Probabilistic graphical models – supportive slides

Jiří Kléma

Department of Computer Science,
FEE, CTU at Prague

http://ida.felk.cvut.cz

Probabilistic reasoning under uncertainty

- uncertainty
  - result of partial observability and/or nondeterminism,
  - sentences cannot be decided exactly,
  - an agent can only have a degree of belief in them,
- probability
  - the main tool for dealing with degrees of belief,
  - fully specified probabilistic model
    - world = atomic event = sample point,
  - every question about a domain can be answered with the full model,
  - the **full joint distribution** is the most common full model
    - for n discrete variables: $Pr(\mathcal{O}_1, \mathcal{O}_2, \ldots \mathcal{O}_n)$.

Probabilistic reasoning under uncertainty

- what questions do we answer?
  - event = sum of atomic events
    - propositions in the absence of any other information,
    - unconditonal or prior probability,
  - dealing with evidence
    - conditonal or posterior probability,
    - this will later be called **inference**.

Notation (binary random variables):

A ... random variable, a ... $A = True$, $\neg a$ ... $A = False$,

$Pr(A, B)$ ... joint probability distribution (a table),

$Pr(a, b) = Pr(A = True, B = True)$ ... probability of a particular event (a single item in table $Pr(A, B)$).

Inference with the full joint model

- every question about the domain can be answered
  - **marginalization** (summing out) to obtain prior probabilities

$$Pr(\mathbf{X}) = \sum_{\mathbf{y} \in \mathbf{Y}} Pr(\mathbf{X}, \mathbf{y}) \quad \text{(}\mathbf{X}\text{ and }\mathbf{Y}\text{ are sets of variables)}$$

  - **normalization** follows to obtain conditional probabilities
    * it either directly follows the definition of conditional probability

$$Pr(\mathbf{X}|\mathbf{Y}) = \frac{Pr(\mathbf{X}, \mathbf{Y})}{Pr(\mathbf{Y})}$$

    * or it works with a normalization constant α,
    * it avoids $Pr(\mathbf{Y})$ enumeration

$$Pr(\mathbf{X}|\mathbf{Y}) = \alpha Pr(\mathbf{X}, \mathbf{Y}), \ \alpha \text{ is set so that} \sum_{\mathbf{x} \in \mathbf{X}} Pr(\mathbf{x}|\mathbf{Y}) = 1.$$

Inference with the full joint model – example

- admission to graduate schools with respect to branch of study and gender
  - real data available, the full joint model can easily be constructed,

Branch	Men		Women	
	Applicants	Admitted	Applicants	Admitted
Engineering	1385	865	133	90
Humanities	1205	327	1702	451

(E)ngineering	(M)an	(A)dmitted	f(E,M,A)	Pr(E,M,A)
T	T	T	865	19.5%
T	T	F	520	11.8%
T	F	T	90	2.0%
T	F	F	43	1.0%
F	T	T	327	7.4%
F	T	F	878	19.8%
F	F	T	451	10.2%
F	F	F	1251	28.3%
	Total		4425	100%

Inference with the full joint model – example

- what is the probability of admission?
- the marginalization task

$$\begin{aligned} Pr(a) &= \sum_{E,M} Pr(E,M,a) = \\ &= Pr(e,m,a) + Pr(e,\neg m,a) + Pr(\neg e,m,a) + Pr(\neg e,\neg m,a) = .392 \end{aligned}$$

Inference with the full joint model – example

- what is the probability of admission given gender?
- marginalization followed by normalization, the direct way used for men

$$Pr(a|m) = \frac{Pr(a,m)}{Pr(m)} = \frac{\sum_E Pr(E,m,a)}{\sum_{E,A} Pr(E,m,A)} =$$
$$= \frac{Pr(e,m,a) + Pr(\neg e,m,a)}{Pr(e,m,a) + Pr(e,m,\neg a) + Pr(\neg e,m,a) + Pr(\neg e,m,\neg a)} = .46$$

- the α trick way used for women, $\alpha = 2.41$, $Pr(a|\neg m) = 0.29$,

$$Pr(A|\neg m) = \alpha Pr(A,\neg m) = \alpha[Pr(e,\neg m,A) + Pr(\neg e,\neg m,A)] =$$
$$= \alpha[\langle .02, .01\rangle + \langle .102, .283\rangle] = \alpha[\langle .122, .293\rangle] = \langle .29, .71\rangle$$

- the university could be (and actually was) sued for bias against women!!!

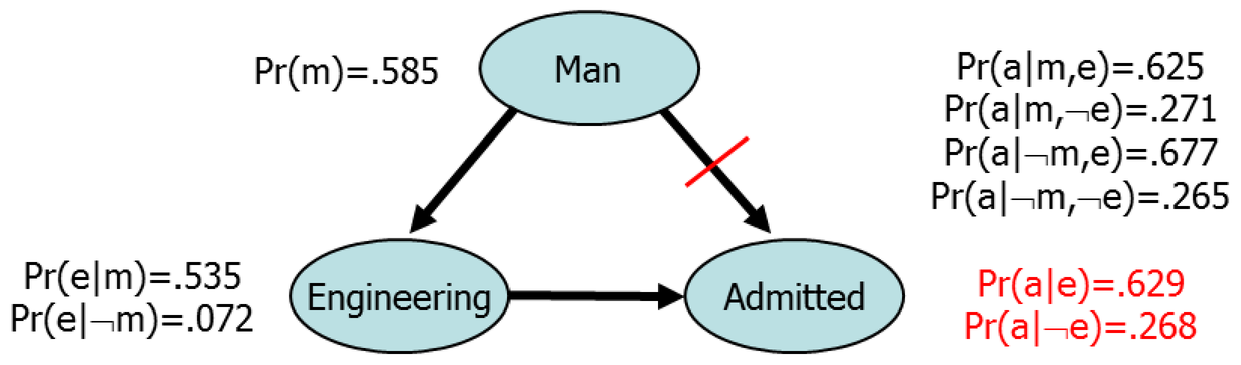
Pros and cons of the full joint distribution model

- universality makes an asset of this model
  - identical and trivial model structure for all problems,
  - for a **sufficient** sample size its learning converges
    - model learning means to estimate (joint) probabilities,
- intractable for real problems
  - $2^n - 1$ probabilities for n propositions,
  - infeasible for experts nor empirical settings based on data,
  - even if probs were known, still exponential in memory and inference time
    - obvious for a joint continuous distribution function,
  - **curse of dimensionality**
    - volume of the space increases fast, the available data become sparse,
- impenetrable for real tasks
  - model gives no explicit **knowledge** about the domain.

The ways to simplify and better organize the model?

- utilize the domain knowledge (or discover it)
  - relationship between the random variables?
  - ex.: gender influences branch of study, it influences admission rate,
  - probabilistic model is enriched with structured knowledge representation,
- graphical probabilistic representation
  - relations posed in terms of directed graph
    - connected means related (edge unconditionally, path conditionally),
  - interpretation in probabilistic context?
    - structured and simplified representation of the joint distribution,
    - edges removed when **(conditional) independence** is employed,
- advantages
  - fewer parameters needed, less data needed for learning,
  - relationships become obvious.

The simplified graphical model – admission example



- still 7 parameters (probability values) in the fully connected graph
  - simplification available, gender and admission conditionally independent,
  - the edge Man $\rightarrow$ Admitted removed, only 5 parameters then,
- branch of study is a **confounder** in gender-admission relationship,
- any joint probability can be approximated by the simplified model (and thus any other probability)

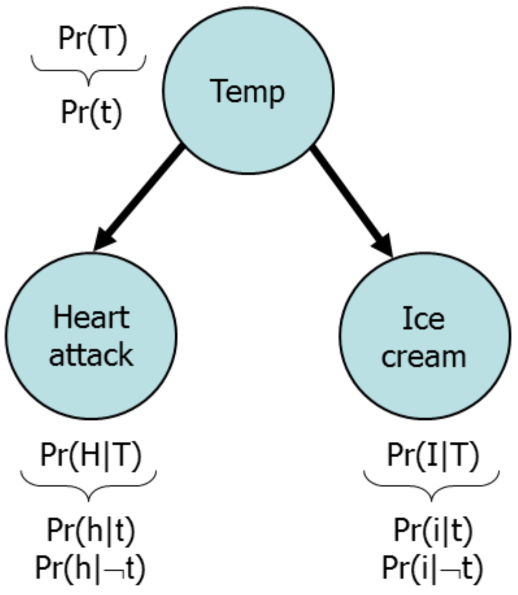
$$Pr(e,m,a) = Pr(m) \times Pr(e|m) \times Pr(a|e,m) = .195 \quad \text{the full model}$$
$$Pr(e,m,a) = Pr(m) \times Pr(e|m) \times Pr(a|e) = .197 \quad \text{the simplified model}$$

(Conditional) independence

- **definition**: A and B are conditionally independent (CI) given C if:
  - $Pr(A, B|C) = Pr(A|C) \times Pr(B|C)$, $\forall A, B, C, Pr(C) \neq 0$
  - denoted as $A \perp\!\!\!\perp B|C$ (conditional dependence $A \not\perp\!\!\!\perp B|C$)
  - (classical independence between A and B: $Pr(A, B) = Pr(A) \times Pr(B)$)
- some observations make other observations uninteresting
  - under assumption of CI it holds:
    $Pr(B|C) = Pr(B|A, C)$ and $Pr(A|C) = Pr(A|B, C)$,
  - observing C, knowledge of A becomes redundant for knowing B,
  - observing C, knowledge of B becomes redundant for knowing A,
- compare with the general formula taking no assumptions
  - $Pr(A, B|C) = Pr(A|C) \times Pr(B|A, C) = Pr(B|C) \times Pr(A|B, C)$

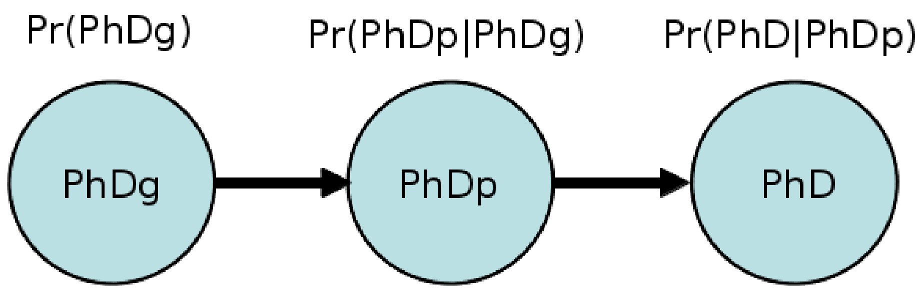
(Conditional) independence

- Example 1:
  - heart attack rate (H) grows with ice cream sales (I),
  - variables H and I are dependent: $Pr(h|i) > Pr(h)$,
  - both grow only because of temperature (T),
  - when conditioned by T, H and I become independent: $Pr(H|I,T) = Pr(H|T)$.



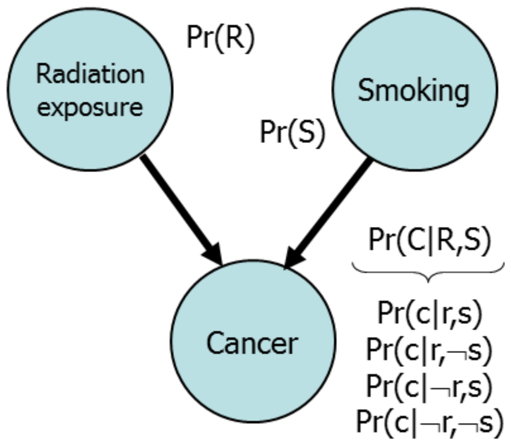
(Conditional) independence

- Example 2:
  - educated grandparents (PhDg) often have educated grandchildren (PhD): $Pr(phd|phdg) > Pr(phd)$
  - parents' education level (PhDp) makes grandparents unimportant: $Pr(PhD|PhDp, PhDg) = Pr(PhD|PhDp)$



(Conditional) independence

- Example 3:
  - both radiation exposure (R) and smoking (S) can cause cancer (C)
  - R and S are obviously independent variables: $Pr(R,S) = Pr(R) \times Pr(S)$
  - R and S become seemingly dependent knowing C! $Pr(r|s,c) < Pr(r|c)$ or $Pr(r|s,\neg c) < Pr(r|\neg c)$



- Summary
  - Ad 1 and 2) conditional independence the intermediate variable explains dependency between the ultimate ones,
  - Ad 3) independence the intermediate variable introduces spurious dependency.

Connection types

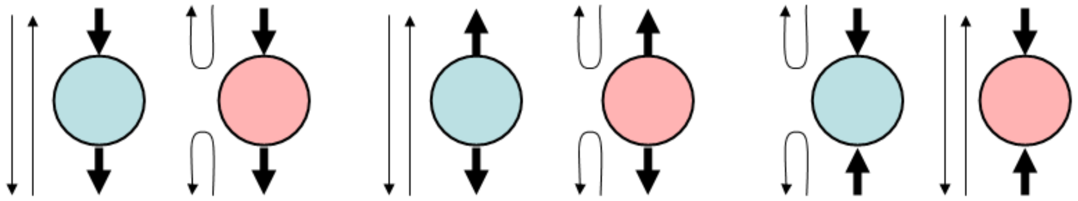
- Nomenclature
  - parent p and child/son c – a directed edge from p to c,
  - ancestor a and descendant d – a directed path from a to d,
- three connection types
  - **diverging**
    - $*$ terminal nodes dependent,
    - $*$ dependence disappears when (surely) knowing middle node,
    - $*$ crime-rate $\leftarrow$ daytime $\rightarrow$ energy consumption (and Ex. 1 – heart attacks).
    - $*$ intermediate variable (daytime) explains dependence,

Connection types

- **linear** (serial)
  - terminal nodes dependent,
  - dependence disappears when (surely) knowing middle node,
  - Simpson's paradox: gender $\rightarrow$ branch of study $\rightarrow$ admission (and Ex. 2 – PhD),
  - intermediate variable (branch of study) explains dependence,
- **converging**
  - terminal nodes independent,
  - spurious dependence introduced with knowledge of middle node,
  - temperature $\rightarrow$ ice cream sales $\leftarrow$ salesperson skills (and Ex. 3 – radiation exposure),
- analogy e.g. with partial correlations.

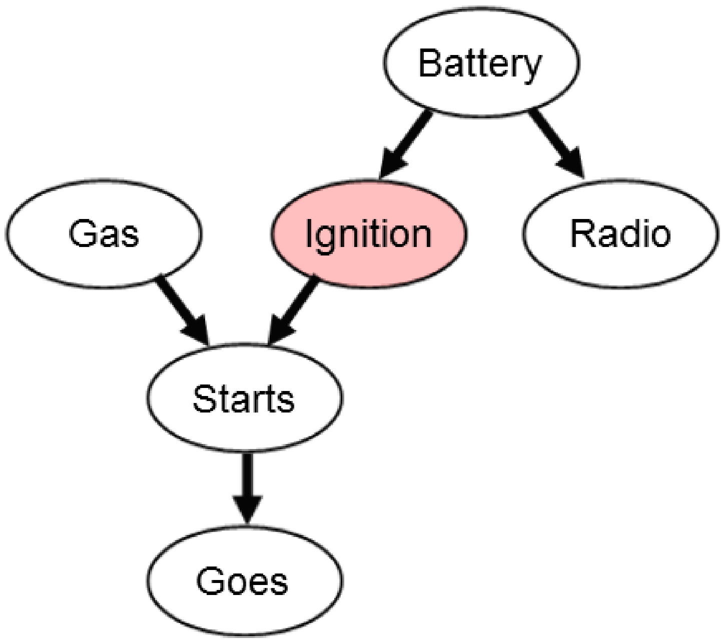
D-separation

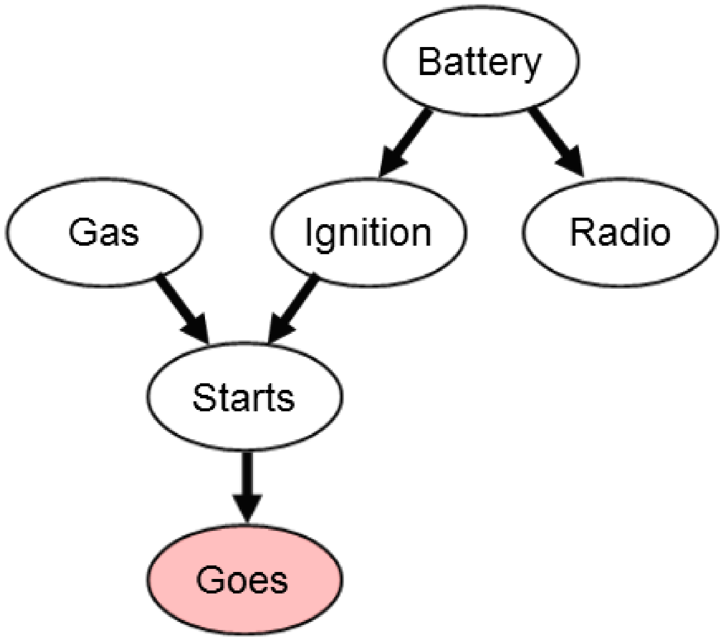
- uses connections to determine CI between sets of nodes
  - linear and diverging con. transmit information **not given** middle node,
  - converging con. transmits information **given** middle node/its descendant,



- two node sets $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by a node set $\mathbf{Z}$ iff
  - all undirected paths between any node pair $x \in \mathbf{X}$ and $y \in \mathbf{Y}$ blocked
    - $*$ there is a linear or diverging node $z \in \mathbf{Z}$ on the path, or
    - $*$ there is a converging node $w \notin \mathbf{Z}$, none of its descendants is in $\mathbf{Z}$,
- d-separation is equivalent of CI between $\mathbf{X}$ and $\mathbf{Y}$ given $\mathbf{Z}$,
- a tool of abstraction
  - generalizes from 3 to multiple nodes when studying information flow.

D-separation – example, car diagnosis BN [Russel: AIMA]





- $Gas, Start, Go \perp\!\!\!\perp Bat, Rad | Ign$
- sets are d-separated
- no open path for any pair of nodes
  - Gas x Battery, Gas x Radio etc.
  - all paths blocked by linear node

- $Gas \not\perp\!\!\!\perp Ign, Bat, Rad | Go$
- sets are not d-separated
- node $Goes$ opens one path at least
  - Starts connects Gas and Ignition
  - observed descendant of converging node

Graphical probabilistic models

- exploit both probability theory and graph theory,
- graph = qualitative part of model
  - nodes represent events / random variables,
  - edges represent dependencies between them,
  - CI can be seen in graph.
- probability = quantitative part of model
  - local information about node and its neighbors,
  - the strength of dependency, way of inference,
- different graph types (directed/undirected edges, constraints), probability encoding and focus
  - Bayesian networks – causal and probabilistic processes,
  - Markov networks – images, hidden causes,
  - data flows – deterministic computations,
  - influence diagrams – decision processes.

Bayesian networks (BNs)

- What is a Bayesian network (also Bayes or belief or causal network)?
  - directed acyclic graph – DAG,
  - nodes represent random variables (typically discrete),
  - edges represent direct dependence,
  - nodes annotated by probabilities (tables, distributions)
    - node conditioned by conjunction of all its parents,
    - $Pr(\mathcal{O}_{j+1}|\mathcal{O}_1, \ldots, \mathcal{O}_j) = Pr(\mathcal{O}_{j+1}|parents(\mathcal{O}_{j+1}))$
    - root nodes annotated by prior distributions,
    - internal nodes conditioned by their parent variables,
    - other (potential) dependencies ignored,
- Network interpretation?
  - concised representation of probability distribution given CI relations,
  - qualitative constituent = graph,
  - quantitative constituent = a set of conditional probability tables (CPTs).

Bayesian networks

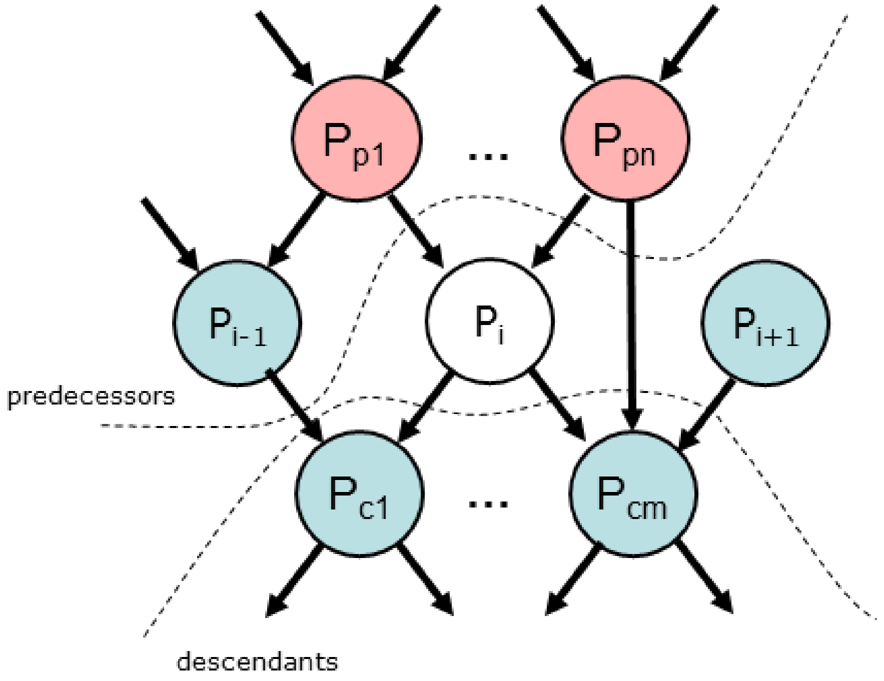
- sacrifice accuracy and completeness – focus on fundamental relationships,
- reduce complexity of representation and subsequent inference,
- full probability model
  - can be deduced by the gradual decomposition (factorization):

$$\begin{aligned} Pr(\mathcal{O}_1, \dots Obsvar_n) &= Pr(\mathcal{O}_1) \times Pr(\mathcal{O}_2, \dots, \mathcal{O}_n | \mathcal{O}_1) = \\ &= Pr(\mathcal{O}_1) \times Pr(\mathcal{O}_2 | \mathcal{O}_1) \times Pr(\mathcal{O}_3, \dots, \mathcal{O}_n | \mathcal{O}_1, \mathcal{O}_2) = \dots = \\ &= Pr(\mathcal{O}_1) \times Pr(\mathcal{O}_2 | \mathcal{O}_1) \times Pr(\mathcal{O}_3 | \mathcal{O}_1, \mathcal{O}_2) \times \dots \times Pr(\mathcal{O}_n | \mathcal{O}_1, \dots, \mathcal{O}_{n-1}) \end{aligned}$$

- BNs simplify the model:
  - $Pr(\mathcal{O}_1, \dots, \mathcal{O}_n) = Pr(\mathcal{O}_1 | parents(\mathcal{O}_1)) \times \dots \times Pr(\mathcal{O}_n | parents(\mathcal{O}_n))$
  - i.e., the other (possible) dependencies are ignored.

Bayesian networks – semantics

- the previous numeric BN definition implies certain CI relationships
  - each node is CI of its other predecessors in the node ordering given its parents,
- the numeric definition matches the topological meaning of d-separation
  - each node is d-separated from its non-descendants given its parents.



Ultimate Bayesian networks

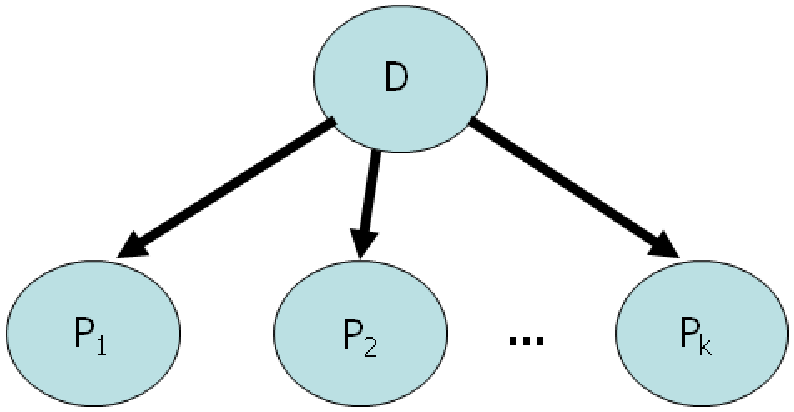
- naïve inference assuming
  - A) variable independence, then empty graph, no edges,
    - no relationship among variables, they are completely independent,
    - $Pr(\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_n) = Pr(\mathcal{O}_1) \times Pr(\mathcal{O}_2) \times \cdots \times Pr(\mathcal{O}_n)$
    - uses marginal probs only – linear complexity in the number of variables,
  - B) CI of variables given diagnosis, $n-1$ of edges only,
    - used in classification, see the next slide,
- fully connected graph assuming direct dependence of all variables
  - no assumptions, same size/complexity as the full joint distribution model,
  - the direction of edges and consequent topological sort of variables selects one of the possible joint probability factorizations,
- reasonable models lie in between
  - sparse enough to be efficient,
  - complex enough to capture the true dependencies.

Naïve Bayes classifier

- a special case of Bayesian network
  - based on purely diagnostic reasoning,
  - assumes CI variables $\mathcal{O}_1, \ldots, \mathcal{O}_k$ given the diagnosis D,
  - the target variable is determined in advance.

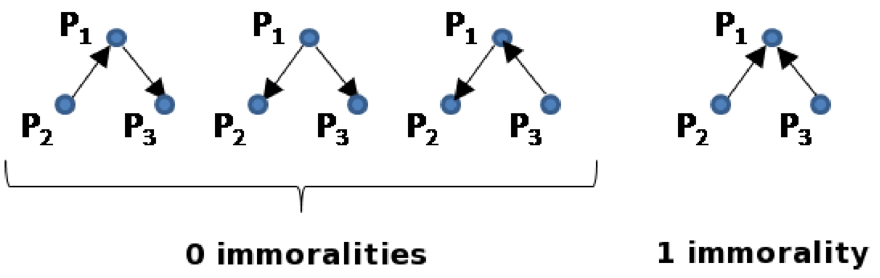
$$Pr(D|\mathcal{O}_1, \ldots \mathcal{O}_k) = \frac{Pr(\mathcal{O}_1, \ldots \mathcal{O}_k|D) \times Pr(D)}{Pr(\mathcal{O}_1, \ldots \mathcal{O}_k)}$$

$$Pr(\mathcal{O}_1, \ldots \mathcal{O}_k|D) = Pr(\mathcal{O}_1|D) \times Pr(\mathcal{O}_2|D) \times \cdots \times Pr(\mathcal{O}_k|D)$$



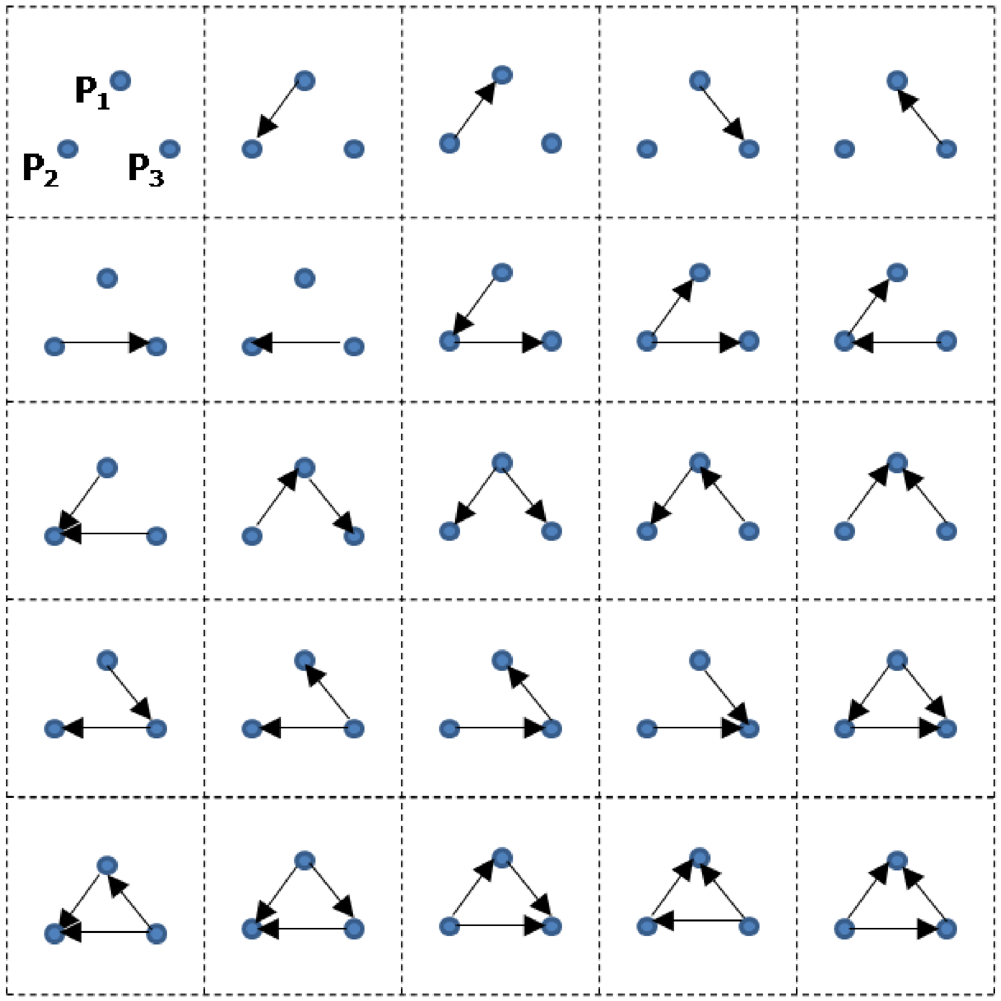
Markov equivalence classes

- DAG classes that define identical CI relationships
  - represent identical joint distribution,
- **Markov equivalence** class is made by directed acyclic graphs which
  - have the identical skeleton
    - $*$ fully match when edge directions removed,
  - contain the same set of **immoralities**
    - $*$ 3 node subgraphs such that: $X \rightarrow Z$ and $Y \rightarrow Z$, no XY arc,
    - $*$ ie. the same sets of uncoupled parents (without an edge between them),
- indistinguishable graphs when learning from data,
- ex.: 2 distinct equivalence classes (first $\mathcal{O}_2 \perp\!\!\!\perp \mathcal{O}_3|\mathcal{O}_1$, second $\mathcal{O}_2 \perp\!\!\!\perp \mathcal{O}_3|\emptyset$)



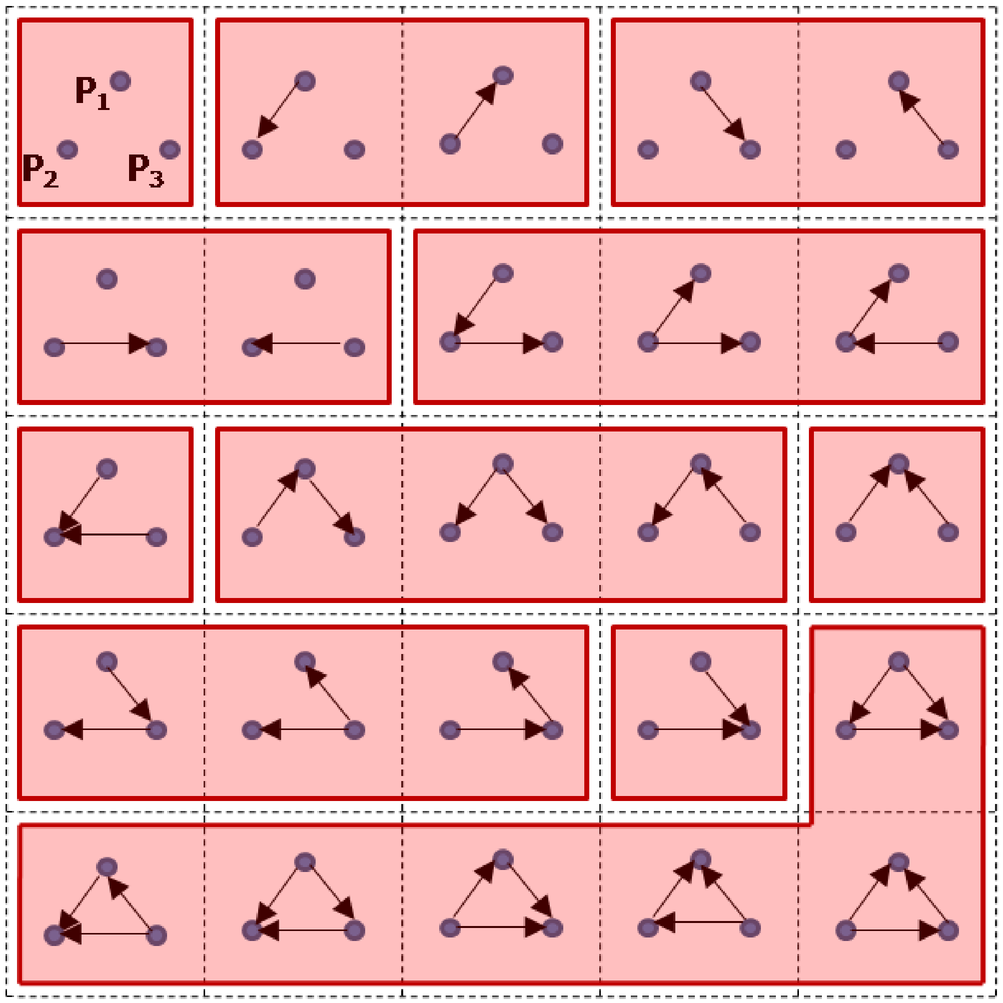
Markov equivalence classes

- let us consider all 25 directed acyclic graphs with 3 **labeled** nodes



Markov equivalence classes

- they make 11 Markov equivalence classes altogether

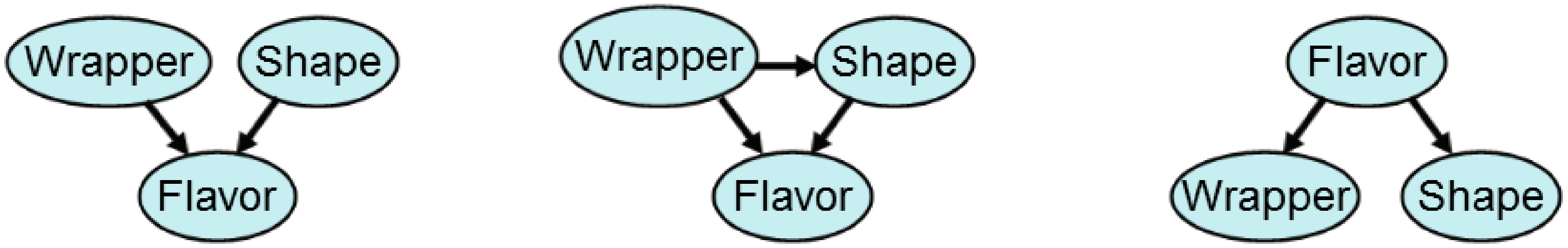


Characteristics of qualitative model

- **correctness**
  - $Pr(\mathcal{O}_{j+1}|\mathcal{O}_1, \ldots \mathcal{O}_j) = Pr(\mathcal{O}_{j+1}|parents(\mathcal{O}_{j+1}))$ holds in reality,
  - each network node is CI of its ancestor given its parents,
- **efficiency**
  - there are no redundant edges,
  - actual CI relations described by the minimum number of edges,
    - extra edges do not violate correctness,
    - but slow down computations and make the model difficult to read,
- **causality**
  - edge directions agree with actual cause-effect relationships,
- consequences
  - graphs from the same M. class have the same correctness and efficiency,
  - fully connected DAG always correct, but very likely inefficient.

Characteristics of qualitative model – example

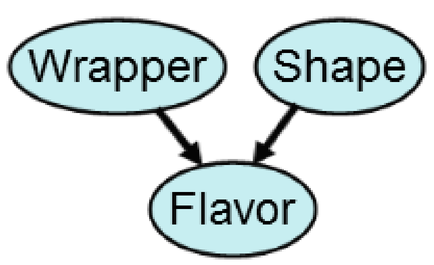
- **The Surprise Candy Company** makes candy in two flavors: 70% are strawberry flavor and 30% are anchovy flavor. Each new piece of candy starts out with a round shape; as it moves along the production line, a machine randomly selects a certain percentage to be trimmed into a square; then, each piece is wrapped in a wrapper whose color is chosen randomly to be red or brown. 80% of the strawberry candies are round and 80% have a red wrapper, while 90% of the anchovy candies are square and 90% have a brown wrapper. All candies are sold individually in sealed, identical, black boxes.



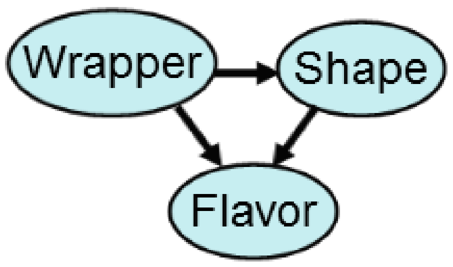
Russell, Norvig: Artificial Intelligence: A Modern Approach.

Characteristics of qualitative model – example

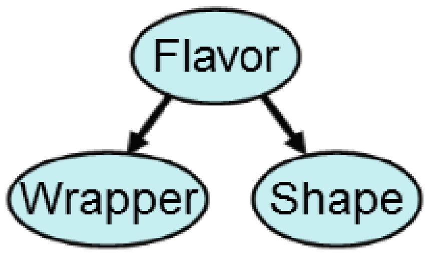
- **The Surprise Candy Company**



incorrect



correct, inefficient



correct, efficient, causal

- $Wrap \perp\!\!\!\perp Shape | \emptyset$
- contradicts reality.

- no indep. relationship,
- thus no unrealistic one.

- $Wrap \perp\!\!\!\perp Shape | Flavor$
- complies with reality.

Summary – BN structure

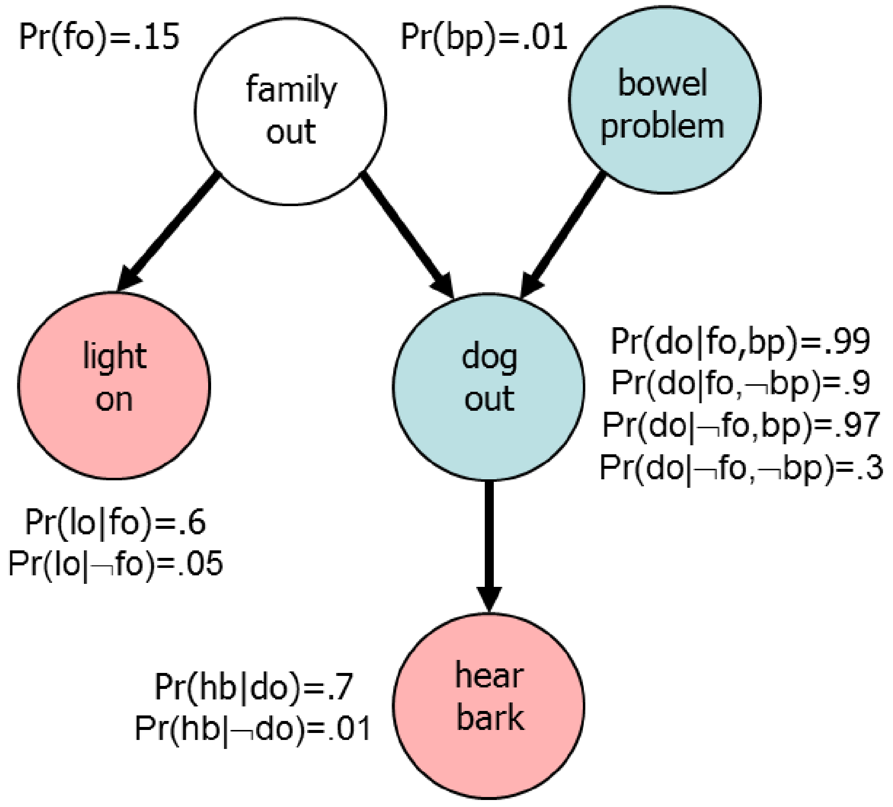
- probability
  - a rigorous tool for uncertainty modeling,
  - each **atomic** event is described by the joint probability distribution,
  - queries answered by enumeration of atomic events
    - (summing, sometimes with final division),
- needs to be simplified in non-trivial domains
  - reason: curse of dimensionality,
  - solution: independence and conditional independence
  - tool: GPM = graph (quality) + conditional probability tables/functions (quantity).

Bayesian networks – fundamental tasks

- inference – reasoning, deduction
  - from observed events assumes on probability of other events,
  - observations (**E** – a set of evidence variables, **e** – a particular event),
  - target variables (**Q** – a set of query variables, Q – a particular query variable),
  - $Pr(\mathbf{Q}|\mathbf{e})$, resp. $Pr(Q \in \mathbf{Q}|\mathbf{e})$ to be found,
  - network is known (both graph and CPTs),
- learning network parameters from data
  - network structure (graph) is given,
  - "only" quantitative parameters (CPTs) to be optimized,
- learning network structure from data
  - propose an optimal network structure
    - which edges of the fully connected graph shall be employed?,
  - too many arcs $\rightarrow$ complicated model,
  - too few arcs $\rightarrow$ inaccurate model.

Probabilistic network – inference by enumeration

- Let us observe the following events:
  - no barking heard,
  - the door light is on.
- What is the prob of family being out?
  - searching for $Pr(fo|lo, \neg hb)$.
- Will observation influence the target event?
  - light on supports departure hypothesis,
  - no barking suggests dog inside,
  - the dog is in house when it is
    - rather healthy,
    - the family is at home.



Probabilistic network – inference by enumeration

- inference by enumeration
  - conditional probs calculated by summing the elements of joint probability table,
- how to find the joint probabilities (the table is not given)?
  - BN definition suggests:

    $$Pr(FO, BP, DO, LO, HB) =$$
    $$= Pr(FO)Pr(BP)Pr(DO|FO, BP)Pr(LO|FO)Pr(HB|DO)$$
- answer to the question?
  - conditional probability definition suggests:

    $Pr(fo|lo, \neg hb) = \frac{Pr(fo,lo,\neg hb)}{Pr(lo,\neg hb)}$
  - by joint prob marginalization we get:

    $Pr(fo, lo, \neg hb) = \sum_{BP,DO} Pr(fo, BP, DO, lo, \neg hb)$

    $Pr(fo, lo, \neg hb) = Pr(fo, bp, do, lo, \neg hb) + Pr(fo, bp, \neg do, lo, \neg hb)+$
    $+Pr(fo, \neg bp, do, lo, \neg hb)+Pr(fo, \neg bp, \neg do, lo, \neg hb) = .15\times .01\times .99\times .6\times .3+.15\times$
    $.01 \times .01 \times .6 \times .99 + .15 \times .99 \times .9 \times .6 \times .3 + .15 \times .99 \times .1 \times .6 \times .99 = .033$

    $Pr(lo, \neg hb) = Pr(fo, lo, \neg hb) + Pr(\neg fo, lo, \neg hb) = .066$

Probabilistic network – inference by enumeration

- after substitution:
  $Pr(fo|lo,\neg hb) = \frac{Pr(fo,lo,\neg hb)}{Pr(lo,\neg hb)} = \frac{.033}{.066} = 0.5$
- posterior probability $Pr(fo|lo,\neg hb)$ higher than prior $Pr(fo) = 0.15$.

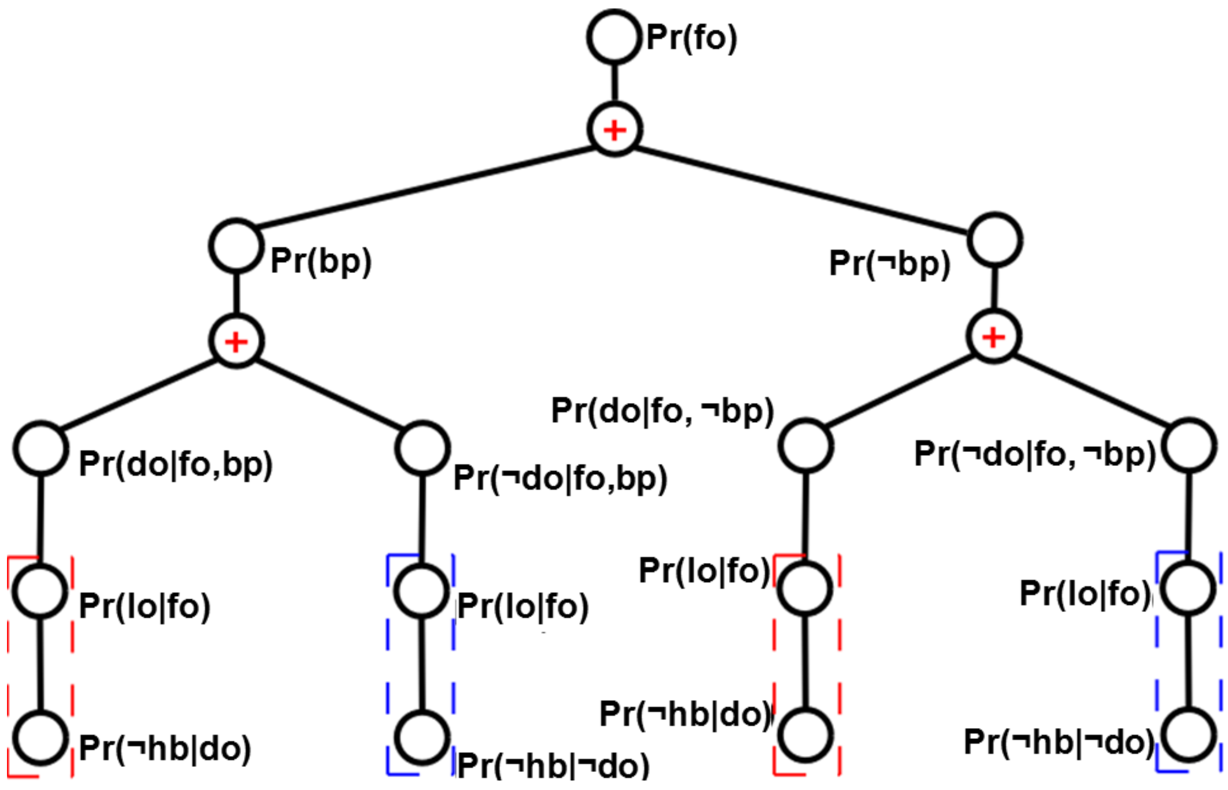
■ can we assume on complexity?

- instead of $2^5 - 1$=31 probs (either conditional or joint) 10 needed only,
- however, joint probs enumerated to answer the query
  - ∗ inference remains a NP-hard problem,
- moving summations left-to-right makes a difference, but not a principal one
  - ∗ see the evaluation tree on the next slide,

$$Pr(fo,lo,\neg hb) = \sum_{BP,DO} Pr(fo,BP,DO,lo,\neg hb) =$$
$$= Pr(fo)\sum_{BP} Pr(BP)\sum_{DO} Pr(DO|fo,BP)Pr(lo|fo)Pr(\neg hb|DO)$$

- inference by enumeration is an intelligible, but inefficient procedure,
- solution: minimize recomputations, special network types or approximate inference.

Inference by enumeration – evaluation tree



- Complexity: time $\mathcal{O}(n2^d)$, memory $\mathcal{O}(n)$
  - n . . . the number of variables, e . . . the number of evidence variables, d=n-e,
- resource of inefficiency: recomputations ($Pr(lo|fo) \times Pr(\neg hb|DO)$ for each BP value)
  - variable ordering makes a difference – $Pr(lo|fo)$ shall be moved forward.

Inference by enumeration – straightforward improvements

- variable elimination procedure
  1. pre-computes factors to remove the inefficiency shown in the previous slide
     - factors serve for recycling the earlier computed intermediate results,
     - some variables are eliminated by summing them out,

       $\sum_P f_1 \times \dots \times f_k = f_1 \times \dots \times f_i \times \sum_P f_{i+1} \times \dots \times f_k = f_1 \times \dots \times f_i \times f_{\bar{P}}$,
       assumes that $f_1, \dots, f_i$ do not depend on P,

       when multiplying factors, the pointwise product is computed
       $f_1(x_1, ..., x_j, y_1, ..., y_k) \times f_2(y_1, ..., y_k, z_1, ..., z_l) = f(x_1, ..., x_j, y_1, ..., y_k, z_1, ..., z_l)$

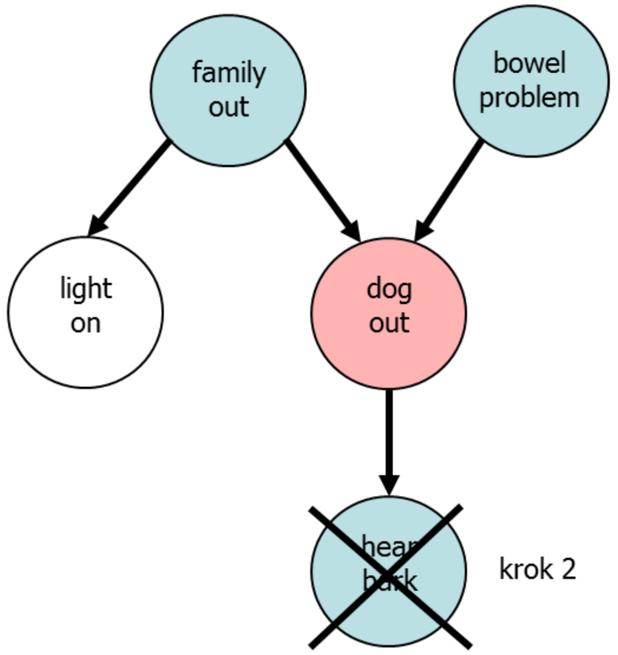
       eventual enumeration over $\mathcal{O}_1$ variable, which takes all (two) possible values
       $f_{\bar{\mathcal{O}}_1}(\mathcal{O}_2, \dots, \mathcal{O}_k) = \sum_{\mathcal{O}_1} f_1(\mathcal{O}_1, \mathcal{O}_2, \dots, \mathcal{O}_k)$,

     - execution efficiency is influenced by the variable ordering when computing,
       (finding the best order is NP-hard problem, can be optimized heuristically too),

Inference by enumeration – straightforward improvements

- variable elimination procedure

  2. does not consider variables irrelevant to the query
     - all the leaves that are neither query nor evidence variable,
     - the rule can be applied recursively.



- example: $Pr(lo|do)$
  - what is prob that the door light is shining if the dog is in the garden?
  - we will enumerate $Pr(LO, do)$, since:

$$Pr(lo|do) = \frac{Pr(lo,do)}{Pr(do)} = \frac{Pr(lo,do)}{Pr(lo,do)+Pr(\neg lo,do)}$$

Inference by enumeration – variable elimination

- HB is irrelevant to the particular query, why?

$\sum_{HB} Pr(HB|do) = 1$

$$Pr(LO, do) = \sum_{FO,BP,HB} Pr(FO)Pr(BP)Pr(do|FO, BP)Pr(LO|FO)Pr(HB|do) =$$
$$= \sum_{FO} Pr(FO)Pr(LO|FO) \sum_{BP} Pr(BP)Pr(do|FO, BP) \sum_{HB} Pr(HB|do)$$

- after omitting the last invariant, **factorization** may take place

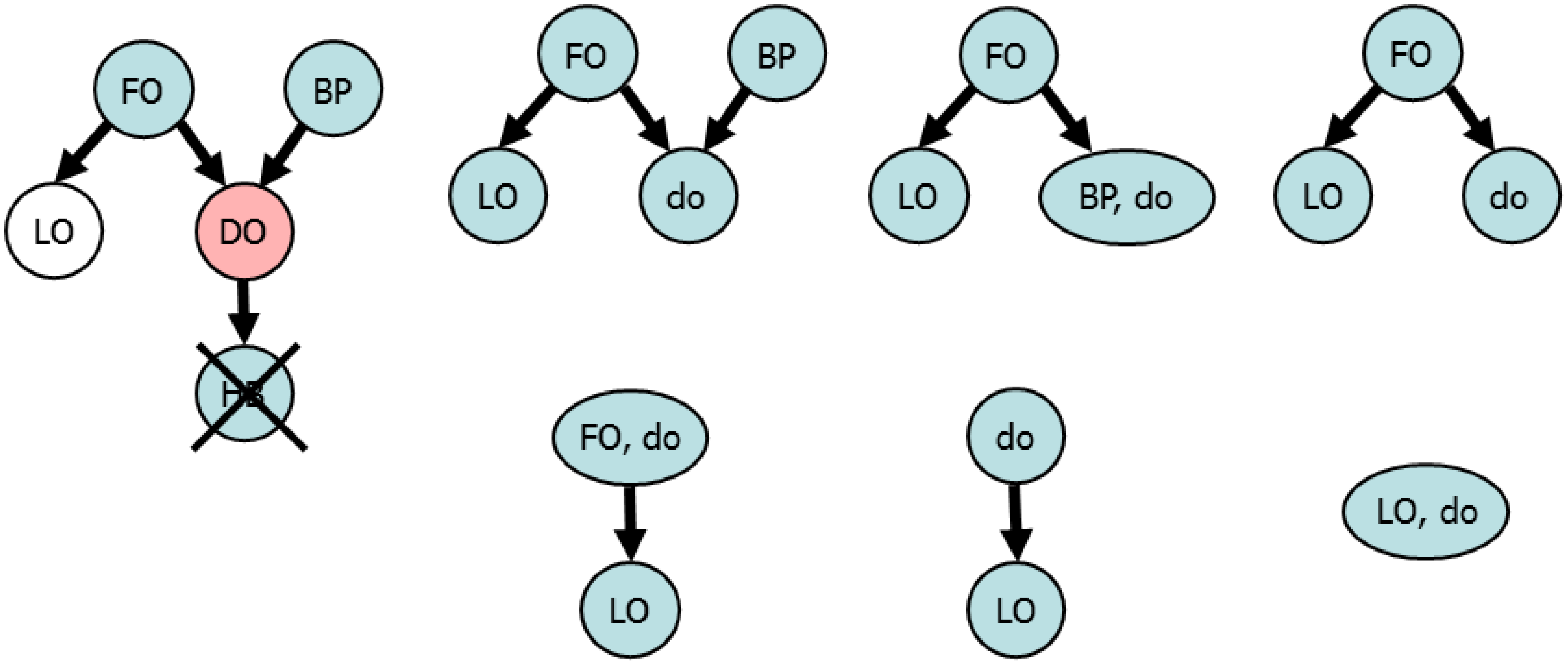
$$Pr(LO, do) = \sum_{FO} Pr(FO)Pr(LO|FO) \sum_{BP} Pr(BP)Pr(do|FO, BP) =$$
$$= \sum_{FO} Pr(FO)Pr(LO|FO) f_{\overline{BP}}(do|FO) = \sum_{FO} f_{\overline{BP},do}(FO)Pr(LO|FO) =$$
$$= f_{\overline{FO},\overline{BP},do}(LO)$$

- having the last factor (a table of two elements), one can read

$$Pr(lo|do) = \frac{f_{\overline{FO},\overline{BP},do}(lo)}{f_{\overline{FO},\overline{BP},do}(lo)+f_{\overline{FO},\overline{BP},do}(\neg lo)} = \frac{0.0941}{0.0941+0.3017} = \frac{0.0941}{0.3958} = 0.24$$

Variable elimination – factor computations

- factors are enumerated from CPTs by summing out variables
  - sum out BP: $CPT(DO)$ & $CPT(BP) \rightarrow f_{\overline{BP}}(do|FO)$
  - reformulate into: $CPT(FO)$ & $f_{\overline{BP}}(do|FO) \rightarrow f_{\overline{BP},do}(FO)$
  - sum out FO: $f_{\overline{BP},do}(FO)$ & $CPT(LO) \rightarrow f_{\overline{FO},\overline{BP},do}(LO)$



Variable elimination – factor computations

BP	$Pr(BP)$
T	0.01
F	0.99

$\times$

FO	BP	$Pr(do\|FO, BP)$
T	T	0.99
T	F	0.9
F	T	0.97
F	F	0.3

$\Rightarrow$

FO	$f_{\overline{BP}}(do\|FO)$
T	$0.9009 = 0.99 \times 0.01 + 0.9 \times 0.99$
F	$0.3067 = 0.97 \times 0.01 + 0.99 \times 0.3$

FO	$Pr(FO)$
T	0.15
F	0.85

$\times$

FO	$f_{\overline{BP}}(do\|FO)$
T	0.9009
F	0.3067

$\Rightarrow$

FO	$f_{\overline{BP},do}(FO)$
T	$0.1351 = 0.15 \times 0.9009$
F	$0.2607 = 0.85 \times 0.3067$

FO	$f_{\overline{BP},do}(FO)$
T	0.1351
F	0.2607

$\times$

LO	FO	$Pr(LO\|FO)$
T	T	0.6
T	F	0.05
F	T	0.4
F	F	0.95

$\Rightarrow$

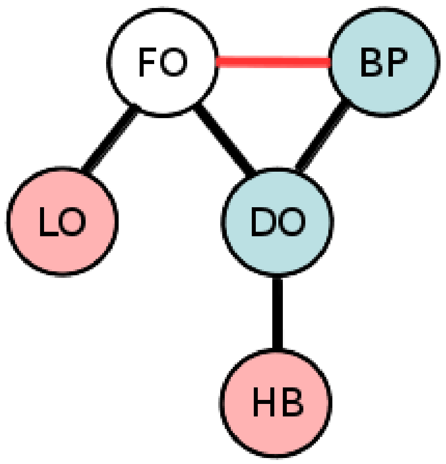
LO	$f_{\overline{FO},\overline{BP},do}(LO)$
T	$0.0941 = 0.1351 \times 0.6 + 0.2607 \times 0.05$
F	$0.3017 = 0.1351 \times 0.4 + 0.2607 \times 0.95$

Inference by enumeration – comparison of the number of operations

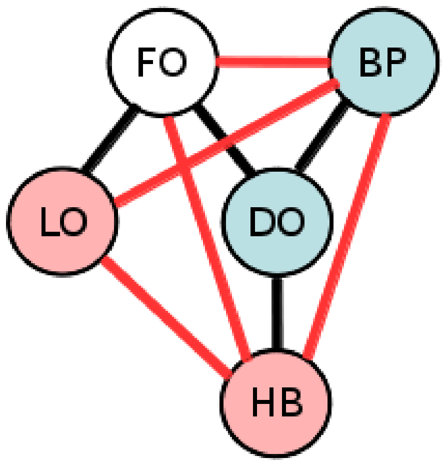
- let us take the last example
  - namely the total number of sums and products in $Pr(LO, do)$,
  - (the final $Pr(lo|do)$ enumeration is identical for all procedures),
- naïve enumeration, no evaluation tree
  - 4 products (5 vars) $\times 2^4$ ($\#$ atomic events on unevidenced variables) $+\ 2^4 - 2$ sums,
  - in total 78 operations,
- using evaluation tree and a proper reordering of variables
  - takes the ordering $Pr(LO, do) = \sum_{FO} Pr(FO)Pr(LO|FO) \sum_{BP} Pr(BP)Pr(do|FO, BP) \sum_{HB} Pr(HB|do)$
  - in total 38 operations,
- with variable elimination on top of that
  - in total 14 operations (6 in Tab1, 2 in Tab2, 6 in Tab3).

Variable elimination – efficiency in general

- Given by the network structure and the variable ordering
  - exponential in the size of the largest clique in the **induced** graph,
  - somewhere between linear and NP-hard,
- induced graph
  - undirected graph, the edge exists if two variables both appear in some intermediate factor induced by the given variable ordering,



$HB \prec BP \prec LO \prec FO \prec DO$



$DO \prec FO \prec LO \prec HB \prec BP$

Variable elimination – variable ordering

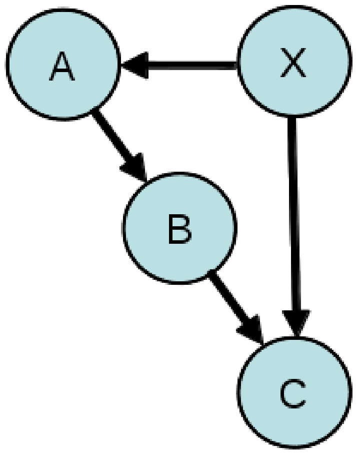
- minimize the number of **fill edges** in induced graph
  - edges introduced in the elimination step,
- NP-hard problem in general
  - greedy local methods often find near-optimal solution,
  - min-fill heuristic
    - vertex cost is the number of edges added to the graph due to its elimination,
  - always take the node that minimizes the heuristic, no backtrack.
- Step 1:

$$Pr(FO, \ldots, HB) = f_{FO}(FO) f_{BP}(BP) f_{DO}(DO, FO, BP) f_{LO}(LO, FO) f_{HB}(HB, DO)$$

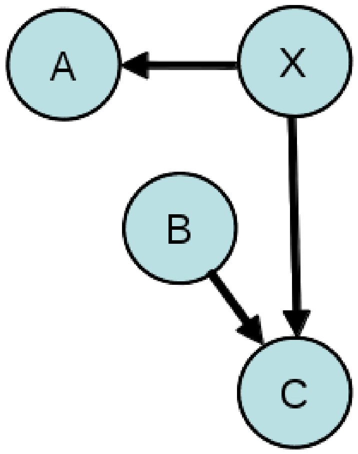
var	intermediate factor	min-fill
FO	$f_{FO}(FO) f_{DO}(DO, FO, BP) f_{LO}(LO, FO)$	3
BP	$f_{BP}(BP) f_{DO}(DO, FO, BP)$	1
DO	$f_{DO}(DO, FO, BP) f_{HB}(HB, DO)$	3
LO	$f_{LO}(LO, FO)$	0
HB	$f_{HB}(HB, DO)$	0

Semantics of factors

- Factors
  - multidimensional arrays (the same as CPTs),
  - often correspond to marginal or conditional probabilities,
  - initialized with CPTs,
  - some intermediate factors differ from any probability in the network
    - eliminate X from the left network,
    - the resulting factor does not agree with any prob in the left network,
    - it gives a conditional prob in the right network.



$f(A, B, C) = \sum_X Pr(X)Pr(A|X)Pr(C|B, X)$



$Pr(A, C|B)$

Approximate inference by stochastic sampling

- a general **Monte-Carlo** method, samples from the joint prob distribution,
- estimates the target conditional probability (query) from a sample set,
- the joint prob distribution is not explicitly given, its factorization is available only (network),

- the most straightforward is direct **forward sampling**
  1. topologically sort the network nodes
     – for every edge it holds that parent comes before its children in the ordering,
  2. instantiate variables along the topological ordering
     – take $Pr(\mathcal{O}_j|parents(\mathcal{O}_j))$, randomly sample $\mathcal{O}_j$,
  3. repeat step 2 for all the samples (the sample size M is given a priori),
- from samples to probabilities?
  – $Pr(q|\mathbf{e}) \approx \frac{N(q,\mathbf{e})}{N(\mathbf{e})}$
  – samples that contradict evidence not used,
  – forward sampling gets inefficient if $Pr(\mathbf{e})$ is small.

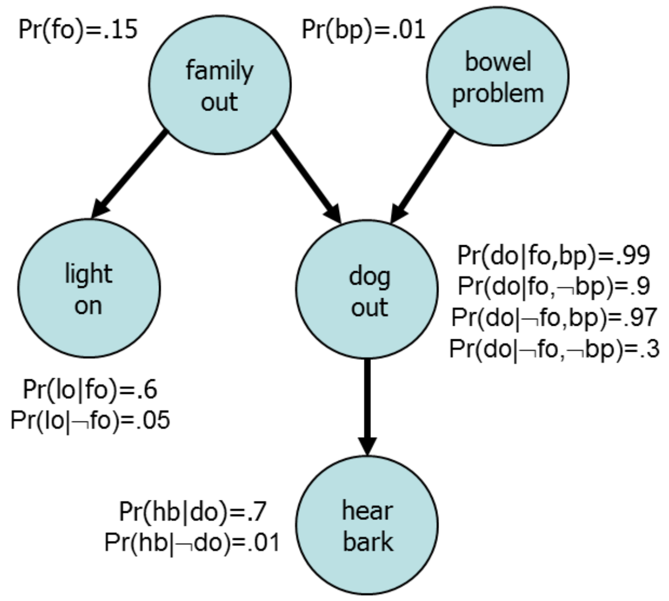
Improved stochastic sampling

- **rejection sampling**
  - rejects partially generated samples as soon as they violate the evidence event $\mathbf{e}$,
  - sample generation may stop early $\rightarrow$ slight improvement,
- **likelihood weighting**
  - avoids necessity to reject samples,
  - the values of $\mathbf{E}$ fixed, the rest of variables sampled only,
  - however, not all events are equally probable, samples must be weighted,
  - the weight equals to the likelihood of the event given the evidence,
- **Gibbs sampling**
  - the previous methods are importance sampling,
  - GS is a **Markov chain** method – the next state depends purely on the current state,
    - state = sample, generates dependent samples!
    - as it is a **Monte-Carlo** method as well $\rightarrow$ MCMC,
  - efficient sampling method namely when some of BN variable states are known
    - it again samples nonevidence variables only, the samples never rejected.

Rejection sampling – example

- FAMILY example, estimate $Pr(fo|lo, \neg hb)$
  1. topologically sort the network nodes
     - e.g., $\langle FO, BP, LO, DO, HB\rangle$ (or $\langle BP, FO, DO, HB, LO\rangle$, etc.)
  2. instantiate variables along the topological ordering
     - $Pr(FO) \rightarrow \neg fo$, $Pr(BP) \rightarrow \neg bp$,
       $Pr(LO|\neg fo) \rightarrow lo$, $Pr(DO|\neg fo, \neg bp) \rightarrow \neg do$, $Pr(HB|\neg do) \rightarrow \neg hb$
     - sample agrees with the evidence $\mathbf{e} = lo \wedge \neg hb$, no rejection needed,
  3. generate 1000 samples, repeat step 2,

- let $N(fo, lo, \neg hb)$ is 491 (the number of samples with the given values of three variables under consideration),
- in rejection sampling $N(\mathbf{e})$ necessarily equals M,
  - $Pr(fo|lo, \neg hb) \approx \frac{N(q,\mathbf{e})}{N(\mathbf{e})} = \frac{491}{1000} = 0.491$



Likelihood weighting – details

- sampling process:

  $\forall$ samples $p^m = \{\mathcal{O}_1 = o_1^m, \ldots \mathcal{O}_n = o_n^m\}$, $m \in \{1, \ldots, M\}$

  1. $w^m \leftarrow 1$ (initialize the sample weight)
  2. $\forall j \in \{1, \ldots, n\}$ (instantiate variables along the topological ordering)
     – if $\mathcal{O}_j \in \mathbf{E}$ then take o_j^m from $\mathbf{e}$ and $w^m \leftarrow w^m \times Pr(\mathcal{O}_j | parents(\mathcal{O}_j))$,
     – otherwise randomly sample o_j^m from $Pr(\mathcal{O}_j | parents(\mathcal{O}_j))$,

- from samples to probabilities?
  – evidence holds in all samples (by definition),
  – weighted averaging is applied to find $Pr(Q =? \mathcal{O}_i | \mathbf{e})$

$$Pr(o_i|\mathbf{e}) \approx \frac{\sum_{m=1}^{M} w^m \delta(o_i^m, o_i)}{\sum_{m=1}^{M} w^m} \quad \delta(i,j) = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases}$$

- nevertheless, samples may have very low weights
  – turns out inefficient in large networks with evidences occuring late in the ordering.

Likelihood weighting – example

- let us approximate $Pr(fo|lo, \neg hb)$ (its exact value computed earlier is 0.5),

	p^1	p^2	p^3	...
FO	F	F	T	
BP	F	F	F	
LO	**T**	**T**	**T**	
DO	F	T	T	
HB	**F**	**F**	**F**	
w	.0495	.015	.18	

FO^1: $Pr(fo) = .15 \rightarrow \neg fo$ sampled
BP^1: $Pr(bp) = .01 \rightarrow \neg bp$ sampled
LO^1: evidence $\rightarrow lo \wedge w^1 = Pr(lo|\neg fo) = .05$
DO^1: $Pr(do|\neg fo, \neg bp) = .3 \rightarrow \neg do$ sampled
HB^1: evidence $\rightarrow \neg hb \wedge w^1 = .05 \times Pr(\neg hb|\neg do) = .0495$

FO^2: $Pr(fo) = .15 \rightarrow \neg fo$ sampled
BP^2: $Pr(bp) = .01 \rightarrow \neg bp$ sampled
LO^2: evidence $\rightarrow lo \wedge w^1 = Pr(lo|\neg fo) = .05$
DO^2: $Pr(do|\neg fo, \neg bp) = .3 \rightarrow do$ sampled
HB^2: evidence $\rightarrow \neg hb \wedge w^2 = .05 \times Pr(\neg hb|do) = .015$

- a very rough estimate having 3 samples only

$$Pr(fo|lo, \neg hb) \approx \frac{.18}{.0495 + .015 + .18} = .74$$

Gibbs sampling

- sampling process:

  $\forall$ samples $o^m = \{\mathcal{O}_1 = o_1^m, \ldots \mathcal{O}_n = o_n^m\}$, $m \in \{1, \ldots, M\}$

  1. fix states of all observed variables from $\mathbf{E}$ (in all samples),
  2. the other variables initialized in o^0 randomly,
  3. generate o^m from o^{m-1} ($\forall \mathcal{O}_i \notin E$)
     - $o_1^m \leftarrow Pr(\mathcal{O}_1 | o_2^{m-1}, \ldots, o_n^{m-1})$,
     - $o_2^m \leftarrow Pr(\mathcal{O}_2 | o_1^m, o_3^{m-1}, \ldots, o_n^{m-1})$,
     - $\ldots$,
     - $o_n^m \leftarrow Pr(\mathcal{O}_n | o_1^m, \ldots, o_{n-1}^m)$,
  4. repeat step 3 for $m \in \{1, \ldots, M\}$,

  ignore samples at the beginning (burn-in period).

Gibbs sampling

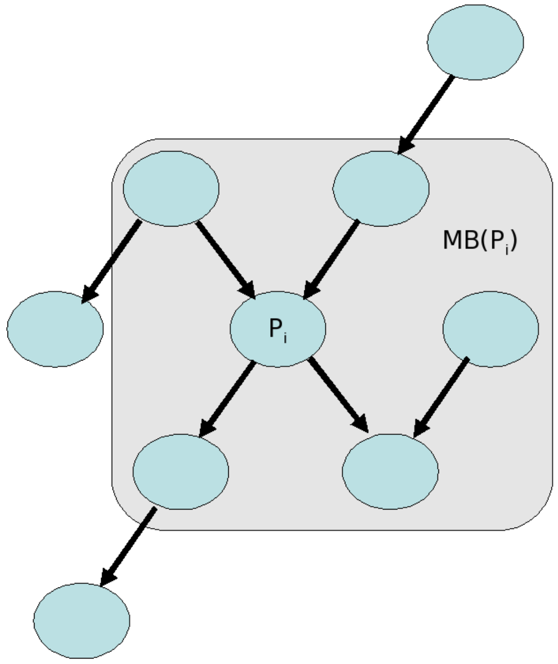
- probs $Pr(\mathcal{O}_i|\mathcal{O}_1,\dots\mathcal{O}_{i-1}\mathcal{O}_{i+1},\dots\mathcal{O}_n) = Pr(\mathcal{O}_i|P\setminus\mathcal{O}_i)$ not explicitly given . . .
  - to enumerate them, only their BN neighborhood needs to be known

$$Pr(\mathcal{O}_i|\mathcal{O}\setminus\mathcal{O}_i) \propto Pr(\mathcal{O}_i|parents(\mathcal{O}_i)) \prod_{\forall\mathcal{O}_j,\mathcal{O}_i\in parents(\mathcal{O}_j)} Pr(\mathcal{O}_j|parents(\mathcal{O}_j))$$

  - the neighborhood is called **Markov blanket** (MB),
  - MB covers the node, its parents, its children and their parents,
  - $MB(\mathcal{O}_i)$ is the minimum set of nodes that d-separates $\mathcal{O}_i$ from the rest of the network.

- from samples to probabilities?
  - evidence holds in all samples (by definition),
  - averaging $\forall m$ is applied to find $Pr(Q|\mathbf{e})$

$$Pr(o_i|\mathbf{e}) \approx \frac{\sum_{m=1}^{M}\delta(o_i^m,o_i)}{M} \quad \delta(i,j) = \begin{cases} 1 & \text{for } i=j \\ 0 & \text{for } i\neq j \end{cases}$$



Gibbs sampling – example

- let us approximate $Pr(fo|lo, \neg hb)$ (its exact value computed earlier is 0.5),

	o^0	o^1	o^2	$\dots$
FO	T	**F**	**F**	
BP	T	**F**	**F**	
LO	**T**	**T**	**T**	
DO	F	F	F	
HB	**F**	**F**	**F**	

o^0: random init of unevidenced variables

FO^1: $Pr^*(fo) \propto Pr(fo) \times Pr(lo|fo) \times Pr(\neg do|fo, bp)$

$Pr^*(\neg fo) \propto Pr(\neg fo) \times Pr(lo|\neg fo) \times Pr(\neg do|\neg fo, bp)$

$Pr^*(fo) \propto .15 \times .6 \times .01 = 9 \times 10^{-4} \rightarrow \times \alpha^1_{FO} = .41$

$Pr^*(\neg fo) \propto .85 \times .05 \times .03 = 1.275 \times 10^{-3} \rightarrow \times \alpha^1_{FO} = .59$

$\alpha^1_{FO} = \frac{1}{Pr^*(fo)+Pr^*(\neg fo)} = 460$

BP^1: $Pr^*(bp) \propto Pr(bp) \times Pr(\neg do|\neg fo, bp) = .01 \times .03 = .0003$

$Pr^*(\neg bp) \propto Pr(\neg bp) \times Pr(\neg do|\neg fo, \neg bp) = .99 \times .7 = 0.693$

$\alpha^1_{BP} = \frac{1}{Pr^*(bp)+Pr^*(\neg bp)} = 1.44 \rightarrow Pr^*(bp) = 4 \times 10^{-4}$

DO^1: by analogy, $|MB(DO)| = 5$

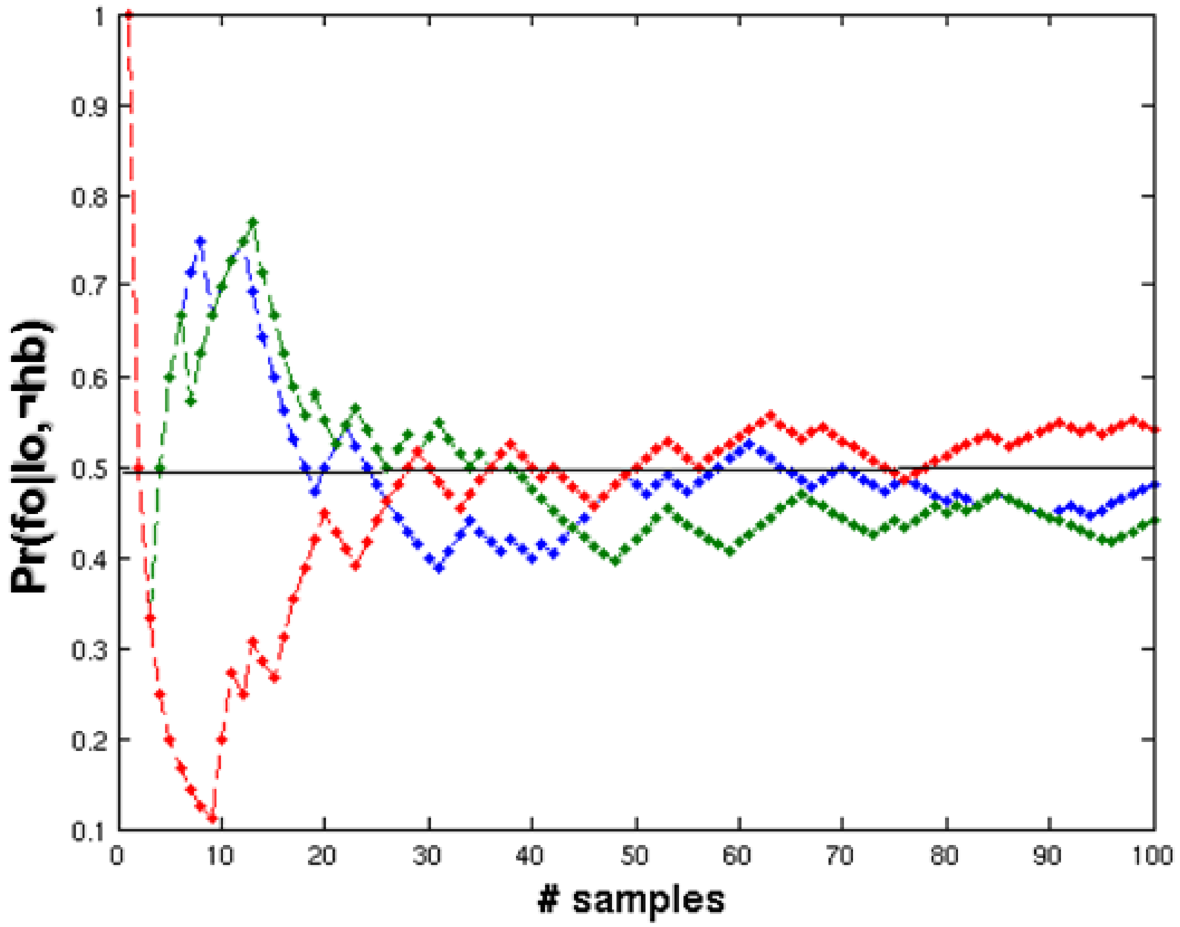
FO^2: BP value was switched, substitution is $Pr(DO|FO, \neg bp)$

$Pr^*(fo) = .21 \quad Pr^*(\neg fo) = .79$

BP^2: the same probs as is sample 1

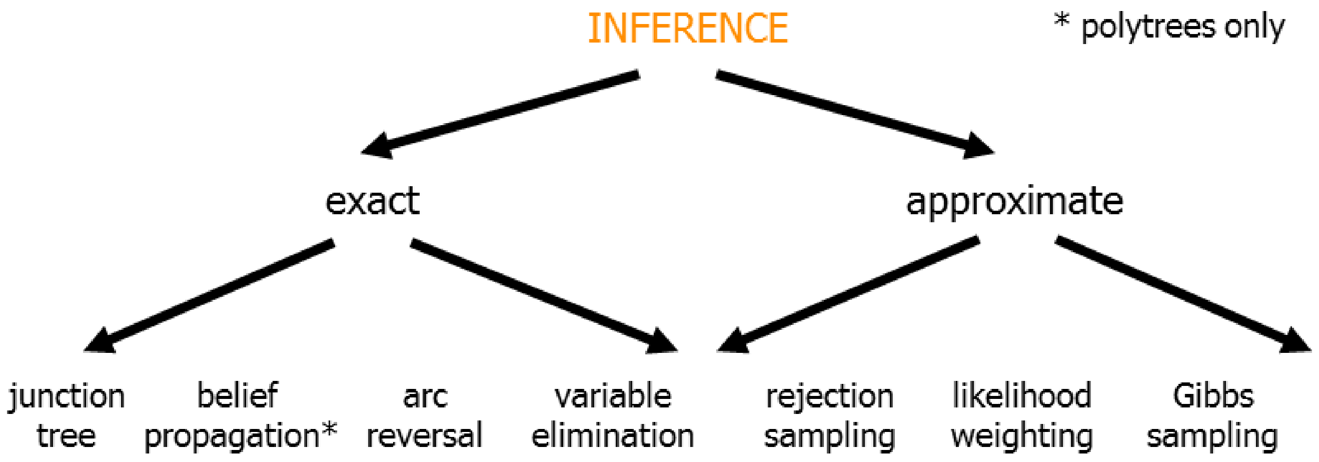
Gibbs sampling – example

- BN Matlab Toolbox, aproximation of $Pr(fo|lo, \neg hb)$,
- gibbs_sampling_inf_engine, three independent runs with 100 samples.



Summary – inference

- independence and conditional independence remarkably simplify prob model
  - still, BN inference remains generally **NP-hard** wrt the number of nodes,
  - inference complexity grows with the number of network edges
    - ∗ naïve Bayes model – linear complexity,
    - ∗ exponential in the size of maximal clique of induced graph,
  - inference complexity can be reduced by constraining model structure
    - ∗ special network types (singly connected), e.g. trees – one parent only,
  - inference time can be shorten when exact answer not required
    - ∗ approximate inference, typically (but not only) stochastic sampling.



Learning Bayesian networks from data

- Motivation for learning from data
  - knowledge is hard to obtain $\times$ data of sufficient size often at hand,
- structure of training data
  - **frequency table** is commonly sufficient,
  - incomplete data make learning harder,
- **parameter learning**
  - easier (sub)task,
  - MLE algorithm (+ EM for incomplete data),
  - data quantity – demonstration of requirements,
- **structure learning**
  - more difficult task,
  - structure selection criteria? likelihood, MAP score, BIC,
  - naïve approach, K2 and MCMC algorithms,
  - illustrative examples.

Learning Bayesian networks from data

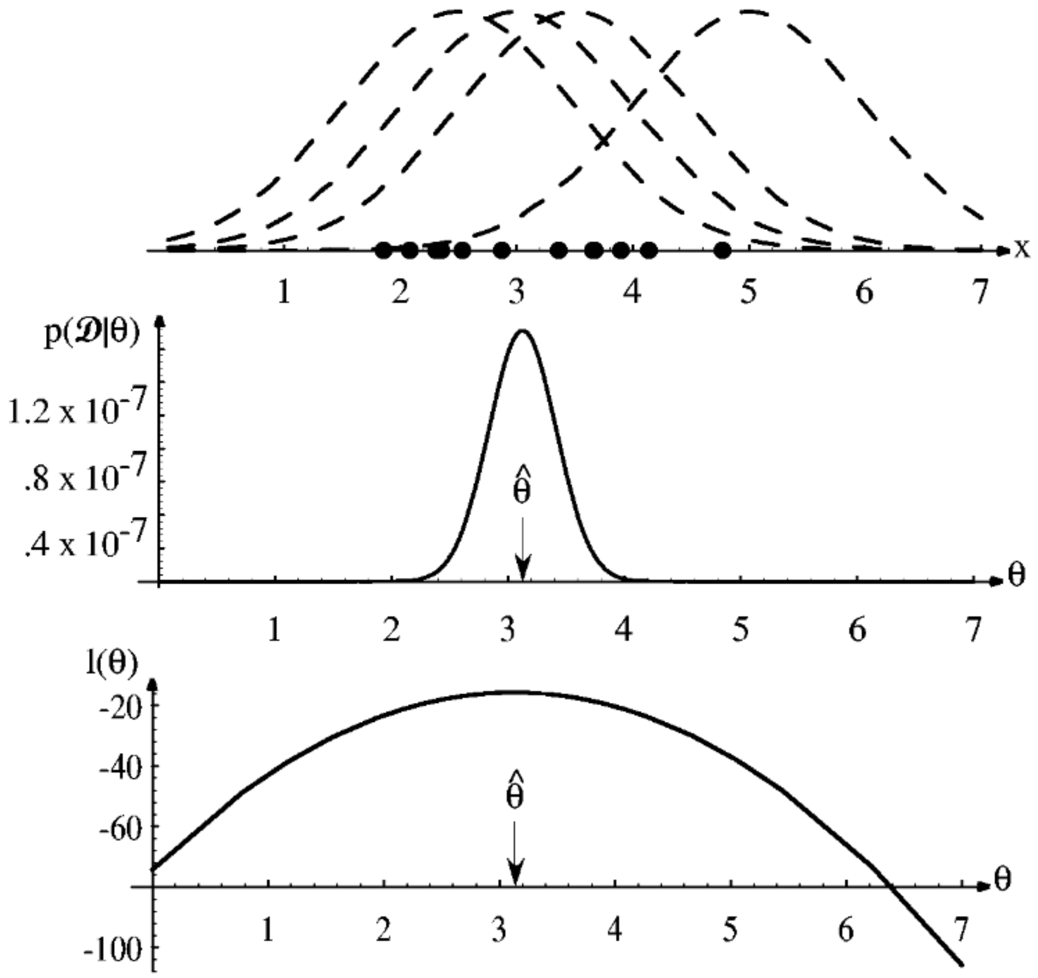
- format of training data?
  - sample set D contains M samples = concurrent observations of all the variables,
  - FAMILY example: $d_m = \{FO_m, BP_m, LO_m, DO_m, HB_m\}$, $m = 1 \ldots M$,
  - no missing values concerned yet for simplicity,
- frequency table (hypercube) provides sufficient statistics (representation)
  - gives the number of samples with particular configuration (frequency over sample space),
  - 2^5 entries $N(\{fo, bp, do, lo, hb\})$, ..., $N(\{\neg fo, \neg bp, \neg do, \neg lo, \neg hb\})$,
  - representation close to the joint probability distribution.

$$d_1 = \{fo_1, \neg bo_1, \neg lo_1, do_1, \neg hb_1\}$$
$$d_2 = \{\neg fo_2, \neg bo_2, \neg lo_2, do_2, hb_2\}$$
$$\ldots$$
$$d_M = \{\neg fo_M, bo_M, \neg lo_M, do_M, hb_M\}$$

			fo		¬fo	
			bp	¬bp	bp	¬bp
lo	do	hb	1	56	0	106
		¬hb	0	24	0	45
	¬do	hb	0	0	0	4
		¬hb	1	9	0	349
¬lo	do	hb	0	37	d_M $\boxed{5}$ +1	d_2 $\boxed{71}$ +1
		¬hb	2	d_1 $\boxed{16}$ +1	2	30
	¬do	hb	0	1	0	2
		¬hb	0	6	0	233

Learning Bayesian network parameters from data

- likelihood review: 1D Gaussian mean estimation (variance assumed to be known)



a set of observations (points)
candidate probabilistic models (dashed)

likelihood as a function of the mean
prob of the observations given the model
the mean value $\hat{\theta}$ maximizes likelihood

log likelihood
the same best value $\hat{\theta}$
easier to handle (underflow)

Duda,Hart,Stork: Pattern Classification

Learning Bayesian network parameters from data

- network structure is known, we search for CPTs in the individual nodes,
- maximum likelihood estimate (MLE) of unknown parameters Θ
  - FAMILY example

$$L(\Theta : D) = \prod_{m=1}^{M} Pr(d_m : \Theta) = \prod_{m=1}^{M} Pr(FO_m, BP_m, LO_m, DO_m, HB_m : \Theta) =$$
$$= \prod_{m=1}^{M} Pr(FO_m : \Theta) Pr(BP_m : \Theta) Pr(LO_m | FO_m : \Theta) \dots Pr(HB_m | DO_m : \Theta)$$

  - for general Bayesian network

$$L(\Theta : D) = \prod_{m=1}^{M} Pr(d_m : \Theta) = \prod_{m=1}^{M} Pr(\mathcal{O}_{1m}\mathcal{O}_{2m}, \dots \mathcal{O}_{nm} : \Theta) =$$
$$= \prod_{j=1}^{n} \prod_{m=1}^{M} Pr(\mathcal{O}_j | parents(\mathcal{O}_j) : \Theta_j) = \prod_{j=1}^{n} L_j(\Theta_j : D)$$

- under the assumption of independence of parameters, likelihood can be decomposed
  - contribution of each network node $L_j(\Theta_j : D)$ is determined (maximized) independently.

Learning Bayesian network parameters from data

- the optimization task: $\widehat{\Theta_j} = \arg\max_{\Theta} L_j(\Theta_j : D)$ is solved for each node,
- let us demonstrate for FO node, where $\Theta_{FO} = \{Pr(fo)\}$
  - let $N(fo)$ be the number of samples, where $FO_j = TRUE$
  - L_{FO} is maximized by putting its first derivative equal to 0

$$L_{FO}(\Theta_{FO} : D) = \prod_{m=1}^{M} Pr(FO : \Theta_{FO}) = Pr(fo)^{N(fo)}(1 - Pr(fo))^{M-N(fo)}$$

$$\frac{\partial L_{FO}(Pr(fo) : D)}{\partial \Pr(fo)} = 0 \rightarrow Pr(fo) = \frac{N(fo)}{M}$$

- the generalized formula for ML parameter estimation is intuitively obvious

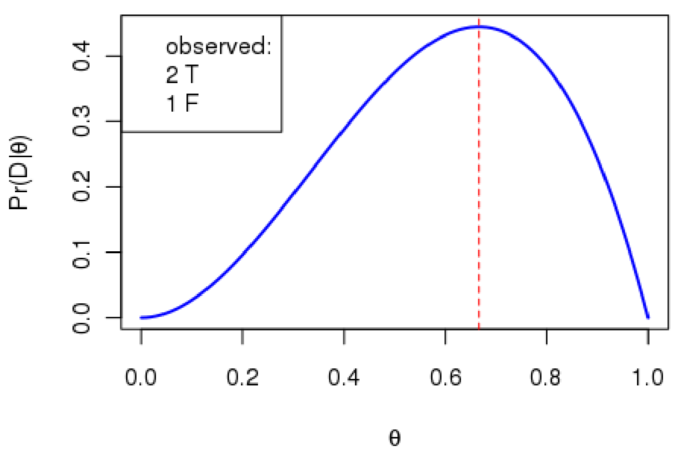
$$\widehat{\theta}_{\mathcal{O}_j|parents(\mathcal{O}_j)} = \frac{N(\mathcal{O}_j, parents(\mathcal{O}_j))}{N(parents(\mathcal{O}_j))} \approx Pr(\mathcal{O}_j|parents(Pj))$$

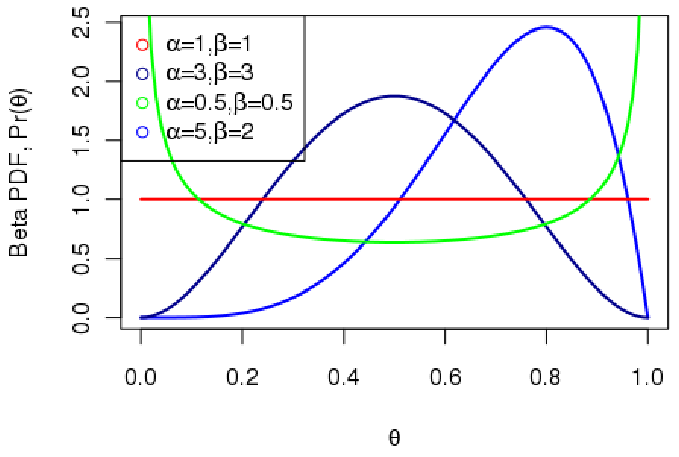
- however, this estimate is imprecise/impossible for sparse/incomplete data
  - sparse data $\rightarrow$ Dirichlet priors and maximum a posteriori (MAP) probability method,
  - missing data $\rightarrow$ Monte-Carlo sampling, or
    $\rightarrow$ EM optimization of multimodal likelihood function.

Parameter learning from a small number of observations

- ill-posed problem
  - overfitting, division by zero or zero probabilities learned,
- regularization
  - introducing additional information in order to resolve an ill-posed problem,
  - Bayesian learning makes use of prior probability

$$Pr(\Theta|D) = \frac{Pr(D|\Theta) \times Pr(\Theta)}{Pr(D)} \Leftrightarrow \text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{prob of data}}$$





- MAP estimate of parameters: $\widehat{\theta}_{o_j|parents(\mathcal{O}_j)} = \frac{N(o_j,parents(\mathcal{O}_j))+\alpha-1}{N(parents(\mathcal{O}_j))+\alpha+\beta-2}$.

Parameter learning from incomplete data

- missing values completely at random
  - the simplest option – independent of variable states, no hidden parameters used,
- it is not advisable to ignore the missing values
  - loses existing observations as well,
- MLE combined with **EM** algorithm:
  1. initialize network parameters (typically using available training data or randomly),
  2. **E step:** take the existing network and compute the missing values (inference),
  3. **M step:** modify the network parameters according to the current complete observations, use MLE,
  4. repeat steps 2 and 3
     (a) for the given prior number of iterations (in this experiment 10),
     (b) until convergence of MLE criterion (log L change between consecutive steps < 0.001).

Parameter learning from incomplete data – example

- consider a linear connection $A \rightarrow B \rightarrow C$,
- learn network parameters, the samples shown in the table below are available,
- use the EM algorithm to learn with missing values (?).

	s_1	s_2	s_3	s_4
A	F	T	T	T
B	T	F	T	?
C	T	F	T	F

Parameter learning from incomplete data – example

- consider a linear connection $A \rightarrow B \rightarrow C$,
- learn network parameters, the samples shown in the table below are available,
- use the EM algorithm to learn with missing values (?).

	s_1	s_2	s_3	s_4
A	F	T	T	T
B	T	F	T	?
C	T	F	T	F

init: $Pr(a)=\frac{3}{4}$, $Pr(b|a)=\frac{1}{2}$, $Pr(b|\neg a)=1$, $Pr(c|b)=1$, $Pr(c|\neg b)=0$,

E$_1$: $Pr(B_4=T)=Pr(b|a,\neg c)=\frac{Pr(a,b,\neg c)}{Pr(a,\neg c)}=\frac{3}{4}\frac{1}{2}0/(\frac{3}{4}\frac{1}{2}0+\frac{3}{4}\frac{1}{2}1)=0 \rightarrow$ estimated F,

M$_1$: $Pr(a)=\frac{3}{4}$, $Pr(b|a)=\frac{1}{3}$, $Pr(b|\neg a)=1$, $Pr(c|b)=1$, $Pr(c|\neg b)=0$,

E$_2$: $Pr(B_4=T)=Pr(b|a,\neg c)=\frac{Pr(a,b,\neg c)}{Pr(a,\neg c)}=\frac{3}{4}\frac{1}{3}0/(\frac{3}{4}\frac{1}{3}0+\frac{3}{4}\frac{2}{3}1)=0 \rightarrow$ estimated F,

M$_2$: necessarily the same result as in M$_1$, converged, STOP.

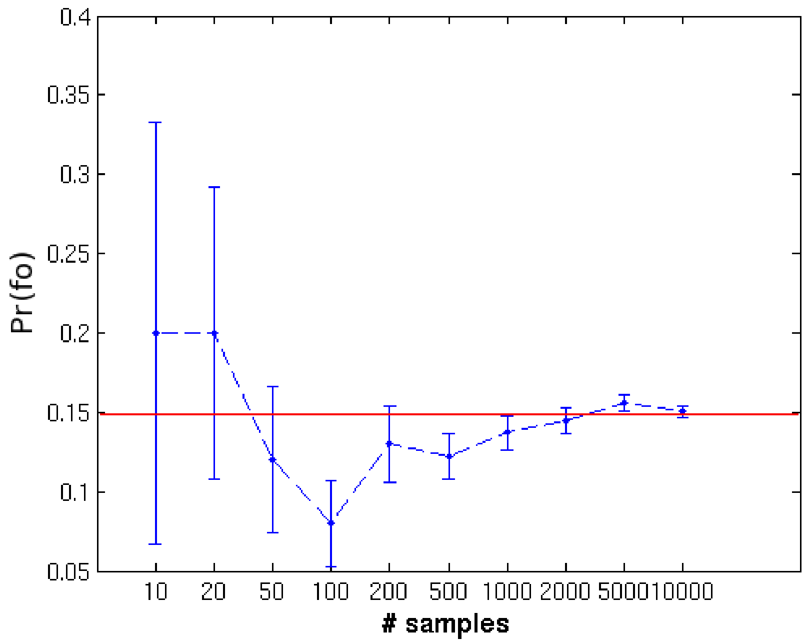
Parameter learning from data – illustration of convergence

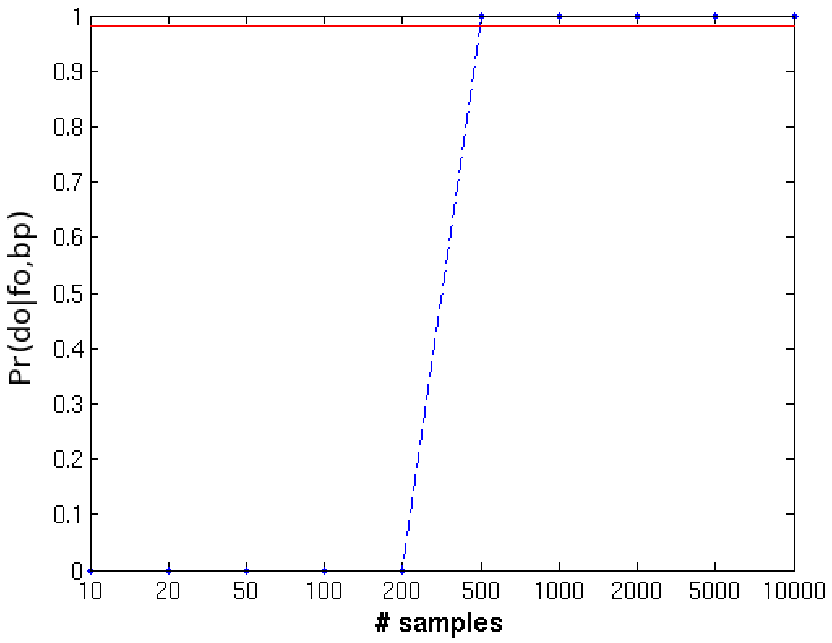
1. take existing (original) network and generate training data (a sample set)
   - FAMILY network, consider different M values (sample set sizes),
   - in which way to generate the data?
     - no evidence, thus **forward sampling**, see inference
     - Gibbs sampling is also possible,
2. randomize quantitative network parameters
   - the network structure is preserved,
   - the original CPTs lost,
3. parameter values are learned from training data
   - complete observations – maximum likelihood estimate (MLE),
   - incomplete observations – combination of MLE and EM algorithm,
4. compare the original and learned CPTs for different sample set sizes M
   - why is it easier to estimate $Pr(fo)$ than $Pr(do|fo, bp)$? see graphs . . .

Parameter learning from data – complete observations

- What is the probability that family is out?
  - $Pr(fo) = ?$
- all samples can be used ...
  - $Pr(fo) = \frac{\sum_{m=1}^{M} \delta(FO^m, fo)}{M}$

- What is the dog out prob given fo and bp?
  - $Pr(do|fo, bp) = ?$
- Condition is met only in 1.5 $^0/_{00}$ of samples.
  - $Pr(fo) = 0.15,\ Pr(bp) = 0.01,$
  - FO and BP independent variables.

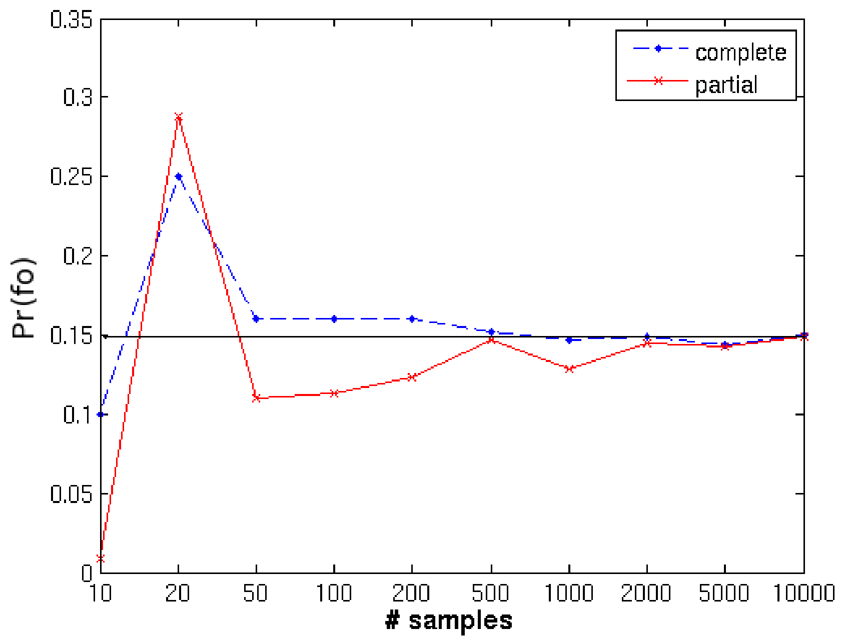


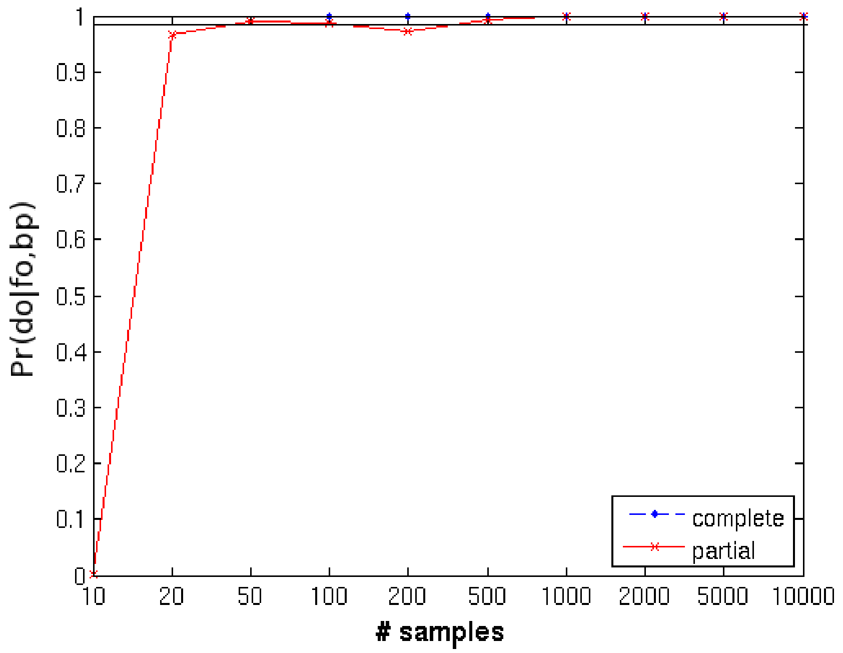


Parameter learning from data – incomplete observations (50% loss)

- What is the probability that family is out?
  - $Pr(fo) = ?$
- Incomplete data = less information
  - considerably longer computational time,
  - the final estimate “a bit less exact only”.

- What is the dog out prob given fo and bp?
  - $Pr(do|fo, bp) = ?$
- Incomplete data = less information
  - comparison is inconclusive.





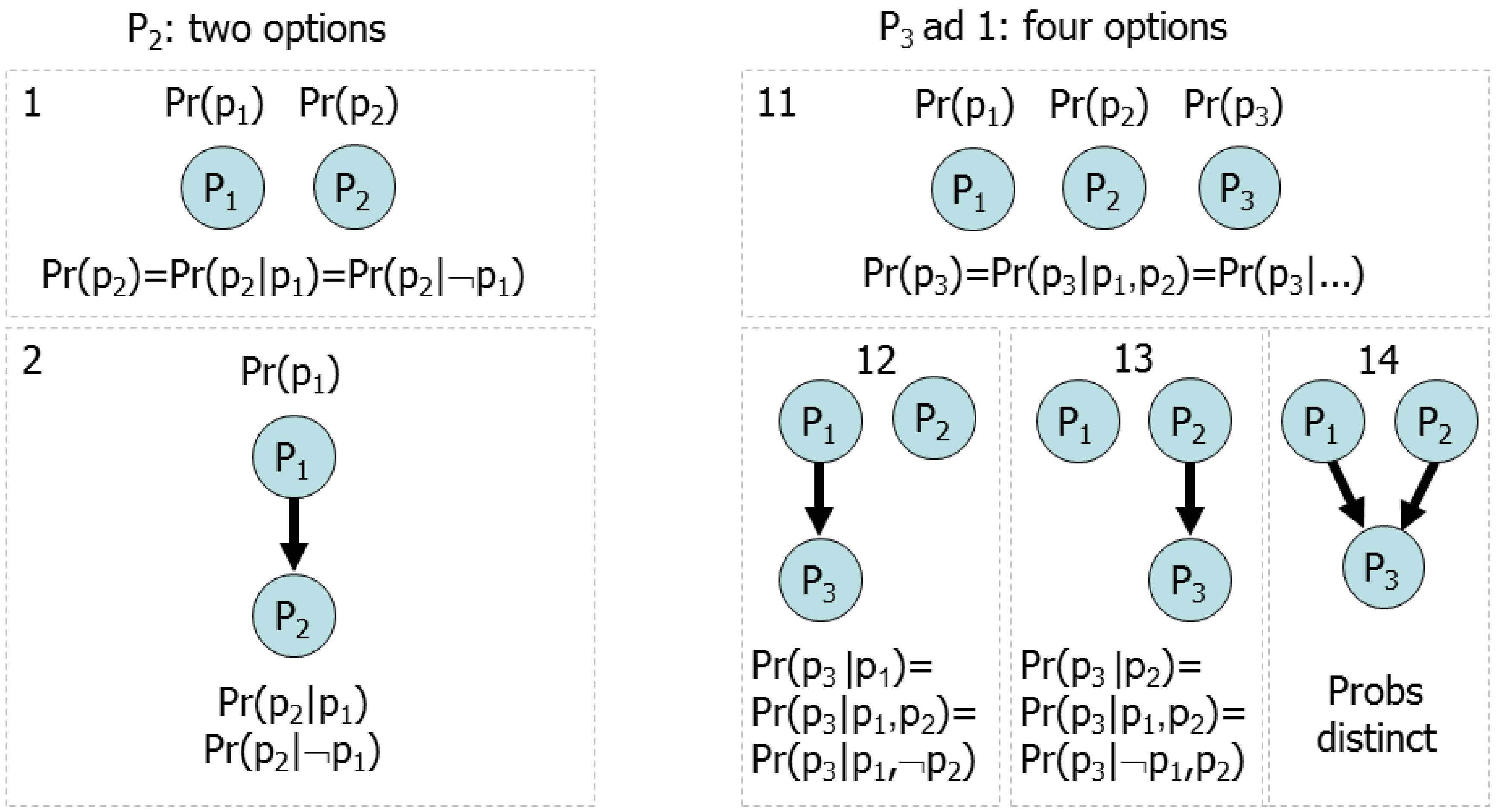
Structure learning – naïve approach

- two steps sufficient to construct the network:
  1. define a sort of n variables,
  2. gradually find subsets of variables that satisfy CI relationship $Pr(\mathcal{O}_{j+1}|\mathcal{O}_1, \dots \mathcal{O}_j) = Pr(\mathcal{O}_{j+1}|parents(\mathcal{O}_{j+1}))$, $parents(\mathcal{O}_{j+1}) \subseteq \{\mathcal{O}_1, \dots \mathcal{O}_j\}$,
- find a network for each of the variable sorts, take the smallest network,
- the algorithm illustrated on a simple three variable example:
  1. select a permutation π: $\pi(\mathcal{O}_1) = 1$, $\pi(\mathcal{O}_2) = 2$ a $\pi(\mathcal{O}_3) = 3$,
  2. gradually build a network, add nodes one by one, CI test underlies the local decision.

Structure learning – naïve approach

- two steps sufficient to construct the network:
  1. define a sort of n variables,
  2. gradually find subsets of variables that satisfy CI relationship
     $Pr(\mathcal{O}_{j+1}|\mathcal{O}_1, \dots \mathcal{O}_j) = Pr(\mathcal{O}_{j+1}|parents(\mathcal{O}_{j+1}))$, $parents(\mathcal{O}_{j+1}) \subseteq \{\mathcal{O}_1, \dots \mathcal{O}_j\}$,
- find a network for each of the variable sorts, take the smallest network,
- the algorithm illustrated on a simple three variable example:
  1. select a permutation π: $\pi(\mathcal{O}_1) = 1$, $\pi(\mathcal{O}_2) = 2$ a $\pi(\mathcal{O}_3) = 3$,
  2. gradually build a network, add nodes one by one, CI test underlies the local decision.
- cannot be implemented in this easy form:
  - variable ordering influences the resulting network
    - ∗ improper ordering $\rightarrow$ redundant edges up to fully connected graph,
    - ∗ however, **n!** distinct permutations cannot be checked,
  - independence tests also non-trivial
    - ∗ for binary variables definitely $\mathcal{O}(2^n)$ operations per single permutation,
    - ∗ among others, $Pr(\mathcal{O}_n|\mathcal{O}_1, \dots \mathcal{O}_{n-1})$ needs to be enumerated.

Structure learning – naïve approach



Score-based structure learning – likelihood and Bayesian score

- score-based learning, maximizes an **evaluation function**
  - the function quantifies how well a structure matches the data,
- straightforward likelihood function selects the fully connected network
  - the more parameters, the better match with data,
  - results in overfitting – improper when comparing structures of different size,

$$\log L(G:D) = \log \prod_{m=1}^{M} Pr(d_m : G) = -M \sum_{i=1}^{n} H(\mathcal{O}_i | parents(\mathcal{O}_i)^G)$$

- evaluation function often based on **Bayesian score** that stems from posterior probability

$$Pr(G|D) = \frac{Pr(D|G)Pr(G)}{Pr(D)} \rightarrow \log Pr(G|D) = \log Pr(D|G) + \log Pr(G) + c$$

  - unlike MLE, it integrates over all parametrizations of given structure

$$Pr(D|G) = \int Pr(D|G,\Theta_G) \times Pr(\Theta_G|G) d\Theta$$

  - MLE concerns solely the best parametrization

$$L(G:D) = Pr(D|G, \widehat{\Theta_G})$$

Score-based structure learning – BIC

- Bayesian Information Criterion (BIC)
  - represents another frequent evaluation function,
  - a heuristic criterion, easier to compute than the Bayesian one,
  - a MDL principle analogy – the best model is both compact and accurate,
  - let us have: q_i …the number of unique instantiations of $\mathcal{O}_i$ parents,
    r_i …the number of distinct $\mathcal{O}_i$ values,
  - then, a network has: $K = \sum_{i=1}^{n} q_i(r_i - 1)$ independent parameters,

$$BIC = -\frac{K}{2}\log_2 M + \log_2 L(G:D) = -\frac{K}{2}\log_2 M - M\sum_{i=1}^{n} H(\mathcal{O}_i|parents(\mathcal{O}_i)^G)$$

  - first addend: network complexity penalty (K ↑ BIC ↓),
  - second addend: network likelihood
    (mutual information between nodes and their parents ↑ $H(|)$ ↓ BIC ↑),

Conditional entropy

- information entropy $H(X)$
  - a measure of the uncertainty in a random variable,
  - the average number of bits per value needed to encode it,
  - $H(X) = -\sum_{x \in X} Pr(x) \log_2 Pr(x)$
- conditional (information) entropy $H(Y|X)$
  - ucertainty in a random variable Y given that the value of random variable X is known,
  - $X \perp\!\!\!\perp Y \Rightarrow H(Y|X) = H(Y)$
  - $H(Y|X) = \sum_{x \in X} Pr(x) H(Y|x) = -\sum_{x \in X} Pr(x) \sum_{y \in Y} Pr(y|x) \log_2 Pr(y|x)$
- how to enumerate conditional entropy?
  - N_{ij} ...the number of samples, where $parents(\mathcal{O}_i)$ take the j-th instantiation of values,
  - N_{ijk} ...the number of samples, where $\mathcal{O}_i$ takes the k-th value and $parents(\mathcal{O}_i)$ the j-th instantiation of values,

$$H(\mathcal{O}_i|parents(\mathcal{O}_i)^G) = -\sum_{j=1}^{q_i} \sum_{k=1}^{r_i} \frac{N_{ij}}{M} \frac{N_{ijk}}{N_{ij}} \log_2 \frac{N_{ijk}}{N_{ij}} = -\sum_{j=1}^{q_i} \sum_{k=1}^{r_i} \frac{N_{ijk}}{M} \log_2 \frac{N_{ijk}}{N_{ij}}$$

Score-based structure learning

- however, no evaluation function can be applied to all 2^{n^2} candidate graphs (simple upper bound),
- heuristics and metaheuristics known for difficult tasks need to be employed
  - metaheuristic example – **local search**
    - it starts with a given network (empty, expert's, random),
    - it construct all the "near" networks, evaluates them and goes to the best of them,
    - it repeats the previous step if the local change increases score, otherwise it stops,
  - auxiliary heuristics examples
    - definition of "near" network,
    - how to avoid getting stuck in local minima or on plateaux
      - random restarts, simulated annealing, TABU search.

Structure learning – K2 algorithm

- Cooper and Herskovitz (1992), it approaches the naïve approach mentioned above,
- advantage
  - complexity is $\mathcal{O}(M, u^2, n^2, r)$, $u \leq n \rightarrow \mathcal{O}(M, n^4, r)$
    - M …the number of samples, n …the number of variables,
    - r …max number of distinct variable values, u …max number of parents,
- disadvantages
  - topological sort of network variables π must be given/known,
  - greedy search results in locally optimal solution.
- it expresses the prob $Pr(G, D)$ as the following function

$$g(\mathcal{O}_i, parents(\mathcal{O}_i)) = \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}!$$

  - q_i …number of unique instantiations of $parents(\mathcal{O}_i)$, r_i …number of distinct $\mathcal{O}_i$ values,
  - N_{ij} …number of samples, where $parents(\mathcal{O}_i)$ take j-th instantiation of values,
  - N_{ijk} …number of samples, where $\mathcal{O}_i$ takes k-th value and $parents(\mathcal{O}_i)$ j-th instantiation of values,

  - **separable** criterion – it can be computed node by node.

Structure learning – K2 algorithm

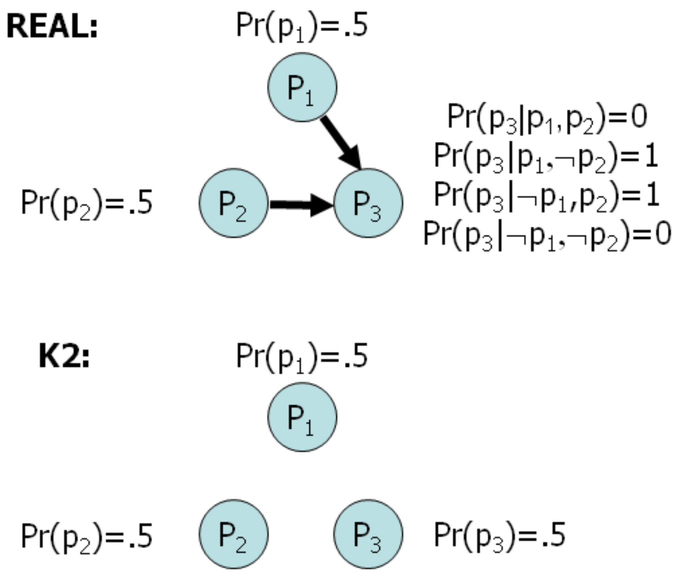
- algorithm K2 (π,u,D):

```
for i=1:n % follow the topological sort of variables $\pi$
    parents($\mathcal{O}_{\pi_i}$)= $\emptyset$ % in the beginning, the set of parents is always empty
    $G_{old}$=g($\mathcal{O}_{\pi_i}$,parents($\mathcal{O}_{\pi_i}$)) % initialize the node value
    while |parents($\mathcal{O}_{\pi_i}$)|$\leq$ u % the number of parents must not exceed u
      $j^* = \underset{j=1\ldots i-1 \mathcal{O}_{\pi_j} \notin parents(\mathcal{O}_{\pi_i})}{\arg\max} g(\mathcal{O}_{\pi_i}, parents(\mathcal{O}_{\pi_i}) \cup \mathcal{O}_{\pi_j})$
      % $\mathcal{O}_{\pi_j^*}$ is the parent maximizing the value of g
      % the parent must have a lower topological index -- by definition
      % omit the candidates already belonging to the set of parents
      $G_{new}$=g($\mathcal{O}_{\pi_i}$,parents$(\mathcal{O}_{\pi_i}) \cup \mathcal{O}_{\pi_j^*}$)
      if $G_{new} > G_{old}$ then
        $G_{old} = G_{new}$
        parents($\mathcal{O}_{\pi_i}$)=parents($\mathcal{O}_{\pi_i} \cup \mathcal{O}_{\pi_j^*}$)
      else
        STOP % the node value cannot be further improved, stop its processing
```

K2 – locality of greedy search, illustration

- let us have binary variables $\mathcal{O}_1$, $\mathcal{O}_2$, $\mathcal{O}_3$, let π={1,2,3} and D is given in the table

$\mathcal{O}_1$	$\mathcal{O}_2$	$\mathcal{O}_3$
F	F	F
F	F	F
F	T	T
F	T	T
T	F	T
T	F	T
T	T	F
T	T	F



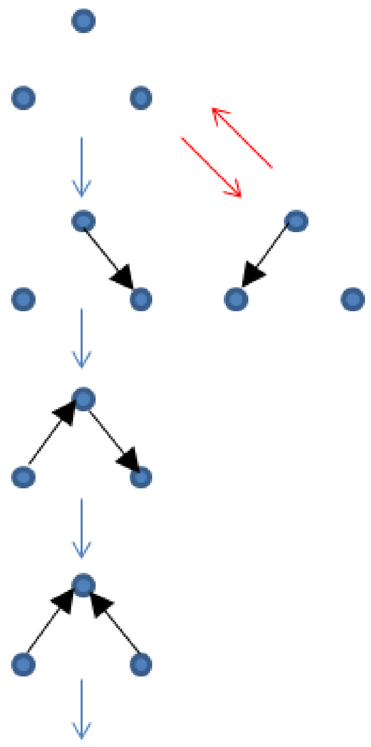
$g(\mathcal{O}_2, \emptyset) = \frac{4!4!}{9!} = \frac{4!}{9\times8\times7\times6\times5} = \frac{1}{630}$
$g(\mathcal{O}_2, \{\mathcal{O}_1\}) = (\frac{2!2!}{5!})^2 = (\frac{1}{180})^2 = \frac{1}{32400}$
K2: STOP, no edge from $\mathcal{O}_1$ to $\mathcal{O}_2$
$g(\mathcal{O}_3, \emptyset) = g(\mathcal{O}_2, \emptyset) = \frac{1}{630}$
$g(\mathcal{O}_3, \{\mathcal{O}_1\}) = (\frac{2!2!}{5!})^2 = (\frac{1}{180})^2 = \frac{1}{32400}$
$g(\mathcal{O}_3, \{\mathcal{O}_2\}) = g(\mathcal{O}_3, \{\mathcal{O}_1\})$
K2: STOP, no edge to $\mathcal{O}_3$, however
$g(\mathcal{O}_3, \{\mathcal{O}_1\mathcal{O}_2\}) = (\frac{2!}{3!})^4 = (\frac{1}{3})^4 = \frac{1}{81}$

- minor improvements
  - apply K2 and K2Reverse and take the better solution
    - ∗ K2Reverse starts with the fully connected graph and greedily deletes edges,
    - ∗ solves the particular problem shown above, but not a general solution,
  - randomly restart the algorithm (various node orderings and initial graphs).

Structure learning – MCMC approach

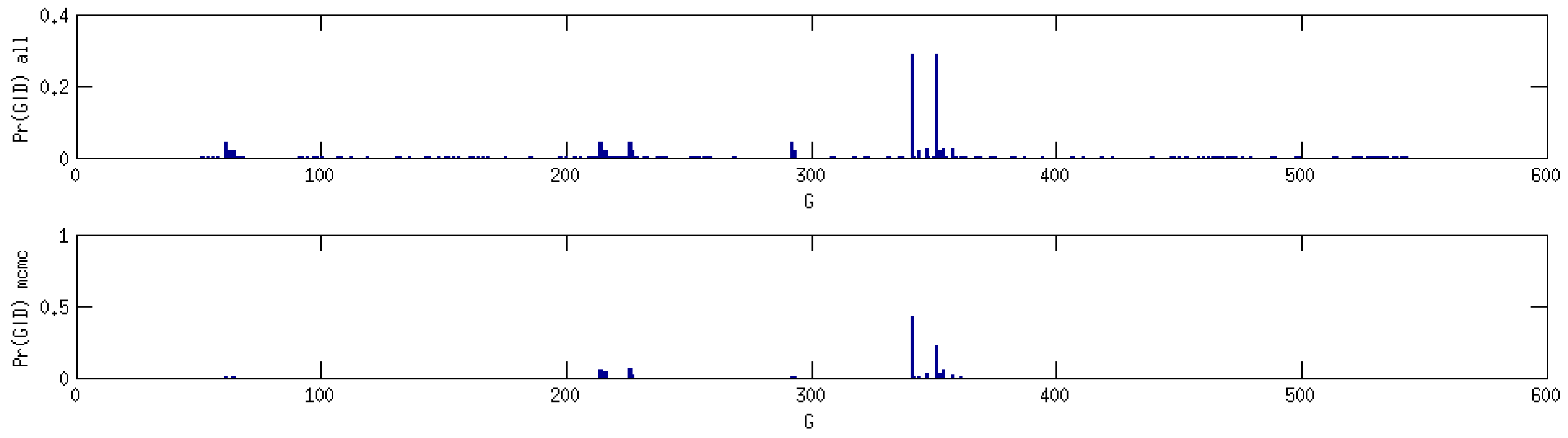
- **MCMC** = Markov chain Monte-Carlo (for meaning see Gibbs sampling),
- applies **Metropolis-Hastings** (MH) algorithm to search the candidate graph/network space

1. take an initial graph G
   - user-defined/informed, random, empty with no edges,
2. evaluate the graph $P(G)$
   - use samples, apply criteria such as BIC or Bayesian,
3. generate a "neighbor" S of the given graph G
   - insert/remove an edge, change edge direction,
   - check the graph acyclicity constraint,
   - prob of transition from G to S is function of $Q(G, S)$,
4. evaluate the neighbor graph $P(S)$,
5. accept or reject the transition to S
   - generate α from U(0,1) (uniform distribution),
   - if $\alpha < \frac{P(S)Q(G,S)}{P(G)Q(S,G)}$ then accept the transition $G \rightarrow S$,
6. repeat steps 3–5 until convergence or the given number of iterations.



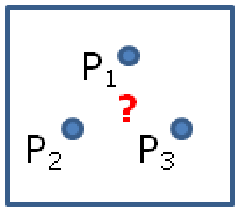
Structure learning – MCMC approach

- graph frequency helps to assume on graph posterior probability
  - a sequence beginning is ignored for random inits,
- the sequence of graphs can be used both for
  - point estimation – e.g., only the network with the highest score is concerned (MAP),
  - Bayesian estimation – all the networks concerned and weighted by their score,
- convergence (frequency proportional to the real score)
  - theoretically converges in polynomial time wrt size of graph space,
  - practically difficult for domains with more than 10 variables.

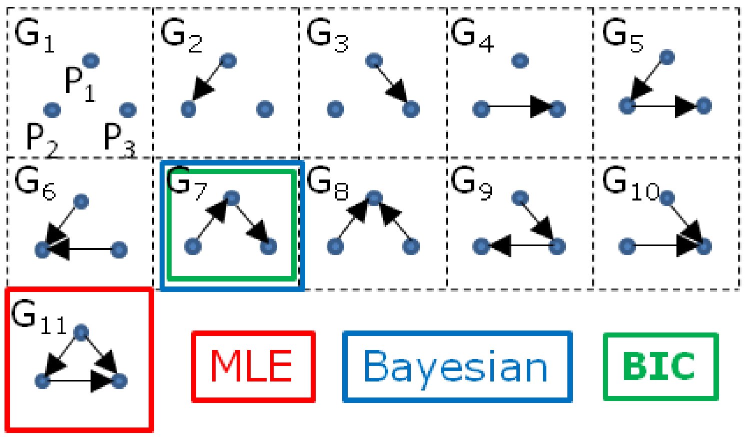


Structure learning – 3DAG example

- initialization:
  - a 3-node trial network taken,
  - 16 samples generated,
  - the network "forgotten",
- learning: (complete search, 11 graphs),
  - score a member of each Markov equivalence class
    - ∗ complete search through a set of 11 graphs/classes,
  - apply 3 distinct criteria to identify the best model
    - ∗ max likelihood, Bayesian MAP and BIC.



		p1	¬p1
p2	p3	0	2
	¬p3	3	4
¬p2	p3	0	0
	¬p3	6	1



Structure learning – 3DAG example

- G_1 gradually evaluated by three criteria:
  - likelihood: ML parameters first $Pr(o_1) = Pr(o_2) = \frac{9}{16}$, $Pr(o_3) = \frac{1}{8}$

$$\ln L(G_1 : D) = \sum_{m=1}^{16} Pr(d_m : G_1) =$$

$$= 2\ln\left(\frac{7}{16}\frac{9}{16}\frac{1}{8}\right) + 3\ln\left(\frac{9}{16}\frac{9}{16}\frac{7}{8}\right) + 10\ln\left(\frac{9}{16}\frac{7}{16}\frac{7}{8}\right) + \ln\left(\frac{7}{16}\frac{7}{16}\frac{7}{8}\right) = -27.96$$

  - the identical likelihood value can also be reached through conditional entropy

$$\ln L(G_1 : D) = -M\sum_{i=1}^{3} H(\mathcal{O}_i|parents(\mathcal{O}_i)^{G_1}) =$$

$$= -16\left[-2\left(\frac{9}{16}\ln\frac{9}{16} + \frac{7}{16}\ln\frac{7}{16}\right) - \left(\frac{1}{8}\ln\frac{1}{8} + \frac{7}{8}\ln\frac{7}{8}\right)\right] = -27.96$$

Structure learning – 3DAG example

- G_1 gradually evaluated by three criteria:
  - BIC – subtract the complexity penalty from the value of network likelihood

$$BIC(G_1 : D) = -\frac{K}{2}\ln M + \ln L(G_1 : D) = -\frac{3}{2}\ln 16 - 27.96 = -32.12$$

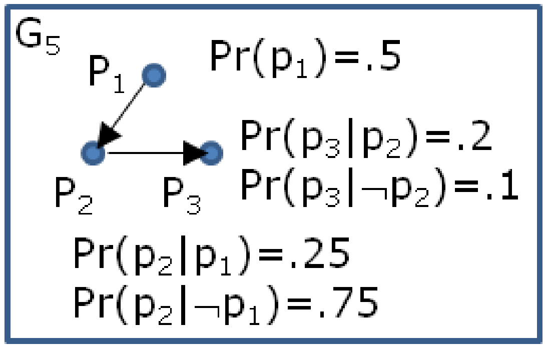
  - Bayesian score

$$\ln Pr(D|G_1) = \ln \prod_{i=1}^{3} g(\mathcal{O}_i, parents(\mathcal{O}_i)^{G_1}) = \sum_{i=1}^{3}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} \ln \frac{(r_i-1)!}{(N_{ij}+r_i-1)!} N_{ijk}! =$$
$$= 2(-\ln 17! + \ln 9! + \ln 7!) - \ln 17! + \ln 2! + \ln 14! = -31.98$$

Natural logarithm is applied to match Matlab BN Toolbox.

Logarithm base change does not change ordering of model evaluations.

Structure learning – 3DAG example



G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}	G_{11}
-27.96	-25.59	-26.12	-26.70	-24.33	-25.32	-23.75	-24.64	-24.86	-25.75	**-23.38**
-31.98	-30.56	-31.78	-32.32	-30.92	-31.03	**-30.36**	-30.56	-31.33	-33.04	-31.62
-32.12	-31.14	-31.67	-32.25	-31.26	-33.64	**-30.68**	-32.96	-31.79	-34.07	-33.08

MLE

Bayesian

BIC

- none of three criteria identified the correct graph class
  - MLE overfits the sample set as expected,
  - BIC and MAP suffer from insufficient data (a too small sample set).

Summary – learning from data

- Estimation of (quantitative) BN parameters
  - relatively easy for large and complete data
    - ML and MAP estimates agree,
    - MAP preferable when a prior distribution exists,
  - gets more difficult with small or incomplete sample sets
    - prior knowledge resp. iterative EM refinement (parameters $\leftrightarrow$ observations),
- BN structure discovery as score-based learning
  - several scores to evaluate how well a structure matches the data
    - likelihood, resp. log likelihood (two ways to compute available) $\rightarrow$ bad idea, overfits,
    - Bayesian score, BIC based on likelihood,
    - other options – among others local CI tests,
  - the space of candidate structures is huge
    - the space cannot be exhaustively searched, i.e., the scores computed for all candidates,
    - consequently, even the naïve approach cannot be considered,
    - K2 – a greedy, locally optimal search,
    - MCMC – a stochastic search similar to simulated annealing.

Recommended reading, lecture resources

- Russell, Norvig: **AI: A Modern Approach**
  - namely uncertainty (chap. 13) and probabilistic reasoning (chap. 14),
  - Norvig's videos on probabilistic inference:
    - http://www.youtube.com/watch?v=q5DHnmHtVmc&feature=plcp,
- Bishop: **Pattern Recognition and Machine Learning.**
  - Chapter 8: Graphical models,
- Charniak: **Bayesian Networks without Tears**
  - popular, AI magazine, 14 pages,
- Koller: **Probabilistic Graphica Models.**
  - book: http://pgm.stanford.edu/, chapter II, inference, variable elimination,
  - Coursera video lectures: https://www.coursera.org/course/pgm,
- Murphy: **A Brief Introduction to Graphical Models and Bayesian Networks.**
  - tutorial: http://www.cs.ubc.ca/~murphyk/Bayes/bayes.html.