Výpočet marginálních podmíněných pravděpodobností v bayesovské síti

Úmluva: Zajímáme se pouze o bayesovské sítě, jejichž graf je spojitý.
Jinak uvažujeme každou komponentu zvlášť.

Tabulky, součin tabulek

Sdružená distribuce definovaná bayesovskou sítí je formátu:

$$P(A_1, A_2, \ldots, A_n) = \Pi_{i=1}^{n} P(A_i | pa(A_i))$$

Podmíněným pravděpodobnostem $P(A_i|pa(A_i))$ budu říkat **tabulky** (potentials).

Součinem tabulek $\phi_1(A, B, C)$ a $\phi_2(B, C, D)$ rozumím tabulku nad sjednocením domén, tj. A, B, C, D, kde se hodnota každého políčka $A = a, B = b, C = c, D = d$ vypočte jako součin odpovídajících políček tabulek ϕ_1, ϕ_2, tj.

$$\phi(A = a, B = b, C = c, D = d) = \phi_1(A = a, B = b, C = c) \cdot \phi_2(B = b, C = c, D =$$

Evidence (hard, jistá evidence)

Pozorováním zjistíme hodnotu některých veličin, např. $A = yes, D = red$ Pak ve všech tabulkách obsahujících veličiny s evidencí snížíme dimenzi vyškrtáním "řádků" (mnoharozměrných), které odpovídají ostatním hodnotám veličiny.

Např.$P(A = a, B, C, D = d) = P(A = a) \cdot P(B|A = a) \cdot P(C|B, A = a) \cdot P(D = d|A$

Pozn: $P(A = a, \ldots)$ budeme zkracovat $P(a, \ldots)$, tj. je–li z kontextu (pořadí) zřejmé, neuvádíme jméno veličiny, ale rovnou její hodnotu.

Alternativně: **Evidence** na A je vektor dimenze rovné počtu možných hodnot A, který obsahuje jednu jedničku a jinak samé nuly. **Vložením evidence** míníme vynásobení sdružené pravděpodobnostní distribuce vektorem evidence.

V tomto případě marginalizujeme i přes proměnnou A – tato marginalizace spočívá s sečtení množství nul a (nejvýše) jednoho nenulového čísla pro každé políčko.

První, méně formální definicí si ušetříme manipulaci se spoustou nul a marginalizaci přes proměnné s evidencí.

Marginály podmíněné evidencí

Nás budou zajímat marginály podmíněné evidencí, tj.

Např.$P(B|A = a, D = d) = \frac{\sum_C P(A = a) \cdot P(B|A = a) \cdot P(C|B, A = a) \cdot P(D = d|}{P(A = a, D = d)}$

Algoritmus eliminace proměnných

- Do seznamu Φ_1 dáme všechny tabulky $P(A_i|pa(A_i), e)$, v každé tabulce odstraním "řádky" nekonzistentní s evidencí, tj. s nulovou pravděpodobností.
- Postupně budeme eliminovat (následujícím algoritmem) všechny proměnné bez evidence, které nás nezajímají (dostaneme $P(A, e)$).
- Nakonec eliminujeme i zbývající proměnné bez evidence, čímž spočteme normalizační konstatnu $\alpha = P(e)$; touto konstantou vydělíme tabulku z předchozího kroku a dostaneme podmíněnou pravděpodobnost $P(A|e)$.

Eliminace proměnné X v kroku i znamená:

1. Vyber z Φ_i všechny tabulky, které mají v doméně X, dej je do Φ_X.
2. Spočti $\phi = \sum_X \prod_{T \in \Phi_X} T$
3. Nové Φ_{i+1} se rovná: $\Phi_i \setminus \Phi_X \cup \{\phi\}$

Pozn: Pokud v Φ_{last} nakonec zbyde více tabulek, musíme je vynásobit.

Pozn2: A může být buď veličina, nebo množina veličin.

Charakteristika algoritmu Eliminace proměnných

- Snadný na pochopení a implementaci.
- Problém: v jakém pořadí eliminovat? Špatné pořadí vede ke zbytečně velkým tabulkám ϕ.
- Pokud nás zajímají **všechny jednorozměrné** marginály, tak bychom nemuseli počítat vše pro každou zvlášť, dost výpočtů se opakuje.

Proto většina software používá jiné algoritmy, my se podíváme, co používá Hugin, ostatní mají různé modifikace.

Příklad: Špatné a lepší pořadí eliminace.

- Různým pořadím eliminace odpovídají různé mezivýsledné tabulky.
- Složitost výpočtu zhruba odpovídá velikosti největší mezivýsledné tabulky.
- Naším cílem je navrhnout pořadí eliminace tak, aby maximální mezivýsledná tabulka byla co nejmenší.

Graf domén

Graf domén v konkrétním kroku eliminace proměnných je takový graf, kde

- uzly jsou právě všechny dosud neeliminované proměnné,
- dva uzly jsou spojeny hranou právě když se odpovídající proměnné vyskytují v aspoň jedné tabulce zároveň.

Moralizace

Moralizací grafu bayesovské sítě rozumíme následující dva kroky:

- Oženit (spojit hranou) rodiče společných dětí. (Neboli: spojit hranou každé dva vrcholy, které se společně vyskytují v některé tabulce dané bayesovské sítě.)
- Zapomenout orientaci hran, tj. vytvořit neorientovaný graf se stejnými hranami.

Graf domén je na počátku moralizovaná bayesovská síť. **Po eliminaci každé proměnné odstraníme jí příslušející uzel a spojíme všechny jeho sousedy** (nově přidané hrany se nazývají **doplněné**, fill–in).

Naším cílem jsou co nejmenší domény, tj. co nejméně doplněných hran.

Perfektní eliminační posloupnost

Perfektní eliminační posloupnost je taková posloupnost eliminace všech proměnných bayesovské sítě, která nevynucuje žádné doplněné hrany.

Lemma 1 *Nechť je $X_1, \ldots, X_k$ perfektní eliminační posloupnost, X_j uzel, jehož každí dva sousedi jsou spojeni hranou. Pak je posloupnost $X_j, X_1, \ldots, X_{j-1}, X_{j+1}, \ldots, X_k$ také perfektní.*

Eliminace X_j nepřidá žádnou hranu, tj. nikomu nepřidá souseda, a při eliminaci se každý stará jen o své sousedy, tj. se na eliminaci ostatních nic nezmění (ledaže by nemuseli přidávat hranu do X_j, ale i původní posloupnost byla perfektní).

Množina maximálních domén je množina všech domén tabulek, vzniklých během výpočtu, ze které vyřadíme ty domény, které jsou vlastní podmnožinou jiného prvku této množiny.

Lemma 2 *Všechny perfektní posloupnosti vytvářejí stejnou množinu maximálních domén, a to* ***množinu klik*** *moralizovaného grafu.*

- Kliky tam musí být, neboť tabulka jejich domény vznikne při eliminaci první proměnné z kliky.
- Nic většího tam nemůže být, neboť by to způsobilo doplněné hranu.

Triangulované grafy

POZOR, něco jiného než triangulovaný planární graf!!!

Pozn: Pro některé moralizované grafy neexistuje perfektní posloupnost.

Graf je **triangulovaný**, pokud pokud pro něj existuje perfektní eliminační posloupnost.

Lemma 3 (Alternativní definice) *Graf je triangulovaný, pokud každý jeho cyklus délky větší než tři má aspoň jednu tětivu.*

Příklady na tabuli.

Notace

Nechť X je uzel neorientovaného grafu. Pak mají označení následující význam:

N_X	sousedé X
F_X	sousedé X včetně X samého (family)

Podgraf je **úplný**, pokud jsou každé dva jeho uzly spojeny hranou.

Vrchol X je **simpliciální**, pokud je N_X úplný podgraf.

Ekvivalentně: vrchol X je simpliciální, je–li F_X klika.

Lemma 4 *Nechť je G triangulovaný graf a X simpliciální uzel. Pak graf $G^{|}$ získaný eliminací X z G je také triangulovaný.*

Důsledek lemmatu 1.

Theorem 1 *Triangulovaný graf s aspoň dvěma vrcholy má aspoň dva simpliciální uzly.*

Navíc: pokud není úplný, tak má aspoň dva simpliciální uzly, které nejsou spojeny hranou.

Důkaz indukcí podle počtu vrcholů.

Pro tři vrcholy platí.

Pro více: První uzel perfektní posloupnosti je simpliciální, vzniklý graf je triangulovaný. Z indukčního předpokladu má aspoň dva nesousední simpliciální uzly, proto aspoň jeden z nich nesousedil s eliminovaným uzlem (sousedi byli propojeni).

Lemma 5 *Pro každý vrchol A triangulovaného grafu existuje perfektní posloupnost, kde je A poslední prvek.*

Důkaz: Vždy eliminuj simpliciální uzel různý od A.

Theorem 2 *Neorientovaný graf je triangulovaný právě když můžeme eliminovat všechny uzly tak, že vždy eliminujeme simpliciální uzel.*

- Je–li graf triangulovaný, eliminací simpliciálního uzlu vznikne opět triangulovaný graf a můžeme pokračovat v eliminaci simpliciálních uzlů.

- Eliminací simpliciálních uzlů tvoříme perfektní posloupnost, tj. graf je triangulovaný.

Stromy spojení (Join trees)

Mějme množinu klik neorientovaného grafu G, kliky jsou organizovány do stromu T. T je **strom spojení**, pokud pro každé dva vrcholy $V, W \in T$ všechny uzly na cestě z V do W obsahují průnik $W \cap V$. Průnik dvou sousedních uzlů nazveme **separátor** těchto uzlů, separátorem V a W je $S_{V,W} = V \cap W$.

Theorem 3 *Pokud kliky grafu G lze organizovat do stromu spojení, pak je G triangulovaný.*

- Vezměme list stromu spojení V, který sousedí jen s W.
- V průnik libovolný uzel je částí W, proto V obsahuje uzel, který není v žádné jiné klice. Ten eliminujeme.
- Pokud byl poslední z $W \setminus V$, odstraníme uzel V a dostaneme zase strom spojení, který má list, pokračujeme prvním bodem.

Theorem 4 *Pokud je G triangulovaný, pak kliky grafu G lze organizovat do stromu spojení.*

Zkonstruujeme strom spojení. Index i je na počátku 1, pak určuje, kolikátou kliku jsme utvořili.

- Začnu eliminací simpliciálního uzlu X, jeho rodina F_X je klika (označíme jí C_i). Pokračuji eliminací všech uzlů, které mají své sousedy pouze v této klice. Zbylé uzly z F_X tvoří separátor, označíme S_i.
- V grafu vzniklém eliminací vyberu další simpliciální uzel a eliminuji dále dle předchozího bodu, dokud neeliminuji všechny kliky.
- Každý separátor S_i připojím k nějaké klice C_j, $j > i$ takové, že $S_i \subset C_j$. S_i je úplná, první uzel z ní eliminovaný musí být v klice – nadmnožině S_i.

Vznikl nám strom spojení?

Graf má $n-1$ hran, je–li souvislý, je strom.

Cesta z C_i do C_j, $j > i$ obsahuje $X \in C_i \cap C_j$, protože X musí být z definice separátoru $X \in S_i$ (S_i neeliminované uzly C_i), a tak dále až do S_{j-1}.

Pokud byl graf G souvislý, vlastnost průniku na cestě nám zaručuje souvislost takto generovaného grafu, proto se jedná o strom.

(Od úvodního slajdu předpokládáme souvislý G. Jinak dostaneme les stromů spojení pro jednotlivé komponenty.)

Strom spojení

Používám termín **strom spojení** ve třech významech:

- viz definice výše, strom klik splňující vlastnost průniků
- strom dle definice výše, kde jsou navíc hrany označeny separátory
- strom dle definice výše, kde je navíc v každé klice ”schránka” na seznam pravděpodobnostních tabulek a v každém separátoru jsou dvě schránky na zprávy – tabulky – jdoucí jednotlivými směry. Tomuto se říká **junction tree**.

Strom spojení reprezentující bayesovskou síť

Mějme bayesovskou síť s množinou pravděpodobnostních tabulek Φ a evidenci e. Nechť množina tabulek Φ_e vznikne z Φ vložením evidence e do příslušných tabulek, tj. "vyříznutím" konkrétních "řádků" v pravděpodobnostních tabulkách.

Strom spojení reprezentuje bayesovskou síť s evidencí e, pokud každou tabulku $\phi \in \Phi_e$ přiřadíme do schránky některé z klik C_i takových, že $dom(\phi) \subseteq C_i$.

Pozn: pokud strom spojení vznikl z moralizovaného a triangularizovaného grafu bayesovské sítě, tak takové klika vždy existuje.

Pokud moralizovaný graf není triangulovaný, doplníme ho hranami na triangulovaný a z něj vytvoříme strom spojení.

Propagace ve stromu spojení

- Propagace (výpočet) ve stromu spojení spočívá v posílání zpráv, kterými se postupně plní schránky separátorů.
- Každý uzel (klika) posílá v každém směru právě jednu zprávu.
- Uzel (klika) může poslat zprávu v daném směru, pokud už ze všech ostatních směrů zprávy dostala.
- Protože se jedná o strom, vždycky někdo může poslat zprávu, nebo jsou již všechny schránky plné.

Poslání zprávy

Uvažujme kliku C se sousedními separátory $S_1, \ldots, S_k$, směr separátoru S_1 (bez újmy na obecnosti). **Poslat zprávu** z C do S_1 znamená zapsat do odchozí schránky S_1 tabulku, která vznikne součinem příchozích zpráv v separátorech $S_2, \ldots, S_k$ a tabulek obsažených v C. Tento součin marginalizujeme přes všechny veličiny $C \setminus S_1$ a výsledek zapíšeme do S_1.

Theorem 5 *Nechť strom spojení reprezentuje bayesovskou síť a evidenci e, všechny schánky byly naplněny. Potom:*

- *Nechť V je klika obsahující tabulky Φ_V a k ní směřující separátory $S_1, \ldots, S_k$ obsahují zprávy $\phi_1, \ldots, \phi_k$.*

$$P(V, e) = \Pi_{\phi \in \Phi_V} \phi \cdot \Pi_{i=1}^{k} \phi_i$$

- *Nechť S je separátor se zprávami ϕ_1, ϕ_2.*

$$P(S, e) = \phi_1 \cdot \phi_2$$

Zprávy směřující do V odpovídají perfektní eliminační posloupnosti, která má V na svém konci.

Pro separátor, odchozí zpráva vznikla marginalizací z V, jen tam nebyla započtena zpráva přicházející z tohoto směru.

$$\begin{aligned}
P(S_1, e) &= \sum_{V \setminus S_1} P(V, e) = \sum_{V \setminus S_1} (\Pi_{\phi \in \Phi_V} \phi \cdot \Pi_{i=1}^{k} \phi_i) \\
&= \sum_{V \setminus S_1} (\Pi_{\phi \in \Phi_V} \cdot \Pi_{i=2}^{k} \phi_i \cdot \phi_1) \\
&= (\sum_{V \setminus S_1} \Pi_{\phi \in \Phi_V} \cdot \Pi_{i=2}^{k} \phi_i) \cdot \phi_1
\end{aligned}$$

což je odchozí krát příchozí zpráva. Poslední řádek plyne z toho, že $dom(\phi_1) = S$.

Výpočet pomocí stromu spojení (shrnutí)

- BN moralizujeme
- doplníme hrany na triangulovaný graf
- vytvoříme strom spojení
- naplníme tabulkami
- vypočteme posíláním zpráv
- pravděpodobnost na veličině A zjistíme tak, že najdeme libovolnou kliku C obsahující A a marginalizujeme, tj. $P(A, e) = \sum_{C \setminus A} P(C, e)$
- pokud nás zajímá sdružená distibuce na množině, která není částí žádné kliky, máme smůlu (musíme použít Eliminaci proměnných. Pokud jde, eliminujeme simpliciální uzly, které nás nezajímají, jinak cokoli (heuristika).

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:

$$\begin{aligned} 0,5 + 3y + 2v &= 1 \\ v &= 0,25 - 1,5y \end{aligned}$$

Přibližný výpočet bayesovské sítě

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

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