Výpočtové učenie

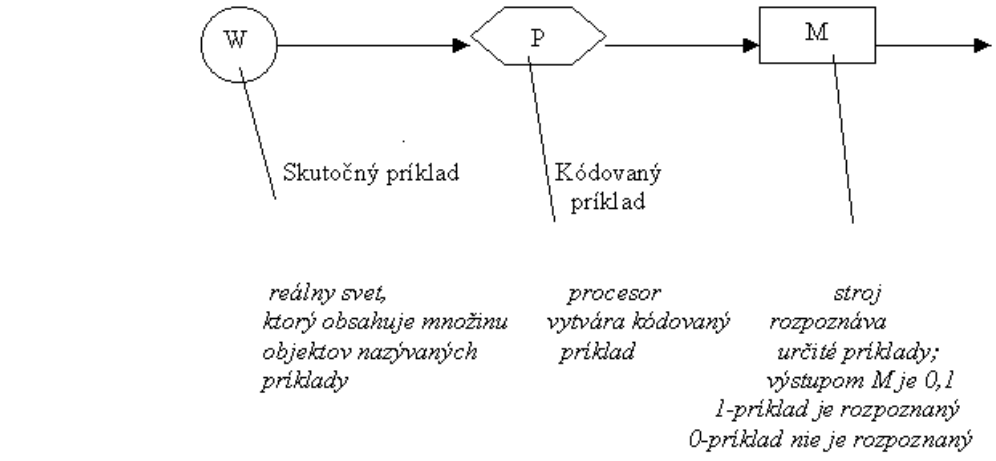
22. februára 2006

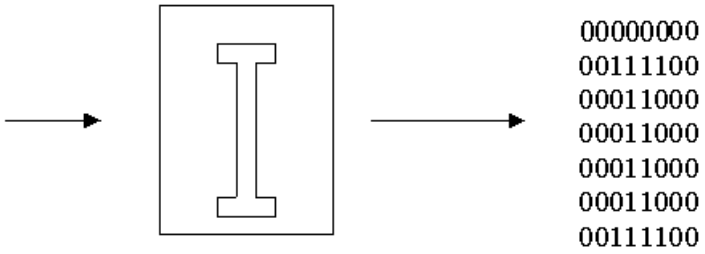
Kapitola 1

Koncepty, hypotézy, učiace algoritmy

1.1 Úvod

Je mnoho typov aktivít označovaných ako "učenie". My budeme študovať matematický model takého procesu. Tento model sa zdá byť použitelný, pretože zachytáva základ určitých aktivít, ktoré boli popísané predtým pomocou nepresných výrazov, a zároveň umožnuje vytvoriť netriviálne matematické tvrdenia, ktoré môžu byť dokázané.





Obrázok 1.1: Spracovanie príkladov reálneho sveta pomocou naténovaného stroja

1.2 Koncepty

Sformalizujeme pojem koncept, ktorý môže byť popísaný ako množina príkladov.Nech je Σ - abeceda na popis príkladov.
Napríklad, $\Sigma = \{0,1\}$, $\Sigma = R$
Množiny Σ^n, Σ^* predstavujú ...

Definícia 1.2.1 *Nech $X \subseteq \Sigma^*$. Koncept v abecede Σ je funkcia c, $c : X \to \{0,1\}$. Množina X sa nazýva priestor príkladov. Prvok x, $x \in X$ sa nazýva príklad. Ak pre $x \in X$ platí $c(x) = 1$, tak x je pozitívny príklad, ak platí $c(x) = 0$, tak x je negatívny príklad.*

Zjednotenie množiny kladných a záporných príkladov je definičný obor funkcie c. Teda za predpokladu, že definičný obor je známy, c určuje a je určované množinou svojich pozitívnych príkladov.

Príklady:

1. Koncept parita
   $\Sigma = \{0,1\}$ $p : \Sigma^* \to \{0,1\}$ $y = y_1 \dots y_n$:

$$p(y) = \begin{cases} 1 & \text{ak v } y \text{ je nepárny počet jedničiek} \\ 0 & \text{ak v } y \text{ je párny počet jedničiek} \end{cases}$$

   1011101 kladný príklad, 1000001 záporný príklad,

2. Koncept palindrom
   $\Sigma = \{0,1\}$ $p : \Sigma^* \to \{0,1\}$ $y = y_1 \dots y_n$:

$$p(y) = \begin{cases} 1 & \text{ak } y_i = y_{n-i+1} \quad i = 1, 2 \dots \frac{n}{2} \\ 0 & \text{inak} \end{cases}$$

3. Koncept n-rozmerná jednotková guľa
   $\Sigma = R$ $u : \Sigma^n \to \{0,1\}$ $y = y_1 \dots y_n$:

$$p(y) = \begin{cases} 1 & \text{ak } {y_1}^2 + {y_2}^2 + \ldots + {y_n}^2 \leq 1 \\ 0 & \text{inak} \end{cases}$$

1.3 Trénovanie a učenie

Sú dve množiny konceptov ukázaných v rámci učenia popísaného na obr. 1.1.

Prvá množina je množina konceptov odvodených z reálneho sveta, ktorá je predkladaná na rozpoznanie. Táto množina môže obsahovať koncepty ako "písmeno A", "písmeno B", ..., z ktorých každé môže byť zakódované. Každý koncept má svoje množiny kladných a záporných príkladov. Keď je množina konceptov určovaná týmto spôsobom, budeme pre ňu používať výraz konceptový priestor.

Druhá množina konceptov obsiahnutých v rámci učenia na obr. 1.1 je množina, ktorú stroj M je schopný rozpoznať. Budeme predpokladať, že M sa môže preradiť do rôznych stavov a v danom stave bude klasifikovať niektoré vstupy ako kladné (výstup 1) a zvyšok ako záporné (výstup 0). Teda stav M určuje koncept, ktorý môžeme chápať ako hypotézu. Množina všetkých konceptov, ktoré M určuje, bude nazývaná hypotézový priestor.

Cieľom učiaceho procesu je vytvoriť hypotézu, ktorá v nejakom zmysle zodpovedá konceptu z konceptového priestoru vzhľadom na vyššie uvedenú úvahu. Detaily, kedy a ako toto môže byť urobené sú ústredným záujmom tejto prednášky.

Máme teda 2 množiny konceptov: C konceptový priestor, H - hypotezový priestor, a **p r o b l é m o m** je nájsť ku každému $c, c \in C$, nejaké $h, h \in H$, ktoré je dobrou aproximáciou pre c.

V reálnych situáciách sú hypotézy tvorené na základe určitých informácií, ktoré neprinášajú explicitnú definíciu c. My budeme predpokladať, že táto informácia je poskytovaná postupnosťou kladných a záporných príkladov X. Nemáme dostatok zdrojov na to, aby sme mohli vybudovať veľmi veľký stroj pre nájdenie c, nemáme dostatok času na to, aby bol vytvorený a spustený program, ktorý by určil, že $h = c$, alebo že h je tak blízko c, ako si prajeme. V praxi sú kladené obmedzenia na zdroje a my sa musíme

uspokojiť s hypotézou h , ktorá "pravdepodobne" reprezentuje c (aproximuje c) v nejakom definovanom zmysle.

Nech $X \subseteq \Sigma^*$ je príkladový priestor. $\Sigma = \{0,1\}$ alebo $\Sigma = R$. Vzorka dlžky m je postupnosť m príkladov, t. j. je to $m-$tica $\overline{x} = (x_1, x_2, \ldots, x_m) \in X^m$, kde x_i sú príklady a b_i vyjadruje, či príklad je kladný alebo záporný. Postupnosť môže obsahovať rovnaké hodnoty viackrát. Niekedy budeme predpokladať, že sú rôzne bez újmy na všeobecnosti.
Tréningová vzorka s je množina $(X \times \{0,1\})^m$, t. j. $\overline{s} = ((x_1, b_1), (x_2, b_2), \ldots, (x_m, b_m))$

Budeme predpokladať, že nie sú žiadne sporné príklady, t.j. ak $x_i = x_j, \quad \Rightarrow \quad b_i = b_j$. To teda znamená, že existuje funkcia s, definovaná ako $s(x_i) = b_i \quad (1 \leq i \leq m)$.

Budeme hovoriť, že $\overline{s}$ je tréningová vzorka pre cieľový koncept t, ak $b_i = t(x_i)$, pre $1 \leq i \leq m$.

Príklady:
Tréningová vzorka pre koncept "palindrom" je
$((0010,0), (1001001001,1), (111,1), (010101,0), (111101,0))$
Cieľový koncept t: $x = (x^1 \ldots x^n)$:

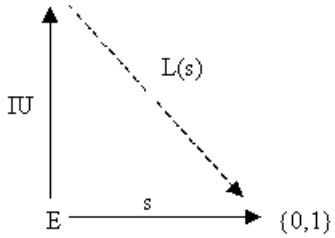
$$t(x) = \begin{cases} 1 & \text{ak } x^i = x^{n-i+1} \quad pre 1 \leq i \leq n \\ 0 & \text{inak} \end{cases}$$

Uvažujme teraz o povahe učiaceho procesu, ktorý tu chceme študovať. Majme dané C - konceptový priestor a H - hypotézový priestor v abecede Σ. **Učiaci algoritmus pre** (C,H) , niekedy nazývaný (C,H)-učiaci algoritmus je procedúra, ktorá akceptuje tréningové vzorky pre funkcie v C a výstupy zodpovedajú hypotézam v H. Aby táto procedúra mohla byť považovaná za algoritmus, musí byť efektívna. Ak ignorujeme problém efektívnosti, tak učiaci algoritmus pre (C,H) je teda funkcia L, ktorá priradí ľubovoľnej tréningovej vzorke $\overline{s}$ pre cielový koncept $t \in C$ funkciu $h \in H$. Píšeme $h = L(\overline{s})$.

Poznamenajme, že $L(\overline{s})$ je definovaná na celom príkladovom priestore X, kde s je funkcia definovaná na konečnej podmnožine $E \subseteq C$ (lebo zahŕňa len príklady vzorky $(x_1, ..., x_m)$.

Hypotéza $h \in H$ je **konzistentná** s s alebo súhlasí s s, ak $h(x_i) = b_i$, pre $1 \leq i \leq m$.

Vo všeobecnosti nerobíme predpoklady, že L (s) je konzistentná s s, ale keď táto podmienka platí pre všetky s, hovoríme že **L je konzistentný**. V tomto prípade je funkcia L(s) ako rozširujúca funkcia s, ako o tom hovorí diagram.



$E \subseteq X$

Vo všeobecnosti, nie každé rozšírenie tréningovej vzorky bude vhodným zovšeobecnením, pretože cieľový koncept je len parciálne definovaný príkladmi vzorky. Ďalej tréningová vzorka môže byť nereprezentatívna, alebo zavádzajúca.

Napríklad: Ak vhodne zakódujeme všetky zvieratá cieľový koncept je "mačka", tak sa môže stať, že tréningová vzorka pozostáva z bezchovstových mačiek. V praxi musíme predpokladať, že nereprezentatívne vzorky sú nepravdepodobné a že väčšina vzoriek je dostatočne reprezentatívna, takže rozšírenia funkcií sú vyhovujúce.

Príklad: Kreslo je možné popísať

(4 roky,	chvost,	sedací priestor,	zafarbenie,	žije)
(1	1	1	0	0),1
(1	1	1	1	0),1

1.4 Učenie pomocou konštrukcie

Uvedieme dva veľmi jednoduché a veľmi všeobecné algoritmy, ktoré sú ale neefektívne. V dalšom budeme venovať pozornosť efektívnejším algoritmom.

Nech X je príkladový priestor, t cieľový koncept, X^+, $X^+ \subseteq X$ množina kladných príkladov. Jeden spôsob učenia t je skonštruovať množinu X^+ explicitne. Môžme začať s prázdnou množinou prechodom cez tréningovú vzorku pridať každý pozitívny príklad. Formálne to môžme vyjadriť

set h (x) = 0 for all x in X;
for i: = 1 to m do if $b_i = 1$ then set h $(x_i) = 1$;
$L(\overline{s}) = h$;

Niektoré otázky, týkajúce sa algoritmu:

1. Čo ak X je nekonečný priestor príkladov?
2. Ako vhodne vyjadriť hypotézový priestor tak, aby hypotézy boli vhodne vyjadriteľné?

Ak dáme bokom otázku efektívnosti, vystúpia nasledujúce poznámky. Zrejme, výstupná hypotéza L(s) je rovná cieľovému konceptu t $\iff$ keď s obsahuje všetky kladné príklady pre t. Pretože s je konečná postupnosť, to znamená, že len koncepty s konečným počtom kladných príkladov môžu byť naučené s úplným úspechom.

Napríklad, koncept "parita" je definovaný nad celým $\{0,1\}^*$, teda algoritmus nemôže skonštruovať celú množinu kladných príkladov. Ak sa obmedzíme na paritu reťazcov dĺžky n, tak koncept "parita" nad $\{0,1\}^n$ je naučiteľný, počet kladných prípadov je $2^{n1} \Rightarrow$ musíme voliť počet príkladov vzorky m aspoň tak veľké.

Tento algoritmus má aj dobré vlastnosti:

1. je **konzistentný** t.j. výstupná hypotéza L(s) klasifikuje všetky príklady vyskytujúce sa v s korektne.
2. každý komponent tréningovej vzorky sa vyskytuje práve 1x. Toto je veľmi silná vlastnosť .... on line vlastnosť. V praxi to znamená, že príklady môžu byť prezentované učiacemu, keď sa vyskytnú, bez nutnosti mať pamäť, ktorá ich uloží pre ďalšie použitie.

Definícia 1.4.1 *Hovoríme, že algoritmus je* **bezpamäťový (on line) algoritmus**, *ak pre danú tréningovú vzorku $\overline{s}$ vytvára postupnosť hypotéz $h_0, h_1, \ldots h_m$, takých, že h_{i+1} závisí len od h_i a od priebežne spracovávaného príkladu vzorky (x_1, b_1).*

1.5 Učenie očíslovaním

Nasledujúca metóda učenia určite nie je bezpamäťový on - line algoritmus. Predpokladáme, že hypotézový priestor H je spočítateľný a má explicitné očíslovanie, $H = \{h^{(1)}, h^{(2)}, \ldots\}$

Predpokladajme, že $\overline{s}$ je tréningová vzorka pre cieľový koncept t. Metóda: Porovnať každú hypotézu s každým príkladom v $\overline{s}$, odmietnuť hypotézu pri každej príležitosti, ak nesúhlasí s hodnotou príkladu. Po odmietnutí hypotézy je ďalšia vzorka testovaná tým istým spôsobom. Proces sa zastaví, keď je nájdená hypotéza, ktorá vyhovuje všetkým príkladom tréningovej vzorky. Formálne,

Nech r - poradové číslo hypotézy, i - poradové číslo vzorky

```
begin
     r: = 1, i: = 1;
     repeat
     if  h(r) (x_{i} )<> b_i then
         begin r: = r+ 1; i : = 1  end
         else  i: = i + 1;
     until  i = m + 1;
     L (s) : = h(r);
end;
```

Množina H môže byť konečná, a teda môže sa stať, že sa vhodná hypotéza nenájde. Modifikáciu algoritmu vieme ľahko urobiť. V praxi sa musíme vyhnúť používaniu neprimeraných veľkých hypotézových priestorov. Počet všetkých hypotéz $h : \{0,1\}^n \to \{0,1\}$ je 2^{2^n}. Ak $n = 10$, $2^{2^n} = 2^{1024} = 4^{512} = 8^{256} = 16^{128}$

Z poznámok vyplýva, že na to, aby sa táto metóda stala vhodnou metódou učenia, je potrebné urobiť určité obmedzenia na hypotézový priestor H a jeho vzťah k priestoru konceptov C.Toto vedie k pojmu **"induktive bias"** predpojatosť.

Je to predpoklad, že učiaci má nejakú vopred predstavenú ideu o tom, akú metódu klasifikácie učiteľ používa, t.j. učiaci vie, alebo má nejaké informácie o konceptovom priestore.

Najjednoduchší spôsob modelovať taký predpoklad je stanoviť $H = C$ a v tomto prípade hovoríme o učiacom algoritme pre H , čo znamená (H, H). Väčšina preberaných algoritmov v ďalšom bude tohto typu.

1.6 Úlohy:

1. Aký je počet kladných príkladov konceptu "palindrom", keď príkladový priestor je $\{0,1\}^n$?
2. Nech w je nasledujúci koncept: $\{0,1\}^n$ $y \in 0,1^n$ $y = y_1 \dots y_n$:

$$w(y) = \begin{cases} 1 & \text{ak y obsahuje najviac 2 jedničky} \\ 0 & \text{inak} \end{cases}$$

   Ukážte, že počet kladných príkladov v tomto koncepte je kvadratickou funkciou n.

3. Predpokladajme, že v konečnom "učení očíslovaním" sme si istí, že hypotézy sú očíslované tak, že tá ktorú chceme, je v prvej polovici. Ak môžeme vybrať 1 milión hypotéz za sekundu a príkladový priestor je $\{0,1\}^9$, koľko to bude trvať v najhoršom prípade?
4. Dokážte, že počet funkcií $f : \{0,1\}^n \to \{0,1\}$ je 2^{2^n}

Kapitola 2

Booleovské formuly a reprezentácie

2.1 Učiaci algoritmus pre monočlenné funkcie

Valiant, 1984

Monočlen je konjunkcia literálov alebo ich negácií. Začíname bez informácií, t.j. predpokladáme výskyt všetkých $2n$ literálov

$$h_u : \qquad u_1\overline{u_1}u_2\overline{u_2}\ldots u_n\overline{u_n}$$

Každý pozitívny príklad $y = y_1 \ldots y_n$ umožňuje odstránenie tých literálov u_j, pre ktoré $y_j = 0$ a tie literály $\overline{u_j}$, pre ktoré $y_j = 1$. Predpokladajme, že $\overline{s}$ je tréningová vzorka

$$\overline{s} = ((x_1, b_1), \ldots (x_m, b_m))$$

$$x_i = ((x_i)_1 (x_i)_2 \ldots (x_i)_n), \quad 1 \leq i \leq m$$

$h_u \ldots$ monočlenná funkcia obsahujúca literály v množine U.

begin
set $U = \{u_1, \overline{u_1}, \ldots, u_n, \overline{u_n}\}$;
for i:=1 to m do
    if b_i=1 then
      for j:=1 to n do
          if $(x_i)_j = 1$ then delete $\overline{u_j}$
                          else delete u_j;
L(s)=h_u;
end;

Tento algoritmus sa nazýva štandardný učiaci algoritmus pre monočleny.

Veta 1 *Štandardný učiaci algoritmus pre monočleny je konzistentný (s výnimkou premenných, na ktorých nezáleží).*

Dôkaz: To znamená, že výsledná neprázdna formula je konzistentná s tréningovou vzorkou.

V každom kroku algoritmu je odstránených niekoľko (možno žiadny) literálov. Literály, ktoré sú vo výslednom t, neboli nikdy odstránené.

Ľubovoľný negatívny príklad pre t je založený na niektorom literále v t, ktorý nebol odstránený. Teda všetky negatívne príklady pre t (a čiastočne tie vo vzorke) sú korektne klasifukované pomocou $L(\overline{s})$.

Ak $h_U(x) = 1$ a $V \subset U$, tak $h_V(x) = 1$. Po každej prezentácii kladného príkladu x, mazacia procedúra zaručí, že $h_U(x) = 1$ a teda klasifikácia x je korektná. Teda koncová hypotéza $L(\overline{s})$ korektne klasifikuje všetky pozitívne príklady.

2.1.1 Disjunktná normálna forma - DNF

$$\mu_1 \vee \mu_2 \vee \ldots \vee \mu_n$$

kde μ_i je monočlenná funkcia, $1 \leq i \leq r$.

2.1.2 Konjuktívna normálna forma

$$\gamma_1 \wedge \gamma_2 \wedge \ldots \wedge \gamma_n$$

kde γ_i je, $1 \leq i \leq n$ je klauzula, tj. disjunkcia literálov.

Označenie:

M_n-množina monočlenov nad $\{0,1\}^n$

$M_{n,k}$-množina monočlenov nad $\{0,1\}^n$, z ktorých každý má najviac k literálov

$D_{n,k}$-množina disjunktných členov z $M_{n,k}$

2.2 Učenie disjunkcií malých monočlenov

Valiant, 1984

```
begin
h:=disjunkcia vsetkych jednoclenov dlzky najviac k;
for i:=1 to m do
if b_{i}=0 and h(x_{i})=1 then vymazat jednocleny $\mu$ pre ktore $\mu(x_i)=1$;
L(s):=h;
end;
```

2.3 Reprezentácia hypotézového priestoru

Učenie prediskutované v tejto časti malo zjednodušené predpoklady, a síce že konceptový priestor je ten istý ako hypotézový. V skutočnosti sme uvažovali o tom, že cieľové koncepty majú nejaký popis pomocou formúl alebo strojov. Hoci tento predpoklad sa môže zdať reštriktívny, je prirodzený pri matematickom štúdiu oblasti.

2.4 Cvičenia:

1. Napíšte postupnosť hypotéz generovaných algoritmom učenia monočlenov, keď na vstupe je prezentovaná tréningová vzorka

   $$(11100101, 1), (00100011, 0), (11001001, 1)$$

   Ak cieľový koncept je $\langle u_2 \overline{u}_4 u_8 \rangle$, doplňte príklady do vzorky, ktoré sú pre to nutné.

2. Napíšte DNF formuly pre koncepty *parita* a *palindrom* na príkladových priestoroch $\{0,1\}^5$.

3. Uveďte príklad booleovskej funkcie 3 premenných, ktorá nie je $D_{3,2}$.

Kapitola 3

Pravdepodobnostné učenie

3.1 Algoritmus pre učenie lúčov

V úvode do najdôležitejších ideí vo výpočtovej teórii učenia sa budeme zaoberať veľmi jednoduchým algoritmom pre učenie v reálnom hypotézovom prietore.

Pre každé reálne číslo Θ lúč r_Θ je koncept definovaný na príkladovom priestore R funkciou

$$r_\Theta(y) \iff y \geq \Theta$$

Algoritmus pre učenie v hypotézovom priestore $H = \{r_\Theta | \Theta \in R\}$ je založený na idee, že za aktuálnu hypotézu vezmeme "najmenší" lúč obsahujúci všetky pozitívne príklady v tréningovej vzorke. Vhodnou default hypotézou v prípade, že neexistujú kladné príklady, je funkcia identicky rovná nule. Vtedy budeme hovoriť o prázdnom lúči. Budeme označovaný r_∞.

Pre danú tréningovú vzorku

$$\overline{s} = ((x_1, b_1), (x_2, b_2), \ldots, (x_m, b_m))$$

výstupná hypotéza L(s) by mala byť r_λ, kde

$$\lambda = \lambda(\overline{s}) = \min_{1 \leq i \leq m} \{x_i | b_i = 1\}$$

$\lambda = \infty$, ak vzorka neobsahuje kladné príklady. Jednoduchá modifikácia algoritmu, ktorý počíta minimum konečnej množiny je postačujúca pre naše účely. Toto poskytuje nasledujúci bezpamäťový on-line algoritmus:

set $\lambda = \infty$;
for i:=1 to m do
    if $(b_i = 1)$ and $(x_i < \lambda)$ then set $\lambda = x_i$;
L(s):=r_λ;

Je ľahké vidieť, že ak tréningová vzorka je pre cieľovú hypotézu r_Θ, potom $L(\overline{s})$ bude lúč r_λ s $\lambda = \lambda(s) \geq \Theta$. Pretože je len konečný počet príkladov v tréningovej vzorke a príkladový priestor je nespočítateľný, nemôžeme očakávať, že $\lambda = \Theta$. Avšak, zdá sa, že ak dĺžka tréningovej vzorky rastie, tak by sa mala pravdepodobnosť, že chyba je malá vyplývajúca z použitia r_λ namiesto r_Θ.

Prakticky táto vlastnosť môže byť charakterizovaná nasledovne. Predpokladajme, že spustíme algoritmus s veľkou tréningovou vzorkou a potom sa rozhodneme použiť výstupnú hypotézu r_λ pre cieľovú (neznámu) hypotézu r_Θ. Inak povedané, uspokojíme sa s tým, že "učiaci sa" bol adekvátne trénovaný. Ak λ nie je blízke Θ, toto indikuje, že pozitívne príklady, ktoré by boli blízke Θ sú relatívne nepravdepodobné a nevyskytovali sa v tréningovej vzorke. Z toho vyplýva, keď teraz klasifikujeme niektoré ďalšie príklady, ktoré sú prezentované podľa toho istého rozloženia, tak môžme urobiť niekoľko chýb ako dôsledok použitia r_λ namiesto r_Θ.

3.2 Pravdepodobnostné aproximačne správne učenie (Probably Approximately Correct learning - PAC)

Uvažujme model, v ktorom trénujúca vzorka s pre cieľový koncept t je generovaná výberom príkladov $x_1, x_2, \ldots, x_m$ z X "náhodne" podľa nejakého známeho, ale pevne daného pravdepodobnostného rozloženia. Učiaci algoritmus L produkuje hypotézu $L(s)$, ktorá je očakávaná ako dobrá aproximácia pre t. Dôslednejšie vyžadujeme, ak počet príkladov m v trénujúcej vzorke vzrastie, tak z pravdepodobnosti vyplynie, že chyba, ktorá je výsledkom použitia $L(s)$ namiesto t je malá.

Základné pojmy:

X - pravdepodobnostný priestor, A - trieda podmnožín množín X μ - pravdepodobnostné rozloženie, miera pravdepodobnosti

$$A \rightarrow \quad [0,1].$$

Od triedy A sa vyžaduje, aby bola uzavretá vzhľadom na operácie komplementu, konečného prieniku a spočítateľného zjednotenia.

$A \in A$, sa nazýva udalosť $\mu(A)$ pravdepodobnosť udalosti A

Od μ sa vyžaduje, aby spĺňovala nasledujúce podmienky:

$$\mu(\emptyset) = 0, \mu(X) = 1,$$

a pre ľub. po dvoch disjunktné množiny $A_1, A_2, \ldots \in A$

$$\mu(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} \mu(A_i).$$

Pre nás:

$$X = \begin{cases} R & \text{- triedu booleovských funkcií v } R^n \\ \text{booleovský priestor} & \text{- konečná alebo spočítateľná} \end{cases}$$

a môžme brať A-triedu všetkých podmnožín X

V oboch prípadoch budeme používať vhodnú triedu bez expl. vyjadrovania detailov. Postačí použiť triedu Boolovských množín v R^n.

Budeme jednoducho hovoriť "pravdepodobnostné rozdelenie μ na X", ktorou mienime funkciu μ definovanú na vhodnej triede A a spĺňujúcej axiómy uvedené vyššie. Musí byť zdôraznené, že v aplikáciách, o ktorých sme sa zmieňovali, nerobíme žiadne predpoklady o μ, okrem podmienok uvedených v definícii. Situácia, ktorú sme modelovali, je že svet príkladov prezentovaných učiacemu sa chová (modeluje) podľa nejakého pevného, ale neznámeho rozloženia. Učiteľovi je povolené klasifikovať príklady ako pozitívne a negatívne, ale nemôže riadiť postupnosť, v ktorej príklady budú prezentované.

Budeme pokračovať s predpokladom, že cieľový koncept patrí do hypotézového priestoru H, ktorý je dostupný učiacemu sa. K danému cieľovému konceptu $t \in H$ definujeme chybu ľubovoľnej hypotézy $h \in H$ vzhľadom na t a bude to pravepodobnosť udalosti $h(x) \neq t(x)$, t.j.

$$er_\mu(h,t) = \mu\{x \in X \quad | \quad h(x) \neq t(x)\}.$$

v kučeravých zátvorkách je error set - chybová množina a predpokladáme, že existuje udalosť taká, že pravdepodobnosť jej môže byť priradená. Keď pôjde o t známe z konceptu, budeme tiež používať označenie $err_\mu(h)$.

Príklad: Nech $X = \{0,1\}^3$, predpokladajme, že cieľov koncept je $\langle u_1 \rangle$. Chybová množina pre hypotézu $\langle u_1 \overline{u_2} \rangle$ obsahuje dva príklady, 110 a 111. Tak

$$err_{<u_1 \overline{u_2}>} = \mu\{110, 111\}.$$

Napríklad, μ - rovnomerné rozloženie na x - $\frac{1}{8}$, potom

$$err_{\langle u_1 \overline{u_2} \rangle} = \frac{1}{4}.$$

Ak z nejakých dôvodov príklady, u ktorých je y_1 sú málo pravdepodobnostné, potom er_μ bude o niečo menšia.

Keď je daná množina X poskytovaná sa štruktúrou pravdepodobnostného priestoru, súčin množín X^m preberá pravdepodobnostnú štruktúru X. Detaily sa nás netýkajú, je postačujúce poznamenať, že konštrukciu nám umožňuje považovať komponenty za nezávislé premenné, rozloženie každej z nich je podľa pravdepodobnostného rozloženia μ na X. Odpovedajúce pravdepodobnostné rozdelenie na X^m je označované μ^m. Neformálne, pre dané $Y \subseteq X^m$ budeme interpretovať hodnotu $\mu^m(Y)$ ako "pravdepodobnosť, že náhodná vzorka m príkladov vybratých z X podľa rozdelenia patrí do Y".

Nech $S(m,t)$ označuje množinu tréningových vzoriek dĺžky m pre daný cieľový koncept t, kde príklady sú vyberané z príkladového priestoru X. Ľubovoľná vzorka $x \in X$ determinuje a je determinovaná tréningovou vzorkou $\overline{s} \in$ S(m,t): ak $\overline{x} = (x_1, x_2, \ldots, x_m)$, potom $\overline{s} = ((x_1, t(x_1), (x_2, t(x_2), \ldots)$. Inak povedané, existuje zobrazenie Φ

$$\Phi : X^m \to S(m,t), \quad \text{pre ktorú} \quad \Phi(x) = \overline{s}$$

Teda môžeme interpretovať pravdepodobnosť, že $\overline{s} \in S(m,t)$ má nejakú danú vlastnosť P, nasledujúcim spôsobom. Definujeme

$$\mu^m\{s \in S(m,t) \quad | \quad \text{s má vlastnosť P } \}$$

to znamená

$$\mu^m\{x \in X^m \quad | \quad \Phi(x) \in S(m,t) \text{má vlastnosť P } \}$$

Z toho vyplýva, že keď príkladový priestor X je vybavený pravdepodobnostným rozložením, môžeme zaviesť precíznejšiu interpretáciu pre

(i) chybu hypotézy, ktorá vznikne, keď učiaci algoritmus L pracuje s $\overline{s}$; Táto veličina pracuje s $er_\mu(L(\overline{s}))$.

(ii) pravdepodobnosti, že táto chyba je menšia než ϵ.

Druhá je pravdepodobnosť vzhľadom na μ_m, že s má vlastnosť $er_\mu(L(\overline{s})) < \epsilon$

3.2.1 PAC - algoritmus

Hovoríme, že algoritmus L je **probably approximately correct** (pravdepodobnostne aproximačne správny) učiaci algoritmus, pre hypotézový priestor H, ak

- k ľubovoľnému reálnemu číslu δ, $0 \leq \delta \leq 1$
- k ľub. reálnemu číslu ϵ, $0 \leq \epsilon \leq 1$
- existuje kladné celé číslo $m_0 = m_0(\delta, \epsilon)$ také ,že
- pre ľub. cieľový koncept $t \in H$,
  pre ľub. pravdepodobnostné rozloženie μ na X pre všetky $m \geq m_0$ platí

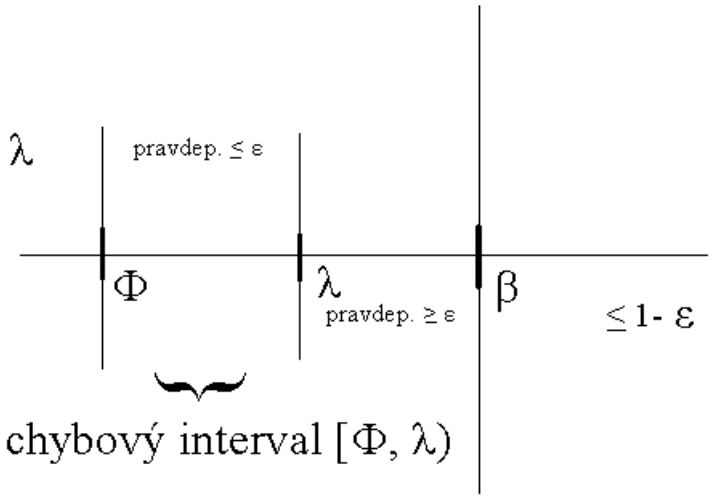
$$\mu^m\{s \in S(m,t) \quad | \quad er_\mu(L(s)) < \epsilon\} > 1 - \delta$$

$$\forall_{(0\leq\delta\leq 1)}\forall_{(0\leq\epsilon\leq 1)}\exists_{(m_0=m_0(\epsilon,\delta))}\forall_{(t\in H)}\forall_{(\mu \text{ na } X)}\forall_{(m\geq m_0)}\mu^m\{s \in S(m,t) \quad | \quad er_\mu(L(s)) < \epsilon\} > 1 - \delta$$

Skutočnosť, že m_0 závisí od δ a ϵ, ale nie od t a μ odráža to, že učiaci sa môže byť schopný špecifikovať predpokladanú úroveň dôvery a presnosti, aj keď cieľový koncept a rozloženie príkladov sú neznáme. Dôvodom k tomu, že je možné splniť podmienku pre ľubovoľné μ je, že vyjadruje vzťah medzi dvoma veličinami, ktoré obsahujú μ: chyba err_μ a pravdepodobnosť vzhľadom na μ^m určitej množiny. PAC učenie je, v istom zmysle, najlepšie, v čo môžeme dúfať pri tomto pravdepodobnostnom pohľade. Nereprezentatívne tréningové vzorky, hoci nepravdepodobné, budú príložitostne prezentované učiacemu algoritmu, a tak môžeme očakávať, že je pravdepodobné, že je prezentovaná použiteľná tréningová vzorka. Naopak, aj keď máme reprezentatívnu trénujúcu vzorku, rozšírenie tréningovej vzorky nebude vo všeobecnosti koincidovať s cieľovým konceptom, takže aj tak výstupná hypotéza je len aproximačne správna.

3.3 Učenie lúčov je PAC.

Veta 2 *Algoritmus L pre učenie lúčov je PAC.*



Definujme

$$\beta_0 = \beta_0(\epsilon, \mu) = \sup\{\beta \quad | \quad \mu[\Theta, \beta) < \epsilon\}$$

Ak uvažujeme $\lambda \leq \beta_0 : er_\mu(L(\overline{s})) = \mu[\Theta, \lambda) \leq \mu[\Theta, \beta_0) \leq \epsilon\}$

Udalosť, že $\overline{s}$ má vlastnosť $\delta \leq \beta_0$ je práve udalosť, že aspoň jeden príklad v $\overline{s}$ je v intervale $[\Theta, \beta_0]$. Pretože $\mu[\Theta, \beta_0] \geq \epsilon$, pravdepodobnosť, že jeden príklad nie je v tomto intervale je najviac $1 - \epsilon$. Preto pravdepodobnosť, že žiadny z m príkladov vzorky $\overline{s}$ nie je v tomto intervale je najviac $(1 - \epsilon)^m$. Keď budeme uvažovať komplementárnu udalosť (existuje príklad, ktorý je z tohto intervalu), z toho vyplýva, že pravdepodobnosť, že $\lambda \leq \beta_0$ je aspoň $1 - (1 - \epsilon)^m$. Ako sme už poznamenali vyššie, že udalosť $\lambda \leq \beta_0$ implikuje udalosť $er_\mu(L(\overline{s})) \leq \epsilon$ a tak $\mu^m\{s \in S(m, r_\Theta) \quad | \quad er_\mu(L(\overline{s})) \leq \epsilon\} \geq 1 - (1 - \epsilon)^m$

Položíme

$$m \geq m_0 = \frac{1}{\epsilon} * ln\frac{1}{\delta}$$

$$(1 - \epsilon)^m \leq (1 - \epsilon)^{m_0} < e^{-\epsilon m_0} < e^{ln\delta} = \delta$$

tento výpočet ukazuje, že algoritmus je PAC.

Dôkaz korektnosti poskytuje explicitnú formulu pre dĺžku vzorky postačujúcu na to, aby boli splnené predpísané hodnoty presnosti a dôveryhodnosti. Predpokladajme, že $\delta = 0.001$ $\epsilon = 0.01$
$m_0 = \frac{1}{0.01} * \ln\frac{1}{0.001} = 100 * \ln 1000 = 691$

Takže aspoň 691 príkladov je treba, aby sme si boli istí na 99,9%, že najviac 1% príkladov bude klasifikovaných nesprávne, za predpokladu, že sú z toho istého zdroja ako tréningová vzorka.

Odvodenie vzťahu:

$\delta = (1 - \epsilon)^m$
$\ln\delta = m * ln(1 - \epsilon) = m * \frac{-\epsilon}{1-\epsilon}$ $\quad \ln\delta \leq m * \frac{-\epsilon}{1-\epsilon}$
$\ln(1 - \epsilon) \leq ln(1) + f'(0) * \frac{\epsilon}{1!} \leq 0 + \frac{1}{1-\epsilon} * (-1) * \frac{\epsilon}{1!}$
$-\ln\delta = m * \frac{\epsilon}{1-\epsilon}$ $\quad \frac{1-\epsilon}{\epsilon} ln\frac{1}{\delta} \leq m \Rightarrow (\frac{1}{\epsilon} - 1)\ln\frac{1}{\delta} \leq m$
$\frac{1}{\epsilon} * ln\frac{1}{\delta} + \ln\delta \leq m$

3.4 Exaktné učenie

Keď príkladový prisetor X je konečný, pojem PAC - učenie má ďalšie dodatočné obmedzenia. Začneme tým, že ľubovoľné pravdepodobnostné rozloženie na konečnej množine X je determinované hodnotami na jej 1-prvkových množinách x, použitím axiómy o aditivite. Budeme písať $\mu(x)$ namiesto $\mu(\{x\})$. Ak budú nejaké príklady, pre ktoré $\mu(x) = 0$, s pravdepodobnosťou 1 sa nebudú vyskytovať v konečnej náhodnej vzorke a môžu byť ignorované. Inými slovami, môžme ak je nutné predefinovať X tak, že $\mu(x) > 0$ pre všetky $x \in X$. Pretože X je konečná, veličina

$$\epsilon_\mu = min_{\{}x \in X\}\mu\{x\} > 0$$

je dobre definovaná.

Predpokladajme, že máme algoritmus L, ktorý je PAC pre hypotetický priestor H definovaný na X. Vo význame definície PAC algoritmu máme dané $\delta, \epsilon, \mu, at$ v ich obvyklom význame

$$m \geq m_0 \Rightarrow \mu^m\{\overline{s} \in S(m,t) \quad | \quad er_\mu(L(\overline{s})) < \epsilon\} > 1 - \delta$$

Predpokladajme, že presnosť ϵ je vybratá tak, aby nebola väčšia než ϵ_μ. Potom podmienka $er_\mu(L(\overline{s})) < \epsilon$ implikuje, že chybová množina pre $L(\overline{s})$ je prázdna, pretože neexistujú žiadne príklady, ktoré majú pravdepodobnosť menšiu než ϵ. Teda podmienka implikuje, že $L(s) = t$, t.j. výstupná hypotéza je presne rovná cieľovému konceptu t. Záver predchádzajúceho argumentu je, že pre učenie na konečnom priestore je "pec-probably exactly correct". Ale je v tom háčik. Jednoduchá vzorka dĺžky m_0 v definícii PAC-učenia závisí od parametrov δ, ϵ ale nezávisí od μ (a t).

Argument uvedený vyššie obsahuje výber ϵ pomocou ϵ_μ, a tak hodnota m_0 vyžadovaná pre exaktné učenie bude závisieť od δ a μ. Toto je v spore s naším originálnym cieľom dokazovania výkonných záruk, ktoré nie sú nezávislé od μ, možno neznáme rozloženie príkladov vo svete bez pomoci.

Príklad: Štandartný učiaci algoritmus pre monočleny na $\{0,1\}^n$ pre pevné n.Uvedieme "PEC" vlastnosť. Kľúčovým zdelením tu je, že algoritmus poskytuje správne hypotézy, poskytované všetkými hľadanými príkladmi, ktoré boli zahrnuté do tréningovej vzorky. Dĺžka tréningovej vzorky rastie a rastie tiež pravdepodobnosť, že vzorka obsahuje všetky kladné príklady; postupne tak urobí pravdepodobnosť, že výstup je korektný. Presnejšie, nech ϵ_μ bude najmenšia hodnota $\mu(x)$, ktorú uvažujeme nad množinou $x \subseteq \{0,1\}^n$ príkladov s nenulovou pravdepodobnosťou. Potom pravdepodobnosť, že trénujúca vzorka dĺžky m neobsahuje daný príklad je najviac $(1 - \epsilon_\mu)^m$. Pravdepodobnosť, že existuje jeden z danej množiny p príkladov, ktoré nie sú v tréningovej vzorke je preto $p * (1 - \epsilon_\mu)^m$. Ak X^+ je množina kladných príkladov pre daný cieľový koncept t, pravdepodobnosť, že ex. člen v X^+, ktorý nie je vo vzorke je najviac

$$|X^+|(1 - \epsilon_\mu)^m$$

Potrebujeme vyjadriť m, teda

$$|X^+|(1 - \epsilon_\mu)^m < \delta$$

$$m \lg(1 - \epsilon_\mu) + \lg|X^+| < \lg\delta$$

$$\lg|X^+| - \lg\delta < -m \lg(1 - \epsilon_\mu) < m\epsilon_\mu$$

Použijeme nejaké známe skutočnosti:

$$|X^t| \leq |X| \leq 2^n \qquad \text{a} \qquad 1 - \epsilon_\mu < exp(-\epsilon_\mu)$$

$$log|X^+| \leq n \qquad log(1 - \epsilon_\mu) < -\epsilon_\mu$$

$$m \geq \left\lceil \frac{n}{\epsilon_\mu} ln2 + \frac{1}{\epsilon_\mu} ln\frac{1}{\delta} \right\rceil$$

Poznamenajme, že dĺžka vzorky je nezávislá od t, ale závisí od rozloženia cez parameter ϵ_μ.

3.5 Ďalšie poznámky

Vo Valiantovom originálnom popise učenia bol predpoklad, že učiaci algoritmus malprístupk"orákulu", ktoré generovalo označené príklady cieľového konceptu brané podľa rozloženia v príkladovom priestore. V takom modeli vstup do algoritmu pozostáva jedine z parametrov *delta* a *epsilon*: algoritmus sám potom používa orákulua na generovanie dostatočne veľa označkovanýchpríkladov na zabezpečenie toho, aby výstupnáhypotéza bola PAC.tento model je všeobecne známy ako model s orákulom, pokým model popísaný v tejto knihe je funkcionálny model. Haussler et al (1988) ukázal, že tieto verzie učiaceho modelu a niekoľko inýchvariantov sú,vzhľadom na všetky zámery a ciele, ekvivalentné.

Predpokladajme, že L je bezpamäťový on-line učiaci algoritmus pre nejaký priestor H a že na vstupe zadávame nejakú trénujúcu vzorku s pre hypotézu t z H. Umožníme aby s bola vybraná ľubovoľne: tj.nemusí byť vyberaná podľa nejakého rozloženia na príkladovom priestore, ale môže, napríklad, byť postupnosťou vybranou zlomyseľne učiteľom, ktorý sa snažídaťučiacemu sa takmálo informácií ako lenmôže. Predpokladajme, že L updatuje jeho aktuálnu hypotézu zakaždým keď urobí chybu na príklade v s. Inak povedané,L prispôsobuje aktuálne hypotézy po prezentácii označkovaného príkladu, s ktorým jeho aktuálna hypotéza nesúhlasí.

Hovoríme, že L má absolútnu chybovú hranicu k, ak na ľub. trénujúcej vzorke, ľub.dĺžky, L urobí najviac k chýb. Chybovo ohraničený učiaci model poskytuje všeobecný rámec pre štúdium tejto situácie; viď, napr. Littlestone (1988). Ex. niekoľko výskumníkov, ktorí študovali tieto modely a ich varianty a dávali ich dovzťahu k PAC modelom popísaných tu, ale tiežk iným modelom učenia: Littlestone (1988),Angluin (1988), Haussler, Littlestone and Wermuth (1988) a Blum (1990).

Je mnoho typov chýb, ktoré sa môžu vyskytnúť počas praktickej implementácie pastikulárnych učiacich algoritmov a mnoho z nich bolo sformulovaných.Sloan (1988). Napríklad, Angluim Laird (1987) vyprodukovali algortimy pre PAC učeniea prítomnosť náhodných nekvalifikovaných chýb, pokiaľ Kearns a Li (1988) študovali tento model a silnejšie učenie sprítomnosťou zlomyseľných chýb a dosiahli výsledky.

Niekoľko variantov PAC - učenia bolo dosiahnutých tak, že bolo umožnené,že učiaci algoritmus a vzorka dostatočnej dĺžky m_0 záviseli nejakým spôsobom buď na rozložení pravdepodobnosti μ alebo na cieľovom koncepte t. Toto nie je umelé: v mnohých učiacich problémoch, niečo je známe ako rozloženie alebo cieľ. Výsledné definície naučiteľnosti sú menej atraktívne než bezkonceptové a bez-rozloženia PAC definície, ale sú veľmi často ľahko splnené. Veľa práce bolo urobenej na takom "neuniformnom" PAC učení. Ben-David et al (1989), Benedek and Itai (1988), Liniál et al (1989), Kearns et al (1987a), Li and Vitanyi (1989), Baum (1990), Natarajan (1988), Bartlett and Williams (1991).

3.6 Úlohy: