Pravděpodobnostní usuzování v čase

Markovské procesy

- příklad: diabetický pacient, hladina inzulinu, léky, jídlo
- předpokládáme, že se množina možných stavů S nemění v průběhu času
- předpokládáme diskrétní čas
- předpokládáme stacionární proces – pravidla přechodu z času t na $t + 1$ nezáleží na čase

Markovská vlastnost

- Markovská vlastnost: aktuální stav závisí jen na konečném počtu předchozích kroků (n–krocích, $n > 0$), tj. stav v čase $t+1$ je nezávislý na stavu v čase $t-n$, při znalosti stavů v mezičase $t, \ldots, t-(i-1)$, tj.

$$S_{t+1} \perp\!\!\!\perp S_{t-n} | S_t, \ldots, S_{t-(n-1)}$$

- Pro Markovský model 1. řádu (často jen Markovský proces, Markovský model)

$$S_{t+1} \perp\!\!\!\perp S_{t-1} | S_t$$

- Pro Markovský proces 2. řádu:

$$S_{t+1} \perp\!\!\!\perp S_{t-2} | S_t, S_{t-1}$$

Pro zadání MP 1. řádu tedy potřebujeme:

- $P(S_0)$
- $P(S_t | S_{t-1})$ pro jedno t, pro ostatní stejná

Skrytý Markovský proces (HMM)

= částečně pozorovaný markovský proces

- Aktuální stav S nemůžeme pozorovat přímo, ale jen částečně
- pozorování O závisí pouze na aktuálním stavu, nikoli na minulosti či budoucnosti
- pro zadání definujeme $P(O_t|S_t)$

Za daných předpokladů máme vlastně speciální bayesovskou síť, tj. pro každé konečné t

$$P(S_0, S_1, \ldots, S_t, O_1, \ldots, O_t) = P(S_0) \cdot \Pi_{i=1}^{t} P(S_t|S_{t-1}) \cdot P(O_t|S_t)$$

Je proces markovský?

- jak který, často ano, nebo zhruba – náhodná procházka, stav pacienta, pozice robota
- pokud ne, tak buď můžeme zkusit zvýšit řád Markovského procesu nebo rozšířit množinu stavů
- nové stavy ale musíme také predikovat, tj. potřebujeme porozumět procesu, který modelujeme
- nebo mít dobré senzory, které nám dobře určí aktuální stav

Příklad: Chřipka trvá 7 dní.
Robot v mřížce: pozice, rychlost, stav baterie.

Inference

Vše je specielní případ propagace v bayesovské síti.

- Filtrování (=monitorování) – zajímá mě pravděpodobnostní rozložení (marginála) na aktuálním stavu $P(S_t|o_1,\ldots,o_t)$
- predikce – zajímá nás budoucí stav, $P(S_{t+k}|o_1,\ldots,o_t)$,$k>0$
- vyhlazování, pohled zpět: nová evidence nám přináší informace o tom, kde jsme byli v minulém čase, $P(S_k|o_1,\ldots,o_t)$,$t>k\geq 0$
- nejpravděpodobnější průchod (vysvětlení)

$$argmax_{s_0,\ldots,s_t} P(s_0,\ldots,s_t|o_0,\ldots,o_t)$$

Filtrování

Definuji: $o_{1:t} = o_1, \ldots, o_t$.
Pravděpodobnost nového stavu při rozložení na starém:

$$\begin{aligned} P(S_{t+1}|o_{1:t}) &= \sum_{s_t} P(S_{t+1}|s_t, o_{1:t}) \cdot P(s_t|o_{1:t}) \\ &= \sum_{s_t} P(S_{t+1}|s_t) \cdot P(s_t|o_{1:t}) \end{aligned}$$

Zapracovnání pozorování:

$$\begin{aligned} P(S_{t+1}|o_{1:t}, o_{t+1}) &= \frac{1}{P(o_{t+1}|o_{1:t})} \cdot P(o_{t+1}, S_{t+1}|o_1, \ldots, o_t) \\ &= \alpha \cdot P(o_{t+1}|S_{t+1}, o_{1:t}) \cdot P(S_{t+1}|o_{1:t}) \\ &= \alpha \cdot P(o_{t+1}|S_{t+1}) \cdot P(S_{t+1}|o_{1:t}) \end{aligned}$$

Dohromady:

$$P(S_{t+1}|o_{1:t}, o_{t+1}) = \alpha \cdot P(o_{t+1}|S_{t+1}) \cdot \sum_{s_t} P(S_{t+1}|s_t) \cdot P(s_t|o_{1:t})$$

Vyhlazování (zpět)

Pro k: $0 \leq k < t$.

$$\begin{aligned} P(S_k|o_{1:t}) &= P(S_k|o_{1:k}, o_{(k+1):t}) \\ &= \alpha \cdot P(S_k|o_{1:k}) \cdot P(o_{(k+1):t}|S_k, o_{1:k}) \\ &= \alpha \cdot f_{1:k} \cdot b_{(k+1):t} \end{aligned}$$

kde $f_{1:k}$ je zpráva zleva (před k), $b_{(k+1):t}$ zpráva zprava (po k).
Zpětná zpráva:

$$\begin{aligned} P(o_{(k+1):t}|S_k) &= \sum_{s_{k+1}} P(o_{(k+1):t}|S_k, s_{k+1}) \cdot P(s_{k+1}|S_k) \\ &= \sum_{s_{k+1}} P(o_{(k+1):t}|s_{k+1}) \cdot P(s_{k+1}|S_k) \\ &= \sum_{s_{k+1}} P(o_{(k+1)}, o_{(k+2):t}|s_{k+1}) \cdot P(s_{k+1}|S_k) \\ &= \sum_{s_{k+1}} P(s_{k+1}|S_k) \cdot P(o_{(k+1)}|s_{k+1}) \cdot P(o_{(k+2):t}|s_{k+1}) \end{aligned}$$

Maticový zápis

$$\begin{aligned} T &= P(S_{t+1}|S_t) \\ O_k &= \begin{cases} P(o_k|S_i) \text{ na diagonále} \\ 0 \;\; \text{jinak} \end{cases} \end{aligned}$$

f, b bereme jako sloupcové vektory.
Potom se dá psát:

$$\begin{aligned} f_{1:(t+1)} &= \alpha \cdot O_{t+1} T f_{1:t} \\ b_{(k+1):t} &= T^T O_{k+1} b_{(k+2):t} \end{aligned}$$

Navíc se dá počítat i v protisměru, tj.:

$$\begin{aligned} \alpha T^{-1} O_{t+1}^{-1} f_{1:(t+1)} &= f_{1:t} \\ O_{t+1}^{-1} (T^T)^{-1} b_{(k+1):t} &= b_{(k+2):t} \end{aligned}$$

Vyhlazování s konstantním zpožděním
např. sledování letadla na radaru

- Mám pevné zpoždění d.
- Zajímá mě $P(S_{t-d}|o_{1:t})$, aniž bych propagovala od začátku a konce.
- tj. $P(S_{t-d}|o_{1:t}) = \alpha f_{1:(t-d)} b_{(t-d+1):t}$
- Dopředná zpráva jako dříve.

Zpětná zpráva – d krát dosadím:

$$b_{(t-d+1):t} = \left(\Pi_{i=t-d+1}^{t} T^T O_i\right) b_{(t+1):t} = B_{(t-d+1):t} 1$$

$$b_{(t-d+2):(t+1)} = \left(\Pi_{i=t-d+2}^{t+1} T^T O_i\right) b_{(t+2):(t+1)} = B_{(t-d+2):(t+1)} 1$$

Odtud vidíme, že B můžeme průběžně aktualizovat:

$$B_{(t-d+2):(t+1)} = O_{t-d+1}^{-1} (T^T)^{-1} B_{(t-d+1):t} (T^T) O_{t+1}$$

Nalezení nejpravděpodobnějšího průchodu

- Pozorované pozice:$[a1, a2, b2, f6, c3, d3]$. Jaký je nejpravděpodobnější průchod?
- používá se např. při zpracování řeči
- NENÍ ekvivalentní najít pro každý krok maximálně pravděpodobný stav! Taková posloupnost může být nemožná, tj. pravděpodobnosti nula.

Viterbi algoritmus
Nalezení nejpravděpodobnějšího průchodu

Algoritmus

- Pro každý stav si pamatujeme pravděpodobnost nejpravděpodobnější cesty do něj.
- Na začátku známe apriorní $P(s_1)$ jednotlivých stavů s_1.

$$max_{s_1,\ldots,s_t} P(s_1,\ldots,s_t,S_{t+1}|o_{1:(t+1)}) =$$

$$\alpha P(o_{t+1}|S_{t+1}) max_{s_t} P(S_{t+1}|s_t) max_{s_1,\ldots,s_{t-1}} P(s_1,\ldots,s_t,S_t|o_{1:t})$$

- čili dopředně šířená zpráva místo sumy používá maximalizaci, tj. zpráva je

$$m_{1:t} = max_{s_1,\ldots,s_{t-1}} P(s_1,\ldots,s_{t-1},S_t|o_{1:t})$$

vždy vynásobíme $P(S_{t+1}|s_t)$ a maximalizujeme přes s_t.

Dynamické Bayesovské sítě

- Příklady jsme už viděli – nákresy Markovských procesů
- Zadáme BN v jednom časovém řezu plus vazby pro přechod z jednoho řezu do druhého a příslušné podmíněné pravděpodobnosti
- předpokládáme, že model časového řezu a přechodu se nemění s časem

- Matematicky je (diskrétní) DBN ekvivalentí HMM.
- výhoda je v rozloženém zadání: pro 20 binárních proměnných, každý 3 rodiče v předchozím čase, máme $20 \cdot 2^3 = 160$ přechodových pravděpodobností, u HMM $2^{20} \cdot 2^{20} = 2^{40}$.
  - HMM potřebuje více prostoru
  - a více času při inferenci, musím matici vždy projít
  - především, odhad takového množství parametrů je velmi diskutabilní (pokud není definován generujícím procesem)
  - v rámci jednoho času je vztah DBN vs. HMM podobný, jako BN a plné rozložení na všech veličinách.

Návrh DBN
Příklad: Baterie robota

- pozice, rychlost, stav baterie, GPS – viz tabule
- model chyby senzorů – klasicky gausovské rozložení okolo správné hodnoty
- přechodné selhání senzoru – když někdo do robota narazí, senzor baterie hlásí nulu (vybitou baterii)
- gausovský chybový model hlásí vybitou baterii, robot přejde do krizového režimu a vypne se.
  - přidáme do modelu: $P(BMeter_t = 0 | Battery_t = 5) = 0.03$
  - modely viz obrázky
  - výhoda: méně věří senzoru, vybitou baterii hlásí až po několika nulových měření, počet závisí na poměru predikce vybití baterie a pravděpodobnosti chyby senzoru.

Trvalé selhání

- Např. u auta jedoucího do prudkého kopce nefunguje měřák paliva
- předchozí model nepomůže, do kopce jedeme docela dlouho
- řešení: modelujeme vadný senzor

Přesná inference DBN

- rozvinout a spočítat jako BN; klasicky příliš velké
- smůla je, že "mezivýsledné tabulky" nemusí faktorizovat, tj. po čase máme jednu tabulku pro celý časový bod, tj. velikost $O(d^{n+1})$ tabulky, složitost aktualizace $O(d^{n+2})$, pořád lepší než $O(d^{2n})$ u HMM, ale obojí příliš náročné.

Přibližný výpočet DBN
částicové filtrování, particle filtering

- Pro BN se používá (mimo jiné) vážené vzorkování (likelihood weighting)
- s délkou času v DBN váhy exponencielně klesají k nule, tj. potřebujeme stále exponencielně větší vzorek; to nejde
- upravíme algoritmus

Částicové filtrování
přibližný výpočet DBN

- vstup: o pozorování, N požadovaný počet vzorků, DBN model
- statické: S, vektor vzorků, na počátku dle $P(S_0)$, W vektor vah
- for $i = 1$ to n
  - $S[i] \leftarrow$ sample from $P(S_1 | S_0 = S[i])$
  - $W[i] \leftarrow P(o | S_1 = S[i])$
- $S \leftarrow$ vážený vzorek s navracením(N, S, W)

Pro N jdoucí k nekonečnu konverguje ke přesnému vyhodnocení.

Zpracování řeči

- Nejmenší jednotka: **foném**
- Liší se podle způsobu a místa tvoření, artikulujícího orgánu nebo sluchového dojmu (fonologie). Celkem ve svět. jazycích jen cca. 12 diferenciálních příznaků.
- Počet fonémů v jazycích je 12 až 60. (ČJ 36, AJ 42, RJ 40).
- Fonémy se spojují co posloupností. Ty lze dělit na slabiky, slabiky tvoří slova. Slovanské jazyky cca. 2500–3000 slabik, 45000 – 50000 slov.
- Člověk při hovoru vysloví 80–130 slov za minutu, tj. cca 10 fonémů za sekundu. Při informaci 3–4 bity na foném je přenos informace 30–40 bit/s; člověk je schopen zpracovat informaci o rychlosti maximálně 50 bit/s.
- V češtině je fonologicky funkční symbol pauza pro hranici mezi slovy, v angličtině ne.

Zpracování signálu

- Snímáme v určité frekvenci (sampling) 8–20 kHz
- kvantování – diskretizujeme velikost signálu 12–14 bitů
- příznaky (features) – např. krátkodobá energie či častěji krátkodobá intenzita, krátkodobá funkce středního poštu průchodů signálu nulou, autokorelační funkce, a Fourierovy transformace pro frekvenční oblast.
- vektorová kvantizace – hodně kombinací příznaků reprezentuji jedním kódem, tím zmenším prostor, ve kterém pak budu pracovat (např. 256 kódů).

Pravděpodobnostní přístup

- Skryté Markovské procesy
- základ – Vintsyuk – každé slovo vlastní model, 40–50 stavů, odpovídajících pruměrnému počtu mikrosegmentů ve slově
- těžko by se trénovalo obecně, proto se učí modely pro jednotlivé fonémy; z modelů pro fonémy složím slovo (transkripce slova pro češtinu celkem snadná)

Viterbiův algoritmus

Skrytý Markovský model $P(S_1), P(S_{t+1}|S_t), P(O_t|S_t)$
S má stavy $i = 1, \ldots, N$
hledám maximálně pravděpodobný průchod.

1. Inicializace: $\delta_1(i) \leftarrow P(S_1 = i) \cdot P(O_1 = o_1|S_1 = i)$
   $\psi_1 = 0$
2. Rekurze v čase $t = 2, \ldots, T$ a nový stav $j = 1, \ldots, N$
   $\delta_t(j) \leftarrow max_i \delta_{t-1}(i) \cdot P(S_{t+1} = j|S_t = i) \cdot P(O_t = o_t|S_t = j)$
   $\psi_t = argmax_i[\delta_{t-1}(i) \cdot P(S_{t+1} = j|S_t = i)]$
3. Výsledná pravděpodobnost a index maximálně pravděpodobného stavu v čase T jsou:
   $P^* = max_i[\delta_T(i)]$
   $i^* = argmax_i[\delta_T(i)]$ O nejpravděpodobnější průchod zpětně vystopujeme z ψ, $i^*_t = \psi_{t+1}(i^*_{t+1})$

Pozn: nejpravděpodobnější průchod není to samé co nejpravděpodobnější posloupnost fonémů.

Učení modelu

- Baum–Welchův algoritmus
- v zásadě EM algoritmus, klasická metoda učení modelu se skrytými parametry ve Strojovém učení.