Graphical Models

Model Estimation and Validation

Marco Scutari

m.scutari@ucl.ac.uk
Genetics Institute
University College London

September 27, 2011

Graphical Models

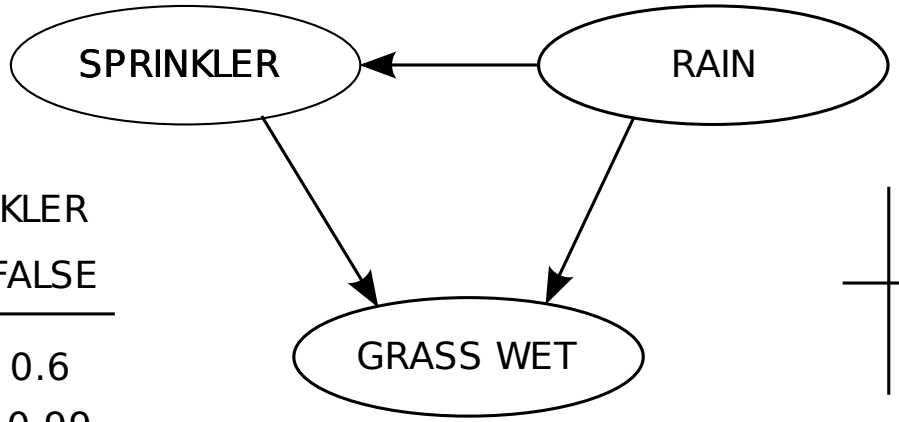
Graphical Models

Graphical models are defined by:

- a network structure, $\mathcal{G} = (\mathbf{V}, E)$, either an undirected graph (Markov networks, gene association networks, correlation networks, etc.) or a directed graph (Bayesian networks). Each node $v_i \in \mathbf{V}$ corresponds to a random variable X_i;
- a global probability distribution, $\mathbf{X}$, which can be factorised into a small set of local probability distributions according to the edges $e_{ij} \in E$ present in the graph.

This combination allows a compact representation of the joint distribution of large numbers of random variables and simplifies inference on the resulting parameter space.

A Simple Bayesian Network: Watson's Lawn



RAIN	SPRINKLER	
	TRUE	FALSE
FALSE	0.4	0.6
TRUE	0.01	0.99

RAIN	
TRUE	FALSE
0.2	0.8

SPRINKLER	RAIN	GRASS WET	
		TRUE	FALSE
FALSE	FALSE	0.0	1.0
FALSE	TRUE	0.8	0.2
TRUE	FALSE	0.9	0.1
TRUE	TRUE	0.99	0.01

Graphical Separation and Independence

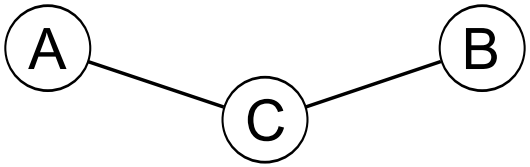
The main role of the graph structure is to express the conditional independence relationships among the variables in the model, thus specifying the factorisation of the global distribution. Different classes of graphs express these relationships with different semantics, which have in common the principle that graphical separation of two (sets of) nodes implies the conditional independence of the corresponding (sets of) random variables.

For networks considered here, separation is defined as:

- (u-)separation in Markov networks;
- d-separation in Bayesian networks.

Graphical Separation

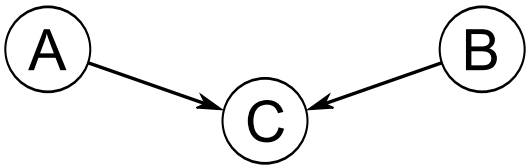
separation (undirected graphs)



$$\mathbf{A} \perp\!\!\!\perp \mathbf{B} \,|\, \mathbf{C}$$

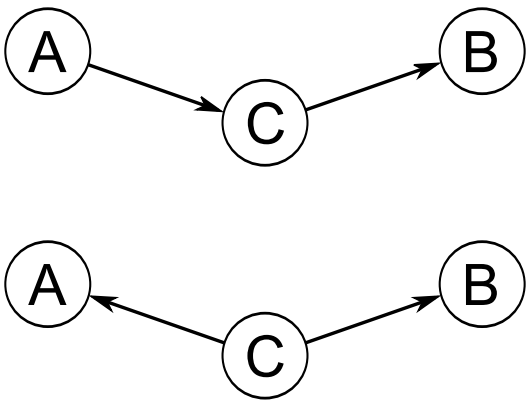
$$\mathrm{P}(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \mathrm{P}(\mathbf{A} \,|\, \mathbf{C})\, \mathrm{P}(\mathbf{B} \,|\, \mathbf{C})\, \mathrm{P}(\mathbf{C})$$

d-separation (directed acyclic graphs)



$$\mathbf{A} \not\perp\!\!\!\perp \mathbf{B} \,|\, \mathbf{C}$$

$$\mathrm{P}(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \mathrm{P}(\mathbf{C} \,|\, \mathbf{A}, \mathbf{B})\, \mathrm{P}(\mathbf{A})\, \mathrm{P}(\mathbf{B})$$



$$\mathbf{A} \perp\!\!\!\perp \mathbf{B} \,|\, \mathbf{C}$$

$$\begin{aligned} \mathrm{P}(\mathbf{A}, \mathbf{B}, \mathbf{C}) &= \\ &= \mathrm{P}(\mathbf{B} \,|\, \mathbf{C})\, \mathrm{P}(\mathbf{C} \,|\, \mathbf{A})\, \mathrm{P}(\mathbf{A}) \\ &= \mathrm{P}(\mathbf{A} \,|\, \mathbf{C})\, \mathrm{P}(\mathbf{B} \,|\, \mathbf{C})\, \mathrm{P}(\mathbf{C}) \end{aligned}$$

Maps and Independence

A graph $\mathcal{G}$ is a dependency map (or D-map) of the probabilistic dependence structure P of $\mathbf{X}$ if there is a one-to-one correspondence between the random variables in $\mathbf{X}$ and the nodes $\mathbf{V}$ of $\mathcal{G}$, such that for all disjoint subsets $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ of $\mathbf{X}$ we have

$$\mathbf{A} \perp\!\!\!\perp_P \mathbf{B} \,|\, \mathbf{C} \Longrightarrow \mathbf{A} \perp\!\!\!\perp_G \mathbf{B} \,|\, \mathbf{C}.$$

Similarly, $\mathcal{G}$ is an independency map (or I-map) of P if

$$\mathbf{A} \perp\!\!\!\perp_P \mathbf{B} \,|\, \mathbf{C} \Longleftarrow \mathbf{A} \perp\!\!\!\perp_G \mathbf{B} \,|\, \mathbf{C}.$$

$\mathcal{G}$ is said to be a perfect map of P if it is both a D-map and an I-map, that is

$$\mathbf{A} \perp\!\!\!\perp_P \mathbf{B} \,|\, \mathbf{C} \Longleftrightarrow \mathbf{A} \perp\!\!\!\perp_G \mathbf{B} \,|\, \mathbf{C},$$

and in this case P is said to be isomorphic to $\mathcal{G}$.

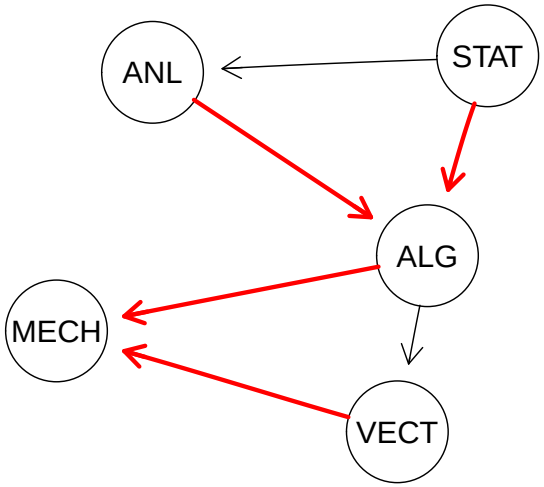
Graphical models are formally defined as I-maps under the respective definitions of graphical separation.

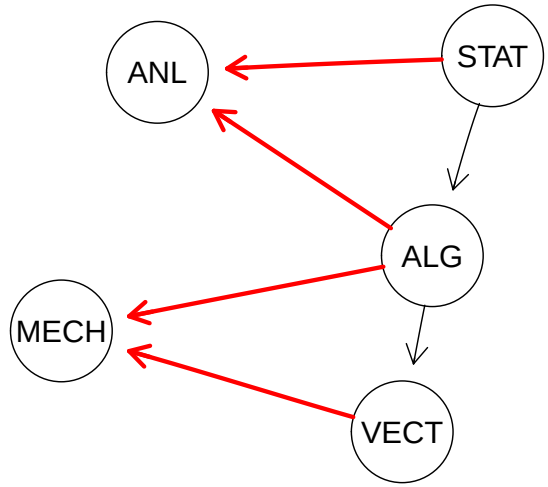
Bayesian Networks, Equivalence Classes and Moral Graphs

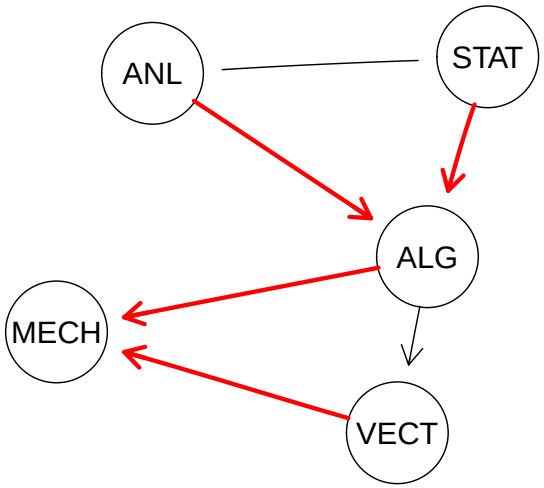
Following the definitions given in the previous couple of slides, the graph associated with a Bayesian network has three useful transforms:

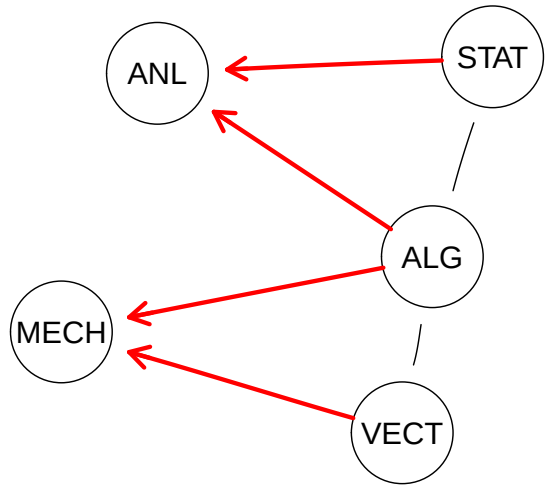
- the skeleton: the undirected graph underlying a Bayesian network, i.e. the graph we get if we disregard edges' direction.
- the equivalence class: the graph in which only edges which are part of a v-structure (i.e. $A \rightarrow C \leftarrow B$) and/or might result in one are directed. All valid combinations of the other edges' directions result in networks representing the same dependence structure P.
- the moral graph: the graph obtained by disregarding edges' direction and joining the two parents in each v-structure with an edge. This is essentially a way to transform a Bayesian network into a Markov network.

Equivalence Classes









Factorisation into Local Distributions

The most important consequence of defining graphical models as I-maps is the factorisation of the global distribution into local distributions:

- in Markov networks, local distributions are associated with the cliques C_i (maximal subsets of nodes in which each element is adjacent to all the others) in the graph,

$$\mathrm{P}(\mathbf{X}) = \prod_{i=1}^{k} \psi_i(\mathbf{C}_i),$$

  and the ψ_k functions are called potentials.

- in Bayesian networks, each local distribution is associated with a single node X_i and depends only on the joint distribution of its parents Π_{X_i}:

$$\mathrm{P}(\mathbf{X}) = \prod_{i=1}^{p} \mathrm{P}(X_i \,|\, \Pi_{X_i})$$

A Note About Potentials

Potentials are non-negative functions representing the relative mass of probability of each clique C_i. They are proper probability or density functions only when the graph is decomposable or triangulated, that is when it contains no induced cycles other than triangles. With any other type of graph inference becomes very hard, if possible at all, because $\psi_1, \psi_2, \ldots, \psi_k$ have no direct statistical interpretation.

In this case the global distribution factorises again according to the chain rule and can be written as

$$\mathrm{P}(\mathbf{X}) = \frac{\prod_{i=1}^{k} \mathrm{P}(\mathbf{C}_i)}{\prod_{i=1}^{k} \mathrm{P}(\mathbf{S}_i)} \tag{1}$$

where $\mathbf{S}_i$ are the nodes of $\mathbf{C}_i$ which are also part of any other clique up to $\mathbf{C}_{i-1}$.

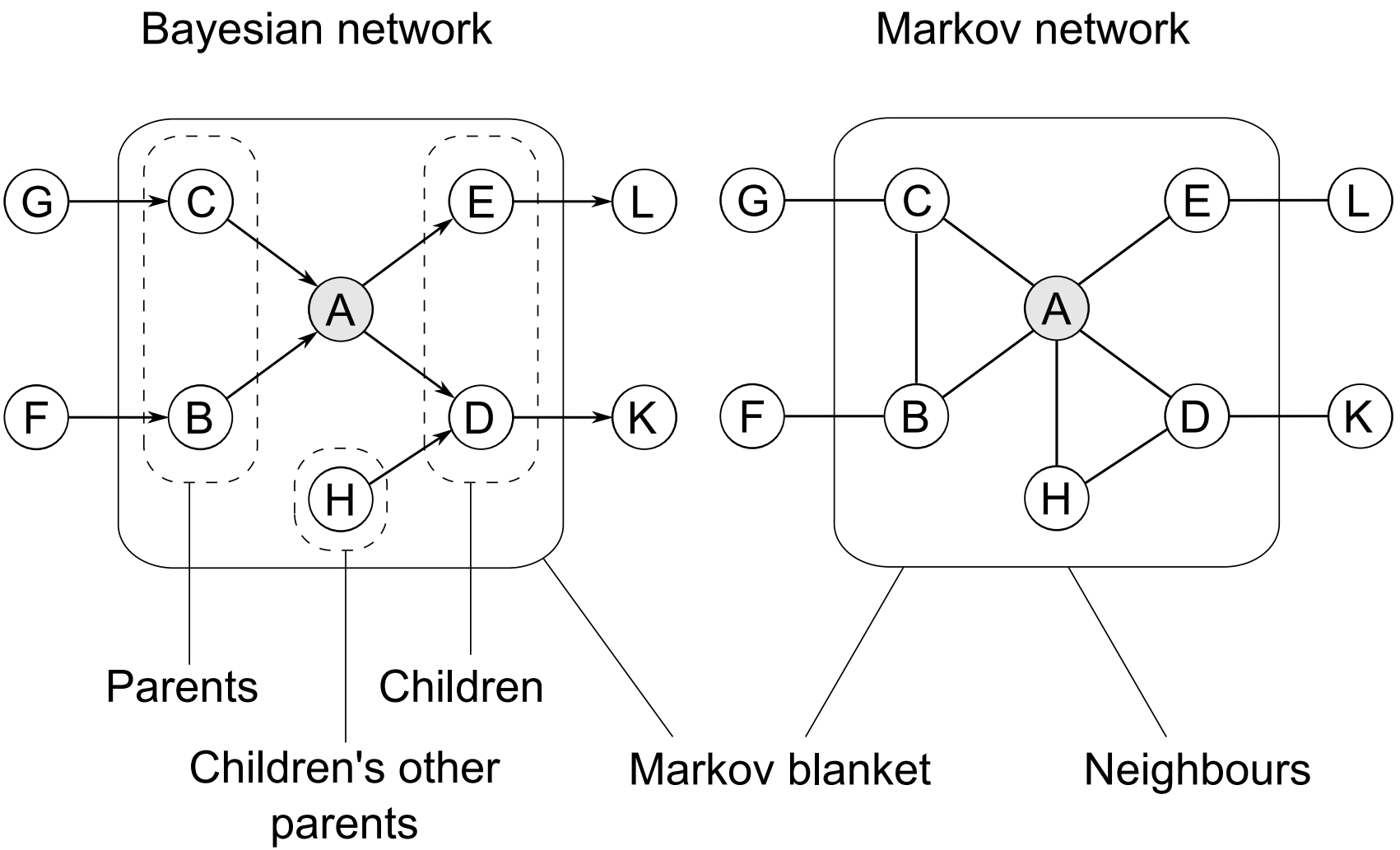
Neighbourhoods and Markov Blankets

Furthermore, for each node X_i two sets are defined:

- the neighbourhood, the set of nodes that are adjacent to X_i. These nodes cannot be made independent from X_i.
- the Markov blanket, the set of nodes that completely separates X_i from the rest of the graph. Generally speaking, it is the set of nodes that includes all the knowledge needed to do inference on X_i, from estimation to hypothesis testing to prediction, because all the other nodes are conditionally independent from X_i given its Markov blanket.

These sets are related in Markov and Bayesian networks; in particular, Markov blankets can be shown to be the same using a moral graph.

Neighbourhoods and Markov Blankets



Markov networks vs Bayesian networks

Markov networks and Bayesian networks do not appear to be closely related, as they are so different in construction and interpretation.

- There are indeed dependency models that have an undirected perfect map but not a directed acyclic one, and vice versa.
- However, it can be shown that every dependency structure that can be expressed by a decomposable graph can be modelled both by a Markov network and a Bayesian network.
- It can also be shown that every dependency model expressible by an undirected graph is also expressible by a directed acyclic graph, with the addition of some auxiliary nodes.

These two results indicate that there is a significant overlap between Markov and Bayesian networks, and that in many cases both can be used to the same effect.

Probability Distributions: Discrete and Continuous

Data used in graphical modelling should respect the following assumptions:

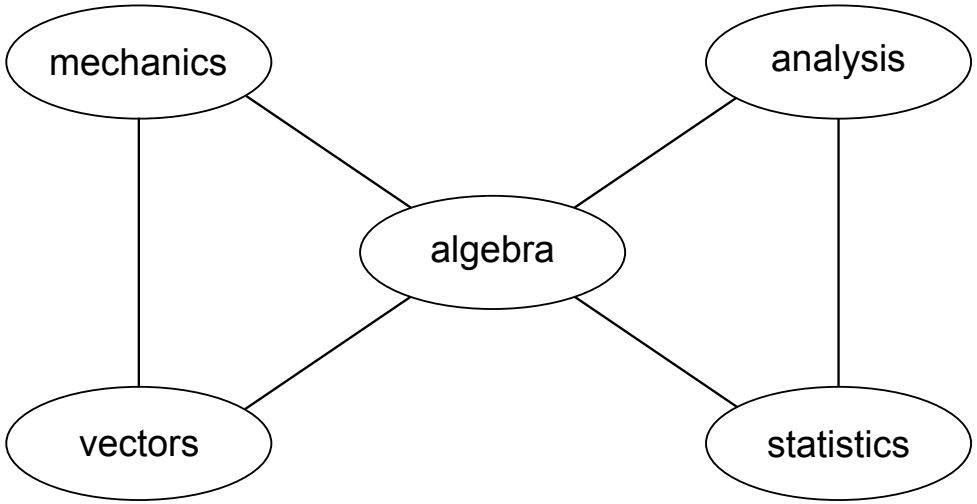
- if all the variables X_i are discrete, both the global and the local distributions are assumed to be multinomial. Local distributions are described using conditional probability tables;
- if all the variables X_i are continuous, the global distribution is assumed to be a multivariate Gaussian distribution, and the local distributions are univariate or multivariate Gaussian distributions. Local distributions are described using partial correlation coefficients;
- if both continuous and discrete variables are present, we can assume a mixture or conditional Gaussian distribution, discretise continuous attributes or use a nonparametric approach.

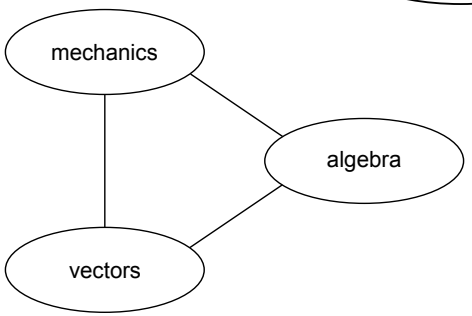
Other Distributional Assumptions

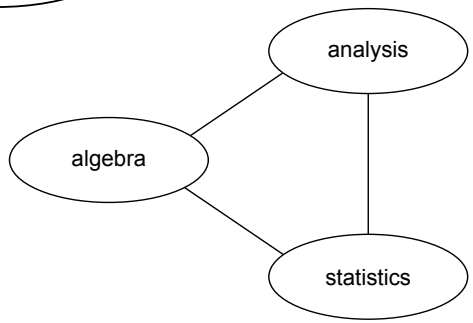
Other fundamental distibutional assumptions are:

- observations must be independent. If some form of temporal or spatial dependence is present, it must be specifically accounted for in the definition of the network (as in *dynamic Bayesian networks*);
- if the model will be used as a causal graphical model, that is, to infer cause-effect relationship from experimental or (more frequently) observational data, there must be no latent or hidden variables that influence the dependence structure of the model;
- all the relationships between the variables in the network must be conditional independencies, because they are by definition the only ones that can be expressed by graphical models.

A Gaussian Markov Network (MARKS)



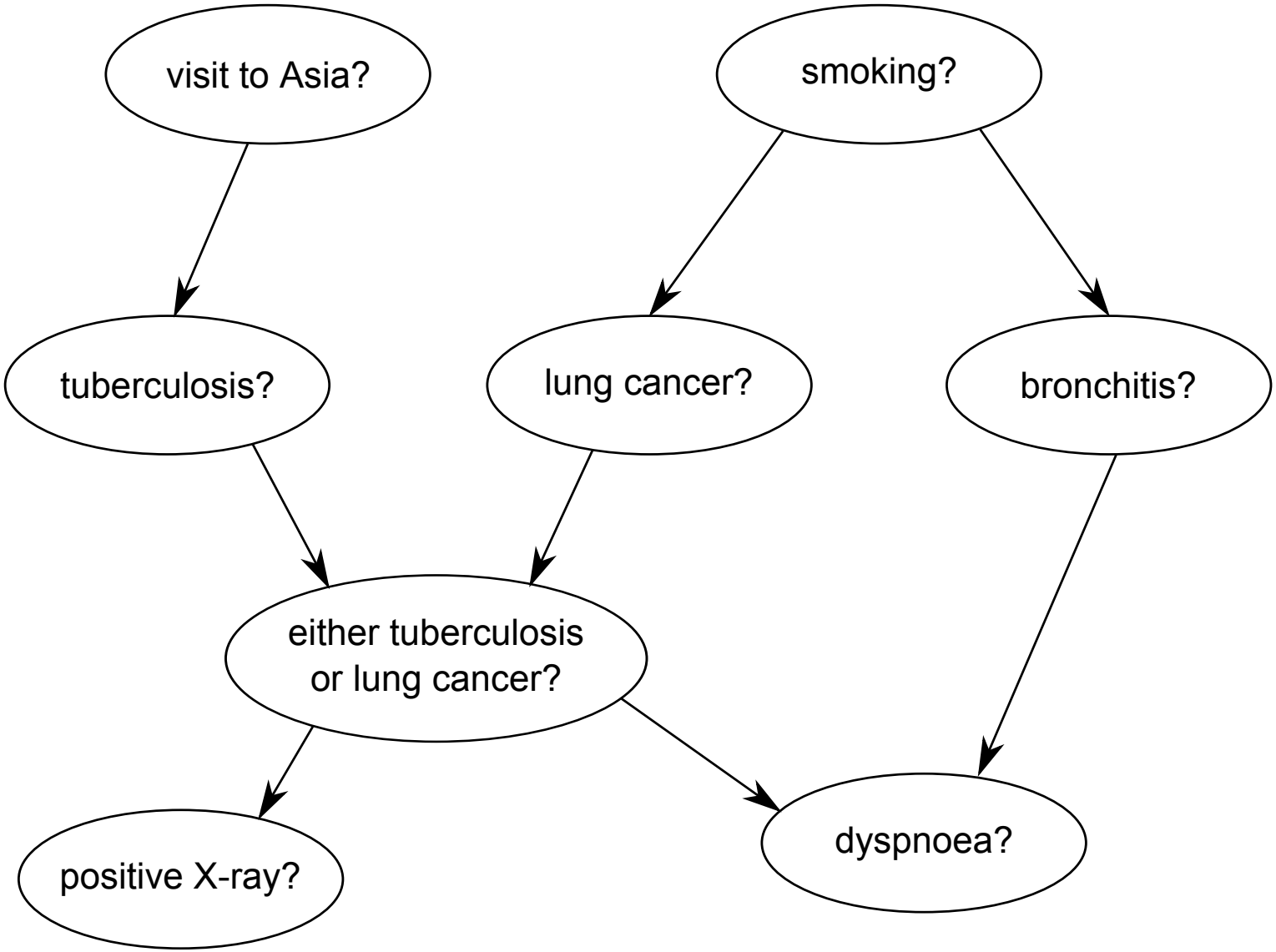




$$\begin{bmatrix} 1.000 & 0.332 & 0.235 \\ 0.332 & 1.000 & 0.327 \\ 0.235 & 0.327 & 1.000 \end{bmatrix}$$

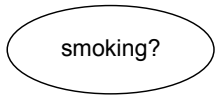
$$\begin{bmatrix} 1.000 & 0.451 & 0.364 \\ 0.451 & 1.000 & 0.256 \\ 0.364 & 0.256 & 1.000 \end{bmatrix}$$

A Discrete Bayesian Network (ASIA)

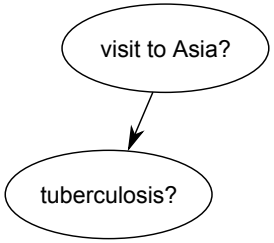


A Discrete Bayesian Network (ASIA)

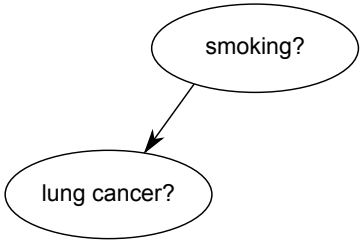
yes	no
0.01	0.99



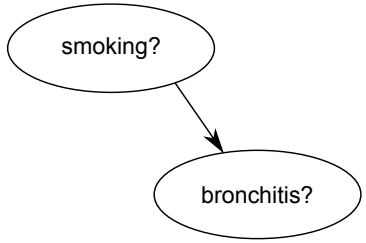
yes	no
0.50	0.50



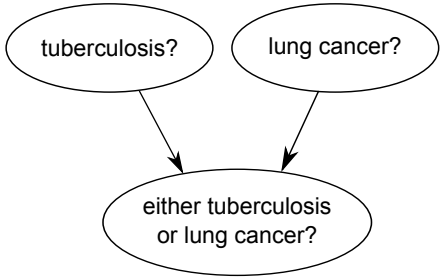
	yes	no
yes	0.05	0.95
no	0.01	0.99



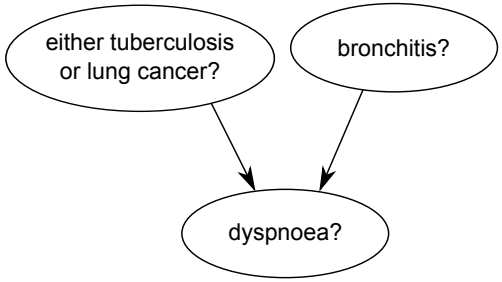
	yes	no
yes	0.10	0.90
no	0.01	0.99



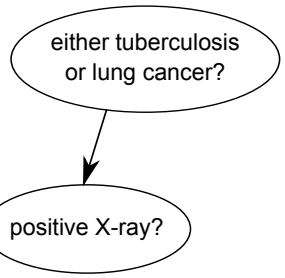
	yes	no
yes	0.60	0.40
no	0.30	0.70



	yes	no
yes:yes	1	0
yes:no	1	0
no:yes	1	0
no:no	0	1



	yes	no
yes:yes	0.90	0.10
yes:no	0.70	0.30
no:yes	0.80	0.20
no:no	0.10	0.90



	yes	no
yes	0.98	0.02
no	0.05	0.95

Limitations of These Probability Distribution

- no real-world, multivariate data set follows a multivariate Gaussian distribution; even if the marginal distributions are normal, not all dependence relationships are linear.
- computing partial correlations is problematic in most large data sets (and in a lot of small ones, too).
- parametric assumptions for mixed data have strong limitations, as they impose constraints on which edges may be present in the graph (e.g. a continuous node cannot be the parent of a discrete node).
- discretisation is a common solution to the above problems, but it discards useful information and it is tricky to get right (i.e. choosing a set of intervals such that the dependence relationships involving the original variable are preserved).
- ordered categorical variables are treated as unordered, again losing information.

Graphical Model Learning

Learning a Graphical Model

Model selection and estimation are collectively known as learning, and are usually performed as a two-step process:

1. structure learning, learning the graph structure from the data.
2. parameter learning, learning the local distributions implied by the graph structure learned in the previous step.

This work-flow is implicitly Bayesian; given a data set $\mathcal{D}$ and if we denote the parameters of the global distribution as $\mathbf{X}$ with Θ, we have

$$\underbrace{\mathrm{P}(\mathcal{M} \mid \mathcal{D})}_{learning} = \underbrace{\mathrm{P}(\mathcal{G} \mid \mathcal{D})}_{structure\,learning} \cdot \underbrace{\mathrm{P}(\Theta \mid \mathcal{G}, \mathcal{D})}_{parameter\,learning}$$

and structure learning is done in practise as

$$\mathrm{P}(\mathcal{G} \mid \mathcal{D}) \propto \mathrm{P}(\mathcal{G})\,\mathrm{P}(\mathcal{D} \mid \mathcal{G}) = \mathrm{P}(\mathcal{G}) \int \mathrm{P}(\mathcal{D} \mid \mathcal{G}, \Theta)\,\mathrm{P}(\Theta \mid \mathcal{G}) d\Theta.$$

Local Distributions: Divide and Conquer

Most tasks related to both learning and inference are NP-hard (they cannot be solved in polynomial time in the number of variables). They are still feasible thanks to the decomposition of $\mathbf{X}$ into the local distributions; under some assumptions (parameter independence) there is never the need to manipulate more than one of them at a time.

In Bayesian networks, for example, structure learning boils down to

$$\begin{aligned} \mathrm{P}(\mathcal{D} \,|\, \mathcal{G}) &= \int \prod \left[\mathrm{P}(X_i \,|\, \Pi_{X_i}, \Theta_{X_i}) \, \mathrm{P}(\Theta_{X_i} \,|\, \Pi_{X_i}) \right] d\Theta \\ &= \prod \left[\int \mathrm{P}(X_i \,|\, \Pi_{X_i}, \Theta_{X_i}) \, \mathrm{P}(\Theta_{X_i} \,|\, \Pi_{X_i}) d\Theta_{X_i} \right] \end{aligned}$$

and parameter learning boils down to

$$\mathrm{P}(\Theta \,|\, \mathcal{G}, \mathcal{D}) = \prod \mathrm{P}(\Theta_{X_i} \,|\, \Pi_{X_i}, \mathcal{D}).$$

Structure Learning

The Big Three: Constraint-based, Score-based and Hybrid

Despite the (sometimes confusing) variety of theoretical backgrounds and terminology they can all be traced to only three approaches:

- constraint-based algorithms: they use statistical tests to learn conditional independence relationships (called *constraints* in this setting) from the data and assume that the graph underlying the probability distribution is a perfect map to determine the correct network structure.
- score-based algorithms: each candidate network is assigned a score reflecting its goodness of fit, which is then taken as an objective function to maximise.
- hybrid algorithms: conditional independence tests are used to learn at least part of the conditional independence relationships from the data, thus restricting the search space for a subsequent score-based search. The latter determines which edges are actually present in the graph and, in the case of Bayesian networks, their direction.

Constraint-based Structure Learning Algorithms

The mapping between edges and conditional independence relationships lies at the core of graphical modelling; therefore, one way to learn the structure of a graphical model is to check which ones of such relationships hold according to a suitable conditional independence test.

Such an approach results in a set of conditional independence constraints that identify a single graph (for a Markov network) or a single equivalence class (for a Bayesian network). In the latter case, the relevant edge directions are determined using more conditional independence tests to identify which v-structures are present in the graph.

The Inductive Causation Algorithm

The Inductive Causation Algorithm

1. For each pair of variables A and B in $\mathbf{X}$ search for set $\mathbf{S}_{AB} \subset \mathbf{X}$ such that A and B are independent given $\mathbf{S}_{AB}$ and $A, B \notin \mathbf{S}_{AB}$. If there is no such a set, place an undirected arc between A and B.
2. For each pair of non-adjacent variables A and B with a common neighbour C, check whether $C \in \mathbf{S}_{AB}$. If this is not true, set the direction of the arcs $A - C$ and $C - B$ to $A \rightarrow C$ and $C \leftarrow B$.
3. Set the direction of arcs which are still undirected by applying recursively the following two rules:
   - 3.1 if A is adjacent to B and there is a strictly directed path from A to B then set the direction of $A - B$ to $A \rightarrow B$;
   - 3.2 if A and B are not adjacent but $A \rightarrow C$ and $C - B$, then change the latter to $C \rightarrow B$.
4. Return the resulting (partially) directed acyclic graph.

Conditional Independence Tests

Classic tests are used because they are fast but are not particularly good.

- asymptotic discrete tests: mutual information/log-likelihood ratio and Pearson's X^2 with a χ^2 distribution.
- asymptotic continuous tests: Fisher's Z, with a $N(0, 1)$ distribution, and mutual information/log-likelihood ratio, with a χ^2 distribution.
- exact continuous tests: t test with a Student's t distribution.

Better alternatives are:

- permutation tests: all of the above, evaluated using the permutation distribution as the null distribution. The resulting structure is better for goodness-of-fit and prediction.
- shrinkage tests: log-likelihood ratio tests can be reworked as shrinkage tests whose behaviour is determined by a regularisation parameter λ. The resulting structure is closer to the "real" one and is therefore better for causal reasoning.

Other Constraint-based algorithms

- Peter & Clark (PC): a true-to-form implementation of the Inductive Causation algorithm, specifying only the order of the conditional independence tests. Starts from a saturated network and performs tests gradually increasing the number of conditioning nodes.
- Grow-Shrink (GS) and Incremental Association (IAMB) variants: these algorithms learn the Markov blanket of each node to reduce the number of tests required by the Inductive Causation algorithm. Markov blankets are learned using different forward and step-wise approaches; the initial network is assumed to be empty (i.e. not to have any edge).
- Max-Min Parents & Children (MMPC): uses a minimax approach to avoid conditional independence tests known *a priori* to accept the null hypothesis of independence.

Pros & Cons of Constraint-based Algorithms

- They depend heavily on the quality of the conditional independence tests they use; all proofs of correctness assume tests are always right. That's why asymptotic tests are bad, and non-regularised parametric tests are not ideal.
- They are consistent, but converge is slower than score-based and hybrid algorithms.
- At any single time they evaluate a small subset of variables, which makes them very memory efficient.
- They do not require multiple testing adjustment in most cases.
- They are embarrassingly parallel, so they scale extremely well.

Score-based Structure Learning Algorithms

The dimensionality of the space of graph structures makes an exhaustive search unfeasible in practice, regardless of the goodness-of-fit measure (called network score) used in the process. However, heuristics can still be used in conjunction with decomposable scores, i.e.

$$\text{Score}(\mathcal{G}) = \sum \text{Score}(X_i \,|\, \Pi_{X_i})$$

such as

$$\text{BIC}(\mathcal{G}) = \sum \log \text{P}(X_i \,|\, \Pi_{X_i}) - \frac{|\Theta_{X_i}|}{2} \log n$$

$$\text{BDe}(\mathcal{G}), \text{BGe}(\mathcal{G}) = \sum \log \left[\int \text{P}(X_i \,|\, \Pi_{X_i}, \Theta_{X_i}) \, \text{P}(\Theta_{X_i} \,|\, \Pi_{X_i}) d\Theta_{X_i} \right]$$

if each comparison involves structures differing in only one local distribution at a time.

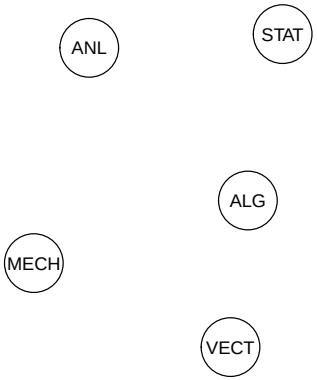
The Hill-Climbing Algorithm

The Hill-Climbing Algorithm

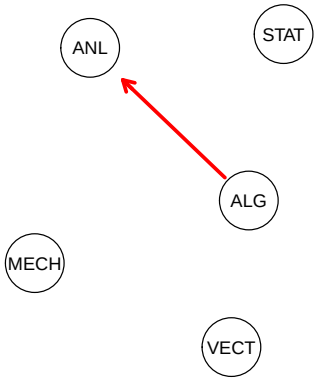
1. Choose an initial network structure $\mathcal{G}$, usually (but not necessarily) empty.
2. Compute the score of $\mathcal{G}$, denoted as $Score_{\mathcal{G}} = \text{Score}(\mathcal{G})$.
3. Set $maxscore = Score_{\mathcal{G}}$.
4. Repeat the following steps as long as $maxscore$ increases:
   - 4.1 for every possible arc addition, deletion or reversal not resulting in a cyclic network:
     - 4.1.1 compute the score of the modified network $\mathcal{G}^*$, $Score_{\mathcal{G}^*} = \text{Score}(\mathcal{G}^*)$:
     - 4.1.2 if $Score_{\mathcal{G}^*} > Score_{\mathcal{G}}$, set $\mathcal{G} = \mathcal{G}^*$ and $Score_{\mathcal{G}} = Score_{\mathcal{G}^*}$.
   - 4.2 update $maxscore$ with the new value of $Score_G$.
5. Return the directed acyclic graph $\mathcal{G}$.

The Hill-Climbing Algorithm

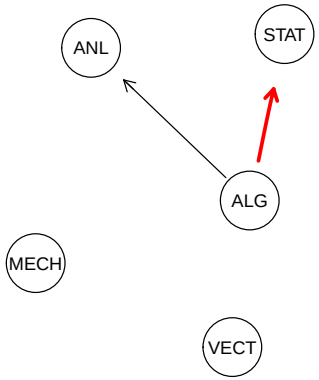
Initial BIC score: −1807.528



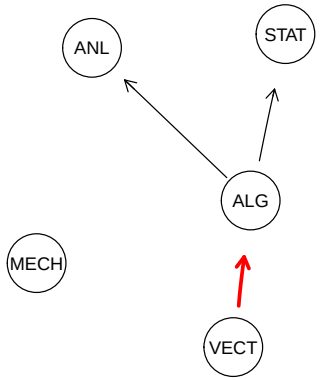
Current BIC score: −1778.804



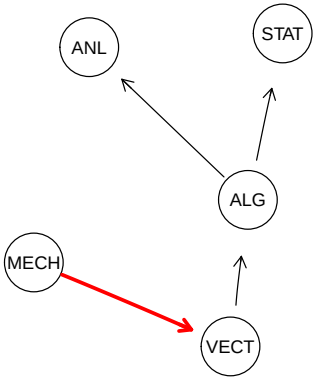
Current BIC score: −1755.383



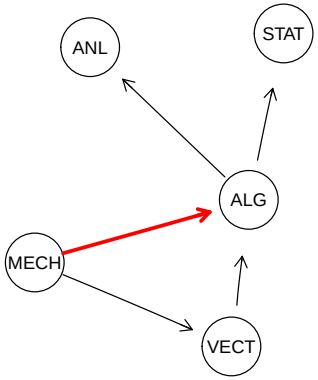
Current BIC score: −1737.176



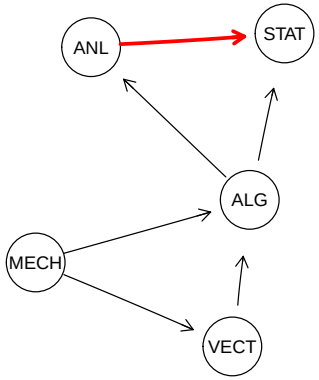
Current BIC score: −1723.325



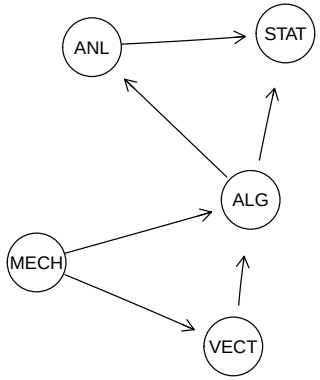
Current BIC score: −1720.901



Current BIC score: −1720.150



Final BIC score: −1720.150



Other Score-based Algorithms

- Hill-Climbing + Random Restart: performs several hill-climbing runs, perturbing the result of each one as the initial network for the next. It does get stuck in local maxima as often as plain hill-climbing.
- Greedy Equivalent Search: hill-climbing over equivalence classes rather than graph structures; the search space is much smaller.
- Tabu Search: a modified hill-climbing that keeps a list of the last k structures visited, and returns only if they are all worse than the current one.
- Genetic Algorithms: they perturb (*mutation*) and combine *crossover* features through several generations of structures, and keep the ones leading to better scores. Inspired by Darwinian evolution.
- Simulated Annealing: again similar to hill-climbing, but not looking at the maximum score improvement at each step. Very difficult to use in practice because of its tuning parameters.

Pros & Cons of Score-based Algorithms

- Convergence to the global maximum (i.e. the best structure) is not guaranteed for finite samples, the search may get stuck in a local maximum.
- They are consistent, and they converge faster than constraint-based algorithms, but this is more due to the properties of the BDe and BGe scores than the algorithms themselves.
- They require a definition of both the global and the local densities, and a matching decomposable, network score.
- Most scores have tuning parameters, whereas conditional independence tests do not.

Hybrid Structure Learning Algorithms

Hybrid algorithms combine constraint-based and score-based algorithms to complement the respective strengths and weaknesses; they are considered the state of the art in current literature.

They work by alternating the following two steps:

- learn some conditional independence constraints to restrict the number of candidate networks;
- find the best network that satisfies those constraints and define a new set of constraints to improve on.

These steps can be repeated several times (until convergence), but one or two times is usually enough.

The Sparse Candidate Algorithm

The Sparse Candidate Algorithm

1. Choose a network structure $\mathcal{G}$, usually (but not necessarily) empty.
2. Repeat the following steps until convergence:
   - 2.1 **restrict:** select a set $\mathbf{C}_i$ of candidate parents for each node $X_i \in \mathbf{X}$, which must include the parents of X_i in $\mathcal{G}$;
   - 2.2 **maximise:** find the network structure $\mathcal{G}^*$ that maximises $\mathrm{Score}(\mathcal{G}^*)$ among the networks in which the parents of each node X_i are included in the corresponding set $\mathbf{C}_i$;
   - 2.3 set $\mathcal{G} = \mathcal{G}^*$.
3. Return the directed acyclic graph $\mathcal{G}$.

Pros & Cons of Structure Learning Algorithms

- Since only the general framework is defined, it is easy to modify them to use newer constraint-based and score-based algorithms.
- You can pick and match conditional independence tests and network scores to create a learning algorithm ranging from frequentist to Bayesian to information-theoretic and anything in between (within reason).
- They are usually faster than the alternatives, and more stable.
- Tuning parameters can be difficult to tune for some configurations of algorithms, tests and scores.

Parameter Learning

The Big Three: Likelihood, Bayesian and Shrinkage

Once the structure of the model is known, the problem of estimating the parameters of the global distribution can be solved by estimating the parameters of the local distributions, one at a time.

Three common choices are:

- maximum likelihood estimators: just the usual empirical estimators. Often described as either maximum entropy or minimum divergence estimators in information-theoretic literature.
- Bayesian posterior estimators: posterior estimators, based on conjugate priors to keep computations fast, simple and in closed form.
- shrinkage estimators: regularised estimators based either on James-Stein or Bayesian shrinkage results.

Maximum Likelihood and Maximum Entropy Estimation

The classic estimators for (conditional) probabilities and (partial) correlations are a bad choice for almost all real-world problems.
They are still around because:

- they are used in benchmark simulations;
- computer scientists do not care much about parameter estimation.

However:

- maximum likelihood estimates are unstable in most multivariate problems, both discrete and continuous;
- for the multivariate Gaussian distribution, James & Stein proved in the 1950s that the maximum likelihood estimator for the mean is not admissible in 3+ dimensions;
- partial correlations are often ill-behaved because of that, even with Moore-Penrose pseudo-inverses;
- maximum likelihood estimates are non-smooth and create problems when using the graphical model for inference.

Maximum a Posteriori Bayesian Estimation

Bayesian posterior estimates are the sensible choice for parameter estimation according to Koller's & Friedman's tome on graphical models. Choices for the priors are limited (for computational reasons) to conjugate distributions, namely:

- the Dirichlet for discrete models, i.e.

$$Dir(\alpha_{k \mid \Pi_{X_i}=\pi}) \quad \xrightarrow{data} \quad Dir(\alpha_{k \mid \Pi_{X_i}=\pi} + n_{k \mid \Pi_{X_i}=\pi})$$

  meaning that $\hat{p}_{k \mid \Pi_{X_i}=\pi} = \alpha_{k \mid \Pi_{X_i}=\pi} / \sum_{\pi} \alpha_{k \mid \Pi_{X_i}=\pi}$.

- the Inverse Wishart for Gaussian models, i.e.

$$IW(\Psi, m) \quad \xrightarrow{data} \quad IW(\Psi + n\Sigma, m + n).$$

In both cases (when a non-informative prior is used) the only free parameter is the equivalent or imaginary sample size, which gives the relative weight of the prior compared to the observed sample.

Bayesian LASSO and Ridge Regression

Gaussian graphical models, being closely related with linear regression, have also used ridge regression (L_2 regularisation) and LASSO (L_1 regularisation) in their Bayesian capacity.

LASSO corresponds to a Laplace prior on the regression coefficients,

$$\beta_k \,|\, \sigma^2 \sim Laplace(0, \sigma^2).$$

Ridge Regression corresponds to a Gaussian prior,

$$\beta_k \,|\, \sigma^2 \sim N(0, \sigma^2).$$

In both cases tuning the σ^2 parameter is crucial, as it takes the role of the λ regularisation parameter found in the original frequentist definitions of these methods.

Other priors are also possible (Student's t, Normal-Exponential-Gamma for HyperLASSO); some are better at controlling sparsity than others.

Shrinkage, James-Stein Estimation

Shrinkage estimation is based on results from James & Stein on the estimation of the mean of a multivariate Gaussian distribution, and takes the form

$$\tilde{\theta} = \lambda t + (1 - \lambda)\hat{\theta} \qquad \lambda \in [0, 1]$$

where the optimal λ (with respect to squared loss) can be estimated in closed form as

$$\lambda^* = \min \left(\frac{\sum_k \mathsf{VAR}(\hat{\theta}_k) - \mathsf{COV}(\hat{\theta}_k, t_k) + \mathsf{Bias}(\hat{\theta}_k)\,\mathrm{E}(\hat{\theta}_k - t_k)}{\sum_k (\hat{\theta}_k - t_k)^2}, 1 \right)$$

The James-Stein estimator $\tilde{\theta}$ dominates the maximum likelihood estimator $\hat{\theta}$ and converges to the latter as the sample size grows. It can be interpreted as an empirical Bayes estimator.

Shrinkage, James-Stein Estimation

For discrete data, conditional probabilities $p_{k\,|\,\pi} = p_{k\,|\,\Pi_{X_i}=\pi}$ end up being estimated as

$$\tilde{p}_{k\,|\,\pi} = \lambda^* t_{k\,|\,\pi} + (1-\lambda^*)\hat{p}_{k\,|\,\pi}, \quad \lambda^* = \min\left(\frac{1-\sum_k \hat{p}^2_{k\,|\,\pi}}{(n-1)\sum_k (t_{k\,|\,\pi} - \hat{p}_{k\,|\,\pi})^2}, 1\right),$$

where t is the uniform (discrete) distribution.

For continuous data, correlations end up being estimated from the shrunk covariance matrix $\tilde{\Sigma}$

$$\tilde{\sigma}_{ii} = \hat{\sigma}_{ii}, \quad \tilde{\sigma}_{ij} = (1-\lambda^*)\hat{\sigma}_{ij}, \quad \lambda^* = \min\left(\frac{\sum_{i\neq j} \mathrm{VAR}(\hat{\sigma}_{ij})}{\sum_{i\neq j} \hat{\sigma}^2_{ij}}, 1\right)$$

where t is $\mathrm{diag}(\hat{\Sigma})$. $\tilde{\Sigma}$ is guaranteed to have full rank, so it can be safely inverted to get partial correlations.

Model Validation

The Big Three: Frequentist, Bayesian and Hybrid

The results of both structure learning and parameter learning should be validated before using a graphical model for inference. Since parameters are learned conditional on the results of structure learning, validating the graph structure learned from the data is an essential step in graphical modelling.

- frequentist: generating network structures using bootstrap and model averaging (aka bagging).
- Bayesian: generating network structures from the posterior $\mathrm{P}(\mathcal{G} \mid \mathcal{D})$ using exhaustive enumeration or Markov Chain Mote Carlo approximations.
- hybrid: generating network structures again using bootstrap, but weighting them with their posterior probabilities when performing model averaging.

A Frequentist Approach: Friedman's Confidence

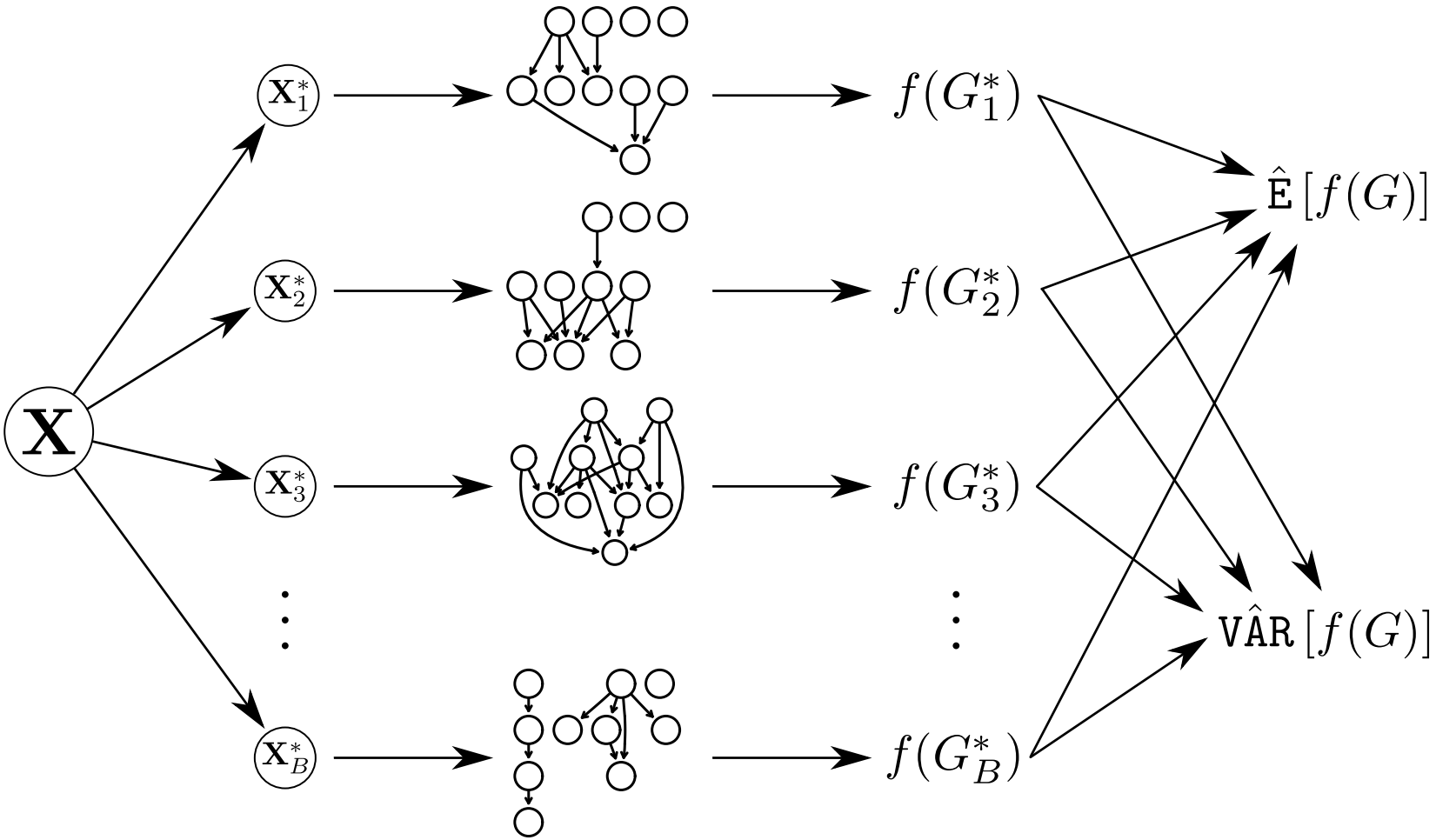
Friedman et al. proposed an approach to model validation based on bootstrap resampling and model averaging:

1. For $b = 1, 2, \ldots, m$:
   1.1 sample a new data set $\mathbf{X}_b^*$ from the original data $\mathbf{X}$ using either parametric or nonparametric bootstrap;
   1.2 learn the structure of the graphical model $\mathcal{G}_b = (\mathbf{V}, E_b)$ from $\mathbf{X}_b^*$.
2. Estimate the confidence that each possible edge e_i is present in the true network structure $\mathcal{G}_0 = (\mathbf{V}, E_0)$ as

$$\hat{p}_i = \hat{\mathrm{P}}(e_i) = \frac{1}{m} \sum_{b=1}^{m} \mathbb{1}_{\{e_i \in E_b\}},$$

where $\mathbb{1}_{\{e_i \in E_b\}}$ is equal to 1 if $e_i \in E_b$ and 0 otherwise.

A Frequentist Approach: Friedman's Confidence



A (Full) Bayesian Approach

Performing a full posterior Bayesian analysis on graph structures, that is, working with

$$\hat{p}_i = \mathrm{E}(e_i|\mathcal{D}) = \sum_{\mathcal{G}} \mathbb{1}_{\{e_i \in E_{\mathcal{G}}\}} \mathrm{P}(\mathcal{G} \,|\, \mathcal{D}),$$

is considered unfeasible for networks with more than ~ 10 nodes because:

- an exhaustive enumeration takes too long, even for Markov networks (and it's even worse for Bayesian networks because of the acyclicity constraint);
- generating graphs from the posterior distribution is feasible but convergence of the MCMC to the stationary distribution is far from certain (mixing is often too slow).

An Hybrid Approach: the "Bayesian confidence"

Friedman's confidence and Bayesian posterior analysis may be combined as follows:

1. For $b = 1, 2, \ldots, m$:
   - 1.1 sample a new data set $\mathbf{X}_b^*$ from the original data $\mathbf{X}$ using either parametric or nonparametric bootstrap;
   - 1.2 learn the structure of the graphical model $\mathcal{G}_b = (\mathbf{V}, E_b)$ from $\mathbf{X}_b^*$.
2. Estimate the confidence for each possible edge e_i as

$$\hat{p}_i = \mathrm{E}(e_i|\mathcal{D}) \simeq \frac{1}{m} \sum_{b=1}^{m} \mathbb{1}_{\{e_i \in E_b\}} \mathrm{P}(\mathcal{G}_b \,|\, \mathcal{D}).$$

The result is a form of approximate Bayesian estimation, whose behaviour depends on how much of the posterior probability mass is concentrated in the subset of graph structures $\mathcal{G}_b$.

Identifying Significant Edges

- The confidence values $\hat{\mathbf{p}} = \{\hat{p}_i\}$ do not sum to one and are dependent on one another in a nontrivial way; the value of the confidence threshold (i.e. the minimum confidence for an edge to be accepted as an edge of $\mathcal{G}_0$) is an unknown function of both the data and the structure learning algorithm.
- The ideal/asymptotic configuration $\tilde{\mathbf{p}}$ of confidence values would be

$$\tilde{p}_i = \begin{cases} 1 & \text{if } e_i \in E_0 \\ 0 & \text{otherwise} \end{cases},$$

  i.e. all the networks $\mathcal{G}_b$ have exactly the same structure.
- Therefore, identifying the configuration $\tilde{\mathbf{p}}$ "closest" to $\hat{\mathbf{p}}$ provides a principled way of identifying significant edges and the confidence threshold.

The Confidence Threshold

Consider the order statistics $\tilde{\mathbf{p}}_{(\cdot)}$ and $\hat{\mathbf{p}}_{(\cdot)}$ and the cumulative distribution functions (CDFs) of their elements:

$$F_{\hat{\mathbf{p}}_{(\cdot)}}(x) = \frac{1}{k} \sum_{i=1}^{k} \mathbb{1}_{\{\hat{p}_{(i)} < x\}}$$

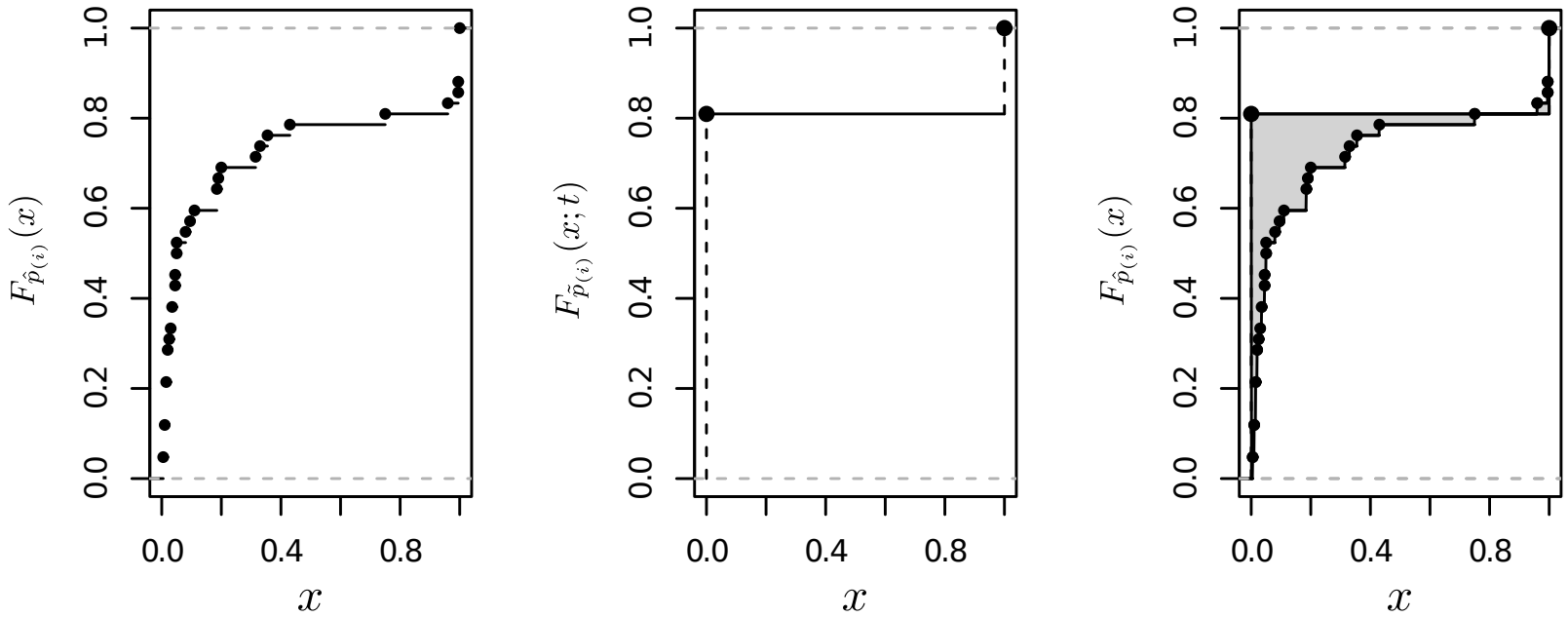
and

$$F_{\tilde{\mathbf{p}}_{(\cdot)}}(x; t) = \begin{cases} 0 & \text{if } x \in (-\infty, 0) \\ t & \text{if } x \in [0, 1) \\ 1 & \text{if } x \in [1, +\infty) \end{cases} .$$

t corresponds to the fraction of elements of $\tilde{\mathbf{p}}_{(\cdot)}$ equal to zero and is a measure of the fraction of non-significant edges, and provides a threshold for separating the elements of $\tilde{\mathbf{p}}_{(\cdot)}$:

$$e_{(i)} \in E_0 \iff \hat{p}_{(i)} > F^{-1}_{\tilde{\mathbf{p}}_{(\cdot)}}(t).$$

The CDFs $F_{\hat{\mathbf{p}}_{(\cdot)}}(x)$ and $F_{\tilde{\mathbf{p}}_{(\cdot)}}(x;t)$



One possible estimate of t is the value $\hat{t}$ that minimises some distance between $F_{\hat{\mathbf{p}}_{(\cdot)}}(x)$ and $F_{\tilde{\mathbf{p}}_{(\cdot)}}(x;t)$; an intuitive choice is using the L_1 norm of their difference (i.e. the shaded area in the picture on the right).

An L_1 Estimator for the Confidence Threshold

Since $F_{\hat{\mathbf{p}}_{(\cdot)}}$ is piece-wise constant and $F_{\tilde{\mathbf{p}}_{(\cdot)}}$ is constant in $[0, 1]$, the L_1 norm of their difference simplifies to

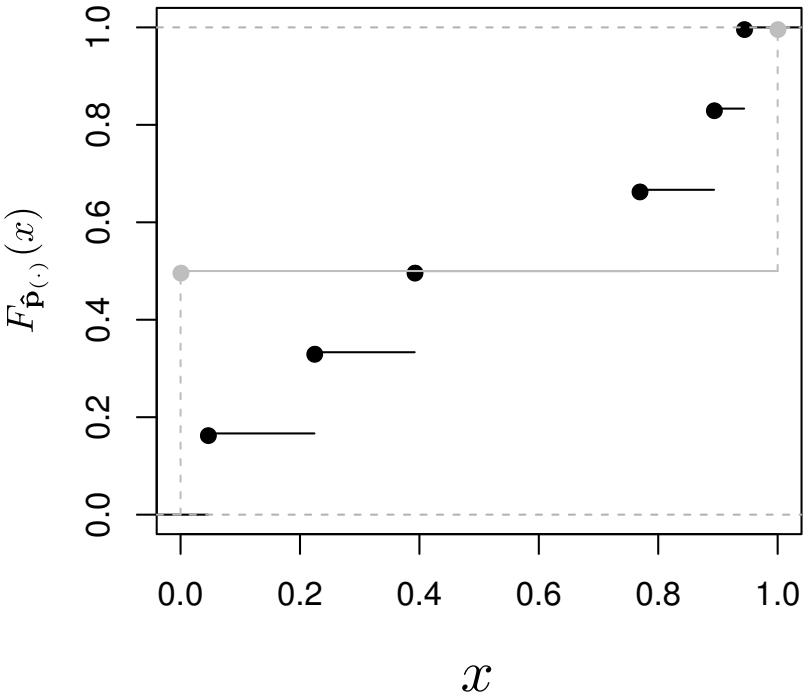
$$\begin{aligned} L_1\left(t; \hat{\mathbf{p}}_{(\cdot)}\right) &= \int \left| F_{\hat{\mathbf{p}}_{(\cdot)}}(x) - F_{\tilde{\mathbf{p}}_{(\cdot)}}(x; t) \right| dx \\ &= \sum_{x_i \in \left\{\{0\} \cup \hat{\mathbf{p}}_{(\cdot)} \cup \{1\}\right\}} \left| F_{\hat{\mathbf{p}}_{(\cdot)}}(x_i) - t \right| (x_{i+1} - x_i). \end{aligned}$$

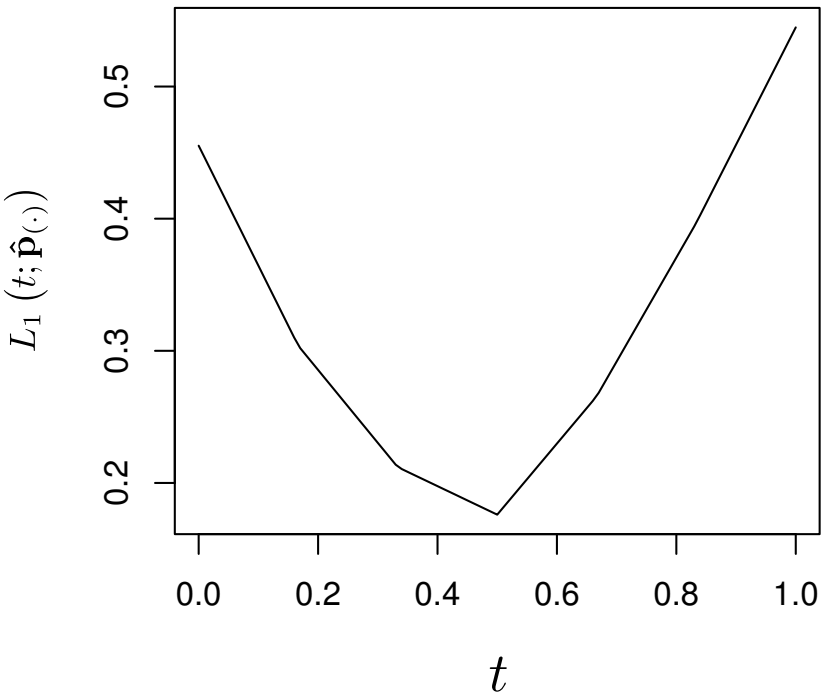
This form has two important properties:

- can be computed in linear time from $\hat{\mathbf{p}}_{(\cdot)}$;
- its minimisation is straightforward using linear programming.

Furthermore, the L_1 norm does not place as much weight on large deviations as other norms (L_2, L_∞), making it robust against a wide variety of configurations of $\hat{\mathbf{p}}_{(\cdot)}$.

A Simple Example





Consider a graph with 4 nodes and confidence values

$$\hat{\mathbf{p}}_{(\cdot)} = \{0.0460, 0.2242, 0.3921, 0.7689, 0.8935, 0.9439\}$$

Then $\hat{t} = \min_t L_1\left(t; \hat{\mathbf{p}}_{(\cdot)}\right) = 0.4999816$ and $F^{-1}_{\tilde{\mathbf{p}}_{(\cdot)}}(0.4999816) = 0.3921$; only three edges are considered significant.

Benchmarking Performance (ALARM network)

n	n/p	TPR	FPR	TNR
100	0.196464	0.563044	0.010129	0.989871
200	0.392927	0.698261	0.010710	0.989290
500	0.982318	0.845652	0.011161	0.988839
1000	1.964637	0.898696	0.012323	0.987677
2000	3.929273	0.911304	0.015387	0.984613
5000	9.823183	0.919130	0.016677	0.983323
10000	19.646365	0.923913	0.016129	0.983871
20000	39.292731	0.952174	0.017129	0.982871

ALARM has 37 nodes, 46 edges and 509 parameters.

Benchmarking Performance (BARLEY network)

n	n/p	TPR	FPR	TNR
100	0.000877	0.332381	0.014655	0.985345
200	0.001754	0.396905	0.008793	0.991207
500	0.004386	0.457143	0.009253	0.990747
1000	0.008772	0.495952	0.009732	0.990268
2000	0.017543	0.544524	0.010651	0.989349
5000	0.043858	0.561905	0.016130	0.983870
10000	0.087715	0.610476	0.018218	0.981782
20000	0.175431	0.638810	0.017950	0.982050

BARLEY has 48 nodes, 84 edges and 114005 parameters.

Conclusions

Conclusions

- Graphical models combine many ideas from different fields to allow an intuitive manipulation of high-dimensional problems and the corresponding multivariate probability distributions.
- A sensible use of Bayesian and shrinkage techniques in structure and parameter learning allows a great deal of flexibility and results in good models.
- Properly validated graphical models can capture the dependence structure of the data even with very small sample sizes.

Thanks for Not Falling Asleep

References

References I

R. R. Bouckaert.
Bayesian Belief Networks: from Construction to Inference.
PhD thesis, Utrecht University, The Netherlands, 1995.

D. M. Chickering.
Optimal Structure Identification with Greedy Search.
Journal of Machine Learning Resesearch, 3:507–554, 2002.

D. I. Edwards.
Introduction to Graphical Modelling.
Springer, 2nd edition, 2000.

J. Friedman, T. Hastie, and R. Tibshirani.
Sparse Inverse Covariance Estimation With the Graphical Lasso.
Biostatistics, 9:432–441, 2007.

N. Friedman, M. Goldszmidt, and A. Wyner.
Data Analysis with Bayesian Networks: A Bootstrap Approach.
In *Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 206 – 215. Morgan Kaufmann, 1999.

References II


N. Friedman and D. Koller.
Being Bayesian about Bayesian Network Structure: A Bayesian Approach to Structure Discovery in Bayesian Networks.
Machine Learning, 50(1–2):95–126, 2003.


N. Friedman, D. Pe'er, and I. Nachman.
Learning Bayesian Network Structure from Massive Datasets: The "Sparse Candidate" Algorithm.
In *Proceedings of 15th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 206–221. Morgan Kaufmann, 1999.


D. Geiger and D. Heckerman.
Learning Gaussian Networks.
Technical report, Microsoft Research, Redmond, Washington, 1994.
Available as Technical Report MSR-TR-94-10.


F. Harary and E. M. Palmer.
Graphical Enumeration.
Academic Press, 1973.

References III


T. Hastie, R. Tibshirani, and J. Friedman.
The Elements of Statistical Learning: Data Mining, Inference, and Prediction.
Springer, 2nd edition, 2009.


J. Hausser and K. Strimmer.
Entropy Inference and the James-Stein Estimator, with Application to Nonlinear Gene Association Networks.
Journal of Machine Learning Resesearch, 10:1469–1484, 2009.


D. Heckerman, D. Geiger, and D. M. Chickering.
Learning Bayesian Networks: The Combination of Knowledge and Statistical Data.
Machine Learning, 20(3):197–243, September 1995.
Available as Technical Report MSR-TR-94-09.


C. J. Hoggart, J. C. Whittaker, M. De Iorio, and D. J. Balding.
Simultaneous Analysis of All SNPs in Genome-Wide and Re-Sequencing Association Studies.
PLoS Genetics, 4(7), 2008.

References IV


J. S. Ide and F. G. Cozman.
Random Generation of Bayesian Networks.
In *Proceedings of the 16th Brazilian Symposium on Artificial Intelligence*, pages 366–375. Springer-Verlag, 2002.


S. Imoto, S. Y. Kim, H. Shimodaira, S. Aburatani, K. Tashiro, S. Kuhara, and S. Miyano.
Bootstrap Analysis of Gene Networks Based on Bayesian Networks and Nonparametric Regression.
Genome Informatics, 13:369–370, 2002.


W. James and C. Stein.
Estimation with Quadratic Loss.
In J. Neyman, editor, *Proceedings of the 4th Berkeley Symposium on Mathematical Statistics and Probability*, pages 361–379, 1961.


D. Koller and N. Friedman.
Probabilistic Graphical Models: Principles and Techniques.
MIT Press, 2009.

References V

K. Korb and A. Nicholson.
Bayesian Artificial Intelligence.
Chapman and Hall, 2nd edition, 2009.

S. Kullback.
Information Theory and Statistics.
Dover Publications, 1968.

O. Ledoit and M. Wolf.
Improved Estimation of the Covariance Matrix of Stock Returns with an Application to Portfolio Selection.
Journal of Empirical Finance, 10:603–621, 2003.

P. Legendre.
Comparison of Permutation Methods for the Partial Correlation and Partial Mantel Tests.
Journal of Statistical Computation and Simulation, 67:37–73, 2000.

References VI

D. Margaritis.
Learning Bayesian Network Model Structure from Data.
PhD thesis, School of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, May 2003.
Available as Technical Report CMU-CS-03-153.

G. Melançon, I. Dutour, and M. Bousquet-Mélou.
Random Generation of DAGs for Graph Drawing.
Technical Report INS-R0005, Centre for Mathematics and Computer Sciences, Amsterdam, 2000.

J. Pearl.
Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.
Morgan Kaufmann, 1988.

F. Pesarin and L. Salmaso.
Permutation Tests for Complex Data: Theory, Applications and Software.
Wiley, 2010.

S. J. Russell and P. Norvig.
Artificial Intelligence: A Modern Approach.
Prentice Hall, 3rd edition, 2009.

References VII


J. Schäfer and K. Strimmer.
A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics.
Statistical Applications in Genetics and Molecular Biology, 4:32, 2005.


G. E. Schwarz.
Estimating the Dimension of a Model.
Annals of Statistics, 6(2):461 – 464, 1978.


M. Scutari and K. Strimmer.
Introduction to Graphical Modelling.
In D. J. Balding, M. Stumpf, and M. Girolami, editors, *Handbook of Statistical Systems Biology*. Wiley, 2011.
In print.


P. Spirtes, C. Glymour, and R. Scheines.
Causation, Prediction, and Search.
MIT Press, 2000.

References VIII

C. Stein.
Inadmissibility of the Usual Estimator for the Mean of a Multivariate Distribution.
In J. Neyman, editor, *Proceedings of the 3rd Berkeley Symposium on Mathematical Statistics and Probability*, pages 197–206, 1956.

I. Tsamardinos, C. F. Aliferis, and A. Statnikov.
Algorithms for Large Scale Markov Blanket Discovery.
In *Proceedings of the 16th International Florida Artificial Intelligence Research Society Conference*, pages 376–381. AAAI Press, 2003.

I. Tsamardinos, L. E. Brown, and C. F. Aliferis.
The Max-Min Hill-Climbing Bayesian Network Structure Learning Algorithm.
Machine Learning, 65(1):31–78, 2006.

T. S. Verma and J. Pearl.
Equivalence and Synthesis of Causal Models.
Uncertainty in Artificial Intelligence, 6:255–268, 1991.

J. Whittaker.
Graphical Models in Applied Multivariate Statistics.
Wiley, 1990.

References IX

S. Yaramakala and D. Margaritis.
Speculative Markov Blanket Discovery for Optimal Feature Selection.
In *Proceedings of the 5th IEEE International Conference on Data Mining*, pages 809–812. IEEE Computer Society, 2005.