Přibližný výpočet bayesovské sítě, Učení parametrů BN

Přibližný výpočet BN

- **Loopy Belief Propagation** - přímo v bayesovské síti posílám zprávy jako kdyby to byl strom spojení,
- **Monte Carlo Metody** - nasimuluji data, z nich počítám pravděpodobnosti jako podíl četností.

Základní odhad parametru BN z dat

- (vyhlazený $smooth = 0.001$) podíl četností:
  $\widehat{P}(A = a|pa(A) = \langle v_1, \ldots, v_{|pa(A)|}\rangle) =$

$$= \frac{\#data[A = a \& pa(A) = \langle v_1, \ldots, v_{|pa(A)|}\rangle] + smooth}{\#data[pa(A) = \langle v_1, \ldots, v_{|pa(A)|}\rangle] + smooth \cdot |dom(A)|}.$$

Simulace dat z BN

- Základní myšlenkou je vygenerovat data dle zadaných podmíněných pravděpodobností a z nich spočítat pravděpodobnosti, které nás zajímají.
- Přesnost výpočtu samozřejmě závisí na počtu vygenerovaných vzorků.
- Metody generující náhodné vzorky se nazývají metody **Monte Carlo**.
- Základem je generátor náhodného výsledku podle zadané pravděpodobnosti, např. $\langle \frac{1}{4}, \frac{1}{2}, \frac{1}{4} \rangle$.

>runif(1)

Přímé vzorkování bez evidence

- Uspořádáme vrcholy BN tak, aby každá hrana začínala v uzlu menšího čísla než končí.
- Vytvoříme N vzorků, každý následovně
  - Pro první uzel A_1 vygenerujeme náhodně výsledek a_1 podle $P(A_1)$.
  - Pro druhý uzel A_2 vygenerujeme náhodně výsledek a_2 podle $P(A_2|A_1 = a_1)$ (je–li hrana, jinak nepodmíněně)
  - Pro n–tý uzel vygenerujeme výsledek podle $P(A_n|pa(A_n))$, na rodičích už známe konkrétní hodnoty.
- Z N vzorků spočteme pravděpodobnost jevu, který nás zajímá. Pro N jdoucí k nekonečnu podíl výskytu jevu konverguje k správné pravděpodobnosti.

Přímé vzorkování s evidencí e (rejection sampling)

- $N(e)$ značí počet vzorků konzistentních s evidencí e, tj. nabývající na příslušných veličinách správné hodnoty.
- Vzorky tvoříme úplně stejně, jako dříve, jen ty, co nejsou konzistentní s e vyšktneme, tj. $\hat{P}(X|e) = \frac{N(X,e)}{N(e)}$
- Problém je v tom, že je–li $P(e)$ malé, tak většinu vzorků zahazujeme.

Vážení věrohodností (Likelihood weighting)

- Generuje jen vzorky konzistentní s e.
- Váhy vzorků jsou různé, podle $P(e|vzorek)$ (což je věrohodnost $L(vzorek|e)$, odtud likelihood weighting).

Algoritmus vytvoření váženého vzorku pro (bn, e)

$w = 1$
v pořadí topologického uspořádání bn, for $i = 1$ to n
    if A_i má evidenci a_i v e
        $w = w \cdot P(A_i = a_i|pa(A_i))$
    else
        a_i vyber podle rozložení $P(A_i = a_i|pa(A_i))$
return $(w, \langle a_1, \ldots, a_n \rangle)$

KL–divergence

Definition (KL–divergence)

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

- KL–divergencí se měří ne–podobnost pravděpodobnostních rozložení.
- Pozor: KL– divergence není symetrická, není definovaná pokud Q je někde nula a P není.

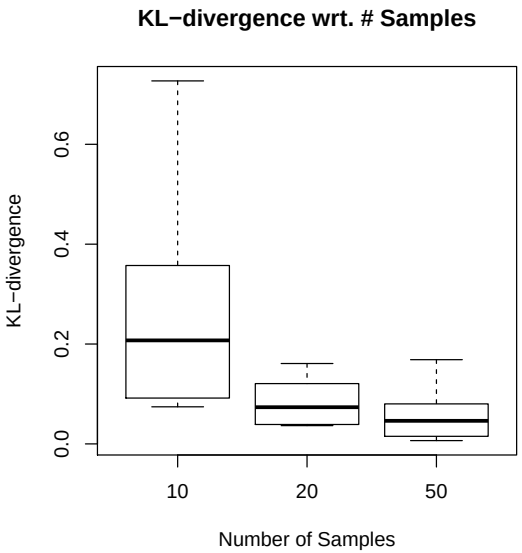
Úkol

- V kódu experimentujte se simulací, 'kontrolou' pravděpodobností/četností.
- Spočítejte KL-divergenci různých vzorků od pravděpodobnosti v originální BN.

```
 pravda=querygrain(bnet,nodes=c('Coin','FirstFlip'),type='joint')
sim.orig=simulate(bnet,n=1000)
novy=xtabs(~Coin+FirstFlip,sim.orig)
KL.empirical(novy,pravda,unit='log2')
```

Příklad (jen dobrovolně)

```
 n.samples=c(rep(10,10),rep(20,10),rep(50,10))
kl=sapply(n.samples
,FUN=function(x)my.sim(bnet,c('Coin','FirstFlip'),x))
boxplot(kl~n.samples,xlab='Number of
Samples',ylab='KL-divergence',main='KL-divergence wrt. Samples')
```



Gibbs Sampling

- První příklad MCMC metody – Markov Chain Monte Carlo

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

$sample_0 = \langle v_{0,1}, \ldots, v_{0,n} \rangle$ 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
            $P(V_l | V \setminus \{V_l\} = \langle v_{(s-1),1}, \ldots v_{(s-1),(l-1)}, v_{(s-1),(l+1)}, \ldots, v_{(s-1),n} \rangle, e)$
        $sample_s = \langle v_{s,1}, \ldots v_{s,(l-1)}, v_{s,l}, v_{s,(l+1)}, \ldots, v_{s,n} \rangle$
return $list(sample_{burned\_in}, \ldots, sample_{last})$

- Pravděpodobnost nových hodnot $P(V_l | \ldots)$ 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.

Konvergence Gibbs Sampling

Theorem

Pokud

- *každou proměnnou bez evidence vybereme s nenulovou pravděpodobností*
- *v bayesovské síti nejsou nulové pravděpodobnosti*

pak Gibbs sampling konverguje, tj.

$$\lim_{i \to \infty} P(sample_i = \boldsymbol{v}) = P(\boldsymbol{V} = \boldsymbol{v} | E = e) \qquad (\boldsymbol{v} \in sp(\boldsymbol{V})).$$

Problémy:

- Vzorky nejsou nezávislé, tj. chyba se nedá odhadnout 'klasickými' intervaly věrohodnosti.
- Není snadné říct, kolik vzorků potřebujeme.

Výhoda:

- U velkých sítí generuje vzorky výrazně rychleji.

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. $q(\mathbf{v'}|\mathbf{v}) = P(\mathbf{V} = \mathbf{v'}|E = e)$.

Úkoly

Úkol

- Srovnejte 'simulate' a primitivní Metropolitan Hastings simulaci.
- Upravte MH simulaci na Gibbs sampling (tj. nezamítejte, jen měňte se správnou pravděpodobností).
- Porovnejte Gibbs s primitivním MH sampling.

```
 startvalue = c(1,1,1)
burnIn = 0
chain = run_metropolis_MCMC(bnet,startvalue, 100)
x1x2=xtabs(~X1+X2,data.frame(chain[-(1:burnIn),]))
KL.empirical(x1x2,pravda,unit='log2')
```

Učení parametrů

- Pokud známe strukturu a všechny veličiny jsou pozorované, odhad parametrů je (skoro) podíl odpovídajících četností.
- 'skoro' se vztahuje na nulové počty a dělení nulou. Proto máme možnost nastavit vyhlazování smooth=0.0001 - přičte ke všem četnostem, tj. nikde nebude nula.

Úkoly

- Načtěte "two_coins_1.net", naučte model s otočenými hranami a srovnejte s původním.
- Načtěte model 'preg4.net'.
- Ze struktury modelu uberte uzel Ho, děti napojte na Pr.
- Naučte parametry nového modelu ze simulovaných dat z původního modelu.
- Porovnejte (podmíněné) pravděpodobnosti v původním a novém modelu,
- najděte příklad, kdy je P(Pr|evid) různá v obou modelech, i když na uzlu Ho není žádná evidence.

```
novy.dag<-dag(~TwiceAHead,~Penny:TwiceAHead,~Dime:TwiceAHead)
md=grain(novy.dag,data=sim.orig,smooth=0)
```

EM algoritmus

používá se pro odhad nepozorovaných veličin.
Jde o iterativní algoritmus opakující dva kroky:

- **Estimate**, který odhadne hodnoty nepozorovaných dat, a
- **Maximize**, který maximalizuje věrohodnost vzhledem k datům přes uvažované modely.

Proč zahrnovat do modelu neznámé veličiny

Protože se to hodí.

- Známe model, některé veličiny nemůžeme pozorovat.
- Neznámá nepozorovaná veličina zaviní, že vše souvisí se vším.
- Často se používají směsi gausovských rozložení: na klastrování, na popis funkce při zpracování obrazu, atd.

Estimate

- Mám model (z předchozího kroku, na počátku volíme parametry např. náhodně či rovnoměrnou distribuci).
- Pro každý řádek dat:
  - vložím do modelu evidenci na veličinách, které jsem pozorovala,
  - podívám se na pravděpodobnost veličin, které pozorované nebyly,
  - řádek dat rozdrobím na spoustu dílků, každý s jinými hodnotami nepozorovaných veličin, váha dílku odpovídá pravděpodobnosti situace, součet vah drobků je 1.

Maximize

- Vybíráme maximálně věrohodný model pro daná vážená data (z E–kroku)
- bayesovská síť: podíl četností

Obecný EM algoritmus

Máme–li počáteční odhady parametrů $\bar{\theta}^{(0)}$, skryté proměnné Z a pozorovaná *data*, pak můžeme jeden krok EM algoritmu zapsat přiřazením:

$$\bar{\theta}^{(i+1)} \leftarrow argmax_{\bar{\theta}^{(i)}} \sum_{z \in Z} P(Z = z | data, \bar{\theta}^{(i)}) \cdot L(data, Z = z | \bar{\theta}^{(i)})$$

EM algoritmus

- Lze dokázat, že v každém kroku zvýší věrohodnost modelu.
- Nakonec (možná) najde model s větší věrohodností, než má model původní. Data jsou generovaná náhodně a nemusí úplně přesně vystihovat původní model.
- Za jistých předpokladů se dá dokázat, že EM konverguje k maximu, obecně jako každá gradientní metoda může zůstat v lokálním maximu.
- Narozdíl od většiny gradientních metod nemáme parametr velikost kroku.
- Spíš je problém, že ke konci konverguje pomalu, než že by zůstal v lokálním maximu.

EM algoritmus pro bayesovské sítě

- Základní princip je stejný – Estimate a Maximize.
- Příklad: Dva pytle bonbónů někdo smíchal dohromady. Každý bonbón má nějaký obal *Wrapper* a příchuť *Flavor* a buď v něm jsou dírky *Holes*, nebo ne. V každém pytli byl jiný poměr příchutí, jiný poměr děravých bonbónů k neděravým atd.

Příklad se dá popsat jako naivní bayesovský model.

Příklad

Snědli jsme 1000 bonbónů a zapsali, co jsme pozorovali:

	W=red		W=green	
	H=1	H=0	H=1	H=0
F=cherry	273	93	104	90
F=lime	79	100	94	167

Počáteční parametry modelu zvolíme:

$$\theta^{(0)} = 0.6,\ \theta^{(0)}_{F1} = \theta^{(0)}_{W1} = \theta^{(0)}_{H1} = 0.6,\ \theta^{(0)}_{F2} = \theta^{(0)}_{W2} = \theta^{(0)}_{H2} = 0.4$$

BN

- Odhad θ: kdyby byla pozorovaná, spočteme podíl bonbónů z prvního balíčku ke všem bonbónům.
- Protože jí nepozorujeme, **sčítáme očekávané počty**

$$\theta^{(1)} = \frac{1}{N}\sum_{j=1}^{N} \frac{P(flavor_j|Bag=1)P(wrapper_j|Bag=1)P(holes_j|Bag=1)P(Ba}{\sum_{i=1}^{2} P(flavor_j|Bag=i)P(wrapper_j|Bag=i)P(holes_j|Bag=i)P(}$$

(normalizační konstanta dole také záleží na hodnotách parametrů).

Pro bonbón *red*, *cherry*, *holes* dostaneme:

$$\frac{\theta_{F1}^{(0)}\theta_{W1}^{(0)}\theta_{H1}^{(0)}\theta^{(0)}}{\theta_{F1}^{(0)}\theta_{W1}^{(0)}\theta_{H1}^{(0)}\theta^{(0)} + \theta_{F2}^{(0)}\theta_{W2}^{(0)}\theta_{H2}^{(0)}\theta^{(0)}} \approx 0.835055$$

takových bonbónů máme 273, tedy je jejich příspěvek $\frac{273}{N} \cdot 0.835055$.
Podobně spočteme příspěvky dalších sedmi políček a dostaneme:

$$\theta^{(1)} = 0.6124$$

BN

- Odhad θ_{F1} by v plně pozorovaném případě byl ...
- My musíme počítat podíl očekávaných počtů $Bag = 1 \& F = cherry$ a $Bag = 1$, tj.

$$\theta_{F1}^{(1)} = \frac{\sum_{j;Flavor_j=cherry} P(Bag = 1|Flavor_j = cherry, wrapper_j, holes_j)}{\sum_j P(Bag = 1|cherry_j, wrapper_j, holes_j)}$$

- Podobně dostaneme:

$$\theta^{(1)} = 0.6124,\ \theta_{F1}^{(1)} = 0.6684, \theta_{W1}^{(1)} = 0.6483, \theta_{H1}^{(1)} = 0.6558,$$

$$\theta_{F2}^{(1)} = 0.3887, \theta_{W2}^{(1)} = 0.3817, \theta_{H2}^{(1)} = 0.3827$$

BN

Pozn: V Bayesovské síti lze učit parametry tak, že postupně vložíme jeden příklad za druhým a sčítáme pravděpodobnosti pro jednotlivé konfigurace dítěte plus jeho rodičů. Tím dostaneme očekávané četnosti (resp. po vydělení počtem příkladů), z očekávaných četností spočteme parametry podílem odpovídajících četností, tj.

$$\theta_{ijk} \leftarrow \frac{\text{četnost } (X_i = x_{ij} \& pa(X_i) = pa_{ik})}{\text{četnost } (pa(X_i) = pa_{ik})}$$