Učení bayesovských sítí

Žádný vzorec na těchto slajdech se nezkouší. Jde o idee.

Bayesovská síť na univerzu U je definovaná:

- orientovaným grafem G, který neobsahuje orientované cykly
- seznamem pravděpodobnostních tabulek, pro každé $X \in U$ tabulka $P(X_i|pa(X_i))$, kde $pa(X_i)$ značí rodiče X_i, tj. uzly, ze kterých vede do X_i hrana.

Známe strukturu, učíme parametry, máme úplná data

- Známe strukturu bayesovské sítě a potřebujeme naučit hodnoty v pravděpodobnostních tabulkách.
- Máme data, tj. hodně případů, pro které známe hodnoty jednotlivých X_i.
- Parametry (hodnoty podmíněných pravděpodobností) můžeme určit následujícím přímým výpočtem:
- pravděpodobnost, že veličina X_i nabyde hodnoty v_i při konfiguraci jejích rodičů $pa(X_i) = pa_i$ se spočte jako podíl

$$\frac{|(X_i = v_i)\&(pa(X_i) = pa_i)|}{|(pa(X_i) = pa_i)|}$$

- takto získáme maximálně věrohodný odhad (který nám říká, že to je v jistém smyslu nejlepší model, který za daných okolností můžeme získat).
- Protože je často problém s tím, když dáme pravděpodobnost 0 velmi řídkému jevu, v praxi často přičítáme 1 pozorování ke každé alternativě, tedy, nabývá–li X_i jednu z k možných hodnot, dostaneme odhad:

$$\frac{|(X_i = v_i)(pa(X_i) = pa_i)| + 1}{|(X_i = v_i)(pa(X_i) = pa_i)| + k}$$

- tento odhad pro počet vzorků jdoucí do nekonečna jde k předchozímu (maximálně věrohodnému) odhadu.

Známe strukturu, učíme parametry, nemáme úplná data

- Výpočet podle četností na datech, které známe, by mohl vést k nekonzistentnímu výsledku,
- používá se iterativní algoritmus na odhad chybějících dat, tzv. EM algoritmus.

Hledáme strukturu

Pro hledání struktury potřebujeme

- prostor možných modelů (všechny DAGy)
- ohodnocovací funkci (**score**)
- strategii prohledávání

Prostor všech DAGů

je VELIKÝ.

- Pro jeden uzel existuje jen jeden DAG.
- Pro tři uzly 25 DAGů.
- Pro devět uzlů 1 213 442 454 842 881 DAGů.

Reálné problémy mají sto/sta uzlů, tj. celý prostor neprojdeme v reálném čase.

Ohodnocovací funkce

- Předpokládáme data **iid**, tj. nezávisle generovaná podle stejného pravděpodobnostního rozložení (např. nám neznámé bayesovské sítě)
- vhodný by byl **nejpravděpodobnější model** za podmínky pozorovaných dat $argmax_M P(M|D)$.
- Dle Bayesova vzorce přepíšeme:

$$P(M|D) = \frac{P(D|M) \cdot P(M)}{P(D)}$$

- $P(D)$ nás nezajímá, protože je pro všechny modely stejné. Kdyby nás zajímalo, je $\sum_M P(D|M) \cdot P(M)$.
- $P(M)$ můžeme volit rovnoměrnou, tj. $\frac{1}{|M|}$, tj. maximalizujeme

věrohodnost (likelihood). Alternativou je **penalizovat složité modely**, abychom se vyhnuli přeučení.

- $P(D|M) = \Pi_{d \in D} P(d|M)$, tj. součin pravděpodobností jednotlivých příkladů v datech v daném modelu. Jenže tady musíme průměrovat (integrovat) přes všechny možné hodnoty parametrů modelu.

Používané ohodnocovací funkce

f značí počet parametrů sítě, LL značí logaritmus věrohodnosti, r_i počet stavů veličiny X_i, q_i počet konfigurací $pa(X_i)$,n počet veličin (uzlů), N_{ijk} počet příkladů mající odpovídající hodnoty veličiny X_i a jejích rodičů, $N_{ij} = \sum_k N_{ijk}$

- bayesovský přístup: předpokládáme Dirichletovské rozložení, ”průměrujeme” přes možné hodnoty parametrů vážené jejich pravděpodobností

$$P(D|M) = \Pi_{i=1}^{n} \Pi_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \Pi_{k=1}^{r_i} N_{ijk}!$$

- zobecnění pro apriorní hodnoty parametrů α_{ijk}:

$$P(D,M) = \Pi_{i=1}^{n} P(M) \Pi_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \Pi_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}$$

- AIC: $LL(D|L) - f$
- BIC: $LL(D|L) - log(|D|) \cdot \frac{f}{2}$
- pro velká data BIC a bayesovský přístup ekvivalentní

Prohledávání

- K2 algoritmus: postupně přidává hrany
- genetické algoritmy
- jinak
- PC algoritmus, založený na postupném ”ubírání” hran na základě testů nezávislosti (není založen na ohodnocovací funkci pro celý model).

K2 algoritmus

- Předpokládá předem dané (známé, zvolené) uspořádání veličin.
- Používá skórovací funkci:

$$P(D|M) = \Pi_{i=1}^{n} \Pi_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \Pi_{k=1}^{r_i} N_{ijk}!$$

  příspěvek uzlu i tedy je:

$$g(i, M) = \Pi_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \Pi_{k=1}^{r_i} N_{ijk}!$$

- postupně prochází uzly v daném pořadí
- pro každý uzel hledá rodiče, který by nejvíce zlepšil $g(i, M)$
- pokud už žádná hrana $g(i, M)$ nezlepší, hledáme rodiče pro další uzel.

K2 nezaručuje nalezení optimální struktury. I když přidání jedné hrany skóre nezlepší, přidání dvou hran současně může skóre zlepšit.

Genetické algoritmy

- DAG se reprezentuje maticí
- za pevně daného uspořádání vrcholů je prostor DAGů uzavřený na křížení
- obecně křížením můžeme dostat graf s orientovaným cyklem
- proto jsou potřeba opravné operátory
- a GA nefungují až tak dobře, jak bychom chtěli, ale celkem fungovaly.

Ekvivalentní DAGy

- Různé DAGy reprezentují stejnou strukturu podmíněných nezávislostí.
- Uvažovat je jako různé modely má řadu nevýhod:
  - větší prostor na prohledávání
  - proč by měla mít struktura nezávislostí reprezentovaná více DAGy větší apriorní pravděpodobnost?
  - pokud ohodnocovací funkce dává jiné skóre ekvivalentním DAGům, je podezřelá (nemáme důvod nějakou z ekvivalentních reprezentací preferovat).

Třídy ekvivalentních DAGů

- Ekvivalentní DAGy mají stejné **imoralities** (šipky směřující do stejného dítěte, jejichž počátky nejsou spojeny hranou) a stejný **skeleton** (hrany, pokud zapomeneme jejich orientaci).
- Třída ekvivalence se dá reprezentovat tzv. esenciálním grafem. Ten vznikne položením všech ekvivalentních grafů na sebe a zrušením orientace těch hran, které jsou v různých modelech orientovány různě.
- V rámci třídy ekvivalence se můžeme pohybovat pomocí **otáčení pokrytých hran**. Hrana $X \rightarrow Y$ je pokrytá, pokud $pa(X) \cup \{X\} = pa(Y)$, tj. pokud mají uzly X a Y stejné rodiče.

Překvapení (Meek, Chickering 2002)

Za předpokladů:

- data byly generovány rozložením odpovídajícím nějaké bayesovské síti
- máme dost dat

Lze:

- definovat řídký prostor (každý uzel má rozumně mnoho sousedů)
- na kterém greedy prohledávání (vem vždy nejslibnějšího souseda)
- najde správný generující model.

Prohledávání

- Prohledáváme prostor tříd ekvivalentích DAGů.
- Začneme DAGem bez jediné hrany.
- Uvažujeme všechny sousedy, které vygenerujeme přidáním hrany k některému DAGu z naší třídy ekvivalence (složitě se ukáže, že to lze).
- Vybereme souseda, který zlepší skóre.
- Až se dostaneme do lokálního maxima, začneme hrany zase ubírat (pokud to zlepší skóre).
- Když už nejde nic ubrat (a byly splněny předpoklady na ohodnocovací funkci, generující distribuci a měli jsme dost dat), máme správný model.