Učení struktury

- Strom: Chow–Liu Tree
- Prohledávání 'všech' struktur
- PC–algoritmus, Orientace hran

Definition (KL–divergence)

KL–divergence dvou pravděpodobnostních rozložení P, Q na stejné doméně $sp(P) = sp(Q)$ je definovaná jako: $D_{KL}(P||Q) = \sum_{i \in sp(P)} P(i) log \frac{P(i)}{Q(i)}$.

Definition (Entropie)

Entropie pravděpodobnostního rozložení P na doméně $sp(P)$ je definovaná jako: $H(P) = -\sum_{i \in sp(P)} P(i) log P(i)$.

Definition (Vzájemná informace (Mutual Information))

Mutual Information dvou veličin X, Y na doménách $sp(X), sp(Y)$ je definovaná jako: $I(X;Y) = \sum_{i \in sp(X)} \sum_{j \in sp(Y)} P(X=i, Y=j) log \frac{P(X=i,Y=j)}{P(X=i)P(Y=j)}$.

Chow–Liu Tree

Lemma

Pro modely s interakcemi maximálně druhého řádu, tj. stromy, platí pro D_{KL} aproximace P' a vzoru P

$$D_{KL}(P||P') = -\sum I(X_i, X_{j(i)}) + \sum H(X_i) - H(X_1, \ldots, X_n)$$

kde $X_{j(i)}$ je rodič vrcholu X_i.

Algoritmus **Chow–Liu**

Calculate $I(A, B)$ for each pair of nodes
Find maximal spanning tree (kostru)
Orient edges (no head–to–head connection)
Learn parameters.

```
library('gRapHD')
bf=minForest(sim.data)
```

Prohledávání 'všech' BN

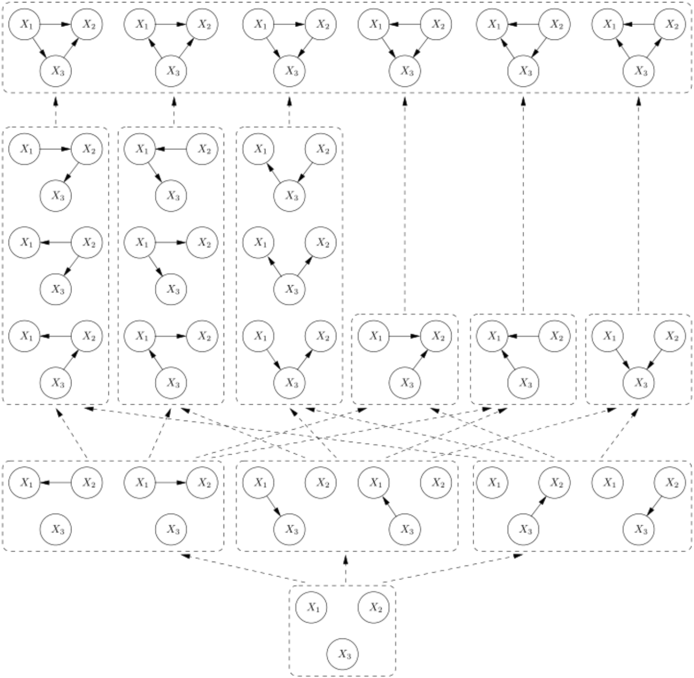
- Budeme prohledávat prostor všech BN.
- Lepší: prohledávat třídy ekvivalence BN – esenciální grafy.
- nebo: triangulované, rozložitelné (neorientované) grafy, orientaci dodat časem.
- Maximalizujeme většinou BIC kriterium, případně AIC a jiná.

Definition (BIC,AIC, loglik)

A BIC criterium for a model S with parameters θ_S compared to N data samples $\mathcal{D}$ is

- Bayesian $BIC(S, \mathcal{D}) = logP_S(\mathcal{D}|\hat{\theta_S}) - \frac{1}{2}|\theta_S|log(N)$
- Akaike's IC $AIC(S, \mathcal{D}) = logP_S(\mathcal{D}|\hat{\theta_S}) - |\theta_S|$
- $loglik(S, \mathcal{D}) = logP_S(\mathcal{D}|\hat{\theta_S})$.

- Kriteria BIC, AIC můžeme vnímat jako maximální věrohodnost penalizovanou složitostí modelu.


X_1
X_2
X_3

Algoritmus **Učení struktury BN**

hc=function($\mathcal{D}$)
  Zvol strukturu S (vše nezávislé, Chow–Liu Tree)
  opakuj
    odhadni parametry θ_s (max. lik., smooth, bayes)
    $currscore = BIC(S, \mathcal{D})$
    S=modifikuj(S) (přidej nejslibnější hranu, uber nejzbytečnější,...)
  dokud chceš (*currscore* se zlepšuje, vyčerpán čas, ...)
  return(S)

```
 chick.bn=stepw(bf,sim.chest) # z lib. gRapHD
chest.net=hc(sim.chest,blacklist=bl) # z library('bnlearn')
```

Orientace hran

- Ne všechny orientace se dají naučit z dat.
- Můžeme použít statistický test na určení podmíněné ne–závislosti veličin,
- pomocí čtyř pravidel (Thomas Nielsen, PC–Algorithm) doplníme orientaci.

Constraint based learning

Some notation

- To denote that A is conditionally independent of B given $\mathcal{X}$ in the database we shall use

$$I(A, B, \mathcal{X}).$$

Some assumptions

- The database is a faithful sample from a Bayesian network M: A and B are d-separated given $\mathcal{X}$ in M if and only if $I(A, B, \mathcal{X})$.
- We have an oracle that correctly answers questions of the type: "Is $I(A, B, \mathcal{X})$?"

The algorithm

Use the oracle's answers to first establish a skeleton of a Bayesian network:

- The skeleton is the undirected graph obtained by removing directions on the arcs.

Next, when the skeleton is found we then start looking for the directions on the arcs.

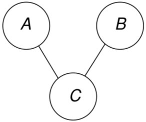
Finding the skeleton II

The idea

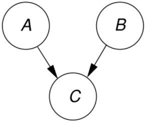
If there is a link between A and B in M then they cannot be d-separated, and as the data is faithful it can be checked by asking questions to the oracle:

- The link $A - B$ is part of the skeleton if and only if $\neg I(A, B, \mathcal{X})$, for all $\mathcal{X}$.

Assume that the only conditional independence found is $I(A, B)$:

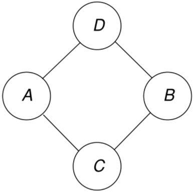


The skeleton



The only possible DAG

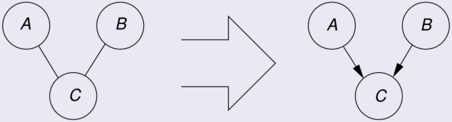
Assume that the conditional independences found are $I(A, B, D)$ and $I(C, D, \{A, B\})$:



Setting the directions on the links I

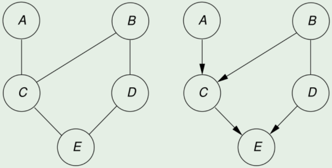
Rule 1

If you have three nodes, A, B, C such that $A - C$ and $B - C$, but not $A - B$, then introduce the v-structure $A \rightarrow C \leftarrow B$ if there exists an $\mathcal{X}$ (possibly empty) such that $I(A, B, \mathcal{X})$ and $C \notin \mathcal{X}$.



Example

Assume that we get the independences $I(A, B)$, $I(A, D)$, $I(A, B, D)$, $I(A, D, B)$, $I(C, D, B)$, $I(A, D, \{B, C\})$, $I(C, D, \{A, B\})$, $I(B, E, \{C, D\})$, $I(A, E, \{C, D\})$, $I(A, D, \{B, C, E\})$, $I(A, E, \{B, C, D\})$, $I(B, E, \{A, C, D\})$.



Setting the directions on the links II

Rule 2 [Avoid new v-structures]

When Rule 1 has been exhausted, and you have $A \rightarrow C - B$ (and no link between A and B), then direct $C \rightarrow B$.

Rule 3 [Avoid cycles]

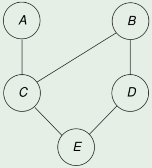
If $A \rightarrow B$ introduces a directed cycle in the graph, then do $A \leftarrow B$.

Rule 4 [Choose randomly]

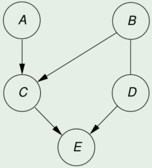
If none of the rules 1-3 can be applied anywhere in the graph, choose an undirected link and give it an arbitrary direction.

Example

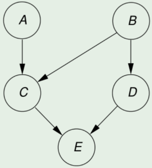
Skeleton:



Rule 1:

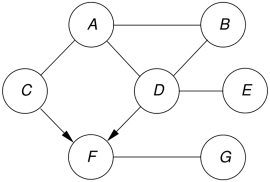


Rule 4:



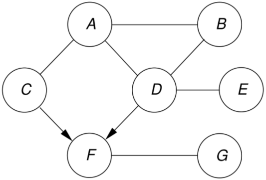
Another example

Consider the graph:

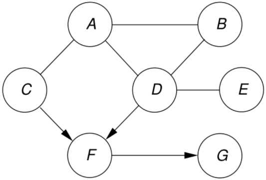


Apply the four rules to learn a Bayesian network structure

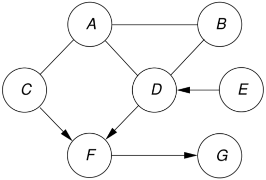
Another example I



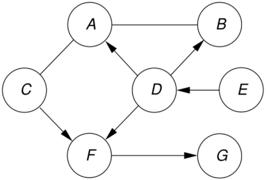
Step 1: Rule 1



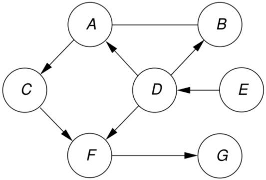
Step 2: Rule 2



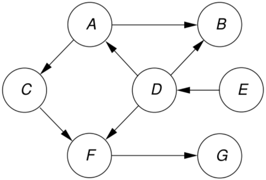
Step 3: Rule 4



Step 4: Rule 2



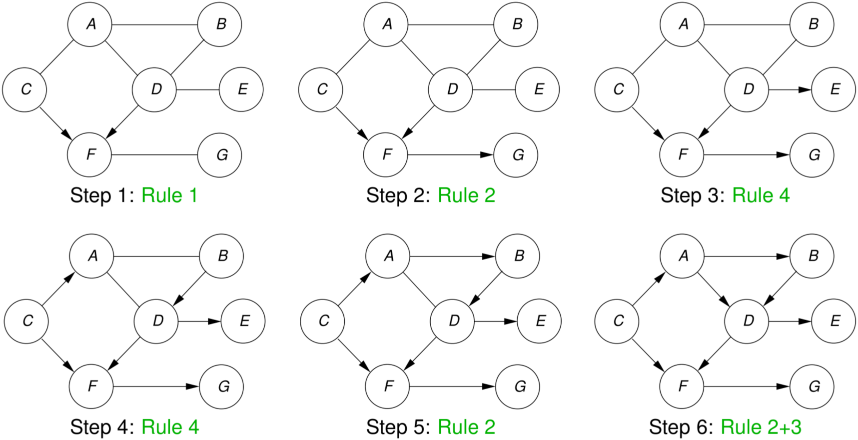
Step 5: Rule 2



Step 6: Rule 4

However, we are not guaranteed a unique solution!

Another example II



Although the solution is not necessarily unique, all solutions have the same d-separation properties!

From independence tests to skeleton

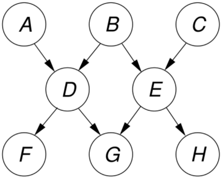
Until now, we have assumed that all questions of the form “Is $I(A, B, \mathcal{X})$?” can be answered (allowing us to establish the skeleton). However, questions come at a price, and we would therefore like to ask as few questions as possible.

To reduce the number of questions we exploit the following property:

Theorem

The nodes A and B are not linked if and only if $I(A, B, \mathrm{pa}(A))$ or $I(A, B, \mathrm{pa}(B))$.

It is sufficient to ask questions $I(A, B, \mathcal{X})$, where $\mathcal{X}$ is a subset of A's or B's neighbors.



An active path from A to B must go through a parent of B.

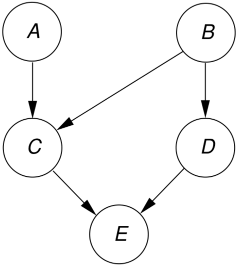
The PC algorithm

The PC algorithm:

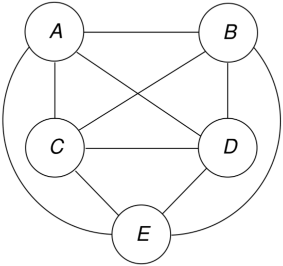
1. Start with the complete graph;
2. $i := 0$;
3. **while** a node has at least $i + 1$ neighbors
   - **for all** nodes A with at least $i + 1$ neighbors
     - **for all** neighbors B of A
       for all neighbor sets $\mathcal{X}$ such that $|\mathcal{X}| = i$ and $\mathcal{X} \subseteq (\mathrm{nb}(A) \setminus \{B\})$
       if $I(A, B, \mathcal{X})$ **then** remove the link $A - B$ and store "$I(A, B, \mathcal{X})$"
   - $i := i + 1$

Example

We start with the complete graph and ask the questions $I(A, B)$?, $I(A, C)$?, $I(A, D)$?, $I(A, E)$?, $I(B, C)$?, $I(B, D)$?, $I(B, E)$?, $I(C, D)$?, $I(C, E)$?, $I(D, E)$?.



The original model



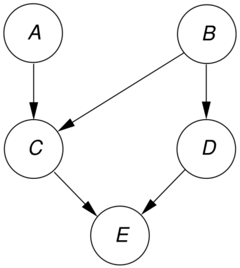
The complete graph

We get a “yes” for $I(A, B)$? and $I(A, D)$?:

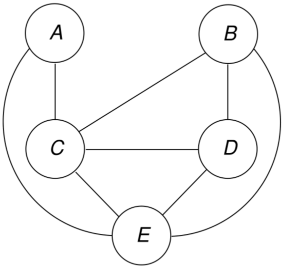
- the links $A - B$ and $A - D$ are therefore removed.

Example

We now condition on one variable and ask the questions $I(A, C, E)$?, $I(A, E, C)$?, $I(B, C, D)$?, $I(B, C, E)$?, $I(B, D, C)$?, $I(B, D, E)$?, $I(B, E, C)$?, $I(B, E, D)$?, $I(C, B, A)$?, $\ldots, I(C, D, A)$?, $I(C, D, B)$?.



The original model



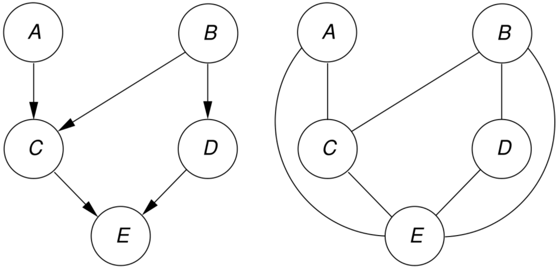
After one iteration

The question $I(C, D, B)$? has the answer "yes":

- we therefore remove the link $C - D$.

Example

We now condition on two variables and ask questions like $I(B, C, \{D, E\})$?.



The original model

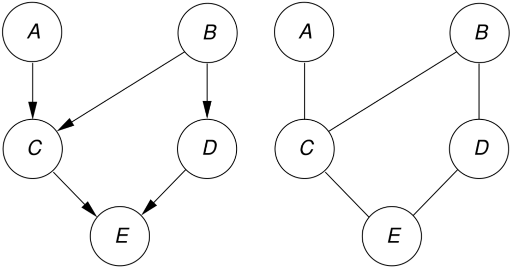
After two iterations

The questions $I(B, E, \{C, D\})$? and $I(E, A, \{D, C\})$? have the answer “yes”:

- we therefore remove the links $B - E$ and $A - E$.

Example

We now condition on three variables, but since no nodes have four neighbors we are finished.



The original model	After three iterations

The identified set of independence statements are then $I(A, B)$, $I(A, D)$, $I(C, D, B)$, $I(A, E, \{C, D\})$, and $I(B, E, \{C, D\})$. They are sufficient for applying rules 1-4.

Real world data

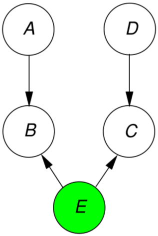
The oracle is a statistical test, e.g. conditional mutual information:

$$CE(A, B|X) = \sum_X P(X) \sum_{A,B} P(A, B|X) \log \frac{P(A, B|X)}{P(A|X)P(B|X)}.$$

$$I(A, B, X) \Leftrightarrow CE(A, B|X) = 0.$$

However, all tests have false positives and false negatives, which may provide false results/causal relations!

Similarly, false results may also be caused by hidden variables:



Algoritmus **Učení struktury BN**

hc=function($\mathcal{D}$)
  Zvol strukturu S (vše nezávislé, Chow–Liu Tree)
  opakuj
    odhadni parametry θ_s (max. lik., smooth, bayes)
    $currscore = BIC(S, \mathcal{D})$
    S=modifikuj(S) (přidej nejslibnější hranu, uber nejzbytečnější,...)
  dokud chceš (*currscore* se zlepšuje, vyčerpán čas, ...)
  return(S)

Pro BN s 24 uzly máme $9.4 * 10^{102}$ možných DAG, esenciánlích grafů méně, ale pořád hodně.

One attempt at scoring a BN with structure S:

$$\begin{aligned}\text{score}(BN, \mathcal{D}) &= -\text{dist}(P_S(\mathcal{U} \mid \hat{\theta}_S), P_{\mathcal{D}}(\mathcal{U})) \\ &= -\sum_{\mathbf{x} \in \text{sp}(\mathcal{U})} (P_S(\mathbf{x} \mid \hat{\theta}_S) - P_{\mathcal{D}}(\mathbf{x}))^2\end{aligned}$$

Problem: The Euclidean distance is difficult to calculate!

Decomposable functions

If the database is complete, then both BIC and BDe are decomposable:

$$\mathrm{score}(\mathcal{D}, S) = \sum_{i=1}^{n} \mathrm{score}(X_i, \mathrm{pa}(X_i), \mathcal{D}).$$

Consider now a greedy search where we only make local changes. Then in order to evaluate if inserting an arc $(X_i \rightarrow X_j)$ is beneficial, we calculate:

$$\begin{aligned}\Delta(X_i \rightarrow X_j) &= \mathrm{score}(\mathcal{D}, S') - \mathrm{score}(\mathcal{D}, S) \\ &= \mathrm{score}(X_j, \mathrm{pa}(X_j) \cup \{X_i\}, \mathcal{D}) - \mathrm{score}(X_j, \mathrm{pa}(X_j), \mathcal{D}).\end{aligned}$$

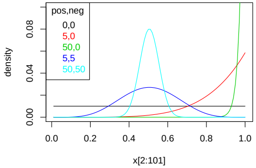
This can be exploited during structure search in DAG space!

Definition (BIC,AIC, loglik)

A BIC criterium for a model S with parameters θ_S compared to N data samples $\mathcal{D}$ is

- Bayesian $BIC(S, \mathcal{D}) = logP_S(\mathcal{D}|\hat{\theta_S}) - \frac{1}{2}|\theta_S|log(N)$
- Akaike's IC $AIC(S, \mathcal{D}) = logP_S(\mathcal{D}|\hat{\theta_S}) - |\theta_S|$
- $loglik(S, \mathcal{D}) = logP_S(\mathcal{D}|\hat{\theta_S})$.

$$beta[a,b](\theta) = \alpha\theta^{a-1}(1-\theta)^{b-1}$$



Dirichlet Distribution

For $W = \{w_1, \ldots, w_k\}$ the Beta distribution is generalized by the Dirichlet distribution:

$$f_{a_1,\ldots,a_k}(p_1,\ldots,p_k) := \frac{\Gamma(a_1+\ldots+a_k)}{\Gamma(a_1)\cdots\Gamma(a_k)} p_1^{a_1}\cdots p_k^{a_k} \qquad (a_i \in \mathbb{R})$$

The BDe score

The BIC score is an approximation to the marginal likelihood of the structure S given the data $\mathcal{D}$. Under the assumptions of parameter independence, Dirichlet priors, complete data, and a few things more, $P(\mathcal{D} \mid S)$ can be calculated exactly as:

$$P(\mathcal{D} \mid S) = \int_{\boldsymbol{\theta}_S} P(\mathcal{D} \mid S, \boldsymbol{\theta}_S) P(\boldsymbol{\theta}_S \mid S) d\boldsymbol{\theta}_S$$

$$= \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij}+N_{ij})} \cdot \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk}+N_{ijk})}{\Gamma(\alpha_{ijk})}$$

Prior distribution over structures

Most prior distributions factorizes over the families in the network:

$$P(S) \propto \prod_{l=1}^{n} \rho(X_i, \mathrm{pa}(X_i)).$$

For complete ignorance we could use:

$$\rho(X_i, \mathrm{pa}(X_i)) = 1.$$

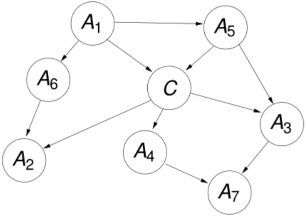
A more informative prior

$$\rho(X_i, \mathrm{pa}(X_i)) = \kappa^{\sum_{i=1}^{n} \delta_i},$$

where δ_i is the number of parents for node X_i for which S disagrees with some prior specified network.

When N gets large the likelihood term will dominate the prior!

A General Bayesian Network Classifier



- Bayesian network can contain information that is irrelevant for the classification task.

$$\log P(D|BN) = \sum_{i=1}^{N} \log P(\mathbf{d}^i \mid BN) = \sum_{i=1}^{N} \log P(c^i \mid \mathbf{a}_1^i, \ldots, \mathbf{a}_n^i, BN) + \sum_{i=1}^{N} \log P(\mathbf{a}_1^i, \ldots, \mathbf{a}_n^i \mid BN)$$

- Precise probability values may not be needed (only ordering on states(C) according to probability).
- Conditional log-likelihood does not decompose over the families.

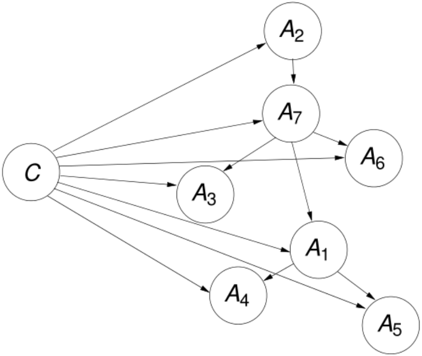
$\rightsquigarrow$: for classification usually use restricted types of Bayesian network structures.

Tree Augmented Naive Bayes

Extend the model to capture some of the correlations among the attributes:

Model: all Bayesian network structures where

- The class node is parent of each attribute node
- The substructure on the attribute nodes is a tree



Learning a TAN Classifier

A TAN of maximal likelihood can be constructed as follows:

1. Calculate $\mathrm{CMI}(A_i, A_j \mid C)$ for each pair (A_i, A_j).
2. Build a maximum-weight spanning tree over the attributes.
3. Direct the resulting tree.
4. Insert C as a parent of all the attributes.
5. Learn the parameters.

$$\mathrm{CMI}(A_i, A_j \mid C) = \sum_{C} P^{\#}(C) \sum_{A_i, A_j} P^{\#}(A_i, A_j \mid C) \log_2 \left(\frac{P^{\#}(A_i, A_j \mid C)}{P^{\#}(A_i \mid C) P(A_j \mid C)} \right)$$

Podmíněná nezávislost

Tabulka ukazuje zadané hodnoty $P(A, B, C)$. Pro která x, y, v, w platí podmíněná nezávislost $A \perp\!\!\!\perp B | C$?

	c1			c2
	b1	b2	b1	b2
a1	x	0.2	v	w
a2	y	0.1	0.1	0.1

$$\frac{P(a_1, b_1, c_1)}{P(a_2, b_1, c_1)} = \frac{P(a_1|c_1) \cdot P(b_1|c_1) \cdot P(c_1)}{P(a_2|c_1) \cdot P(b_1|c_1) \cdot P(c_1)} = \frac{P(a_1, b_2, c_1)}{P(a_2, b_2, c_1)}$$

tedy $x = 2 \cdot y$
obdobně $v = w$.
Navíc celkový součet musí být 1, tedy:

$$\begin{aligned} 0.5 + 3y + 2v &= 1 \\ v &= 0.25 - 1.5y \end{aligned}$$

Separace, Markovské vlastnosti

Definition (Separace)

Buď G neorientovaný graf nad V. Řekneme $\mathcal{C}$ separuje A a B, psáno $A \perp\!\!\!\perp_G B|\mathcal{C}$ v G, pokud každá cesta z A do B v G vede přes $\mathcal{C}$.

Definition (Markovská vlastnost, Globální, lokální, párová)

Buď G neorientovaný graf nad V.

(GM) Pravděpodobnostní míra P nad V je **(globálně) markovská** vzhledem k G, jestliže:

$$\forall (A, B \in V, \mathcal{C} \subseteq V)\ A \perp\!\!\!\perp_G B|\mathcal{C} \Rightarrow \mathcal{A} \perp\!\!\!\perp \mathcal{B}|\mathcal{C} \text{ v P.}$$

(LM) Míra je **lokálně markovská**, pokud $\forall A \in V\ A \perp\!\!\!\perp V \setminus Fa_A|N_A\ [P]$

(PM) Míra je **párově markovská**, pokud $\forall A, B \in V, A \neq B$, nespojené hranou, $A \perp\!\!\!\perp B|V \setminus \{A, B\}\ [P]$

Theorem

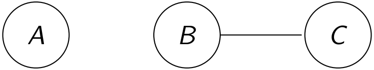
Pro strikně pozitivní míry (bez nul) jsou všechny tři vlastnosti ekvivalentní. [Milan Studený:Struktury podmíněné nezávislosti, Matfyzpress 2014]

Příklady

Example

$V = \{A, B, C\}, E = \{(b, c)\}$. Zvolíme binární pravděpodobnostní míru nad V sídlící v bodech $(0, 0, 0)$ a $(1, 1, 1)$ viz Studený p.101.

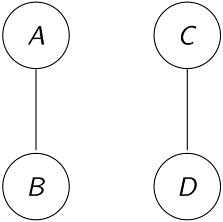
$\begin{array}{l} A \perp\!\!\!\perp B|\{C\} \\ A \perp\!\!\!\perp C|\{B\} \end{array}$ & neimplikuje $A \perp\!\!\!\perp BC|\{\}$.



Example

$V = \{A, B, C\}, E = \{(b, c)\}$. Zvolíme binární pravděpodobnostní míru nad V sídlící v bodech $(0, 0, 0, 0)$ a $(1, 1, 1, 1)$ viz Studený p.101.

$\begin{array}{l} A \perp\!\!\!\perp CD|\{B\} \\ B \perp\!\!\!\perp CD|\{A\} \\ C \perp\!\!\!\perp AB|\{D\} \\ D \perp\!\!\!\perp AB|\{C\} \end{array}$ & neimplikuje $A \perp\!\!\!\perp C|\{\}$.



Rozložitelnost

Definition (Rozklad)

Buď G neorientovaný graf nad V. Jestliže $S, T \subseteq V$ jsou takové, že

1. $S \cup T = V$
2. $S \cap T$ je úplná množina v G a
3. $S \setminus T \perp\!\!\!\perp_G T \setminus S | S \cap T$,

Pak dvojici indukovaných podgrafů G_S, G_T nazveme **rozkladem** G. Tento rozklad nazveme netriviální, jestli že $S \setminus T \neq \emptyset \neq T \setminus S$.
Neorientovaný graf G nazveme **rozložitelný**

- buď G je úplný
- nebo existuje netriviální rozklad G_S, G_T grafu G takový, že G_S a G_T jsou rozložitelné.

Ekvivalence

Theorem

Buď G neorientovaný graf nad V. Následující podmínky jsou ekvivalentní:

- *G je rozložitelný graf,*
- *G je chordální (=chordální) graf,*
- *existuje perfektní eliminiační posloupnost pro G,*
- *existuje strom spojení pro množinu klik grafu G.*

Definition (d-separace)

Dvě veličiny $A, B \in V$ bayesovské sítě $G = (V, E)$ jsou **d-separované** $A \perp\!\!\!\perp_d B|\mathcal{C}$ množinou $\mathcal{C} \subseteq V \setminus \{A, B\}$ právě když pro každou (neorientovanou) cestu z A do B platí aspoň jedno z následujících:

- cesta obsahuje uzel $Blocking \in \mathcal{C}$ a hrany se v $Blocking$ **nesetkávají 'head-to-head'**,
- cesta obsahuje uzel $Blocking$ kde se hrany **setkávají** 'head-to-head' a ani on ani nikdo z jeho následníků není v $\mathcal{C}$, $\{Blocking, succ(Blocking)\} \cap \mathcal{C} = \emptyset$.

Gibbs Sampling

- První příklad MCMC metody – Markov Chain Monte Carlo

Algoritmus Gibbs Sampling $(bn, E = e)$ **with** n **variables** $V_j \in V$

$sample_0 = \langle v_{0,1}, \dots, v_{0,n} \rangle$ libovolné přiřazení hodnot $V_j \in V$ konzistentní s e,
for s in $1 : last$
  vyber $V_l \in V \setminus E$ jednu proměnnou bez evidence ke změně
    generuj novou hodnotu $v_{s,l} \in V_l$ dle pravděpodobnosti
      $P(V_l | V \setminus \{V_l\} = \langle v_{(s-1),1}, \dots v_{(s-1),(l-1)}, v_{(s-1),(l+1)}, \dots, v_{(s-1),n} \rangle, e)$
    $sample_s = \langle v_{s,1}, \dots v_{s,(l-1)}, v_{s,l}, v_{s,(l+1)}, \dots, v_{s,n} \rangle$
return $list(sample_{burned\_in}, \dots, sample_{last})$

- Pravděpodobnost nových hodnot $P(V_l | \dots)$ zjistíme z BN.
  - Pro výpočet stačí Markov Blanket - rodiče V_l, děti a rodiče dětí.
  - Ostatní veličiny jsou d–separované od V_l dáno Markov Blanket (ověřte).
- Vzorky nejsou nezávislé; většinou se prvních $burn\_in - 1$ vzorků zahazuje.

Metropolis Hastings Algorithm

- Jiná náhodná procházka, MCMC metoda.
- Hodí se např. při hledání struktury BN.
- Mějme libovolnou funkci pravděpodobnosti přechodu (**proposal probabilities**) v prostoru hodnot bn: $\{q(\mathbf{v'}|\mathbf{v})|\mathbf{v}, \mathbf{v'} \in sp(\mathbf{V})\}$.
- Definujme pravděpodobnosti přijetí (**acceptance probabilities**)

$$\begin{aligned}\alpha(\mathbf{v'}|\mathbf{v}) &= min\left(1, \frac{P(\mathbf{V}=\mathbf{v'}|E=e)q(\mathbf{v}|\mathbf{v'})}{P(\mathbf{V}=\mathbf{v}|E=e)q(\mathbf{v'}|\mathbf{v})}\right) \\ &= min\left(1, \frac{P(\mathbf{V}=\mathbf{v'}, E=e)q(\mathbf{v}|\mathbf{v'})}{P(\mathbf{V}=\mathbf{v}, E=e)q(\mathbf{v'}|\mathbf{v})}\right)\end{aligned}$$

Algoritmus **Metropolis Hastings sampling** $(bn, E = e)$ **with** n **variables** $V_j \in V$

$sample_0 = \langle v_{0,1}, \ldots, v_{0,n} \rangle$ libovolné přiřazení hodnot $V_j \in V$ konzistentní s e,
for s in 1 : *last*
    vyber kandidáta na nový stav **v'** podle $q(\mathbf{v'}|sample_{s-1})$
        přijmi ho s pravděpodobností $\alpha(\mathbf{v'}|sample_{s-1})$
        if (přijatý) $sample_s = \mathbf{v'}$
            else $sample_s = sample_{s-1}$
return $list(sample_{burned\_in}, \ldots, sample_{last})$

Theorem

Pokud $q(\mathbf{v'}|\mathbf{v}) > 0$ *pro každé* $\mathbf{v}, \mathbf{v'}$*, pak Metropolis Hastings sampling konverguje* $lim_{i\to\infty} P(sample_i) = P(\mathbf{V}|E = e)$.

- Pro dobré fungování potřebujeme α pravděpodobnost přijetí blízkou 1,

$$\alpha(\mathbf{v'}|\mathbf{v}) \quad = \quad min\left(1, \frac{P(\mathbf{V} = \mathbf{v'}, E = e)q(\mathbf{v}|\mathbf{v'})}{P(\mathbf{V} = \mathbf{v}, E = e)q(\mathbf{v'}|\mathbf{v})}\right)$$

- ideálně q 'trefí' cílové rozložení $P(\mathbf{V}|E = e)$, tj. $q(\mathbf{v'}|\mathbf{v}) = P(\mathbf{V} = \mathbf{v'}|E = e)$.