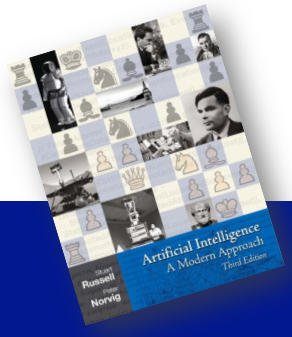


Umělá inteligence II

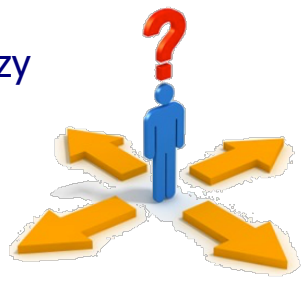
11

Roman Barták, KTIML

roman.bartak@mff.cuni.cz
http://ktiml.mff.cuni.cz/~bartak

Dnešní program

- V reálném prostředí převládá **neurčitost**.
- Neurčitost umíme zpracovávat **pravděpodobnostními modely** jako jsou Bayesovské sítě.
- **Jak ale získat vhodný model** (jak vyplnit tabulky podmíněných pravděpodobností)?
- Učení pravděpodobnostních modelů
  - Bayesovské učení
    - vypočtení pravděpodobnosti každé hypotézy
  - učení naivních Bayesovských modelů
  - učení modelů se skrytými proměnnými



Modelový příklad

- Máme bonbóny dvou příchutí – třešeň (cherry) a citrón (lime), které výrobce balí do stejného neprůhledného obalu.
- Bonbóny jsou prodávány v nerozlišitelných (velkých) baleních pěti typů:
  - h_1: 100% třešeň
  - h_2: 75% třešeň + 25% citrón
  - h_3: 50% třešeň + 50% citrón
  - h_4: 25% třešeň + 75% citrón
  - h_5: 100% citrón

- **Náhodná proměnná** H (hypotéza) označuje typ balení (není přímo pozorovatelná).
- K dispozici jsou **vstupní data** $D_1,\ldots,D_N$ – otevřené a „prozkoumané" bonbóny z jednoho balení.
- Základní úlohou je **předpovědět příchuť dalšího bonbónu**.

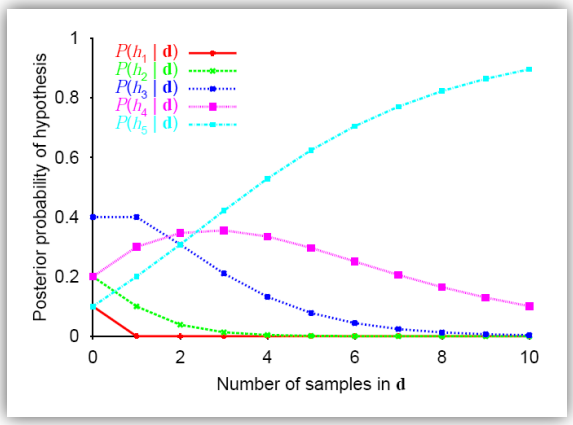
Bayesovské učení

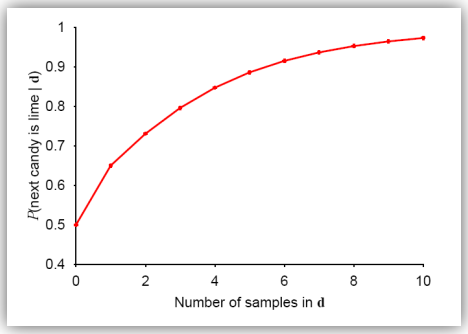
- Na základě dat spočteme pravděpodobnost každé hypotézy.
- Podle získané pravděpodobnosti uděláme předpověď.
- **Předpověď se dělá na základě všech hypotéz**, ne podle jedné nejlepší hypotézy.
- Formálně $P(h_i|\mathbf{d}) = \alpha\, P(\mathbf{d}|h_i)\, P(h_i)$
  kde **d** jsou pozorované hodnoty
- Předpověď se udělá následovně:
  $P(X|\mathbf{d}) = \Sigma_i\, P(X|\mathbf{d},h_i).\, P(h_i|\mathbf{d}) = \Sigma_i\, P(X|h_i).\, P(h_i|\mathbf{d})$
  - Předpověď je váženým průměrem předpovědí jednotlivých hypotéz.
- Hypotéza je prostředníkem mezi daty a předpovědí.
- Základní prvky Bayesovského učení
  - $P(h_i)$ je apriorní pravděpodobnost hypotéz
  - $P(\mathbf{d}|h_i)$ je věrohodnost dat podle hypotézy

Modelový příklad

řešení

- Nechť od výrobce známe pravděpodobnost výskytu jednotlivých balení $\langle 0.1; 0.2; 0.4; 0.2; 0.1 \rangle$
- Za předpokladu nezávislosti pozorování (velká balení):
  $P(\mathbf{d}|h_i) = \Pi_j P(d_j|h_i)$
- Pokud jsme rozbalili deset citronových bonbónů, máme například $P(\mathbf{d}|h_3) = 0.5^{10}$.





Bayesovské učení

vlastnosti

- Bayesovská předpověď po určité době **souhlasí s pravdivou hypotézou**
  - posteriorní pravděpodobnosti nepravdivých hypotéz se blíží 0
- Bayesovská předpověď je **optimální** nezávisle na velikosti dat
  - každá jiná předpověď bude správná méně často
- **Prostor hypotéz** je ale často **velmi velký** (až nekonečný).
  - řešíme různými aproximacemi

MAP učení

- Častou aproximací Bayesovského učení je dělání předpovědi na základě **nejpravděpodobnější hypotézy** (**maximum a posteriori hypothesis**).
  - $P(X|d) \approx P(X|h_{MAP})$
- V našem příkladě je po třech citronových bonbónech $h_{MAP} = h_5$.
  - další citronový bonbón předpoví s pravděpodobností 1.0, což je riskantnější předpověď než Bayesovská předpověď 0.8
- Hledání MAP hypotézy znamená **řešení optimalizačního problému**, což je snazší než dlouhé součty (integrály) u Bayesovského učení.

Přeučení

- Již víme, že pokud je prostor hypotéz příliš bohatý, může docházet k přeučení (overfitting).
- Bayesovské a MAP učení používá místo omezení složitosti hypotéz metodu penalizace složitosti.
  - složitější hypotézy mají menší apriorní pravděpodobnost
  - použití apriorní pravděpodobnosti hypotézy tedy přirozeně vyvažuje složitost hypotézy a míru mapování na data
- Pokud budeme brát deterministické hypotézy ($P(\mathbf{d}|h_i) = 1$, je-li hypotéza konzistentní, a 0 jindy) potom je h_{MAP} nejjednodušší hypotézou konzistentní s daty.
  - jedná se o přirozené vyjádření principu Ockhamovy břitvy

MDL učení

- Při hledání h_{MAP} maximalizujeme $P(\mathbf{d}|h_i)\,P(h_i)$
- To je ekvivalentní minimalizaci $(-\log_2 P(\mathbf{d}|h_i) - \log_2 P(h_i))$
  - $-\log_2 P(h_i)$ můžeme brát jako počet bitů nutných pro specifikaci hypotézy h_i
  - $-\log_2 P(\mathbf{d}|h_i)$ jsou potom dodatečné bity pro specifikaci dat z dané hypotézy (speciálně, pokud hypotéza přesně odpovídá datům, máme $-\log_2 1 = 0$, tj. není potřeba další informace)
- MAP učení tedy vybírá hypotézu, která maximalizuje kompresi dat.
- Podobný přístup přímo realizuje metoda MDL (**minimum description length**)
  - přímo počítá bity potřebné pro binární kódování hypotézy i dat a hledá hypotézu s nejmenším počtem bitů

ML učení

- Jiné zjednodušení Bayesovského učení uvažuje stejné apriorní pravděpodobnosti všech hypotéz.
- Hledáme hypotézu maximalizující $P(\mathbf{d}|h_i)$ - **maximum likelihood hypothesis**
  - potlačuje subjektivní hodnocení rozložení hypotéz
  - dobrá aproximace pro hodně dat (data časem převáží nad apriorním rozdělením hypotéz)
  - pro malý počet příkladů problém

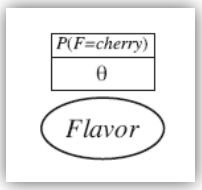
Učení parametrů

Jak se naučit pravděpodobnostní model?

- Uvažujme, že struktura modelu je dána (např. máme Bayesovskou síť)
- Potřebujeme vědět, jak do tabulek doplnit podmíněné pravděpodobnosti – **učení parametrů.**
- Nejprve předpokládejme, že máme **úplná data.**
  - každý příklad určuje hodnoty všech náhodných proměnných v modelu

Maximální věrohodnost

- Uvažujme opět výrobu bonbónů, ale tentokrát neznáme poměr příchutí
- Parametr k učení je pravděpodobnost θ výskytu třešňového bonbónu.
  - hypotéza je h_θ
  - modelujeme Bayesovskou sítí s jedním uzlem



- Po rozbalení N bonbónů, nechť c je třešňových a l (= N-c) citronových bonbónů

$P(\mathbf{d}|h_\theta) = \Pi_j P(d_j|h_\theta) = \theta^c (1-\theta)^l$

- Použití ML učení je zde vhodné, protože apriorní pravděpodobnost hypotéz je stejná
  - maximalizujeme $P(\mathbf{d}|h_\theta)$ což je stejné jako maximalizovat

$L(\mathbf{d}|h_\theta) = \log P(\mathbf{d}|h_\theta) = \Sigma_j \log P(d_j|h_\theta) = c \log\theta + l \log(1-\theta)$

  - použijeme derivaci rovnou nule

$\partial L(\mathbf{d}|h_\theta) / \partial\theta = c/\theta - l/(1-\theta) = 0 \Rightarrow \theta = c/(c+l) = c/N$

  - h_{ML} je tak vlastně dána podílem třešňových bonbónů ve vyzkoušeném vzorku!

Maximální věrohodnost

obecněji

- Učení metodou ML obecně funguje takto:
  - zapíšeme vztah pro pravděpodobnost dat jako funkci parametrů
  - logaritmus vztahu derivujeme
  - najdeme hodnoty parametrů tam, kde je derivace nulová
    - tento bod bývá nejkomplikovanější, někdy je potřeba řešit numericky
- ML učení parametrů Bayesovské sítě se rozpadne na učení se jednotlivých parametrů.

Maximální věrohodnost

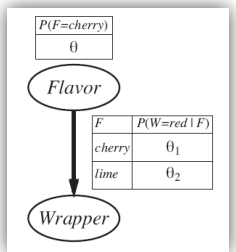
rozšířený příklad

- Uvažujme nyní, že výrobce bonbónů napovídá příchuť barvou obalu (červený nebo zelený).
  - síť má teď tři parametry θ, θ_1, θ_2

    θ - bonbón je třešňový

    θ_1 - je-li bonbón třešňový, potom má červený obal

    θ_1 - je-li bonbón citronový, potom má červený obal



$P(Flavor=cherry, Wrapper=green \mid h_{\theta,\theta_1,\theta_2})$

$= P(Flavor=cherry \mid h_{\theta,\theta_1,\theta_2}) \; P(Wrapper=green \mid Flavor=cherry, h_{\theta,\theta_1,\theta_2})$

$= \theta \, (1- \theta_1)$

- Po rozbalení N bonbónů
  (c třešňových, l citronových,
  r_c třešňových v červeném obalu, g_c třešňových v zeleném obalu,
  r_l citronových v červeném obalu, g_l citronových v zeleném obalu)

$$P(\mathbf{d} \mid h_{\theta,\theta_1,\theta_2}) = \theta^c(1-\theta)^l \, \theta_1^{r_c} \, (1- \theta_1)^{g_c} \, \theta_2^{r_l} \, (1-\theta_2)^{g_l}$$

$$L = c.\log \theta + l.\log(1-\theta) + r_c.\log \theta_1 + g_c.\log(1- \theta_1) + r_l.\log \theta_2 + g_l.\log(1-\theta_2)$$

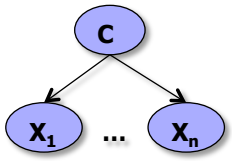
$$\partial L / \partial \theta = c / \theta - l / (1- \theta) \quad \Rightarrow \quad \theta = c / (c+l)$$

$$\partial L / \partial \theta_1 = r_c / \theta_1 - g_c / (1- \theta_1) \quad \Rightarrow \quad \theta_1 = r_c / (r_c + g_c)$$

$$\partial L / \partial \theta_2 = r_l / \theta_2 - g_l / (1- \theta_2) \quad \Rightarrow \quad \theta_2 = r_l / (r_l + g_l)$$

Naivní Bayesovské modely

- **Naivní Bayesovský model** je jeden z nejběžněji používaných modelů Bayesovské sítě ve strojovém učení.
  - proměnná třídy C, kterou předpovídáme
  - atributové proměnné X_i v listech, které jsou podmíněně nezávislé vzhledem k C

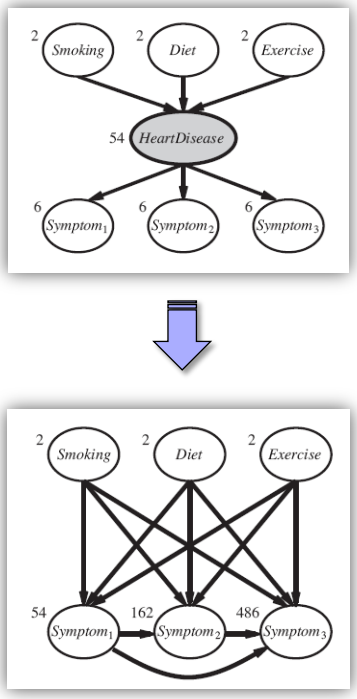


- Pro Booleovské náhodné proměnné máme **parametry**
  - θ = P(C=true)
  - θ_{i1} = P(X_i=true | C=true)
  - θ_{i2} = P(X_i=true | C=false)
- Po naučení tabulek podmíněné distribuce je **pravděpodobnost zařazení do tříd** podle pozorování $x_1,\ldots,x_N$

  $P(C \mid x_1,\ldots,x_N) = \alpha\, P(C) \prod_i P(x_i \mid C)$
- **Vlastnosti**
  - dobrá **škálovatelnost** (pro n atributů máme 2n+1 parametrů pro učení)
  - **není problém** se **šumem** v datech
  - **pravděpodobnostní předpověď**

Skryté proměnné

- V řadě reálných problémů se vyskytují tzv. **skryté proměnné** – náhodné proměnné, které nejsou součástí pozorovaných dat.
  - např. lékařské záznamy obsahují symptomy, diagnózu, léčbu a výsledek léčby, ale málokdy lze přímo pozorovat nemoc samotnou
- Pokud nemůžeme skryté proměnné pozorovat, proč neudělat model bez nich?
  - model bez skrytých proměnných má mnohem více parametrů
  - uvažujeme-li v příkladu tři hodnoty, máme 708 parametrů modelu bez skrytých proměnných oproti původním 78 parametrům



EM algoritmus

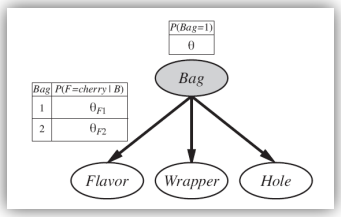
Jak se naučit podmíněné distribuce skryté proměnné, když v příkladech nejsou její hodnoty?

- můžeme předstírat, že hodnoty parametrů známe
- potom jsme schopni **spočítat očekávané hodnoty** skrytých proměnných, čímž data doplníme na úplný model (E-step, expectation)
- **upravíme hodnoty parametrů**, abychom zvýšili věrohodnost modelu (M-step, maximization)
- celý proces iterujeme

EM algoritmus

modelový příklad

- Uvažujme, že v příkladu s bonbóny přibyl třetí atribut – díra – a že bonbóny byly původně ve dvou sadách, které se ale promíchaly.
  - modelujeme naivní Bayesovskou sítí, kde proměnná Bag je skrytá, protože nevíme, do které sady bonbón patřil



  - parametry
    - θ bonbón je z první sady
    - θ_{F1}, θ_{F2} bonbón je třešňový z první nebo druhé sady
    - θ_{W1}, θ_{W2} bonbón je zabalený červeně z první nebo druhé sady
    - θ_{H1}, θ_{H2} bonbón má díru z první nebo druhé sady
  - prozkoumali jsme 1000 bonbónů s následujícím výsledkem

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

EM algoritmus

řešení modelového příkladu

- Začneme s náhodným nastavením parametrů
  $\theta^{(0)} = \theta^{(0)}_{F1} = \theta^{(0)}_{W1} = \theta^{(0)}_{H1} = 0.6$
  $\theta^{(0)}_{F2} = \theta^{(0)}_{W2} = \theta^{(0)}_{H2} = 0.4$
- Protože proměnná Bag je skrytá, spočteme její očekávanou hodnotu z příkladů a parametrů (použitím inference pro Bayesovské sítě):
  $N(Bag=1) = \Sigma_j P(Bag=1 \mid flavor_j, wrapper_j, holes_j)$
- Upravíme hodnoty parametrů (N je počet příkladů)
  $\theta^{(0)} = N(Bag=1) / N$
  - obecný princip updatu parametrů
    - nechť $\theta^{(0)}_{i,j,k}$ je parametr $P(X_i = x_{ij} \mid \mathbf{U}_i = \mathbf{u}_{ik})$ distribuce pro X_i s rodiči $\mathbf{U}_i$
    - $\theta_{ijk} \leftarrow N(X_i = x_{ij}, \mathbf{U}_i = \mathbf{u}_{ik}) / N(\mathbf{U}_i = \mathbf{u}_{ik})$

  $\theta^{(1)} = 0.6124$, $\theta^{(1)}_{F1} = 0.6684$, $\theta^{(1)}_{W1} = 0.6483$, $\theta^{(1)}_{H1} = 0.6558$
  $\theta^{(1)}_{F2} = 0.3887$, $\theta^{(1)}_{W2} = 0.3817$, $\theta^{(1)}_{H2} = 0.6558$

EM algoritmus

shrnutí

- EM algoritmus pracuje ve dvou krocích
  - počítá očekávané hodnoty skrytých proměnných (expectation)
  - přepočítá hodnoty parametrů (maximization)
- V kostce
  - **x** jsou pozorované hodnoty
  - **Z** jsou skryté proměnné
  - θ jsou parametry pravděpodobnostního modelu

$$\theta^{(i+1)} \leftarrow \text{argmax}_{\theta(i)} \Sigma_{\mathbf{z}} P(\mathbf{Z}=\mathbf{z} \mid \mathbf{x}, \theta^{(i)}) L(\mathbf{x}, \mathbf{Z}=\mathbf{z} \mid \theta^{(i)})$$

M-step (maximization)

E-step (expectation)