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.

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. $\langle \frac{1}{4}, \frac{1}{2}, \frac{1}{4} \rangle$.

Základní odhad parametru BN z dat

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

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

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.

$$\widehat{P}(A = a|pa(A) = \langle v_1, \ldots, v_{|pa(A)|}\rangle, e) =$$

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

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 $(w, \langle a_1, \ldots, a_n \rangle)$

$$\widehat{P}(A = a|pa(A) = \langle v_1, \ldots, v_{|pa(A)|} \rangle, e) =$$

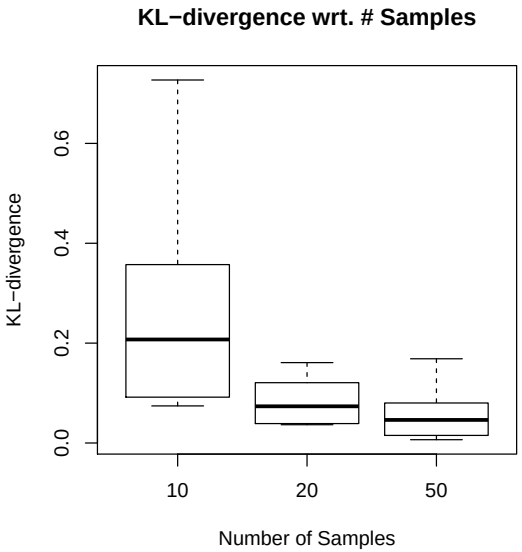
$$= \frac{\sum_{vzorky} w_{vzorek} \cdot \delta_{[A=a \& pa(A)=\langle v_1,\ldots,v_{|pa(A)|}\rangle]} + smooth}{\sum_{vzorky} w_{vzorek} \cdot \delta_{[pa(A)=\langle v_1,\ldots,v_{|pa(A)|}\rangle]} + smooth \cdot |dom(A)|}.$$

KL–divergence - Určení kvality aproximace

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.




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

Markovský obal

Definition (Markov blanket)

Markovský obal (Markov blanket) uzlu A je definován jako množina A, dětí A a rodičů A i jeho dětí.

Theorem

Markovský obal je nejmenší množina, která d-separuje uzel A od všech ostatních uzlů.

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.

Bayesian Learning

Complicated derivation of known things.

- Maximal aposteriory probability hypothesis (MAP) (nejpravděpodobnější hypotéza)
- Maximum likelihood hypothesis (ML) (maximálně věrohodná hypotéza)
- Bayesian optimal prediction (Bayes Rate)
- **EM algorithm**
- **Naive Bayes model (classifier)**

Candy Example (Russel, Norvig: Artif. Intell. a MA)

- Our favorite candy comes in two flavors: cherry and lime, both in the same wrapper.
- They are in a bag in one of following rations of cherry candies and prior probability of bags:

hypothesis (bag type)	h_1	h_2	h_3	h_4	h_5
cherry	100%	75%	50%	25%	0%
prior probability h_i	10%	20%	40%	20%	10%

- The first candy is cherry.

MAP Which of h_i is the most probable given first candy is cherry?

ves estimate What is the probability next candy from the same bag is cherry?

Maximum Aposteriory Probability Hypothesis (MAP)

- We assume large bags of candies, the result of one missing candy in the bag is negligable.
- Recall Bayes formula:

$$P(h_i|B=c) = \frac{P(B=c|h_i)\cdot P(h_i)}{\sum_{j=1,\ldots,5} P(B=c|h_j)\cdot P(h_j)} = \frac{P(B=c|h_i)\cdot P(h_i)}{P(B=c)}$$

- We look for the MAP hypothesis **maximálně aposteriorně pravděpodobná** $argmax_i P(h_i|B=c) = argmax_i P(B=c|h_i)\cdot P(h_i)$.
- Aposteriory probabilities of hypotheses are in the following table.

Candy Example: Aposteriory Probability of Hypotheses

index	prior	cherry ratio	cherry AND h_i	aposteriory prob. h_i
i	$P(h_i)$	$P(B = c\|h_i)$	$P(B = c\|h_i) \cdot P(h_i)$	$P(h_i\|B = c)$
1	0.1	1	0.1	0.2
2	0.2	0.75	0.15	0.3
3	0.4	0.5	0.2	**0.4**
4	0.2	0.25	0.05	0.1
5	0.1	0	0	0

- Which hypothesis is most probable?

$$h_{MAP} = argmax_i P(data|h_i) \cdot P(h_i)$$

- What is the prediction of a new candy according the most probable hypothesis h_{MAP}?

MAP and Penalized Methods

- MAP hypothesis maximizes:

$$h_{MAP} = argmax_i P(data|h_i) \cdot P(h_i)$$

- therefore minimizes:

$$\begin{aligned} h_{MAP} &= argmax_h P(data|h)P(h) \\ &= argmin_h[-log_2 P(data|h) - log_2 P(h)] \\ &= argmin_h[-loglik + \text{complexity penalty}] \\ &= argmin_h[RSS + \text{complexity penalty}] \text{ Gaussian models} \\ &= argmax_h[loglik - \text{complexity penalty}] \text{ Categorical models} \end{aligned}$$

Bayesian Learning, Bayesian Optimal Prediction

- **Bayesian optimal prediction** is weigthed average of predictions of all hypotheses:

$$\begin{aligned} P(N = c|data) &= \sum_{j=1,\ldots,5} P(N = c|h_j, data) \cdot P(h_j|data) \\ &= \sum_{j=1,\ldots,5} P(N = c|h_j) \cdot P(h_j|data) \end{aligned}$$

- If our model is correct, no prediction has smaller expected error then Bayesian optimal prediction.
- We always assume i.i.d. data, independently identically distributed.
- We assume the hypothesis fully describes the data behavior. Observations are mutually conditionally independent given the hypothesis. This allows the last equation above.

Candy Example: Bayesian Optimal Prediction

i	$P(h_i \mid B = c)$	$P(N = c \mid h_i)$	$P(N = c \mid h_i) \cdot P(h_i \mid B = c)$
1	0.2	1	0.2
2	0.3	0.75	0.225
3	0.4	0.5	0.2
4	0.1	0.25	0.02
5	0	0	0
$\sum$	1		0.645

Maximum Likelihood Estimate (ML)

- Usually, we do not know prior probabilities of hypotheses.
- Setting all prior probabilities equal leads to **Maximum Likelihood Estimate, maximálně věrohodný odhad**

$$h_{ML} = argmax_i P(data|h_i)$$

- Probability of data given hypothesis = likelihood of hypothesis given data.
- Find the ML estimate:

index	prior	cherry ration	cherry AND h_i	Aposteriory prob. h_i
i	$P(h_i)$	$P(B = c\|h_i)$	$P(B = c\|h_i) \cdot P(h_i)$	$P(h_i\|B = c)$
1	0.1	1	0.1	0.2
2	0.2	0.75	0.15	0.3
3	0.4	0.5	0.2	0.4
4	0.2	0.25	0.05	0.1
5	0.1	0	0	0

- In this example, do you prefer ML estimate or MAP estimate?
- (Only few data, overfitting, penalization is usefull. AIC, BIC)

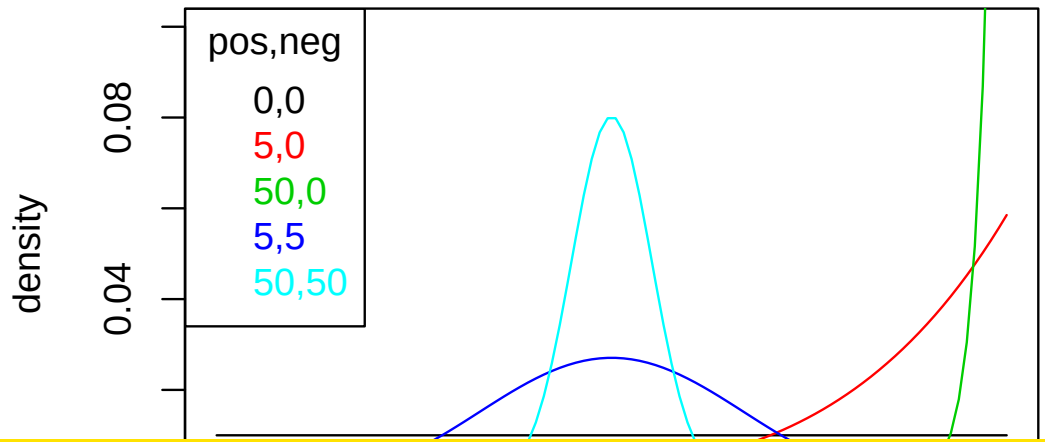
Remark: Bayesian Parameter Learning

- We represent probability distribution on parameters.
- For binary features, Beta function is used, a is the number of positive examples, b the number of negative examples.

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

- (For categorical features, Dirichlet priors and multinomial distribution is used. (Dirichlet-multinomial distribution).
- For Gaussian, μ has Gaussian prior, $\frac{1}{\sigma}$ has gamma prior (to stay in exponential family).)

Beta Function:



Maximum Likelihood: Continuous Parameter θ

- New producer on the market. We do not know the ratios of candies, any h_θ, kde $\theta \in \langle 0; 1 \rangle$ is possible, any prior probabilities h_θ are possible.
- We look for maximum likelihood estimate.
- For a given hypothesis h_θ, the probability of a cherry candy is θ, of a lime candy $1 - \theta$.
- Probability of a sequence of c cherry and l lime candies is:

$$P(data|h_\theta) = \theta^c \cdot (1 - \theta)^l.$$

ML Estimate of Parameter θ

- Probability of a sequence of c cherry and l lime candies is:

$$P(data|h_\theta) = \theta^c \cdot (1-\theta)^l$$

- Usual trick is to take logarithm:

$$LL(h_\theta; data) = c \cdot \log_2 \theta + l \cdot \log_2(1-\theta)$$

- To find the maximum of LL (log likelihood of the hypothesis) with respect to θ we set the derivative equal to 0:

$$\frac{\partial LL(h_\theta; data)}{\partial \theta} = \frac{c}{\theta} - \frac{l}{1-\theta}$$

$$\frac{c}{\theta} = \frac{l}{1-\theta}$$

$$\theta = \frac{c}{c+l}.$$

ML Estimate of Multiple Parameters

- Producer introduced two colors of wrappers - red r and green g.
- Both flavors are wrapped in both wrappers, but with different probability of the red/green wrapper.
- We need three parameters to model this situation:

$P(B = c)$	$P(W = r \mid B = c)$	$P(W = r \mid B = l)$
θ_0	θ_1	θ_2

- Following table denotes observed frequences:

wrapper\ flavor	cherry	lime
red	r_c	r_l
green	g_c	g_l

ML Estimate of Multiple Parameters

Parameters are:

$P(B=c)$	$P(W=r\|B=c)$	$P(W=r\|B=l)$
θ_0	θ_1	θ_2

Probability of data given the hypothesis $h_{\theta_0,\theta_1,\theta_2}$ is:

$$\begin{aligned} P(data|h_{\theta_0,\theta_1,\theta_2}) &= \theta_1^{r_c} \cdot (1-\theta_1)^{g_c} \cdot \theta_0^{r_c+g_c} \cdot \theta_2^{r_l} \cdot (1-\theta_2)^{g_l} \cdot (1-\theta_0)^{r_l+g_l} \\ LL(h_{\theta_0,\theta_1,\theta_2}; data) &= r_c \log_2 \theta_1 + g_c \log_2(1-\theta_1) + (r_c+g_c)\log_2 \theta_0 \\ &\quad + r_l \log_2 \theta_2 + g_l \log_2(1-\theta_2) + (r_l+g_l)\log_2(1-\theta_0) \end{aligned}$$

We look for maximum:

$$\begin{aligned} \frac{\partial LL(h_{\theta_0,\theta_1,\theta_2}; data)}{\partial \theta_0} &= \frac{r_c+g_c}{\theta_0} - \frac{r_l+g_l}{1-\theta_0} \\ \theta_0 &= \frac{(r_c+g_c)}{r_c+g_c+r_l+g_l} \\ \frac{\partial LL(h_{\theta_0,\theta_1,\theta_2}; data)}{\partial \theta_2} &= \frac{r_l}{\theta_2} - \frac{g_l}{1-\theta_2} \\ \theta_2 &= \frac{r_l}{r_l+g_l}. \end{aligned}$$

Discrete Variables

- Maximum Likelihood estimate is the ratio of fequences.
- **Naive Bayes Model, Bayes Classifier** assumes independent features given the class variable.
  - Caculate prior probability of classes $P(c_i)$
  - For each feature f, calculate for each class the probability of this feature $P(f|c_i)$
  - For a new observation of features f predict the most probable class $argmax_{c_i} P(f|c_i) \cdot P(c_i)$.
- Bayesian Networks learn more complex (in)dependencies between features.

Missing data (T.D. Nielsen)

Die tossed N times. Result reported via noisy telephone line. When transmission not clearly audible, record missing value:

$$4, 2, ?, 6, 5, 4, ?, 3, 4, 1, \ldots$$

"2" and "3" sound similar, therefore:

$$P(Y_i = ?|X_i = k) = P(M_i = 1|X_i = k) = \begin{cases} 1/4 & \text{k} = 2,3 \\ 1/8 & \text{k} = 1,4,5,6 \end{cases}$$

Distribution of the Y is (for fair die):

?	$\frac{1}{3}\frac{1}{4} + \frac{2}{3}\frac{1}{8} = \frac{1}{6}$
2,3	$\frac{1}{6}\frac{3}{4} = \frac{1}{8}$
1,4,5,6	$\frac{1}{6}\frac{7}{8} = \frac{7}{48}$

If we simply ignore the missing data items, we obtain as the maximum likelihood estimate for the parameters of the die:

$$\theta^* = (\frac{7}{48}, \frac{1}{8}, \frac{1}{8}, \frac{7}{48}, \frac{7}{48}, \frac{7}{48}) * \frac{6}{5} = (0.175, 0.15, 0.15, 0.175, 0.175, 0.175)$$

Incomplete data

How do we handle cases with missing values:

- Faulty sensor readings.
- Values have been intentionally removed.
- Some variables may be unobservable.

How is the data missing?
We need to take into account how the data is missing:

- **Missing completely at random** The probability that a value is missing is independent of both the observed and unobserved values (a monitoring system that is not completely stable and where some sensor values are not stored properly).
- **Missing at random** The probability that a value is missing depends only on the observed values (a database containing the results of two tests, where the second test has only performed (as a "backup test") when the result of the first test was negative).
- **Non-ignorable** Neither MAR nor MCAR (an exit poll, where an extreme right-wing party is running for parlament).

EM - Algorithm

- EM algorithm is used for learning a model with unobserved variables (for example, cluster membership).
- We assume (hope) they are missing at random.
- It is an iterative algorithm with two steps:
  - **Estimate**, fills in the unobserved data based on current M model, and
  - **Maximize**, finds maximum (log)likelihood model given the data filled in E step.

Example: T.D. Nielsen

Learning by EM - Algorithm

- Clustering (observed may be of categorical and/or continuous)
- Hidden Markov Models
- Latent Dirichlet Allocation
- Hierarchical Mixtures of Experts
- and others.

ML Estimate of Gaussian Distribution Parameters

- Assume x to have Gaussian distribution with unknown parameters μ a σ.
- Our hypotheses are $h_{\mu,\sigma} = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$.
- We have observed $x_1, \ldots, x_n$.
- Log likelihood is:

$$\begin{aligned} LL &= \sum_{j=1}^{N} log \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \\ &= N \cdot (log \frac{1}{\sqrt{2\pi}\sigma}) - \sum_{j=1}^{N} \frac{(x_j - \mu)^2}{2\sigma^2} \end{aligned}$$

- Find the maximum.

Linear Gaussian Distribution

- Assume random variable (feature) X.
- Assume goal variable Y with linear gaussian distribution where $\mu = b \cdot x + b_0$ and fixed variance σ^2 $p(Y|X = x) = N(b \cdot x + b_0; \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(y-((b \cdot x + b_0))^2}{2\sigma^2}}$.
- Find maximum likelihood estimate of b, b_0 given a set of observations $data = \{\langle x_1, y_1 \rangle, \ldots, \langle x_N, y_N \rangle\}$.
- (Look for maximum of the logarithm of it; change the max to min with the opostite sign. Do you know this formula?)

$$argmax_{b,b_0}(log_e(\Pi_{i=1}^{N}(e^{-(y_i-(b \cdot x_i+b_0))^2}))) = argmin_{b,b_0}(?)$$

Reasons for Modelling Unobserved Variables

- We know the model structure, observations are missing.
- Unobserved variable makes many features conditionally independent (that is, simplifies the model).
- Often, mixtures of Gaussians are used. It is also our example: clustering.
- Also used to learn Hidden Markov Models.

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.

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.

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.

Příklad EM algoritmu pro bayesovské sítě

- 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**

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

(normalizační konstanta dole také záleží na hodnotách parametrů).

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

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

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:

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

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

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.

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