wedge B) \neq P(A)P(C) \equiv A \not\perp\!\!\!\perp C|B$

Causality and Probability

Dependency signature

Note that the conditional dependency structures are exact opposite btw chains/common ancestry and “collisions”.

- Marginal dependence: marginal independence
- Conditional independence: conditional dependence

This is key for causal discovery.

Bayesianism

The Dubious Reverend Bayes (1702-1761)

Are Bayesian networks Bayesian?

Bayesianism

The Bayesian Proposal:

Use probability theory to represent uncertainty

Bayesianism

Bayes' Theorem (1763)

$$P(h|e) = \frac{P(e|h)P(h)}{P(e)}$$

Bayesianism

Conditionalization

$$P'(h) = P(h|e)$$

I.e., it claims we can read Bayes' theorem as:

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Prob of evidence}}$$

Two basic assumptions:

1. Joint priors over $\{h_i\}$ and e exist.
2. Total evidence: e, and only e, is learned.

Bayesianism

Bayesianism

Asserts that conditionalization is a key to understanding scientific inference.

Two key technologies liberated Bayesianism in the last two decades:

- Bayesian networks
- Stochastic sampling (computer oomph)

Learning Bayesian Networks

LBN

Learning Bayesian networks = causal discovery = **structure** *Learning*

Orthodox Mantra

Correlation Does Not Imply Causation!

Not even a little bit:

RA Fisher: Learning a probabilistic dependency will not advance our causal understanding even one step.

Causal Discovery

Some Non-Examples

- Firemen cause Fires: the larger the fire the more fire trucks there are.
- Ice Cream causes Drowning/Shark Attacks.
- Volume causes Surface Area; Height causes Weight.
- CO_2 causes Human Population Growth

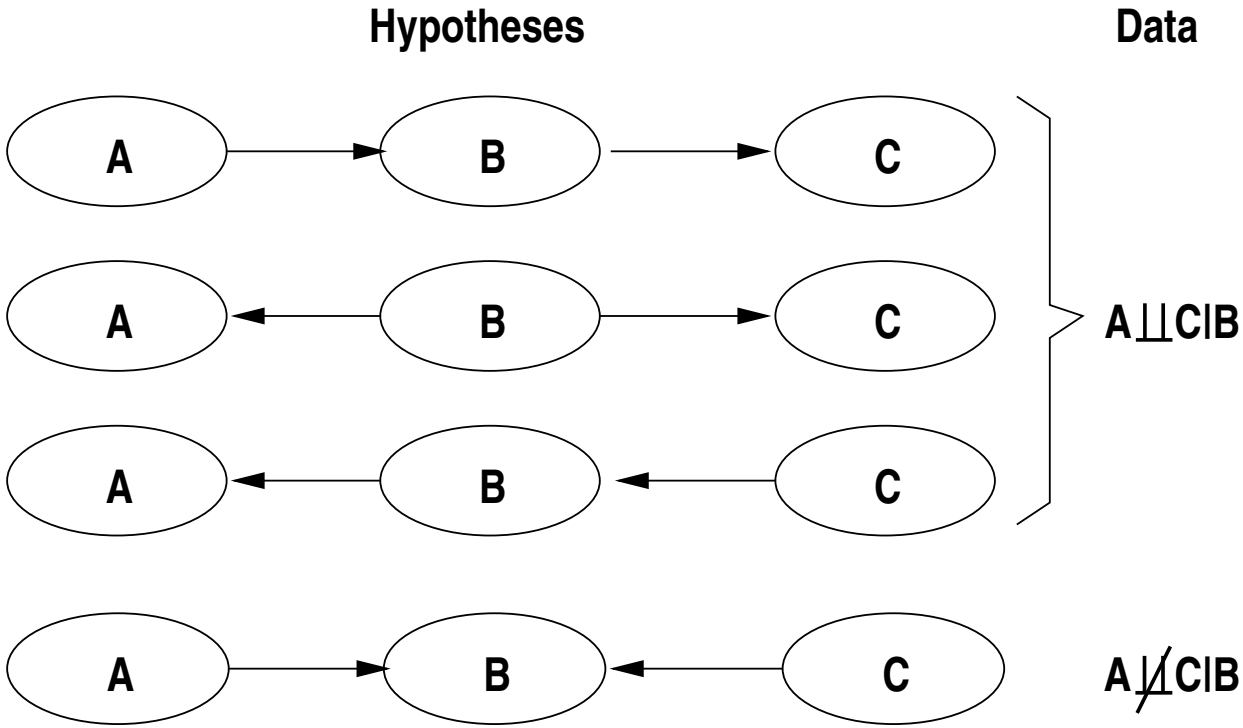
Despite this we have:

Reichenbach's Common Cause Principle (1956)

No Correlation without Causation

Causal Discovery: Possible

There are four types of undirected 3-chains:



In Popperian language, we can “falsify” the one causal pattern or the other.

Causal Discovery: Possible

Definition (Pattern)

A pattern is (equivalently):

1. A set of statistically equivalent dags
2. A maximal set of dags having the same variables and probabilistic depedencies (assuming each arc is "working", i.e., carries dependence)

Example (Patterns on Last Slide)

- $A \rightarrow B \rightarrow C$, $A \leftarrow B \leftarrow C$, $A \leftarrow B \rightarrow C$
- $A \rightarrow B \leftarrow C$

Causal Discovery: Possible

So, 3-chains are partially learnable, and this can be scaled up.

- At least to large scale patterns

  With all but a few arcs directed

- With experimental data, in the ideal case with *all* arcs directed (Korb & Nyberg, 2006)

$\Rightarrow$ For a comprehensive argument, see Glymour, et al. (1987, Part I)

Constraint-Based Learning

Verma-Pearl Algorithm

IC algorithm (Verma and Pearl, 1991)

0. Given an Oracle who can answer any (in)dependency question, e.g., $X \perp\!\!\!\perp Y | \mathbf{S}$?
1. Link any two variables X and Y s.t. for every $\mathbf{S}$ s.t. $X, Y \notin \mathbf{S}$ $X \not\perp\!\!\!\perp Y | \mathbf{S}$
2. For every undirected, uncovered collision $X - Z - Y$ orient $X \rightarrow Z \leftarrow Y$ iff $X \not\perp\!\!\!\perp Y | \mathbf{S}$ for **every** $\mathbf{S}$ s.t. $X, Y \notin \mathbf{S}$ and $Z \in \mathbf{S}$.
3. Remove potential inconsistencies.

PC: TETRAD

Spirtes, Glymour and Scheines (1993) made this approach practical. Replace the Oracle with statistical tests:

- for linear models a significance test on partial correlation

$$X \perp\!\!\!\perp Y | \mathbf{S} \text{ iff } \rho_{XY \cdot \mathbf{S}} = 0$$

- for discrete models a χ^2 test on the difference between CPT counts expected with independence (E_i) and observed (O_i)

$$X \perp\!\!\!\perp Y | \mathbf{S} \text{ iff } \sum_i O_i \ln \left(\frac{O_i}{E_i} \right)^2 \approx 0$$

Implemented in their **PC Algorithm**

PC Algorithm

Versions of this (“PC” algorithm) are found in:

- Hugin
- GeNIe
- Weka
- Tetrad IV

Problems:

- When stat sig tests err, mistakes cascade
- Can’t distinguish between some alternatives, so does not direct all the arcs!

Metric Causal Discovery

A very different approach is *metric* learning of causality:

- Develop a score function which evaluates any Bayesian network *as a whole* relative to the evidence.
- Originally this was done in a brute force Bayesian computation of

$$P(dag|data)$$

  by Cooper & Herskovits (1991) in their K2 program
- CD then means: search the space of dags looking for that dag which maximizes the score.

Metric Discovery Programs

K2 (Cooper & Herskovits)
Greedy search. Mediocre performance.

MDL (Lam & Bacchus, 1993; Friedman, 1997)
An information-theoretic scoring function with various kinds of search, such as beam search. Friedman allows for hybrid local structure.

BDe/BGe (Heckerman & Geiger, 1995)
A Bayesian score; edit-distance priors supported; returns a pattern. Good performance.

CaMML (Korb & Nicholson, 2010; Part II)
A Bayesian information-theoretic scoring function with MCMC (Metropolis search); returns dags and patterns. Performance similar to BDe/BGe. Supports priors and hybrid local structure.

CaMML

Minimum Message Length (Wallace & Boulton 1968) uses **Shannon's information measure**:

$$I(m) = -\log P(m)$$

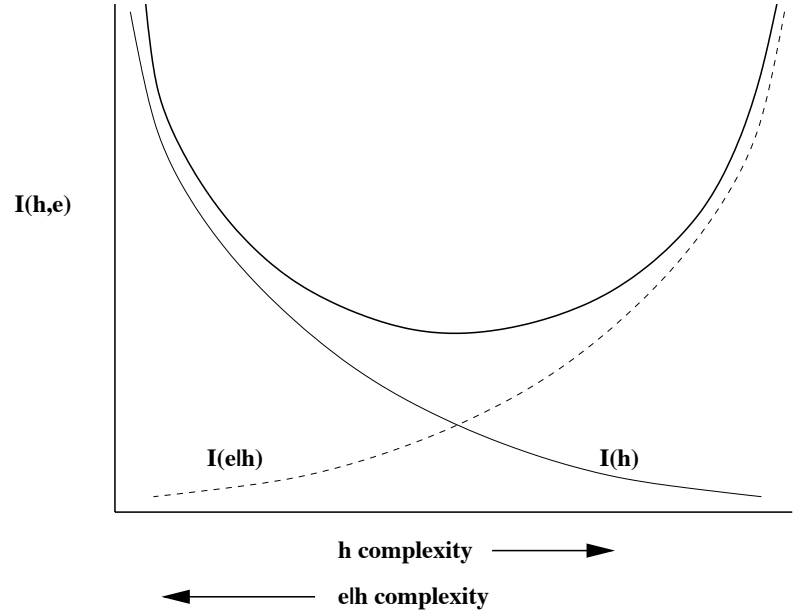
Applied in reverse, we can compute $P(h, e)$ from $I(h, e)$.
Given an *efficient* joint encoding method for the hypothesis & evidence space (i.e., satisfying Shannon's law), MML:

> *Searches* $\{h_i\}$ *for that hypothesis h that minimizes* $I(h) + I(e|h)$.

Applies a trade-off between

- Model simplicity
- Data fit

MML Metric



Equivalent to that h that maximizes $P(h)P(e|h)$ — i.e., $P(h|e)$.

$$
\begin{aligned}
I(h, e) &= I(h) + I(e|h) \\
-\log P(h, e) &= -\log P(h) - \log P(e|h) \\
-\log P(h, e) &= -\log P(h)P(e|h) \\
P(h, e) &= P(h)P(e|h)
\end{aligned}
$$

Hence, $\min I(h, e) \equiv \max P(h, e)$.

MML Metric for Linear Models

- Network:

$$\log n! + (-\log p \times a) + \left(-\log(1-p) \times \left(\frac{n(n-1)}{2} - a\right)\right) - \log E$$

  - $\log n!$ for variable order
  - $(-\log p \times a)$ for a arcs, with prob p
  - $\left(-\log(1-p) \times \left(\frac{n(n-1)}{2} - a\right)\right)$ for pairs lacking arcs, with prob $1-p$
  - $-\log E$ restore efficiency by subtracting the estimated cost of selecting a linear extension

- Parameters given dag h:

$$\sum_{X_j} -\log \frac{f(\theta_j|h)}{\sqrt{F(\theta_j)}}$$

where θ_j are the parameters for X_j and $F(\theta_j)$ is the Fisher information. $f(\theta_j|h)$ is assumed to be $N(0, \sigma_j)$ (vs. MDL's fixed length for parms).

MML Metric for Linear Models

- For X_j given h and θ_j:

$$-\log P(e|h,\theta_j) = \prod_{k=1}^{K} \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\epsilon_{jk}^2/2\sigma_j^2}$$

  where K is the number of sample values and ϵ_{jk} is the difference between the observed value of X_j and its linear prediction.

In short, $MML(h|e) = I_{MML}(h) + I_{MML}(\theta|h) + I_{MML}(e|\theta,h)$

MML Metric for discrete models

We can use $P_{CH}(h_i, e)$ (from Cooper & Herskovits) to define an MML metric for discrete models.

Difference between MML and Bayesian metrics:

> *MML partitions the parameter space and selects optimal parameters.*

Equivalent to a penalty of $\frac{1}{2} \log \frac{\pi e}{6}$ per parameter (Wallace & Freeman 1987); hence:

$$I(e, h_i) = \frac{s_j}{2} \log \frac{\pi e}{6} - \log P_{CH}(h_i, e) \qquad (1)$$

Applied in MML Sampling algorithm.

MML search algorithms

MML metrics need to be combined with search. This has been done three ways:

1. Wallace, Korb, Dai (1996): greedy search (linear).
   - Brute force computation of linear extensions (small models only).
2. Neil and Korb (1999): genetic algorithms (linear).
   - Asymptotic estimator of linear extensions
   - GA chromosomes = causal models
   - Genetic operators manipulate them
   - Selection pressure is based on MML
3. Wallace and Korb (1999): MML sampling (linear, discrete).
   - Stochastic sampling through space of totally ordered causal models (TOMs)
   - No counting of linear extensions required

MML Sampling

Search space of totally ordered models (TOMs).
Sampled via a Metropolis algorithm (Metropolis et al. 1953).

From current model M, find the next model M' by:

- Randomly select a variable; attempt to swap order with its predecessor.
- Or, randomly select a pair; attempt to add/delete an arc.

Attempts succeed whenever $P(M')/P(M) > U$ (per MML metric), where U is uniformly random from $[0 : 1]$.

MML Sampling

Metropolis: this procedure samples TOMs with a frequency proportional to their posterior probability.

To find posterior of dag *h*: keep count of visits to all TOMs consistent with *h*

> *Estimated by counting visits to all TOMs with identical max likelihoods to h*

Output: Probabilities of

- Top dags
- Top statistical equivalence classes
- Top MML equivalence classes

Extensions to CaMML

Two significant enhancements:

Expert priors (O'Donnell et al., 2006b)

- Being Bayesian, it is relatively easy to incorporate non-default priors into CaMML. We've done this in various ways, specifying strengths for:
  - A prior dag, computing a prior distribution via edit distance
  - Arc densities
  - Topological orders, total or partial

Hybrid model learning (O'Donnell et al., 2006a)

- Allowing varying representations of local structure (CPTs, d-trees, logit model) throughout the network

Expert priors

Support for priors is crucial for most practical applications of CD:

- Many real-world data sets lead to “crazy” learned models: Age causing Gender, Job Success causing Education Level, etc.
- Learned models may be absurdly dense.

CaMML can incorporate a wide variety of prior info:

- *Tiers* of variables (also PC, GES, K2).
- Edit-distance priors based on a single expert model (also BDe/BGe).
- Specific relations (direct or indirect) between variables, as well as an arc density prior.

Camml Availability

- Freeware discrete CaMML:
  `https://github.com/rodneyodonnell/CaMML`
- Linear CaMML(CaMMLL, executables only):
  `https://www.bayesian-intelligence.com/software`

Evaluation Theory

Causal Evaluation Theory

This is a major weakness in the literature. The basic idea is to compare a sequence of learned models with generating models. Whoever's sequence is most similar wins. But what is similarity?

- Most common answer: edit distance. E.g., 1 for each arc omission, 1 for each "commission", 1 (or 2) for reversals (except within a pattern).
- Problem: Not all arcs are created equal.
- Solution: Kullback-Leibler divergence (KL).
- New problem: Fails to discriminate dags within a pattern.
- New solution: Causal Kullback-Leibler divergence (CKL). (See Korb & Nicholson, 2010, ch 9.)

Classifier Evaluation Theory

Given a classifier $f(X_1, \ldots, X_n) \rightarrow C_i$, how can we evaluate it?

- A very basic principle: Having found a classifier using a training set, evaluate it using a (different) test set.
- Since we are often trying to predict class membership, test set predictive accuracy suggests itself.

Predictive Accuracy

	T (p)	F (1-p)
“T”	TP	FP
“F”	FN	TN

Predictive Accuracy

	T (p)	F (1-p)
"T"	0.9	0.2
"F"	0.1	0.8

$$
\begin{aligned}
PA &= p(0.9) + (1-p)(0.8) \\
&= 1 - \text{error rate} \\
&= 1 - (p \times 0.1 + (1-p)0.2)
\end{aligned}
$$

Predictive Accuracy

	Edible	Poison
“Edible”		y
“Poison”	x	

Predictive Accuracy

	Edible	Poison
“Edible”		y
“Poison”	x	

But, $v(x) \neq v(y)$

Failing to eat a good mushroom hurts a lot less than eating a poisonous mushroom!!

Predictive Accuracy

Note that predictive accuracy is also invariant to the confidence of predictions.

In a binary task, a prediction with probability 0.51 is treated the same as a prediction with probability 0.99.

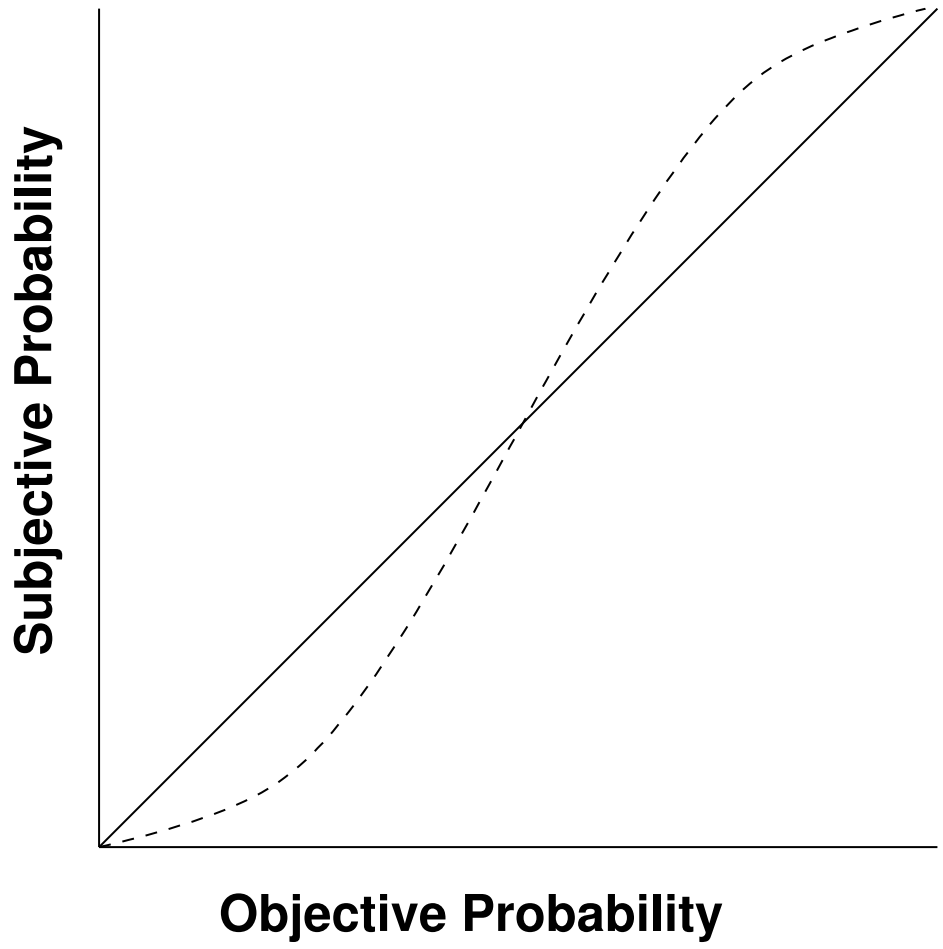
But that's absurd!

- P(Mushroom A is edible) = 0.51
- P(Mushroom B is edible) = 0.99

Those who were indifferent are no longer with us. . . Calibration is a big issue in prediction.

Classification should always be thought of as probabilistic, not categorical.

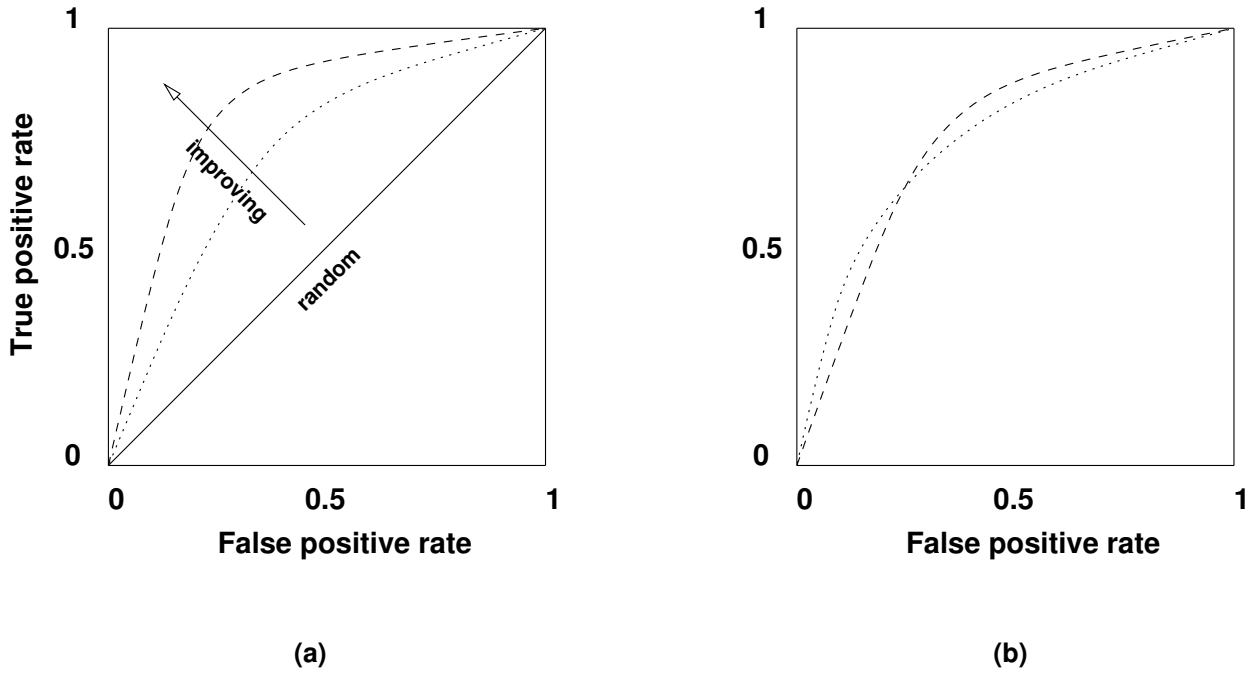
Calibration



ROC

ROC curves have become popular; but they also fail to address these issues.

Maps TPRs to FPRs $P(\text{“}T\text{”}|T)$ *v* $P(\text{“}T\text{”}|F)$:



- PA (for given FPR): $p \times TPR + (1 - p)(1 - FPR)$
- AUC: Integrate under the curve

Cost-Based Classification

Classify (discretize) to maximize expected value of classification. I.e.,

	T (p)	F (1-p)
"T"	tp, u(tp)	fp, u(fp)
"F"	fn, u(fn)	tn, u(tn)

$$\max \sum_i (tp \times u(tp) + fp \times u(fp) + fn \times u(fn) + tn \times u(tn))$$

Note: $tp = P(\text{“}T''|T)p$, etc.

This *ought* to be the gold standard for classification, when turning from training to test sets!

Cost-Based Classification

This has become moderately popular since the work of Peter Turney on cost-sensitive learning of classification trees; e.g., Turney (1995).

- The potential for very substantial improvements in classification + test costs was made clear.
- *Actual* improvements are more elusive, since true classification costs were not estimated, only a sensitivity to costs study was done.
- In general, expected utility studies are hard, because finding justifiable utilities is hard.

Cost-Based Evaluation

The Bayesian Gold Standard

Regardless of the classification/learning method, the Bayesian gold standard for evaluation is/ought to be maximizing expected value in test sets:

$$\max \sum_i (tp \times u(tp) + fp \times u(fp) + fn \times u(fn) + tn \times u(tn))$$

Note that this evaluation combines accuracy and calibration:

- Both greater accuracy and better calibration mean landing more often in the higher utility outcome cells.

TBD_1

Cost-Based Discretization

New Discretization Method 1

Optimize search for the discretization which maximizes expected classification utility.

This should work when utilities are available. It won't otherwise, so we need something else...
Meanwhile, notice this this is a *scoring rule*, not a direct assessment of a data model (such as MDL, MML, etc). I.e., there is NO complexity control on offer.

Bayesian Information Reward

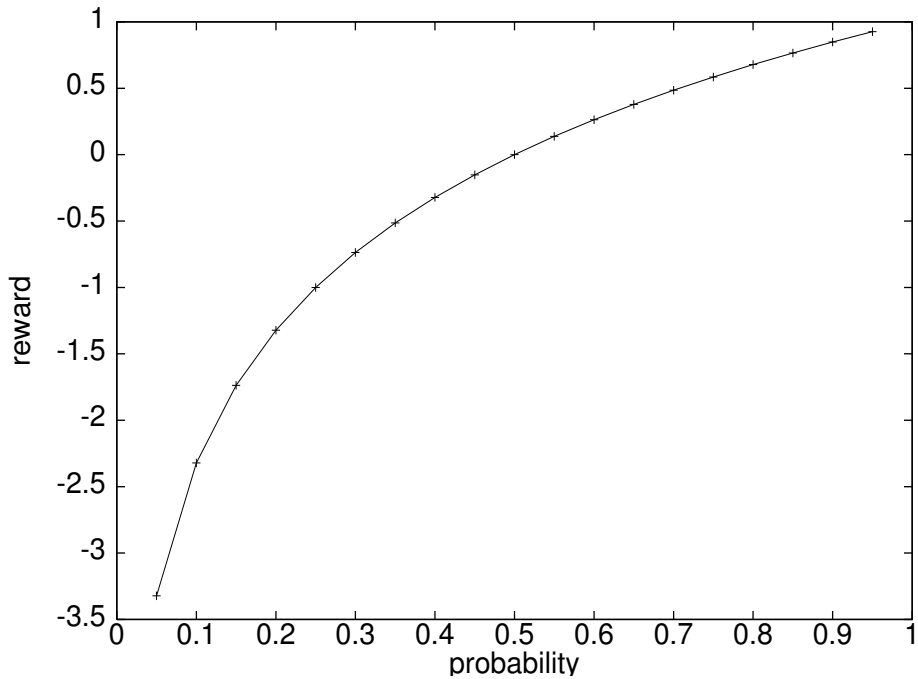
Bayesian information reward (Good, 1952; Korb & Nicholson, 2011) is a log scoring rule that provides a probability-weighted reward for every test instance which simultaneously

- Rewards classification accuracy
- And calibration
- *Maximally* rewards the true probability distribution (i.e., it's strictly proper)
- Again, this is a scoring rule, not a model measure

Good's Binomial Information Reward

$$IR_G = \sum_i [1 + \log_2 P(c_i)]$$

where c_i is the actual class.



$-\log_2 P(c_i)$ is the bit length of a message reporting the true class assuming the reported probability. Note the penalty for falsely asserting probability zero!

TBD_2

Bayesian Information Reward

BIR generalizes this by:

- Generalizing to multinomial classes $\{C = c_i\}, P(c_i) = \hat{p}_i$
- Relativizing reward for $\hat{p}_i$ to the prior probability p_i

New Discretization Method 2

Search of discretization space optimizing:

$$BIR = \frac{1}{n}\left(\log\frac{\hat{p}_i}{p_i} + \sum_j \log\frac{1-\hat{p}_j}{p_j}\right)$$

where n is arity, i indexes true classes and j indexes false classes.

TBD_2

Bayesian Information Reward

In our study we actually use:

$$BIR = \frac{1}{m} \sum_{k=1,m} \left[\frac{1}{n} \left(\log \frac{\hat{p}_i}{p_i} + \sum_j \log \frac{1 - \hat{p}_j}{p_j} \right) \right]$$

where m is the test set size.

GA-Slicer

- GA search for the optimal multivariate discretization
  - Weka plugin
- Classifiers: J48 (C4.5), NB, AODE
  - Seeded with random discretizations (1-3 cutpoints)
- Reproduction:
  Crossover (0.25) XOR clone & mutate (0.75)
- Optimizing: PA, AUC, Cost, BIR

GA-Slicer

Results

- BIR approximately the same as Entropy-MDL; a slight win on the AUC measure

Current work for fog prediction with the Bureau of Meteorology

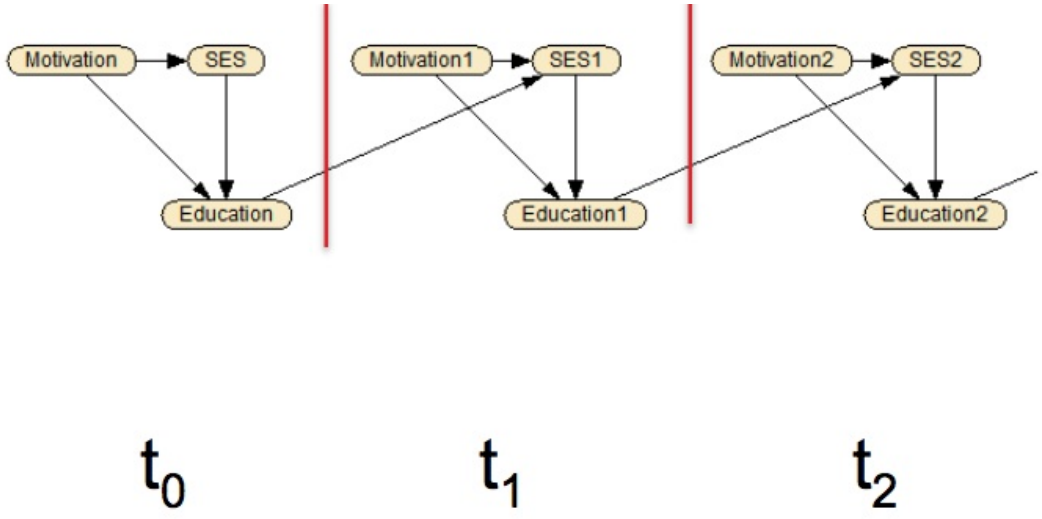
- Extend independently multiple variable discretization to
- Joint multiple variable discretization, requiring function discovery over the joint variable space

Learning Dynamic Bayesian Networks

DBN

Dynamic Bayesian networks are static, duplicated BNs linked by temporal arcs across one time step.

- Same structure for each slice (stationary)
- Arcs cannot span multiples steps (order 1)
- Longer term: look at non-stationary DBNs



LDBN

$$DBN \subset BN$$

So, why not use existing static learners?

- We have and it works. However, you have to restrict LDBN:
  - t nodes precede $t+1$ nodes
  - static nets are identical
- Prior constraints can be used, e.g., tiers (done; equiv performance to Friedman, Murphy and Russell, 1998)
- Restricting the search space is more efficient (current work)

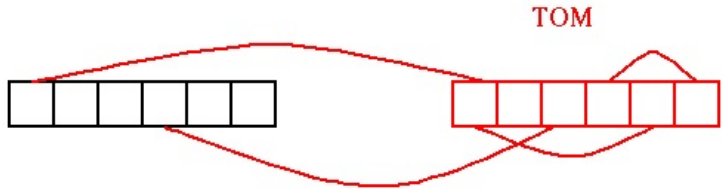
MML for DBNs

Assumptions:

- t_0 static structure is the same as t_i; temporal arcs are also the same
  - Static structure needs to be coded once
  - Temporal arcs need to be coded once
  - Weights between the two need not be identical (open issue)

MML for DBNs

DTOM representation:



$$MML(DTOM|e) = I_{MML}(DTOM) + I_{MML}(\theta|DTOM) I_{MML}(e|\theta, DTOM)$$

$$I_{MML}(DTOM) = I_{MML}(TOM) + I_{MML}(t-arcs|TOM)$$

$$I_{MML}(t-arcs|TOM) = (-\log p_t \times A_t) + (-\log(1-p_t) \times (N^2 - A_t))$$

MML for DBNs

Parameter and data cost:

- Codes are unchanged
- As a simplification, we don't cost the first time slice data (using t_0), but cost the remained using t_1, conditioned on appropriate parental values from t_0.

Results

eMilk

		DBN		Tier Prior		CaMML (No Priors)	
		Mean	Stdev	Mean	Stdev	Mean	Stdev
100	ED	5.58	1.57	6.37	2.00	7.51	2.09
	KL	1.22	0.25	1.24	0.23	1.25	0.25
	CKL	1.50	0.68	2.34	0.72	2.38	0.86
500	ED	2.44	0.72	2.55	0.90	3.26	1.69
	KL	0.52	0.08	0.52	0.07	0.51	0.07
	CKL	0.53	0.08	1.45	0.58	0.96	0.50
1000	ED	1.74	0.85	1.82	0.86	2.27	1.43
	KL	0.44	0.05	0.43	0.05	0.43	0.05
	CKL	0.44	0.06	0.93	0.50	0.63	0.18
2000	ED	0.42	0.57	0.52	0.61	0.60	0.83
	KL	0.36	0.04	0.37	0.03	0.37	0.03
	CKL	0.36	0.04	0.68	0.40	0.51	0.18

Results

eMetastatic

	ED	9.75	1.13	10.61	0.62	11.20	0.90
100	KL	1.45	0.43	1.47	0.41	1.46	0.41
	CKL	1.49	0.42	1.49	0.39	1.62	0.42
	ED	5.76	1.44	6.62	1.70	7.32	1.83
500	KL	0.84	0.18	0.85	0.19	0.85	0.19
	CKL	0.94	0.23	1.02	0.25	1.11	0.28
	ED	4.09	1.37	4.87	1.47	5.24	1.77
1000	KL	0.74	0.10	0.75	0.11	0.76	0.11
	CKL	0.83	0.16	0.89	0.17	0.94	0.20
	ED	2.65	1.07	3.37	1.28	3.80	1.41
2000	KL	0.67	0.06	0.68	0.07	0.68	0.07
	CKL	0.74	0.11	0.73	0.10	0.80	0.14
	ED	1.03	0.81	1.51	1.12	2.05	1.59
5000	KL	0.64	0.04	0.64	0.04	0.64	0.04
	CKL	0.69	0.09	0.66	0.05	0.72	0.11

Knowledge Engineering
Bayesian Networks

KEBN

We emphasis rapid prototyping (Boehm's cyclical model), integrating both expert elicitation and machine learning. Main steps:

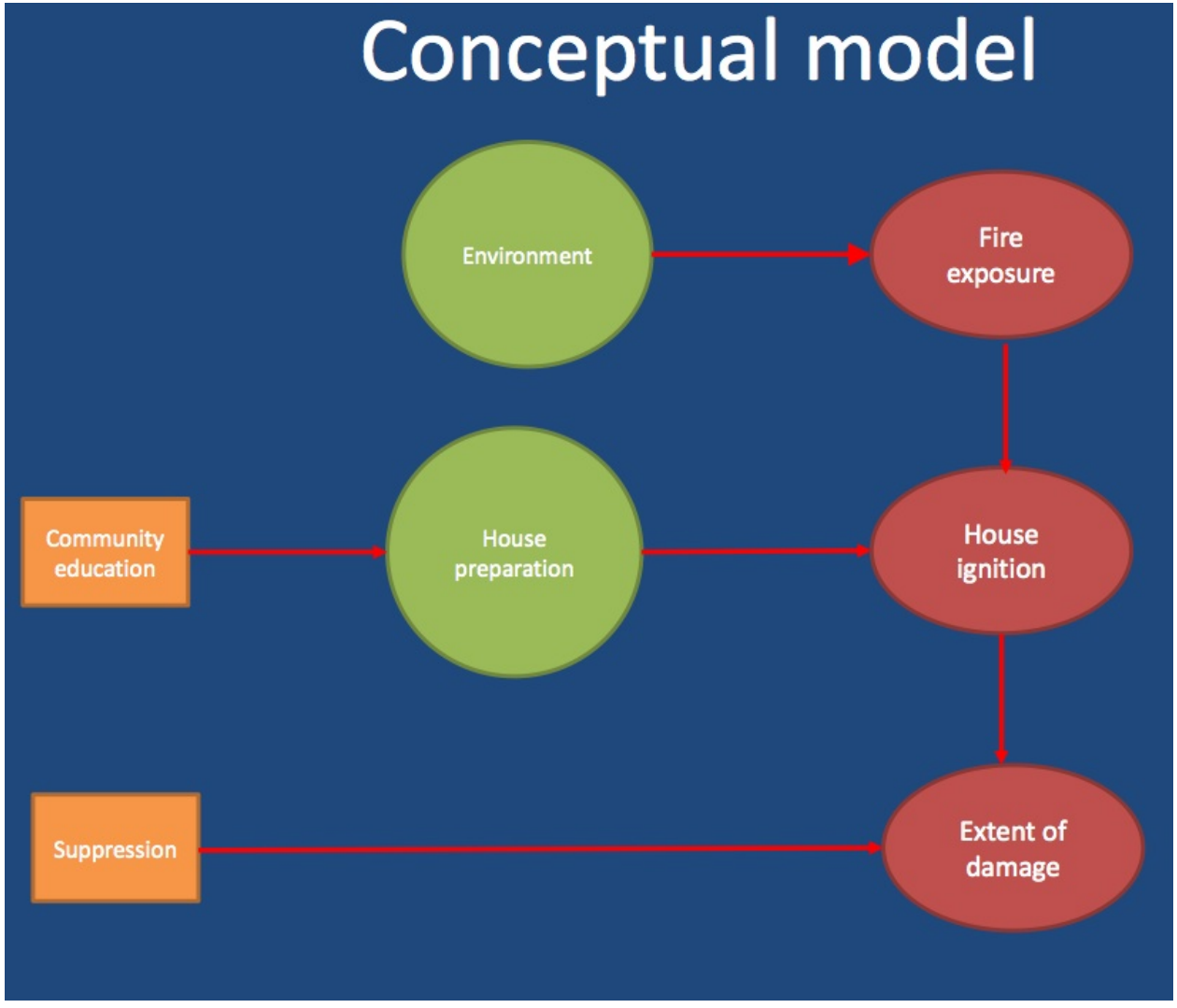
- Expert elicitation
  - Structure
  - Parameters
- Machine learning with priors
- Evaluation
  - Sensitivity analysis
  - Scenario testing
  - Statistical testing

BN Applications

Fire Risk Management: NSW Rural Fire Service

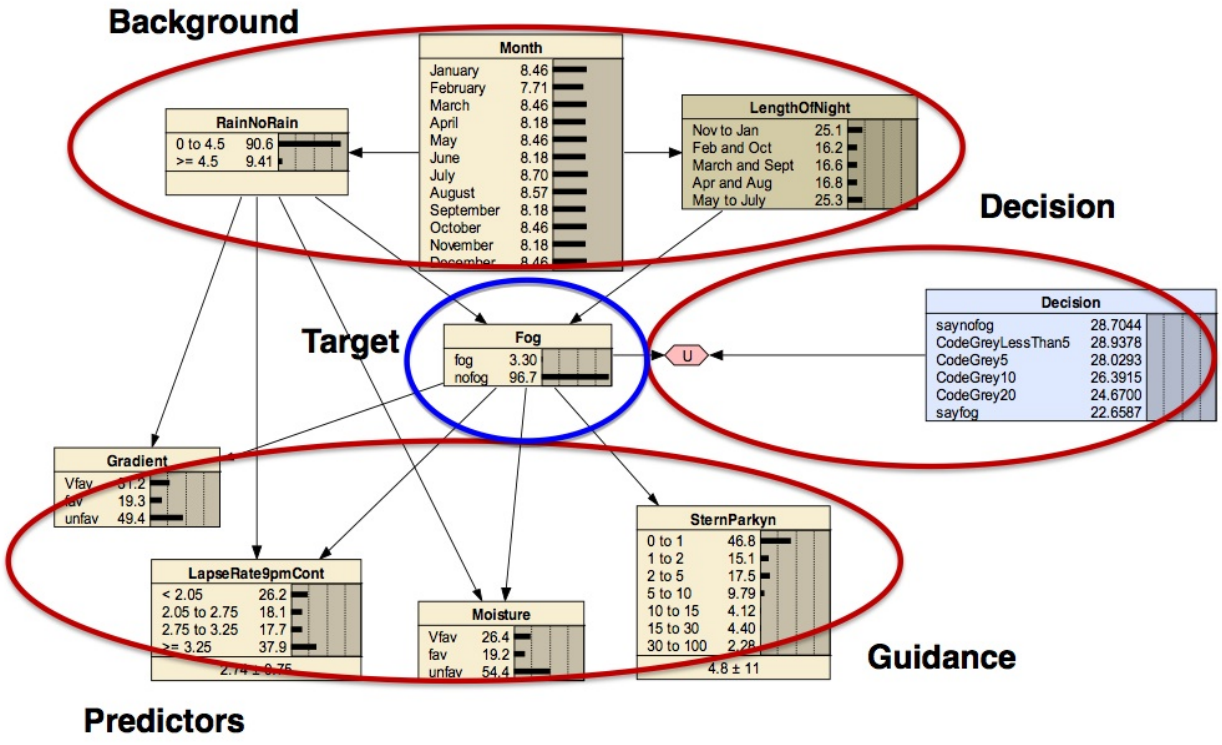
Penman & Nicholson

GIS Bayesian nets to assess and inform about fire risk:



Fog Forecasting

Bureau of Meteorology & Monash



Current efforts:

- Improve timing of predictions using DBN
- Improve predictions via improved multivariate discretization

Forestry Management

New Zealand Forest Research Institute & BI

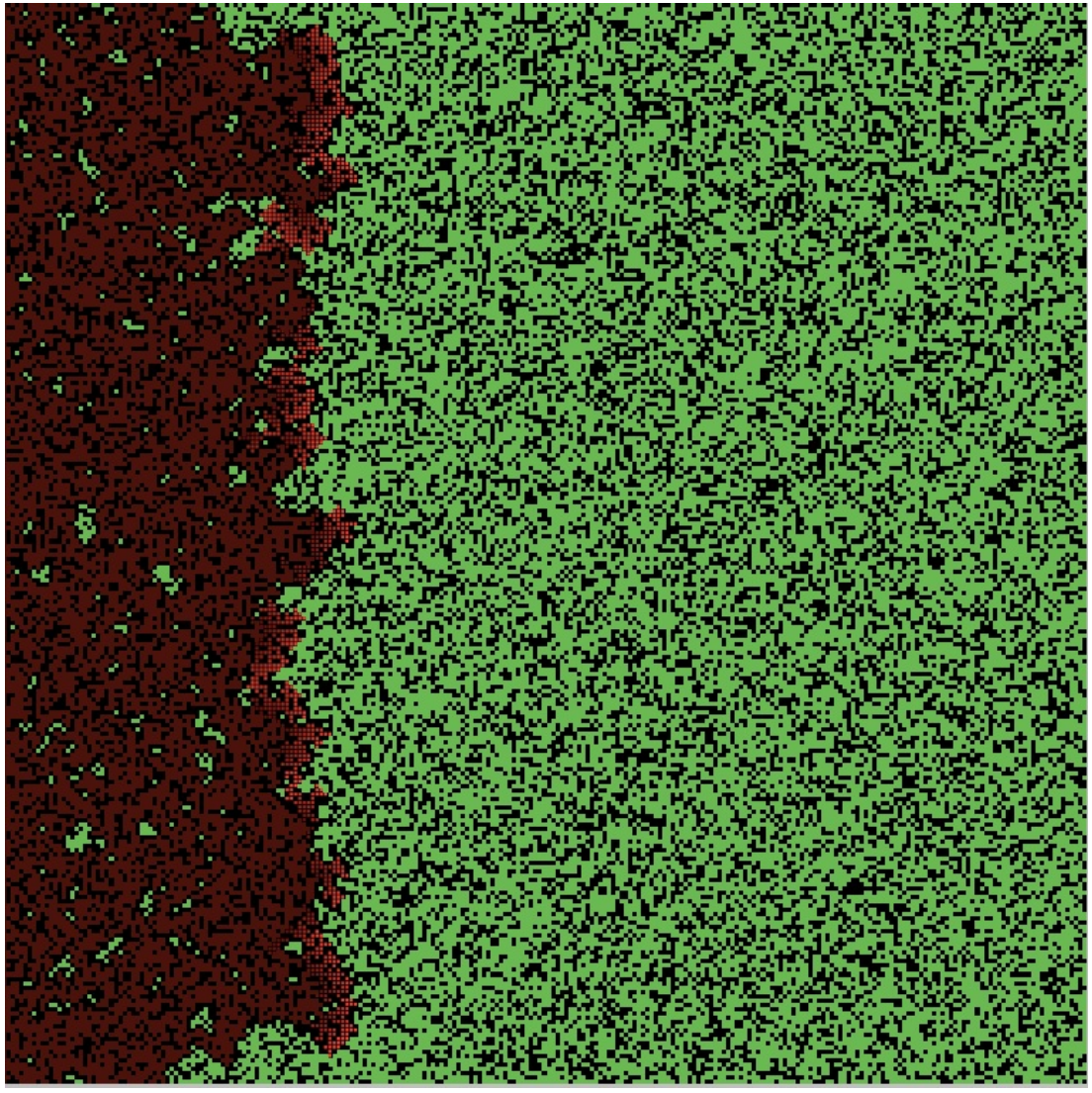
BNs to better predict/justify pesticide treatments of wood exports.

Other Research

Agent-Based Models

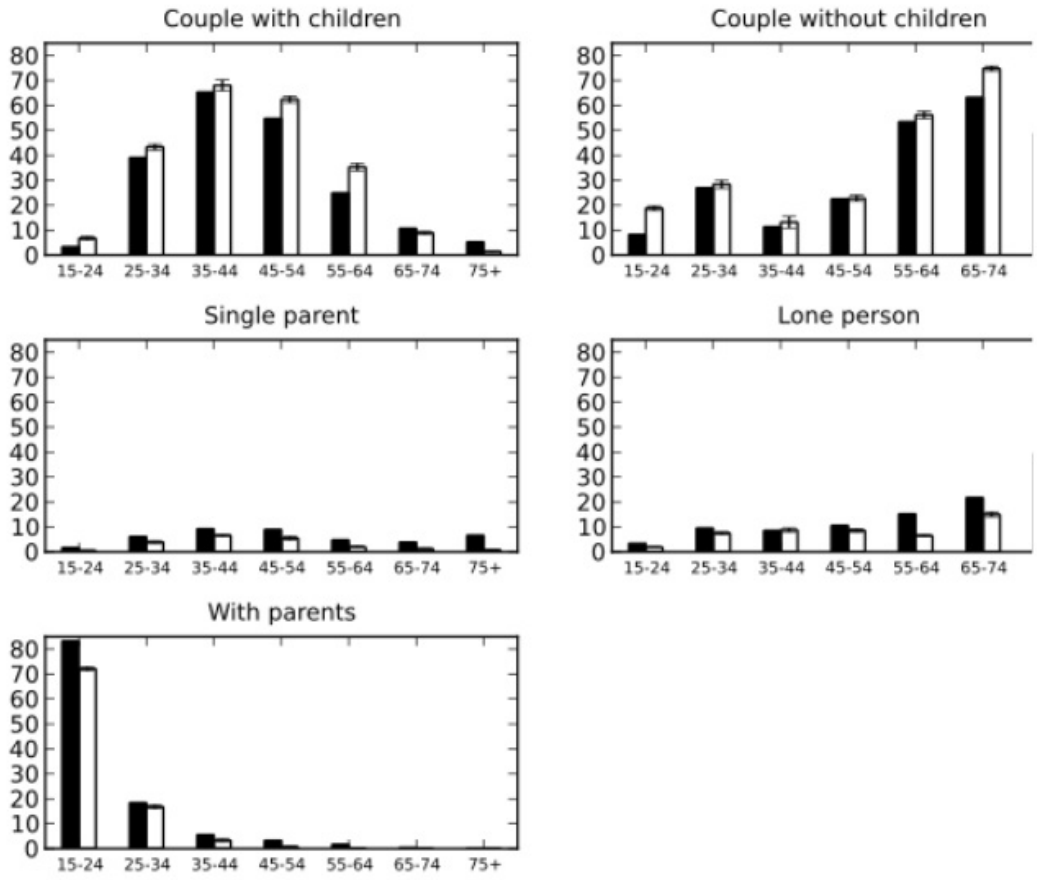
Agent-Based Models

Netlogo Fire Model



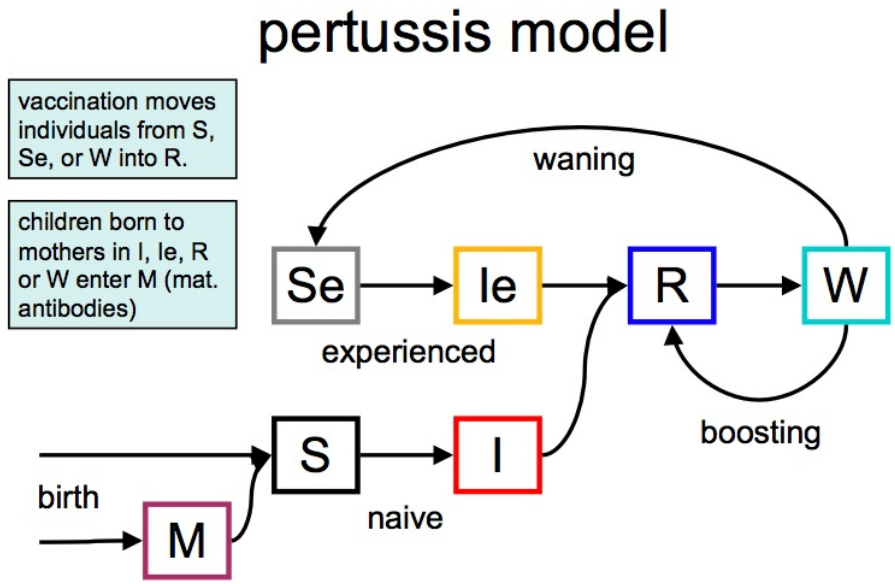
Agent-Based Models

Demographics



Agent-Based Models

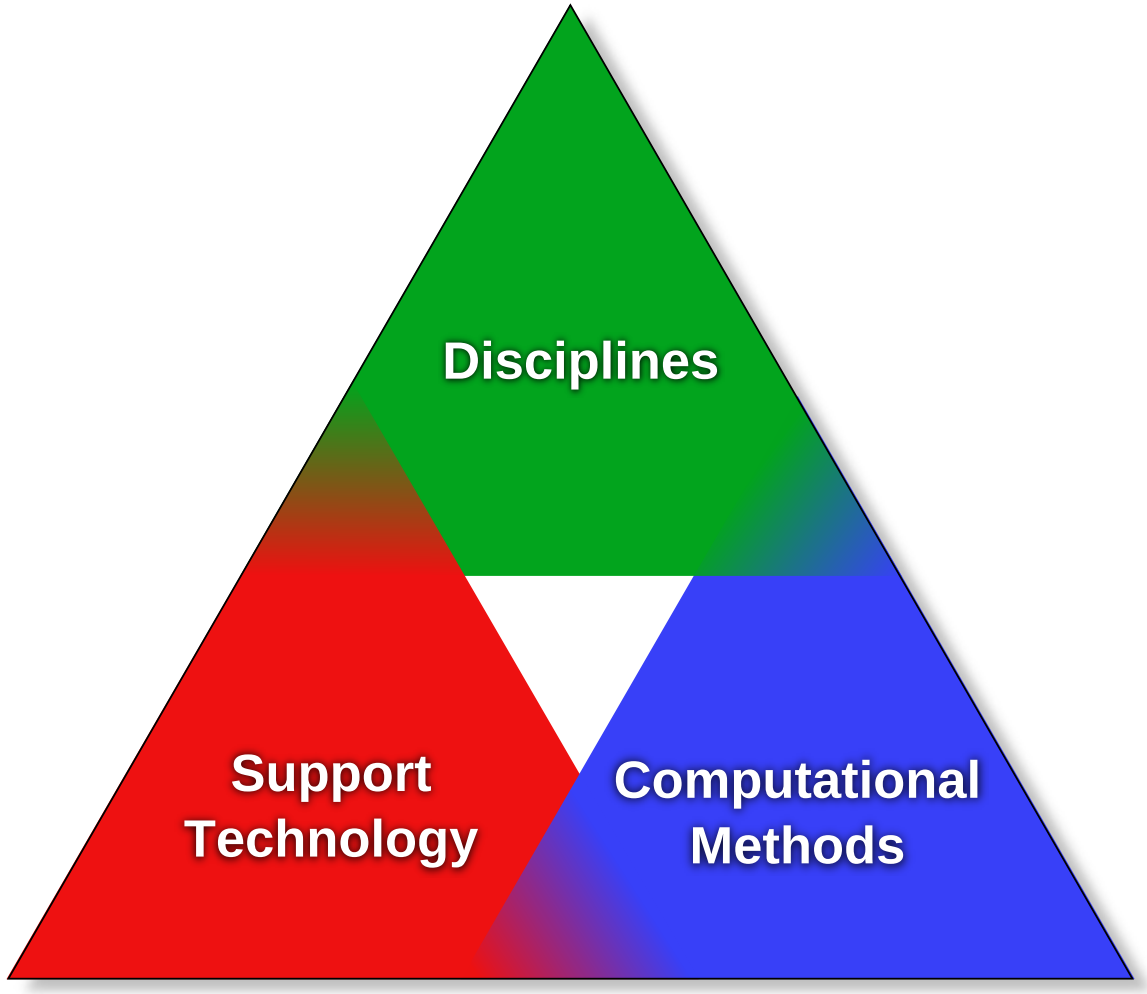
- Epidemiology, waning immunity model (w School of Public Health, Uni Melb):



- Evolutionary Models (Alan Dorin, Jon McCormack, David Green)
- Stochastic Optimization (Bernd Meyer)
- Biocomplexity (Dorin, Lloyd Allison)

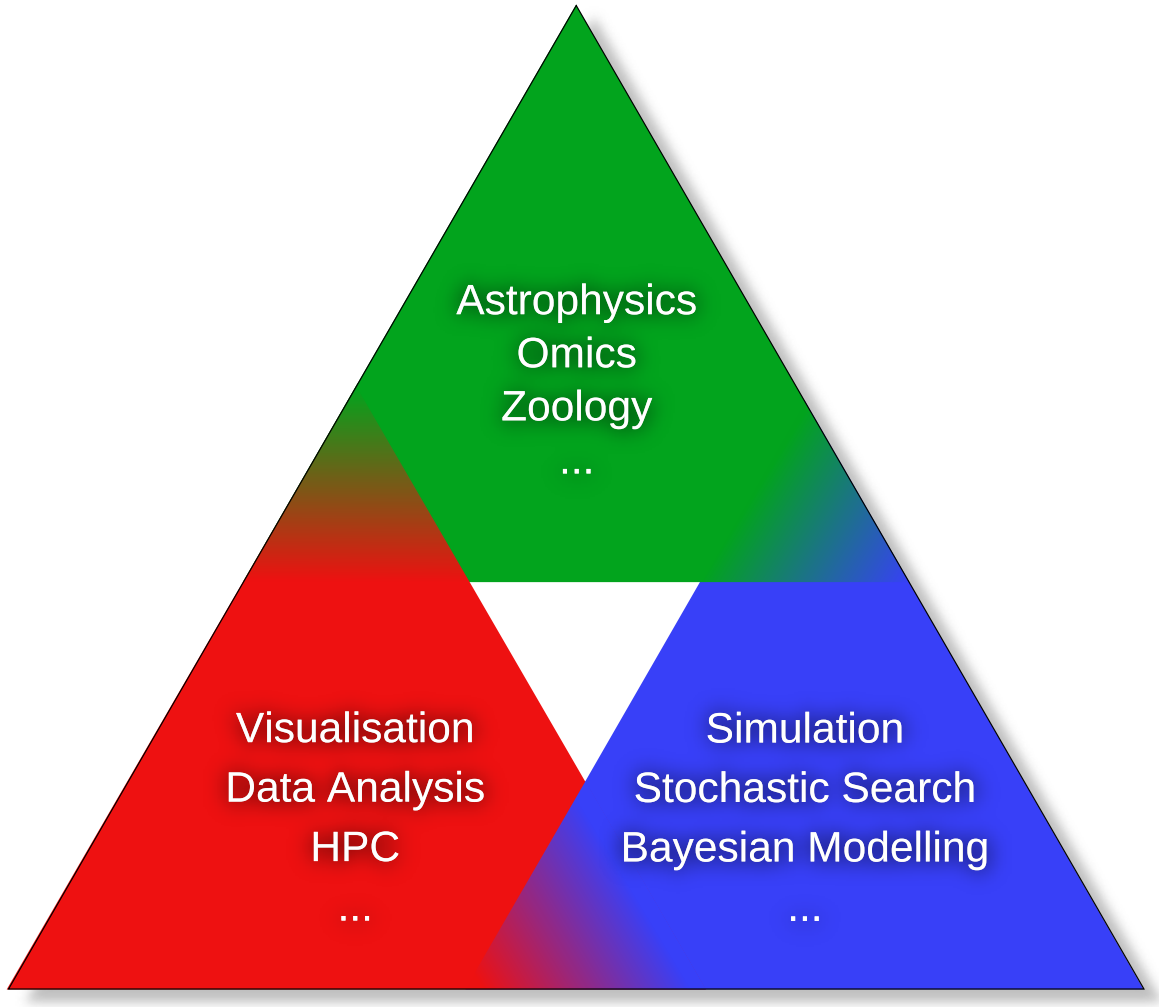
New Centre for Computational Science

Monash FIT



New Centre for Computational Science

Monash FIT



New Centre for Computational Science

Monash FIT

Existing strengths:

- Optimization (Kim Marriott, Maria de la Banda, Guido Tack, Aldeida Aleti)
- Visualization (Dorin, Marriott, Michael Wybrow)
- Bayesian network technology (Nicholson, Albrecht)
- Machine learning, data analysis (Webb, Konagurthu, Albrecht, Martinez, Carmen, Haffari, . . .)

Looking for research partnerships

Centre for Research in Intelligence Systems

Monash FIT

- Top AI group in Australia (as measured by research grants)
- 31 academics
- Strengths: machine learning, computational statistics, Bayesian networks, classification, computational biology, NLP

References I

G.F. Cooper and E. Herskovits (1991) "A Bayesian Method for Constructing Bayesian Belief Networks from Databases," in D'Ambrosio, Smets and Bonissone (eds.) *UAI 1991,* 86-94.

N. Friedman (1997) "The Bayesian Structural EM Algorithm," in D. Geiger and P.P. Shenoy (eds.) *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence* (pp. 129-138). San Francisco: Morgan Kaufmann.

N. Friedman, K. Murphy, and S. Russell (1998). Learning the structure of dynamic probabilistic networks. In *Uncertainty in Artificial Intelligence 14.*

C. Glymour, P. Spirtes, R. Scheines, K. Kelly (1987). *Discovering Causal Structure.* Academic.

D. Heckerman and D. Geiger (1995) "Learning Bayesian networks: A unification for discrete and Gaussian domains," in Besnard and Hanks (eds.) *UAI 1995,* 274-284.

Kevin B. Korb and Ann E. Nicholson *Bayesian Artificial Intelligence, 2nd edition*, Chapman & Hall/CRC, 2010.

K Korb & E Nyberg (2006). The power of intervention. *Minds and Machines, 16,* 289-302.

W. Lam and F. Bacchus (1993) "Learning Bayesian belief networks: An approach based on the MDL principle," *Jrn Comp Intelligence, 10,* 269-293.

References II

N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller and E. Teller (1953) "Equations of state calculations by fast computing machines," *Jrn Chemical Physics, 21,* 1087-1091.

J.R. Neil and K.B. Korb (1999) "The Evolution of Causal Models: A Comparison of Bayesian Metrics and Structure Priors," in N. Zhong and L. Zhous (eds.) *Methodologies for Knowledge Discovery and Data Mining: Third Pacific-Asia Conference* (pp. 432-437). Springer Verlag.

R O'Donnell, A E Nicholson, B Han, K B Korb, M J Alam and L Hope (2006). Causal discovery with prior information, Lecture Notes in Artificial Intelligence, Springer-Verlag, Vol 4304, pp. 1162-1167.

R O'Donnell, L Allison and K Korb (2005). Learning hybrid Bayesian networks by MML, Lecture Notes in Computer Science: AI 2006 - Advances in Artificial Intelligence, Springer-Verlag, Vol: 4304, pp. 192-203.

P. Spirtes, C. Glymour and R. Scheines (1993) *Causation, Prediction and Search.* Springer.

T.S. Verma and J. Pearl (1991) "Equivalence and Synthesis of Causal Models," in P. Bonissone, M. Henrion, L. Kanal and J.F. Lemmer (eds) *Uncertainty in Artificial Intelligence 6* (pp. 255-268). Elsevier.

C.S. Wallace and D. Boulton (1968) "An information measure for classification," *Computer Jrn, 11,* 185-194.

References III

C. S. Wallace and K. B. Korb (1999) "Learning Linear Causal Models by MML Sampling," in A. Gammerman (ed.) *Causal Models and Intelligent Data Management.* Springer Verlag.

C. S. Wallace, K. B. Korb, and H. Dai (1996) "Causal Discovery via MML," in L. Saitta (ed.) *Proceedings of the Thirteenth International Conference on Machine Learning* (pp. 516-524). San Francisco: Morgan Kaufmann.