Výpočtové učenie

10. mája 2005

Kapitola 1

Sylabus a odporúčaná literatúra

Forma: Prednáška, cvičenie – 2/1 hod. Školský rok: 2004/2005, LS
Výučbu zabezpecujú: Ústav informatiky, Doc. RNDr. G. Andrejková, CSc.

Obsah predmetu:

1. Učiace algoritmy, koncepcie, hypotézy. Tréning a učenie, učenie konštrukciou a očíslovaním.
2. Booleovské formuly a ich reprezentácia. Učiace algoritmy pre monočleny. Reprezentácia hypotézového priestoru.
3. Pravdepodobnostné učenie.
4. Konzistentné algoritmy a učenie. Potenciálna naučitelnosť. Konzistentný algoritmus pre rozhodovacie zoznamy.
5. Efektívne učenie. Čas behu učiaceho algoritmu. Problém konzistencie. Veľkosť reprezentácie. Occam algoritmus.
6. Hľadanie najmenšej konzistentnej hypotézy. Occam algoritmy.
7. Test č. 1
8. VC (Vapnik - Cervonenkis) dimenzia jej vzťah k perceptrónom. Sauerova lema.
9. Učenie vo vzťahu k VC dimenzii.
10. VC dimenzia a efektívne učenie.
11. Výpočtové učenie a lineárne prahové jednotky.
12. Test č. 2
13. Výpočtové učenie a neurónové siete.
14. Rekapitulácia učiva.

Odporúčaná literatúra:

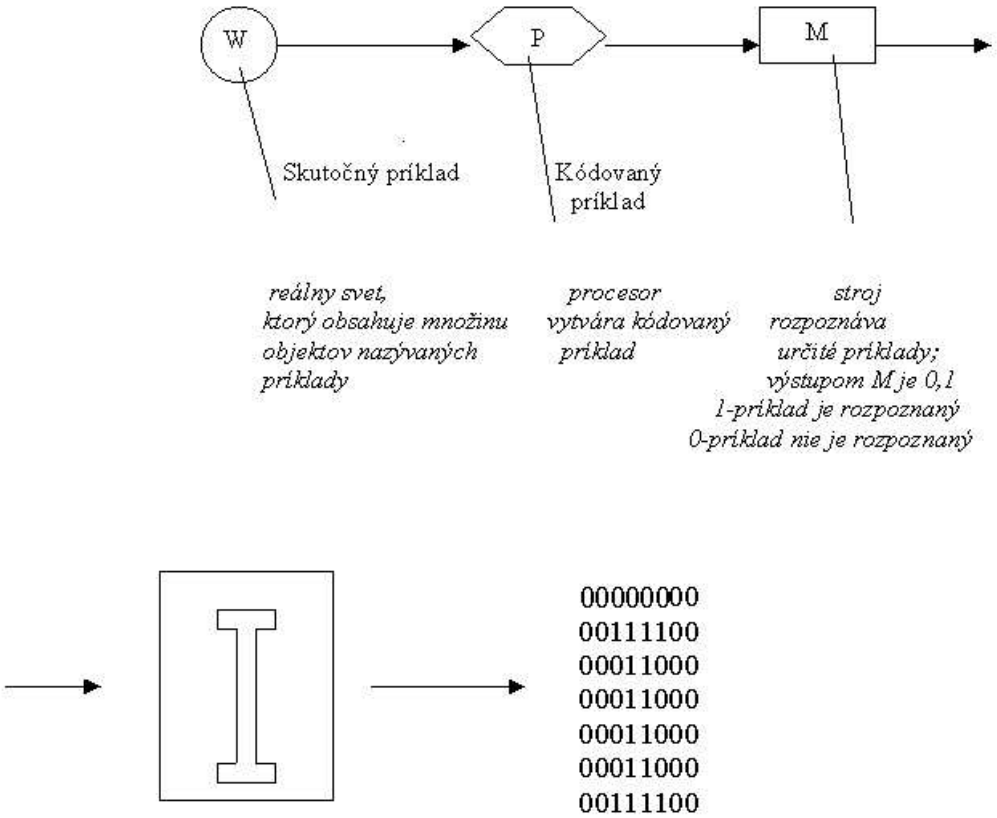
1. M. Anthony, N. Biggs: Computational Learning Theory, Cambridge University Press, 1991, 1997.
2. S. A. Goldman: Computational Learning Theory, Lecture Notes for CS 582, Washington University, 1991.
3. S. J. Russell, P. Norvig: Artificial Intelligence, Prentice-Hall International, Inc., 1995.
4. M. J. Kearns,U. V. Vazirani: An Introduction to Computational Learning Theory, The MIT Press, London, 1994.
5. M. I. Schlesinger, V. Hlaváč: Deset prednášek z teorie štatistického a strukturovaného rozpoznávání. CVUT, Praha, 1999.

Kapitola 2

Koncepty, hypotézy, učiace algoritmy

2.1 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 2.1: Spracovanie príkladov reálneho sveta pomocou naténovaného stroja

2.2 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 2.2.1 *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
   $\Sigma = R$ $u : \Sigma^n \to \{0, 1\}$ $y = y_1 \dots y_n$:

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

2.3 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 $x_ı = x_j, \quad \Rightarrow \quad b_ı = b_j$. To teda znamená, že existuje funkcia s, definovaná ako $s(x_ı) = b_ı \quad (1 \leq i \leq m)$.

Budeme hovoriť, že $\overline{s}$ je tréningová vzorka pre cieľový koncept t, ak $b_ı = t(x_ı)$, 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: $x = (x^1 \ldots x^n)$:

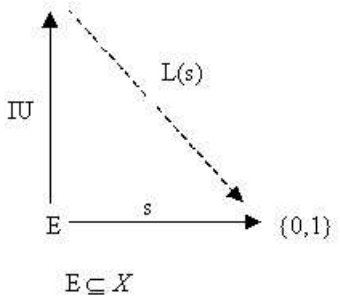
$$t(x) = \begin{cases} 1 & \text{ak } x^ı = x^{n-i+1} \quad pre 1 \leq i \leq n \\ 0 & \text{inak} \end{cases}$$

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 $h(x_ı) = b_ı$, 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

2.4 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 2.4.1 *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* $h_ı$ *a od priebežne spracovávaného príkladu vzorky* $(x_ı, b_ı)$.

2.5 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.

2.6 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 3

Booleovské formuly a reprezentácie

3.1 Učiaci algoritmus pre monočlenné funkcie

Valiant, 1984 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 literálovu_j, pre ktoré $y_j = 0$ a tie literály $\overline{u_j}$, pre ktoré $y_j = 1$. Predpokladajme, že $\overline{s}$ 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: Štandardný učiaci algoritmus pre monočleny je konzistentný s výnimkou premenných, na ktorých nezáleží.

3.1.1 Disjunktná normálna forma - DNF

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

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

3.1.2 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 $M_{n,k}$

3.2 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;
```

3.3 Reprezentácia hypoézového 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.

3.4 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 vzork, ktoré sú pre to nutné.

Kapitola 4

Pravdepodobnostné učenie

4.1 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.

4.2 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 splňovala 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 splňujúcej 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 $er_\mu(L(\overline{s})) < \epsilon$

4.2.1 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

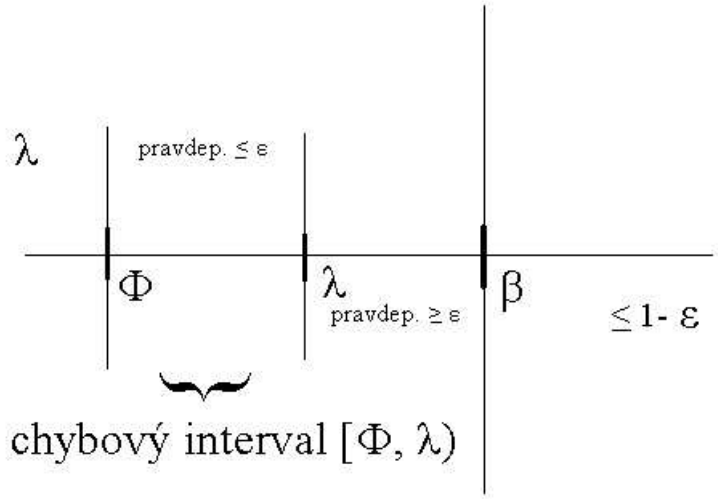
$$\mu^m\{s \in S(m,t) \quad | \quad er_\mu(L(s)) < \epsilon\} > 1 - \delta$$

$$\forall_{(0\leq\delta\leq1)}\forall_{(0\leq\epsilon\leq1)}\exists_{(m_0=m_0(\epsilon,\delta))}\forall_{(t\in H)}\forall_{(\mu \text{ na } X)}\forall_{(m\geq m_0)}\mu^m\{s \in S(m,t) \quad | \quad er_\mu(L(s)) < \epsilon\} > 1 - \delta$$

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.

4.3 Učenie lúčov je PAC.

Veta 1 *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:

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

4.4 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.

4.5 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).

4.6 Úlohy:

Kapitola 5

Konzistentné algoritmy a naučiteľnosť

5.1 Potenciálna naučiteľnosť

Učenie v zmysle PAC je vlastnosť algoritmu. Ak je algoritmus daný, môžeme sa pokúšať dokázať, že je PAC, ale môžeme tiež uvažovať vo všeobecnejšej rovine.

V tejto časti budeme popisovať **vlastnosť hypotézového priestoru H**, ktorá zaručí, že konzistentný algoritmus pre učenie H podľa H je **PAC** a ukážeme, že mnoho priestorov má túto vlastnosť.

Definícia 5.1.1 *Nech H je hypotézový priestor funkcií definovaných na príkladovom priestore X. Učiaci algoritmus L pre H je **konzistentný**, ak pre ľubovoľnú trénujúcu vzorku s a cieľový koncept $t \in H$, výstupná hypotéza $h = L(s) \in H$ súhlasí s t na príkladoch v $\overline{s}$, t. j. $h(x_i) = t(x_i) \quad (1 \leq i \leq m)$.*
Pre dané $\overline{s} \in S(m,t)$ je obvyklé označenie $H[\overline{s}] = \{h \in H| \quad h(x_i) = t(x_i), \quad 1 \leq i \leq m\}$, $H[\overline{s}]$ je množina všetkých hypotéz konzistentných s $\overline{s}$.

Teda L je konzistentný vtedy a len vtedy, keď $L(\overline{s}) \in H[\overline{s}]$ pre všetky trénujúce vzorky $\overline{s}$.

Z toho vyplýva, že zabezpečiť, aby konzistentný učiaci algoritmus bol PAC, postačí zadať podmienky na množiny $H[\overline{s}]$.

Ako predtým predpokladajme, že je dané pravdepodobnostné rozloženie μ na X. Na moment zafixujme cieľový koncept $t \in H$.
K danému $\epsilon \in (0,1)$ položíme $B_\epsilon = \{h \in H| \quad er_\mu(h) \geq \epsilon\}$ čo môže byť popísané ako množina ϵ-zlých hypotéz pre t. Konzistentný algoritmus pre H dáva výstup, ktorý je v $H[\overline{s}]$ a PAC vlastnosť vyžaduje, aby taký výstup, ktorý je nepravdepodobný, bol ϵ-zlý; inak povedané, zdôrazníme, že nepravdepodobné, že zlá hypotéza je korektná na vzorke. Toto vedie k nasledujúcej definícii:

Definícia 5.1.2 *Hovoríme, že hypotézový priestor H je **potenciálne naučiteľný**, ak k daným reálnym číslam $\delta, \epsilon, 0 < \delta, \epsilon < 1$ existuje kladné celé číslo $m_0 = m_0(\delta, \epsilon)$ také, že pre všetky $m \geq m_0$, platí*

$$\mu^m\{s \in S(m,t)| \quad H[s] \cap B_\epsilon = \emptyset\} > 1 - \delta$$

pre ľubovoľné pravdepodobnostné rozloženie μ na X a ľub. $t \in H$.

Veta 2 *Ak H je potenciálne naučiteľný a L je konzistentný učiaci algoritmus pre H, potom L je PAC.*

Dôkaz: L je konzistentný, teda $L(\overline{s}) \in H[\overline{s}]$.

$$H[\overline{s}] = \{h \in H| \quad h(x_i) = t(x_i), \quad 1 \leq i \leq m\}$$

$$B_\epsilon = \{h \in H| \quad err_\mu(h) \geq \epsilon\}$$

$L(\overline{s}) \in H[\overline{s}]$ a zároveň $H(s) \cap B_\epsilon = \emptyset$, teda chyba $L(s)$ je menšia ako ϵ.

$$\forall_\delta \, \forall_\epsilon \, \exists_{m_0} \, \forall_{m>m_0} \, \forall_\mu \, \forall_{t \in H} \; \mu^m\{\overline{s} \in S(m,t)| \quad H[\overline{s}] \cap B_\epsilon = \emptyset\} > 1 - \delta$$

$$\forall_\delta \, \forall_\epsilon \, \exists_{m_0} \, \forall_{m>m_0} \, \forall_\mu \, \forall_{t \in H} \; \mu^m\{\overline{s} \in S(m,t)| \quad err_\mu(L(\overline{s})) < \epsilon\} > 1 - \delta$$

$$err_\mu(L(\overline{s})) = \mu\{x \in X| \quad t(x) \neq L(\overline{s})\}$$

□

5.2 Konečný prípad

Definícia potenciálnej naučiteľnosti je celkom zložitá a môže byť dokázané, že viac-menej je to popis PAC učenia. Naša úloha je teraz potvrdiť definíciu tým, že ukážeme jej významné aplikácie.

Veta 3 *Ľubovoľný konečný hypotézový priestor H je potenciálne naučiteľný.*

Dôkaz: Predpokladajme, že H je konečný hypotézový priestor a $\delta,\ \epsilon,\ t$ a μ sú dané. Dokážeme, že pravdepodobnosť udalosti $H[\overline{s}] \cap B_\epsilon \neq \emptyset$ (komplement udalosti v definícii) môže byť zvolená menšia než δ vybratím dostatočne veľkej dĺžky vzorky $\overline{s}$.

Pretože B_ϵ je definovaná tak, že obsahuje ϵ-zlé hypotézy, z toho vyplýva, že pre ľub. $h \in B_\epsilon = \{h \in H|\quad \mu\{x \in X, h(x) \neq t(x)\} \geq \epsilon\}$ platí

$$\mu\{x \in X|\quad h(x) = t(x)\} = 1 - err_\mu(h) \leq 1 - \epsilon$$

Teda pre celú vzorku dĺžky m máme

$$\mu^m\{s|\quad h(x_i) = t(x_i),\ 1 \leq i \leq m\} \leq (1-\epsilon)^m$$

Toto je pravdepodobnosť, že hypotéza h je ϵ-zlá pre celú vzorku, a teda je to pravdepodobnosť, že nejaká ϵ-zlá hypotéza je v $H[\overline{s}]$.
Pravdepodobnosť, že nejaká ϵ-zlá hypotéza je v $H[\overline{s}]$, je vyjadriteľná

$$\mu^m\{s|\quad H[\overline{s}] \cap B_\epsilon \neq \emptyset\}$$

a je preto menšia než $|H| \cdot (1-\epsilon)^m$. Toto bude menej ako δ za predpokladu, že položíme $m \geq m_0 = \left\lceil \frac{1}{\epsilon} \cdot \ln \frac{|H|}{\delta} \right\rceil$, pretože v tomto prípade

$$|H| \cdot (1-\epsilon)^m \leq |H| \cdot (1-\epsilon)^{m_0} < |H| \cdot \exp(-\epsilon \cdot m_0) \leq |H| \cdot e^{\ln \frac{\delta}{|H|}} = |H| \cdot \frac{\delta}{|H|} = \delta$$

Dokázali sme, že pre ľub. $\delta,\ \epsilon,\ t,\ \mu$ existuje m_0

$$\forall_{m \geq m_0} \forall_t \quad \mu^m\{s \in S(m,t)|\quad H[s] \cap B_\epsilon \neq \emptyset\} < \delta$$

Ak vezmeme komplementárnu udalosť, dostaneme správny záver. □

Je zrejmé, že toto je použiteľná veta. Pokrýva všetky bool. prípady, kde príkladový priestor je $\{0,1\}^n$ (alebo podmnožina) s pevným n. V ľubovoľnej takej situácii konzistentný algoritmus je a utomaticky PAC. Napríklad algoritmus pre učenie monočlenov a disjunkcií malých monočlenov prezentovaných sú PAC. Dôkaz nám ďalej hovorí, koľko príkladov postačí na dosiahnutie predpísaných úrovní dôvery a presnosti.

Pre algoritmus monočlenov vieme, že veľkosť $|M_n|$ hypotézového priestoru je 3^n. Preto

$$m_0 = \left\lceil \frac{1}{\epsilon} \ln \frac{|M_n|}{\delta} \right\rceil = \left\lceil \frac{1}{\epsilon} \left(n \cdot \ln 3 + \ln \frac{1}{\delta} \right) \right\rceil$$

je postačujúci počet príkladov na zabezpečenie, že pre pravdepodobnosť väčšiu než $1-\delta$ výstup algoritmu má chybu menšiu než ϵ.

Navyše pre ľub. konečný hypotetický priestor existuje konzistentný učiaci algoritmus: metda učenia očíslovaním, ktorú sme popísali. Teda bezprostredným dôsledkom vety je, že k ľub. konečnému hypotézovému priestoru H existuje učiaci algoritmus, ktorý je PAC.

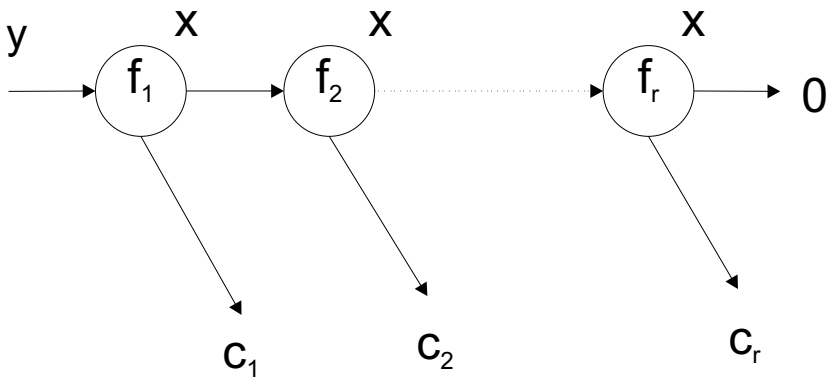
V tomto bode by sa mal čitateľ zadiviť, prečo je to tak. Vytvorili sme komplikovanú podmienku, aby sme dokázali, že je vždy splnená v konečnom prípade, ktorý je najdôležitejší v praxi. Ale praktické predpoklady zavádzajú dodatočné ohraničenie, že počet príkladov by mal byť "ovládateľný" a teda nie je nutný prípad s metódou učenia očíslovaním. Predpokladajme, že napríklad ak hypotézový priestor je množina B_n všetkých booleovských funkcií n premenných, potom hranica pre vzorku je $m_0 = \left\lceil \frac{2^n}{\epsilon} \ln \frac{2}{\delta} \right\rceil$.

5.3 Rozhodovacie zoznamy

Jeden spôsob popisovania zložitých konceptov je ich vytváranie z menších jednotiek. Viď vytváranie $D_{n,k}$.

V tejto sekcii popíšeme inú metódu konštrukcie, ktorá môže byť aplikovaná na ľub. danú množinu vytvárajúcich blokov.

Definícia 5.3.1 *Nech K je ľubovoľná množina bool. funkcií na $\{0,1\}^n$, n je pevné. Booleovská funkcia f s tým istým oborom ako K sa nazýva **rozhodovacím zoznamom** (podľa Rivesta) založeným na K, ak môže byť vyhodnotená nasledovne:*
Nech je daný príklad y. Najprv vyhodnotíme $f_1(y)$ pre nejaké pevné $f_1 \in K$.
Ak $f_1(y) = 1$, priradíme do $f(y)$ pevnú hodnotu c_1.
Ak $f_1(y) \neq 1$, vyhodnotíme $f_2(y)$ pre nejaké pevné $f_2 \in K$.
Ak $f_2(y) = 1$, priradíme do $f(y)$ pevnú hodnotu c_2,
inak vyhodnotíme $f_3(y)$, atď.



Obrázok 5.1: Vyhodnotenie funkcie pomocou rozhodovacieho zoznamu

V inom zápise

```
    if f_1(y) = 1 then set f(y) = c_1
else if f_2(y) = 1 then set f(y) = c_2
⋮
else if f_r(y) = 1 then set f(y) = c_r
else set f(y) = 0
```

Formálnejšie môžeme definovať:

Definícia 5.3.2 *$DL(K)$ je priestor **rozhodovacích zoznamov na množine** K - je to množina konečných postupností $f = (f_1, c_1), (f_2, c_2), \ldots, (f_r, c_r)$ takých, že $f_i \in K$, $c_i \in \{0,1\}, (1 \leq i \leq r)$. Hodnoty f sú definované*

$$f(y) = \begin{cases} c_j, & akj = min\{i| \quad f_i(y) = 1\} \ existuje \\ 0, & inak \end{cases}$$

Bez újmy na všeobecnosti, keď budeme vyžadovať, aby všetky termy v rozhodovacom zozname boli rôzne, pretože opakovanie danej funkcie $g \in K$ môže byť odstránené bez účinku na vyhodnotenie. Teda dĺžka rozhodovacieho stromu založeného na konečnej množine K je najviac $|K|$ a $|DL(K)|$ je ohraničená funkciou $|K|$.

Príklad 1 *Predpokladajme, že $K = M_{3,2}$ je priestor monočlenov dĺžky najviac 2 z 3 premenných. Rozhodovací zoznam*

$$(\langle u_2 \rangle, 1), (\langle u_1 \overline{u_3} \rangle, 0), (\langle \overline{u_1} \rangle, 1)$$

môže byť spracovávaný nasledujúcim spôsobom na príkladovom priestore $\{0,1\}^3$.
Najprv sú vybrané tie príklady, pre ktoré $\langle u_2 \rangle$ *dáva hodnotu 1. Sú to 010, 011, 110, 111.*
ďalej zostanú len tie, ktoré pre $u_1\overline{u_3}$ *dajú 0. Sú to len 100.*
ďalej tie, ktoré pre $\langle \overline{u_1} \rangle$ *dávajú hodnotu 1. Sú to 000, 001.*
Zostáva jediný príklad 101, ktorému je priradená hodnota 0.

Je dôležité poznamenať, že disjunkcia 2 funkcií v K je špeciálny prípad rozhodovacieho zoznamu založeného na K. Explicitne $f \vee g$ je reprezentovaná rozhodovacím zoznamom $(f,1),(g,1)$.

To znamená, že rozhodovací zoznam je zovšeobecnenie disjunkcie. Napríklad, priestor $D_{n,k}$ je obsiahnutý v $DL(M_{n,k})$; v skutočnosti je vlastnou podmnožinou. Iný dôsledok je, že pre dané n, ľubovoľná bool. funkcia n premenných je v nejakom $DL(M_{n,k})$ pre k dostatočne veľké. (Bezprostredne, toto je pravda pre $k = n$ podľa existencie disjunktnej normálnej formy.) Ďalšie detaily v cvičení na konci odseku.

5.4 Konzistentný algoritmus pre rozhodovacie zoznamy

Popíšeme učiaci algoritmus pre $DL(K)$, ktorý pracuje, keď K je ľub. konečná množina. Algoritmus je konzistentný, ale nie je bezpamťový on-line algoritmus. Samozrejme, učiaci algoritmus očíslovaním má podobné vlastnosti a teda zostáva zistiť, či tento algoritmus je podstatným vylepšením. Tento bod bude prediskutovaný neskôr.

Algoritmus môže byť popísaný nasledovne.

Nech s je trénujúca vzorka označených príkladov $(x_i, b_i), \quad 1 \leq i \leq m$. V každom kroku konštrukcie požadovaného rozhodovacieho zoznamu niektoré príklady budú vymazané, iné zostanú. Procedúra prebehne cez K hľadajúc funkciu $g \in K$ a bit c taký, že pre všetky zostávajúce príklady x_i, vždy keď $g(x_i) = 1$, tak b_i je konštantná booleovská hodnota c. Dvojica (g,c) je potom vybratá ako ďalší term postupnosti definujúcej rozhodujúci zoznam, a všetky príklady spajúce g sú vymazané. Procedúra je opakovaná, pokým všetky príklady v s neboli vymazané.

Nech $\{g_1, g_2, \ldots g_p\}$ je očíslovanie K.

Algoritmus:

```
set I= {1,2, ..., m };
j:=1;
repeat
if forall i in I, g[j](x[i])=1 implikuje b[i]=c
    then begin select (g[j],c);
               delete from I all i for which g[j](x[i])=1;
               j:=1;
         end
   else j:=j+1;
until I is empty set;
```

Príklad 2 $K = M_{5,2}$ *a predpokladajme, že* K *je zaevidované v "slovníkovom poradí" založenom na usporiadaní* $u_1\ u_2\ u_3\ u_4\ u_5\ \overline{u_1}\ \overline{u_2}\ \overline{u_3}\ \overline{u_4}\ \overline{u_5}$ *ako literálov.*

Prvých niekoľko funkcií v slovníku je: monočleny ident. 1

$$\langle\rangle\ \langle u_1 \rangle\ \langle u_1 u_2 \rangle\ \langle u_1 u_3 \rangle\ \langle \overline{u_1} u_4 \rangle$$

Predpokladajme, že trénovacia vzorka je:

$x_1 = 10000 \quad b_1 = 0$
$x_2 = 01110 \quad b_2 = 0$
$x_3 = 11000 \quad b_3 = 0$
$x_4 = 10101 \quad b_4 = 1$
$x_5 = 01100 \quad b_5 = 1$
$x_6 = 10111 \quad b_6 = 1$

Na začiatku vyberieme prvú položku zo slovníka, ktorá spĺňuje požiadavky podmienky.
$\langle\rangle$ *vylúčime*
$\langle u_1 \rangle$ *vylúčime* x_1 *a* x_4 *majú* $b_1 \neq b_4$
$\langle u_1 u_2 \rangle$ *je splnené len pre* x_3 *a* $b_3 = 0$, *teda vyberieme ako prvý term* $(\langle u_1 u_2 \rangle, 0)$ *do rozhodovacieho zoznamu*

a vymažeme x_3.
ďalšie kroky:
$\langle u_1 u_3 \rangle$ *je splnené pre* x_4 *a* x_6 *a* $b_4 = b_6 = 1$, *teda vyberieme* $(\langle u_1 u_3 \rangle, 1)$; *vymažeme* x_4 *a* x_6
$\langle u_1 \rangle$ *je splnené pre* x_1, $b_1 = 0$, *vyberieme* $(\langle u_1 \rangle, 0)$
$(\langle \overline{u_1} u_4 \rangle, 0)$ *vymaže* x_2
$(\langle\rangle, 1)$ *vymaže* x_5

Vytvorený rozhodovací zoznam je:

$$(\langle u_1 u_2 \rangle, 0), (\langle u_1 u_3 \rangle, 1), (\langle u_1 \rangle, 0), (\langle \overline{u_1} u_4 \rangle, 0), (\langle\rangle, 1)$$

Treba poznamenať, že rôzne usporiadania $M_{5,2}$ *dávajú rôzne odpovede.*

Zdá sa nám, že tu pôsobí nejaký záhadný faktor (v príklade), pretože nie je bezprostredne zrejmé, prečo hľadanie g a c je vždy úspešné. Aby sme dokázali, že algoritmus pracuje, je nutné ukázať, že vždy keď je daná trénujúca vzorka s pre cieľový koncept v $DL(K)$, potom vždy bude nejaká dvojica (g, c), ktorá má požadované vlastnosti.

O trénujúcej vzorke danej vyššie nebolo na začiatku známe, či je kompatibilná s konceptom v $DL(M_{5,2})$, napriek tomu úspešné dokončenie algoritmu ukazuje, že je v skutočnosti v tomto tvare.

Tvrdenie 1 *Predpokladajme, že* K *je hypotézový priestor obsahujúci identicky 1-kovú funkciu. Nech* t *je funkcia v* $DL(K)$ *a nech* S *je konečná množina príkladov. Potom existuje* $g \in K$ *a* $c \in \{0, 1\}$ *také, že:*

1. *množina* $S^g = \{x \in S | \quad g(x) = 1\}$ *je neprázdna;*
2. *pre všetky* $x \in S^g, \quad t(x) = c$.

Dôkaz: Je dané $t \in DL(K)$ také, že je reprezentácia t ako rozhodovacieho zoznamu:

$$t = (f_1, c_1), (f_2, c_2), \ldots, (f_r, c_r)$$

Ak $f_i(x) = 0$ pre všetky $x \in S$ a všetky $i \in \{1, 2, \ldots, r\}$, potom všetky príklady v S sú negatívne príklady pre t. V tomto prípade vezmeme g také, že je identicky 1-ková funkcia pre $c = 0$.

Na druhej strane, ak je nejaké i také, že množina $\{x_{i_1}, x_{i_2}, \ldots\}$, $x \in S$, pre ktoré $f_i(x_{i_j}) = 1$ nie je prázdna, potom nech q bude najmenším takým indexom, t.j. $q = min\{i_1, i_2, \ldots\}$.
Z definície rozhodovacieho zoznamu vyplýva, že $t(x) = c_q$ pre všetky x také, že $f_q(x) = 1$. V tomto prípade môžeme vybrať $g = f_q$ a $c = c_q$. □

Z tohto tvrdenia vyplýva, že k ľub. trénovacej vzorke pre funkciu v $DL(K)$ existuje vhodný výber dvojice (g, c) pre "prvý term" rozhodovacieho zoznamu. Aplikáciou tohto výsledku rekurzívne vidíme, že algoritmus popísaný vyššie bude vždy úspešný.

5.5 Ďalšie poznámky

Je dôležité zaručiť, že všetky pravdepodobnosti, ktoré sa vyskytujú v definícii tohto odseku, budú dobre definované. Nie sú žiadne problémy, ak príkladový priestor X je spočítateľný, ale pre reálny X musíme stanoviť nejaké merateľné teoretické obmedzenia na hypotézový priestor, ktorý uvažujeme. Pre ľubovoľné dve hypotézy $t, h \in H$ potrebujeme priradiť pravdepodobnosť chybovej množiny $\{x | \quad h(X) \neq t(x)\}$. Toto sa dá, ak chyby hypotézy sú merateľné funkcie.

Aby sme zaručili, že pre všetky m a pre všetky $t \in H$ množina

$$\{s \in S(m, t) | \quad H[s] \cap B_\epsilon \neq \emptyset\}$$

má dobre definovanú pravdepodobnosť (vzhľadom na μ^m), musia byť zavedené niektoré dodatočné obmedzenia. Postačuje mať H **univerzálne separovateľný**; čitateľa odkážeme na Pollarda (1984) a Blumera et al. (1989).

Poznamenajme, že najviac používané hypotézové priestory majú túto vlastnosť a určite sú prediskutované v týchto knihách.

Teria potenciálnej naučiteľnosti môže byť ľahko rozšírená na algoritmy, ktoré sú **"takmer konzistentné"**. Pre potenciálnu naučiteľnosť musí byť záruka konzistencie na trénujúcej vzorke dostatočnej dĺžky, čo implikuje dobrú aproximáciu.

Podmienka vyžadovaná pre rozšírenú definíciu je nasledujúca:
Pre ľubovoľnú pevnú konštantu $\alpha < 1$ existuje kladné celé číslo $m_0(\alpha, \delta, \epsilon)$ také, že ak hypotéza h nesúhlasí s najviac $\alpha \cdot \epsilon$ časťou trénujúcej vzorky dĺžky m_0, potom s pravdepodobnosťou aspo $1 - \delta$ má h skutočnú chybu menšiu ako ϵ.

Táto podmienka môže byť splnená pre ľubovoľný konečný hypotézový priestor H; dôkaz vyplýva celkom jednoducho použitím hraníc Chybinovej a Valianta (1979) na určité súčty binárnych čísel.

5.6 Cvičenia:

1. Dokážte, že priestor $H = \{r_\Theta | \quad \Theta \in R\}$ lúčov je potenciálne naučiteľný.
2. Ukážte, že pre hypotézový priestor $D_{n,k} \quad (n \geq k > 1)$ je postačujúce vziať hodnotu $m_0(\delta, \epsilon) = \left\lceil \frac{k}{\epsilon} \cdot \ln 2n + \frac{1}{\epsilon} \cdot \ln \frac{1}{\delta} \right\rceil$ v def. potenciálnej naučiteľnosti.
3. Dokážte, že pre všetky $n \geq k \geq 1$, $D_{n,k} \subset DL(M_{n,k})$. Odvoďte, že ľubovoľná bool. funkcia môže byť reprezentovaná rozhodovacím zoznamom.
4. Skonštruujte bool. funkciu 3 premenných, ktorá nie je $D_{3,2}$, ale je v $DL(M_{3,2})$.
5. Skonštruujte bool. funkciu 3 premenných, ktorá nie je v priestore $DL(M_{3,2})$.
6. Dokážte, že algoritmus pre DL je konzistentný.
7. Dokážte, že pre ľubovoľnú množinu K booleovských funkcií, $3^{|K|} \cdot |K|!$ je horná hranica na $|DL(K)|$.
8. Komplement bool. funkcie h je bool. funkcia $\overline{h}$ taká, že $\overline{h}(x) = 1 \Leftrightarrow h(x) = 0$. Dokážte, že pre ľub. množinu K bool. funkcií obsahujúcu identickú funkciu 1, $h \in DL(K) \Leftrightarrow \overline{h} \in DL(K)$.
   Toto znamená, že $DL(K)$ je uzavretá vzhľadom na komplement.

Kapitola 6

Efektívne učenie I

6.1 Pohľad na teóriu zložitosti

Predmet *Výpočtová zložitosť* študuje vzťahy medzi veľkosťou vstupu do algoritmu a časom, ktorý algoritmus spotrebuje, aby vypočítal výstup pre vstup tejto veľkosti teda zaoberá sa efektívnosťou (účinnosťou) algoritmov.

Veľkosť vstupu do algoritmu môže byť meraná rôznymi spôsobmi.

Ak sa zaoberame booleovskými funkciami - počet bitov, ktoré sú na vstupe, ale ak uvažujeme reálne čísla, tak môžu vzniknúť problémy ?

Čas behu algoritmu je závislý od toho, ako rýchlo môže byť výpočet vykonaný. Pretože toto budeme robiť nezávisle od zariadenia, budeme počítať počet potrebných operácií (obvykle pre najhorší prípad!). Budeme sa zaujímať len o závislosti, preto budeme používať nasledujúcu definíciu:

Nech A je algoritmus, ktorý akceptuje vstupy rôznej veľkosti s. Hovoríme, že čas behu A je $O(f(s))$, ak pre ľubovoľný vstup veľkosti s, počet operácií požadovaných na získanie výstupu algoritmu A je najviac $K * f(s)$, kde K je nejaká konštanta, $K > 0$.

6.1.1 Splniteľnosť booleovskej formuly (satisfiability)

Inštancia: Booleovská formula Φ o n premenných
Otázka: Existuje pozitívny príklad pre $\langle\Phi\rangle$?

NP-úplný, ak $\in$ NP a každý problém z NP je naň redukovateľný;

Budeme aplikovať idey teórie výpočtovej zložitosti:

Predpokladajme, že Π je problém, o ktorý sa zaujímame, Π_0 je problém o ktorom je známe, že je NP-úplný. Predpokladajme, že môžme demonštrovať, že ak existuje polynomiálny algoritmus pre Π, potom existuje aj pre Π_0. V tomto prípade náš problém sa nazýva NP-ťažký problém.
V prípade, že platí $P \neq NP$, potom dôkaz, že M je NP-ťažký znamená, že preň neexistuje polynomiálny algoritmus.

6.1.2 Čas behu učiacich algoritmov

Väčšina učiacich algoritmov prediskutovaných v predchádzajúcom texte sa zaoberá booleovskými konceptami. V týchto prípadoch príkladový priestor je $\{0,1\}^n$ pre nejaké pevné n a hypotézový priestor je množina funkcií definovaných na príkladovom priestore. Pre každý z týchto algoritmov parameter n je ľubovoľný v zmysle, že algoritmus je definovaný pre ľubovoľné n a navyše operuje v podstate tým istým spôsobom pre každú hodnotu n. Napríklad, štandartný učiaci algoritmus pre priestor M_n monočlenov je definovan'y celkom všeobecne, hoci potrebujeme špeciálny "stroj" na jeho implementáciu pre danú hodnotu n. Chceme kvantifikovať chovanie sa učiacich algoritmov vzhľadom na n, a je obvykle používať nasledujúce definície.

Definícia 6.1.1 *Hovoríme, že zjednotenie hypotézových priestorov $H = \bigcup H_n$ je odstupňované (graded) príkladmi veľkosti n, ak H_n označuje hypotézový priestor definovaný na príkladoch z X^n.*

Definícia 6.1.2 *Učiaci algoritmus pre $H = \bigcup H_n$ je funkcia L z množiny tréningových príkladov pre hypotézy v H do priestoru H taký, že ak $\overline{s}$ je tréningová vzorka pre $h \in H_n$, tak z toho vyplýva $L(\overline{s})) \in H_n$. Teda L podporuje stupňovanie (grading) priestoru H.*

Uvažujme učiaci algoritmus L pre bool. hypotézový priestor $H = \bigcup H_n$, odstupňovaný veľkosťou príkladov. Vstup do L je tréningová vzorka, ktorá pozostáva z m n-bitových vektorov spolu s m 1-bitovými označeniami. Celkový počet bitov na vstupe je $m(n+1)$ a bolo by možné použiť toto jediné číslo ako mieru veľkosti vstupu. Avšak je výhodné sledovať aj m aj n oddelene a použijeme označenie $R_L(m,n)$ **na označenie najhoršieho času behu L na tréningovej vzorke m n-bitových vektorov.**

Príklad 1: Nech L je učiaci algoritmus pre monočleny popísaný vyššie. Hypotézový priestor je zjednotenie $\bigcup M_n$. Hlavný krok algoritmu vyžaduje kontrolu každého bitu u každého pozitívneho príkladu a možno vymazanie niektorých literálov. V najhoršom prípade, každý príklad v tréningovej vzorke môže byť pozitívny príklad, a tak by sme mali ošetriť tento krok m-krát, každý krok obsahuje kontrolu n bitov. Iné časti výpočtu vyžadujú porovnateľne toľko operácií, takže môžme hovoriť, že čas behu $R_L(m,n)$ je v tomto prípade $O(m * n)$.

Príklad 2: Vyššie sme popísali učiaci algoritmus pre priestor $D_{n,k}$ disjunkcií malých mnohočlenov. Ako obvykle, považujeme k za pevné a n za premennú. Každý krok algoritmu obsahuje kontrolu, či jeden z m príkladov v tréningovej vzorke je pozitívny alebo negatívny a ak je negatívny, vyhodnotenie niektorých monočlenov v $M_{n,k}$. Na začiatku zoznam relevantných monočlenov má dĺžku okolo $(2n)^k$ a v každom stave môže byť niektorý z nich vymazaný. Pretože k je pevné, 2^k je konštanta, a teda čas behu je $O(m * n^k)$.

Oba vyššie prediskutované algoritmy sú bezpamäťové on-line algoritmy, a to znamená, že výpočet času behu je veľmi jednoduchý. V takom algoritme L vyžaduje najviac $S_L(n)$ operácií na spracovanie jediného n-bitového príkladu, potom jeho čas behu je $R_L(m,n) \leq mS_L(n)$. Pre algoritmy, ktoré nie sú tohto typu, výpočet času behu môže byť zložitejší.

Príklad 3: Analyzujme algoritmus na učenie v rozhodovacích zoznamoch $DL(K_n)$. Toto zrejme nie je bezpamäťový on-line algoritmus, pretože v každom kroku je nutné kontrolovať všetky zostavajúce príklady so zoznamom dvojíc (g,c), kde $g \in K_n$ a $c \in \{0,1\}$. Ak je na začiatku m príkladov v trénujúcej vzorke, bude tu $2|K_n|m$ kontrôl v 1. kroku v najhoršom prípade. Aspoň 1 príklad bude vymazaný, takže ďalší krok vyžaduje $2|K_n|(m-1)$ kontrôl. Opakovaním tých istých argumentov dostávame celkový počet kontrôl najviac

$$2(m + m - 1 + \ldots + 3 + 2 + 1)|K_n| = O(m^2|K_n|).$$

Ak K_n je $M_{n,k}$, pre ktorého kardinalitu máme ohraničenie $(2n)^k$, čas behu je $O(m^2 n^k)$.

6.2 Prístup k efektívnosti PAC učenia

Všeobecný prístup k dokazovaniu vlastnosti PAC pre nejaký učiaci algoritmus bol uvedený vyššie. V kontexte odstupňovaného hypotézového priestoru $H = \bigcup H_n$ bool. funkcií môžme popĭsať procedúru schématicky nasledovne:

$$H_n \text{ konečný} \qquad \Rightarrow \qquad H_n \text{ potencionálne naučiteľný}$$

H_n potencionálne naučiteľný $\wedge$ L je konzistentný pre $H_n \Rightarrow L$ PAC učí H_n

S ohľadom na účinnosť vzniká prirodzená otázka: pre dané požadované úrovne presnosti a dôvery, ktorá (aká) podmienka zaručí, že čas behu, v ktorom L PAC učí H_n, je polynomiálny v n?

V tomto bode nám pomôže zavedenie ďalšej terminológie na popis podobných pojmov. Predpokladajme, že sú dané reálne čísla $0 < \delta, \epsilon < 1$ a nech L je učiaci algoritmus pre konceptový priestor C a hypotézový priestor H. (Predpoklad $C = H$ tu nie je požadovaný.) Hovoríme, že zložitosť vzorky pre L na podmnožine $T \subseteq C$ je najmenšia hodnota $m_L(T,\delta,\epsilon)$ taká, že pre všetky cieľové koncepty $t \in T$ a všetky pravdepodobnostné rozloženia μ

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

pre všetky $m \geq m_L(T,\delta,\epsilon)$; inak povedané vzorka dĺžky $m_L(T,\delta,\epsilon)$ postačuje k záruke, že výstupná hypotéza je PAC pre dané ϵ, δ. V praxi sa skôr zaoberáme obvyklou dolnou hranicou $m_0 \geq m_L$, než

m_L samotným; teda $m_0(T, \delta, \epsilon)$ bude označovať ľub. hodnotu postačujúcu k záruke, že PAC (ako bolo uvedené vyššie) platí pre všetky $m \geq m_0$.

Úplné zovšeobecnenie definície bude použiteľné na prípady v ďalších kapitolách, ale vo väčšine aplikácií budeme uvažovať $T' = C = H$. Napríklad, použitím tejto terminológie veta ukazuje, že pre konzistentný učiaci algoritmus na konečnom priestore H, dolná hranica pre zložitosť vzorky $m_L(H, \delta, \epsilon)$ je $m_0 H, \delta, \epsilon) = \lceil \frac{1}{\epsilon} ln \frac{|H|}{\delta} \rceil$.

Zložitosť vzorky poskytuje prepojenie medzi časom behu $R_L(m, n)$ učiaceho algoritmu (tj. počet operácií potrebných produkovať svoj výstup na vzorke dĺžky m, ak príklady sú dĺžky n) a jeho časom behu ako PAC učiaceho algoritmu (tj. počet operácií potrebných na vytvorenie výstupu, ktorý je PAC s danými parametrami). Pretože vzorka dĺžky $m_0(H_n, \delta, \epsilon)$ postačuje pre vlastnosť pac, počet vyžadovaných operácií je najviac $R_L(m_0(H_n, \delta, \epsilon), n)$. V prípade konzistentného algoritmu toto poskytuje odpoveď na otázku položenú vyššie.

Veta 4 *Predpokladajme, že L je konzistentý učiaci algoritmus pre hypotézový priestor $H = \bigcup H_n$. Ak*

- *$R_L(m, n)$ je polynóm v m a n,*
- *$ln|H_n|$ je polynóm v n,*

potom pre dané hodnoty parametrov pre presnosť ϵ a pre dôveru δ čas behu, počas ktorého L bude produkovať pravdepodobnostne aprixomavanú správnu hypotézu je polynomiálny v n.

Dôkaz: Pretože L je konzistentný, horná hranica pre zložitosť vzorky algoritmu L na H_n je

$$\lceil \frac{1}{\epsilon} ln \frac{|H_n|}{\delta} \rceil$$

Teda je nutné potvrdiť, že keď podmienky platia, tak výraz $R_L(\lceil \frac{1}{\epsilon} \ln \frac{|H_n|}{\delta} \rceil, n)$ sa dá upraviť na polynóm vzhľadom na n. □

Tento výsledok vrhá svetlo na účinnosť učiacich algoritmov prediskutovaných skôr. Napríklad, priestor monočlenov má mohutnosť $|M_n| = 3^n$ a tak $\ln |M_n| = n \ln 3$. Štandartný učiaci algoritmus pre monočleny je konzistentný a má čas behu $O(mn)$. Z toho vyplýva, že algoritmus PAC učí M_n v polynomiálnom čase vzhľadom na n - špecificky, čas behu je $O(mn^k)$ a teda vieme, že $|D_{n,k}|$ je najviac $2^{(2n)^k}$. Z toho vyplýva, že $\ln |D_{n,k}|$ je ohraničený konštantným násobkom n^k a teda implikuje, že algoritmus PAC učí $D_{n,k}$ v čase $O(n^{2k})$.

Predchádzajúca veta nám však neumožňuje načrtnúť žiadny záver o účinnosti všeobecnejších učiacich algoritmov. Napríklad, algoritmus pre učenie očíslovaním, vyžaduje zakaždým m označených príkladov v tréningovej vzorke na kontrolu (porovnanie) s hypotézami v H_n. Teda je to konzistentný algoritmus, ktorého čas behu $R_L(m, n)$ je $O(m|H_n|)$. V tomto prípade prvá podmienka predchádzajúcej vety vyžaduje, že $|H_n|$ samotné (skôr než jeho logaritmus) rastie polynomiálne. Toto je veľmi obmedzujúca podmienka, pretože aj celkom ohraničené hypotézové priestory ako M_n a $D_{n,k}$ majú kardinalitu, ktorá rastie exponenciálne.

V dôsledku toho, hoci algoritmus očíslovaním môže byť použitý pre ľubovoľný konečný priestor, existuje mnoho prípadov kde minulá veta nemôže byť aplikovaná k tomu, aby ukázala, že alg. bude produkovať pravdepodobnostne aproximované korektné hypotézy v polynomiálnom čase.

Aplikujme vetu na učiaci algoritmus pre $DL(K_n)$, pre ktorý čas behu je $O(m^2|K_n|)$. Prvá podmienka vyžaduje, že mohutnosť základného priestoru K_n je polynomiálna funkcia v n. V skutočnosti to je všetko, čo je potrebné, pretože druhá podmienka vyplýva z toho automaticky; tj. $\ln |DL(K_n)|$ je polynóm v $|K_n|$. Aby sme to overili, odhadneme, že $\ln(N!) \leq N \ln N$, a tak máme

$$\ln |DL(K_n)| \leq |K_n|(\ln |K_n| + \ln 3)$$

a zrejme toto je ohraničené polynómom v n vždy, keď je aj $|K_n|$ - napríklad $K_n = M_{n,k}$.

6.3 Problém konzistencie tréningovej vzorky

Budeme študovať odraz teórie o NP-ťažkých problémoch v tejto oblasti. Nech $H = \bigcup H_n$ je hypotézový priestor bool. funkcií odstupňovaný príkladmi veľkosti n. Problém konzistencie pre H môže byť stanovený nasledovne:

H - KONZISTENCIA:

- Inštancia: Tréningová vzorka s vyjadrená n-bitovými označenými vektormi.

- Otázka: Existuje hypotéza v H_n konzistentná s $\overline{s}$?

Ukážeme, že v niektorých netriviálnych prípadoch je tento problém NP-ťažký. Na to, aby sme vyjadrili praktický dosah tohto výsledku, potrebujeme urobiť niekoľko všeobecných komentárov.

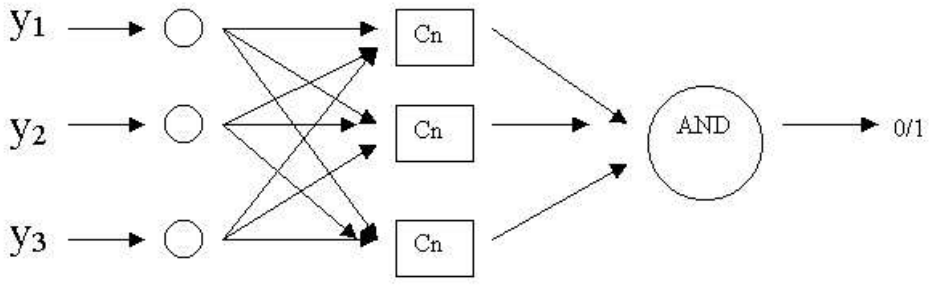
Prvý komentár, ak uvažujeme len tie inštancie problému, v ktorých dĺžka vzorky $\overline{s}$ je ohraničená nejakým fixným polynómom v n, potom máme ohraničený tvar problému konzistencie. Sú také ohraničené tvary, ktoré budeme sledovať v tejto prednáške a uvidíme, že niektoré z týchto problémov sú NP-ťažké. Upozorňujeme, že ak ohraničený tvar H-kozistencie je NP-ťažký, potom aj H-kozistencia samotná je NP-ťažká.

Ďalší komentár, v praxi si skôr prajeme nájsť konzistentnú hypotézu, než len vedieť o tom, či existuje. Inak povedané, máme riešiť problém "hľadania" skôr než problém "existencie". Ale tieto problémy sú v priamom vzťahu. Predpokladajme, ako vyššie, že uvažujeme len tie $\overline{s}$ s dĺžkou ohraničenou nejakým polynómom. Potom, ak môžme nájsť konzistentnú hypotézu v polynomiálnom čase v n, tak môžeme odpovedať na existenčnú otázku pomocou nasledujúcej procedúry:

Spustiť hľadací algoritmus na čas (polynomiálny v n), v ktorom je zaručené nájdenie hypotézy, ak existuje. Potom skontrolovať výstupnú hypotézu explicitne s príkladmi zo vzorky $\overline{s}$, čo nám povie, či je konzistentná alebo nie. Táto kontrola môže byť tiež urobená v polynomiálnom čase v n. Teda, ak ukážeme, že ohraničený tvar existenčného problému je NP-ťažký, to znamená, že neex. polynomiálny algoritmus pre odpovedajúci vyhľadávací problém (pokiaľ neplatí P=NP).

6.4 Výsledok o zložitosti

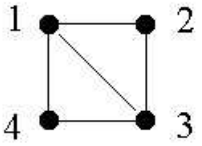
Pitt a Valiant (1988) boli prví, ktorí ukázali príklad hypotézového priestoru H, pre ktorý ohraničený tvar problému konzistencie je NP-ťažký. Nasleduje o niečo zjednodušený popis ich metódy. Nech C_n je priestor klauzúl - množina bool. fukcií n premenných, ktoré môžu byť reprezentované v tvare klauzúl; tj. formuly tvaru $u_2 \vee \overline{u_3} \vee u_6$ čo sú disjunkcie literálov. Nech C_n^k je priestor bool. funkcií, ktoré môžu byť reprezentované ako konjukcie k klauzúl. Môžme si predstaviť C_n ako hypotézový priestor stroja duálneho k stroju pre monočleny a C_n^k ako hypotézový priestor stroja vytvoreného spojením k $C_n - strojov$ paralelne a prechodom ich výstupov cez AND jednotku s n vstupmi.



Ukážeme, že pre pevné $k \geq 3$, problém konzistencie pre $C^k = \bigcup C_n^k$ je NP-ťažký. Teda nie je pravda, že ex. polynomiálny učiaci algoritmus pre C_n^k, ktorý produkuje konzistentnú hypotézu. Dôkaz spočíva v prevedení problému farbenia grafu na tento problém s n vrcholmi pomocou k farieb, čo je NP-úplný problém pre $k \geq 3$ (Gray, Johnson 1979).

Problém farbenia: $G = (V, E), k$-farbenie je funkcia $\chi : V \to \{1, 2, \ldots, k\}$ s vlast. $\langle v_i, v_j \rangle \in E$ potom $\chi(v_i) \neq \chi(v_j)$. Predp., že máme $G = (V, E), V = \{1, 2, \ldots k\}$. Skonštruujeme tréningovú vzorku $s(G)$ nasledovne: Pre každý vrchol $i \in V$ určíme záporný príklad vektor v_i, ktorý má 1 v pozícii i-tej súradnice a 0 inde. Pre každú hranu $\langle i, j \rangle \in E$ vezmeme ako pozitívny príklad vektor $v_i + v_j$.

Príklad:



v_i	
1000	0
0100	0
0010	0
0001	0
1100	1
1010	1
1001	1
0110	1
0011	1

Tvrdenie: Existuje funkcia $h \in C_n^k$, ktorá je konzistentná so vzorkou $s(G) \iff$ graf G je k-zafarbiteľný.
Dôkaz:

$\Rightarrow$

Predpokladajme, že $h \in C_n^k$ a je konzistentná s tréningovou vzorkou. Podľa definície h je konjunkcia $h = h_1 \wedge h_2 \wedge \ldots h_k$ klauzúl. Pre každý vrchol $i \in V$, $h(v_i) = 0$ a teda musí ex. aspoň 1 klauzula h_f, pre ktorú $h_f(v_i) = 0$. Funkcia $\chi : V \to \{1, 2, \ldots k\}$ taká, že:

$$\chi(i) = min\{f \quad | \quad h_f(v_i) = 0\}$$

Zostáva ukázať, že χ je farbenie grafu G; inak povedané, ak i a j sú dva vrcholy, pre ktoré $\chi(i) = \chi(j)$, potom $\langle i, j\rangle \notin E$. Predp., že $\chi(i) = \chi(j) = f$ a teda $h_f(v_i) = h_f(v_j) = 0$. Pretože h_f je klauzula, každý literál, ktorý sa v nej vyskytuje musí byť 0 na v_i a na v_j. Teraz v_i má 1 len v i-tej pozícii a tak $h_f(v_i) = 0$ implikuje, že len jeden negovaný literál, ktorý sa môže vyskytnúť v h_f je $\overline{u_i}$. Pretože to isté platí pre $\overline{u_j}$, dostávame, že h_f obsahuje len niektoré literály u_z, pre ktoré $z \neq i, j$. Teda $h_f(v_i + v_j) = 0$ a $h(v_i + v_j) = 0$. Ak by $\langle i, j\rangle$ bola hranou v G, potom by platilo $h(v_i + v_j) = 1$, pretože sme predpokladali, že h je konzistentná s $s(G)$. Teda $\langle i, j\rangle$ nie je hranou v G a χ je farbenie.

$\Leftarrow$

Predpokladajme, že je dané farbenie $\chi : V \to \{1, 2, \ldots, k\}$. Pre $1 \leq f \leq k$ definujme h_f ako klauzulu

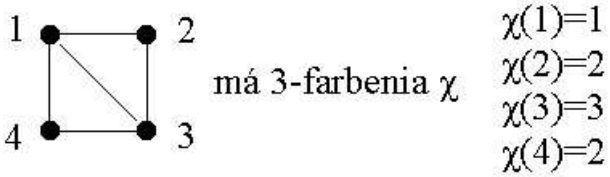
$$\langle \vee u_{i_{\chi(i) \neq f}} \rangle$$

a definujme $h = h_1 \wedge h_2 \wedge \ldots \wedge h_k$. Tvrdíme, že h je konzistentná s $s(G)$.

Najprv, predpokladajme, že pre vrchol i $\chi(i) = g$. Klauzula h_g je definovaná tak, že obsahuje len tie (nie negované) literály, ktoré zodpovedajú vrcholom nezafarbeným farbou g, a teda u_i sa nenachádza v h_g. Teda $h_g(v_i) = 0$ a $h(v_i) = 0$.

Ďalej, nech ij je hrana v G. Pre každú farbu f aspoň jedno v_i alebo v_j nemá farbu f; označme vhodný výber $i(f)$. Potom h_f obsahuje literál $u_{i(f)}$, ktorý je 1 na $v_i + v_j$. Teda klauzula h_f je 1 na $v_i + v_j$ a $h(v_i + v_j) = 1$, ako sme požadovali. □

Príklad:



Teda je funkcia h v C_4^3 je konzistentná s odpovedajúcou tréningovou vzorkou

$$h = h_1 \wedge h_2 \wedge h_3 = \langle (u_2 \vee u_3 \vee u_4) \wedge (u_1 \vee u_3) \wedge (u_1 \vee u_2 \vee u_4) \rangle$$

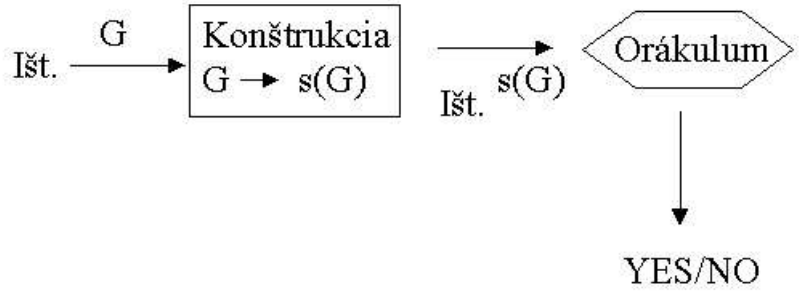
Tento graf nemôže byť zafarbený 2 farbami a teda môžme povedať, že neex. C_4^2, ktorá je konzistentná s trén. vzorkou.

Predchádzajúce tvrdenie vyjadruje súvislosť medzi k-farbením grafov a problémom C^k -konzistencie. Ak zvolíme kladné celé číslo k pevne, potom môžeme tieto 2 problémy vyjadriť formálnejšie:

k-FARBENIE GRAFU
Je daný graf G s n vrcholmi. Existuje k - farbenie grafu G?

C^k - KONZISTENCIA
Je daná tréningová vzorka $\overline{s}$ s označenými n-bitovými vektormi. Existuje funkcia v C_n^k, ktorá je konzistentná s $\overline{s}$?

Dôkaz, že C^k - konzistencia je NP-ťažký problém je pozorovateľný podľa obr.



Najprv k danej inštancií G skonštruujeme (v polynomiálnom čase) inštanciu $s(G)$ pre C^k konzistenciu. Poznamenajme, že počet hrán v grafe s n vrcholmi je najviac $n(n-1)/2$, a tak počet príkladov v $s(G)$ je najviac $n + n(n-1)/2$, čo je $O(n^2)$. Teraz predpokladajme, že existuje algoritmus (môžeme si myslieť, že je to orákulum), ktorý môže dávať odpovede na otázky C^k konzistencie. Ak orákulum operuje v polynomiálnom čase vzhľadom na n, potom by mohlo odpovedať na pôvodnú otázku tiež v polynomiálnom čase. Avšak pôvodný problém je NP-ťažký. Teda už ohraničený tvar C^k - konzistencie, v ktorom je $m \sim O(n^2)$ je NP-ťažký. Predchádzajúci dôkaz o tom, že C^k problém konzistencie je NP-ťažký, platí len pre $k \geq 3$, pretože problém k-farbenia grafu je NP-ťažký len pre $k \geq 3$. Keď $k = 1$, máme $C_1^n = C_n$ a ex. polynomiálny algoritmus pre C_n duálny k známemu algoritmu pre monočleny. Keď $k = 2$, môžme ukázať inými metódami, že problém konzistencie je NP-ťažký.

6.5 Ďalšie poznámky

Prediskutovali sme učiace algoritmy pre odstupňované priestory $H = \bigcup H_n$, kde hypotézový priestor a konceptový priestor koincidujú. Je ľahké definovať učiaci algoritmus pre odstupňovaný konceptový priestor $\bigcup C_n$ pomocou (možno rôznych) stupňového hypotézového priestoru $H = \bigcup H_n$; taký algoritmus vezme vstupné vzorky pre hypotézy v $C = \bigcup C_n$ a výstupné hypotézy v $H = \bigcup H_n$ s vlastnosťou, že ak $\overline{s}$ je tréningová vzorka pre hypotézu v C_n, potom $L(\overline{s}) \in H_n$. Hausler, Littlestone a Warmuth (1988) zaviedli pojem účinná predikcia postupne študovaný Pittom a Warmuthom (1988,1990) a Hausslerom (1988). Zhruba povedané, odstupňovaný konceptový priestor $C = \bigcup C_n$ je účinne predikovateľný, ak existuje nejaký stupňový hypotézový priestor $H = \bigcup H_n$ taký, že existuje PAC učiaci algoritmus pre $(\bigcup C_n, \bigcup H_n)$ v polynomiálnom čase n. (Aby sme boli presnejší, od H požadujeme málo - "polynomiálne vyhodnotiteľnú" reprezentáciu. Inak, povedané, ak učiaci algoritmus je prezentovaný tréningovou vzorkou pre cieľ v C_n, potom výstupom L je reprezentácia ω hypotézy $h_\omega \in H_n$ taká, že možno určiť v polynomiálnom čase či je daný príklad pozitívny pre h_ω. (Nebudeme o tom ďalej hovoriť, budeme v ďalšom predpokladať, že všetky reprezentácie majú túto vlastnosť.

6.6 Úlohy:

1. Prečo je obyčajne vhodné uvažovať veľkosť vstupu v tvare $\lg n$ namiesto n, keď uvažujeme o otázkach efektívnosti?

2. Nasledujúci algoritmus je rýchlym algoritmom pre výpočet m-tej mocniny daného čísla u. (výstupom je finálna hodnota uložená v *bot*.)

```
bot:=1; top:=u; q:=m;
while q>0 do
begin
   if q mod 2 =1 then bot:=top*bot;
   top:=sqr(top);
   q:=q div 2;
end;
```

   Ukážte, že efektívnosť algoritmu je $O(s)$, kde s je miera veľkosti m ako bolo povedané v predchádzajúcom príklade.

3. Navrhnite algoritmus, ktorý rozhodne, či daný n-bitový reťazec je palindrom a odhadnite jeho efektívnosť vzhľadom na veľkosť vstupu n.

4. Nech $G = (V, E)$ je graf s množinou vrcholov $V = \{1, 2, 3, 4, 5\}$ a množinou hrán $E = \{12, 13, 15, 23, 25, 34, 35\}$. Vytvorte odpovedajúcu tréningovú vzorku $s(G)$ podľa postupu v dôkaze. Nájdite najmenšiu možnú hodnotu k, pre ktorú je funkcia $h \in C_5^k$ konzistentná s $s(G)$ a danú formulu vyjadrite explicitne.

5. Nech $C_n = C_n^1$, čo je priestor booleovských funkcií na $\{0, 1\}^n$, ktorý môže byť reprezentovaný jednou klauzulou. Sformulujte konzistentný učiaci algoritmus pre C_n, ktorý je "duálny" k štandardnému algoritmu pre monočleny a vyhodnodtťe tvrdenie, že jeho čas behu je polynomiálny v m a n.

6. Nasledujúci problém je známy ako NP-úplný (Lovász, 1973).

   SET SPLITTING
   Inštancia: Dvojica (U, S), kde U je konečná množina a S je zbierka množín so zjednotením U.
   Otázka: Existujú $U_1, U_2 \subset$ také, že $U = U_1U_2$ a také, že žiadna množina v S neleží celá v U_1 alebo U_2?

   Redukujte SET SPLITTING na C^2-CONSISTENCY a uvažujte o tom, či problem pre C^2 je NP-ťažký (Pitt a Valiant, 1988).

Kapitola 7

Efektívne učenie II

7.1 Účinnosť versus dôveryhodnosť a presnosť

Diskusia v predchádzajúcej kapitole sa sústredila na chovanie sa učiaceho algoritmu v čase behu, ktorý závisel (bol funkciou) len od veľkosti príkladov n. Zrejme sú aj **iné faktory, ktoré určujú čas** behu učiaceho algoritmu, a mohli by sme zaviesť pojem efektívnosti vzhľadom na **úroveň dôveryhodnosti a úroveň presnosti**. Potom prediskutujeme efektívnosť vzhľadom na veľkosť reprezentácie cieľového konceptu. Tieto úvahy sú relevantné pre ľubovoľný hypotézový priestor a môžu byť kombinované s ideami nasledujúcej kapitoly, aby bola zavedená celkom všeobecná definícia pojmu **efektívny PAC učiaci algoritmus**.

V predchádzajúcej kapitole sme zaviedli pevné, ale ľubovoľné parametre, a síce parameter dôveryhodnosti δ a parameter presnosti ϵ. Je zrejmé, že zníženie aspoň jednej hodnoty z týchto veličín urobí učenie ťažším, a preto čas behu efektívneho PAC učiaceho algoritmu by mohol byť ohraničovaný nejakým vhodným spôsobom pomocou rastúcich δ^{-1} a ϵ^{-1}. Mohli by sme sa jednoducho pýtať, či čas behu rastie polynomiálne vzhľadom na δ^{-1} a ϵ^{-1}, ale táto závislosť na δ^{-1} nie je celkom vhodná z nasledujúcich dôvodov. Ak dĺžka tréningovej vzorky, ktorá vstupuje do efektívneho učiaceho algoritmu je zdvojnásobená, možeme očakávať pravdepodobnosť, že výstupná hypotéza je zlá približne kvadratickú. Inak povedané, **vzťah medzi zložitosťou vzorky a δ^{-1} je logaritmický**. Motivovaní týmto, mohli by sme povedať, že učiaci algoritmus L je **efektívny vzhľadom na dôveryhodnosť**, ak jeho čas behu je polynomiálny v m a zložitosť vzorky $m_L(H, \delta, \epsilon)$ závisí polynomiálne od veličiny $\ln(\delta^{-1})$, čo budeme označovať δ^*. V prípade parametra presnosti budeme hovoriť, že L **je efektívny vzhľadom na presnosť**, ak jeho čas behu je polynomiálny v m a zložitosť vzorky závisí polynomiálne od ϵ^{-1}. Ak obe tieto podmienky platia, potom čas behu potrebný na vytvorenie PAC výstupnej hypotézy je polynomiálny v δ^* a ϵ^{-1}.

Napríklad, ak H je ľubovoľný konečný hypotézový priestor a L je konzistentný učiaci algoritmus pre H, potom teória navrhnutá predtým implikuje, že dolná hranica pre zložitosť vzorky je $m_0(H, \delta, \epsilon) = \lceil \epsilon^{-1} \cdot \ln(|H|/\delta) \rceil$.

V tomto prípade m_0 je zrejme ohraničené polymomiálnou funkciou vzhľadom na δ^* a ϵ^{-1}. Ak čas behu L je polynóm v m, potom L je PAC učiaci algoritmus pre H, ktorý beží v polynomiálnom čase vzhľadom na δ^* a ϵ^{-1}. Ten istý argument platí v odstupňovanom prípade. Ak $H = \bigcup H_n$ je hypotézový priestor booleovských funkcií odstupňovaný veľkosťou príkladov, potom dolná hranica pre zložitosť vzorky je

$$m_0(H_n, \delta, \epsilon) = \left\lceil \frac{1}{\epsilon} \cdot \ln\left(\frac{|H_n|}{\delta}\right) \right\rceil.$$

V tomto prípade, ak čas behu $R_L(m, n)$ je polynóm v m a n a ak $\ln |H_n|$ je polynóm v n, potom L PAC učí H_n v polynomiálnom čase behu nielen v n, ale tiež v δ^* a ϵ^{-1}.

7.2 PAC učenie a problém konzistencie

Analýza urobená na konci predchádzajúcej sekcie je motivovaná doteraz známymi vzťahmi medzi konzistenciou a PAC učením. Obrátením našej pozornosti na neodstupňovaný prípad výsledok je jednoduchý, a síce ak existuje konzistentný učiaci algoritmus L pre konečný hypotézový priestor H, ktorý beží v

polynomiálnom čase vzhľadom na dĺžku vzorky m, potom L PAC učí H v polynomiálnom čase vzhľadom na δ^* a ϵ^{-1}. Zhruba povedané, môžeme hovoriť, že efektívny "konzistentný-hľadač-hypotéz" je efektívny "PAC-learner". V tejto sekcii uvedieme, k čomu vedie opačná úvaha.

Toto by znamenalo, že efektívne PAC učenie implikuje efektívne hľadanie konzistentných hypotéz za predpokladu, že sme pripravení akceptovať **náhodný algoritmus**. Úplné vysvetlenie tohto pojmu možno nájsť v knihe Cormena, Leiersona a Rivesta (1990), ale pre naše účely postačí vysvetlenie uvedené v nasledujúcich odsekoch.

Predpokladajme, že máme daný nejaký **generátor náhodných čísel**, ktorý pre dané ľubovoľné celé číslo $I \geq 2$ produkuje náhodné čísla i v intervale $1 \leq i \leq I$, pričom každá hodnota je rovnako pravdepodobná.

Náhodný algoritmus A má povolené brať tieto čísla ako časť svojho vstupu. **Výpočet algoritmu A je riadený jeho vstupom** tak, že závisí od partikulárnej postupnosti produkovanej generátorom náhodných čísel. Z toho vyplýva, že môžeme hovoriť o pravdepodobnosti, že A má daný výsledok, čím je mienená relatívna frekvencia postupností, ktoré produkuje tento výsledok vzhľadom na celkový počet možných postupností.

Hovoríme, že náhodný algoritmus A "rieši" problém vyhľadávania Π, ak sa chová nasledujúcim spôsobom: Algoritmus vždy zastaví a produkuje výstup. Ak A padol pri hľadaní riešenia pre Π, dá jednoducho výstup *nie*. Ale s pravdepodobnosťou aspoň $\frac{1}{2}$ (v zmysle vyjadrenom vyššie), A je úspešný pri hľadaní riešenia pre π a výstupom je jeho riešenie.

Praktická použiteľnosť náhodného algoritmu vyplýva z faktu, že opakovaním algoritmu niekoľkokrát veľmi rýchlo rastie pravdepodobnosť úspechu. Ak algoritmus padne pri prvom pokuse, čo sa stane s pravdepodobnosťou najviac $\frac{1}{2}$, potom jednoducho skúšame ďalej. Pravdepodobnosť, že padne dvakrát, je najviac $\frac{1}{4}$, že padne k-krát je najviac $(\frac{1}{2})^k \to 0$. Teda v praxi náhodný algoritmus je takmer tak dobrý ako obyčajný - samozrejme za predpokladu, že má polynomiálny čas behu. Máme nasledujúcu vetu Pitta a Valianta (1988) (tiež Matarjan (1989) a Haussler et. al. (1988)).

Veta 5 *Nech H je hypotézový priestor a predpokladajme, že existuje PAC učiaci algoritmus pre H s časom behu polynomiálnym v ϵ^{-1}. Potom existuje náhodný algoritmus, ktorý rieši problém hľadania hypotézy v H konzistentnej s danou tréningovou vzorkou a ktorý má čas behu polynomiálny v m (dĺžka tréningovej vzorky).*

Dôkaz. Predpokladajme, že $\overline{s}$ je tréningová vzorka pre cieľovú hypotézu $t \in H$ a $\overline{s}$ obsahuje m rôzne označených príkladov. Ukážeme, že je možné nájsť hypotézu konzistentnú s $\overline{s}$ spustením daného PAC algoritmu L na odpovedajúcej tréningovej vzorke. Definujme pravdepodobnostné rozdelenie μ na príkladovom priestore X takto:

$$\mu(x) = \begin{cases} \frac{1}{m}, & \text{ak } x \text{ sa vyskytuje v } \overline{s} \\ 0, & \text{inak.} \end{cases}$$

Môžeme použiť generátor náhodných čísel s výstupnými hodnotami $\in \langle 1, m \rangle$ na výber príkladov z X podľa tohoto rozdelenia: pridelíme každé náhodné číslo ako návestie $1..m$ rovnako pravdepodobného príkladu. Teda výber tréningovej vzorky dĺžky m pre t na základe rozdelenia μ môže byť simulovaný generovaním postupnosti m náhodných čísel v danom intervale.

Nech L je PAC učiaci algoritmus, ako bolo uvedené vyššie. Potom ak máme dané 4 veličiny δ, ϵ, μ, t, tak môžeme nájsť celé číslo $m_0(\delta, \epsilon)$

$$\overset{\forall}{\epsilon} > 0 \quad \overset{\forall}{\delta} > 0 \quad \overset{\exists}{m_0}(\delta, \epsilon) \quad \overset{\forall}{\mu} \quad \overset{\forall}{t} \quad \mu^m\{s \in S(m,t) : \quad h(s,t) < \epsilon\} > 1 - \delta$$

Predpokladajme, že špecifikujeme dôveryhodnosť $\delta = \frac{1}{2}$ a presnosť $\epsilon = \frac{1}{m^*}$.

Ak spustíme učiaci algoritmus L na tréningovej vzorke dĺžky $m_0(\frac{1}{2}, \frac{1}{m^*})$, získanej náhodne podľa rozdelenia μ, vlastnosť PAC algoritmu zaručí, že pravdepodobnosť, že chyba výstupu je menšia než $\frac{1}{m^*}$, je väčšia než $1 - \frac{1}{2} = \frac{1}{2}$. Pretože nie sú žiadne príklady s pravdepodobnosťou striktne medzi 0 a $\frac{1}{m^*}$, z toho vyplýva, že pravdepodobnosť, že výstup súhlasí presne s tréningovou vzorkou je väčší než $\frac{1}{2}$.
□

Procedúra uvedená vo vyššom odseku je základom pre náhodný algoritmus L^* pre hľadanie hypotézy, ktorá je konzistentá s danou tréningovou vzorkou $\overline{s^*}$.

Zhrnutie v krokoch pre L^*:

- Vyhodnotiť $m_0 = m_0(\frac{1}{2}, \frac{1}{m^*})$.

- Použitím gnč na skonštrukciu tréningovej vzorky $\overline{s}$ dĺžky m_0 podľa pravdepodobnostného rozdelenia μ.
- Spustiť daný PAC učiaci algoritmus L na $\overline{s}$.
- Skontrolovať výslednú hypotézu $L(\overline{s})$, či je konyistentná s $\overline{s^*}$.
- Ak hypotéza nie je konzistentná s $\overline{s^*}$, výstup "nie". Ak hypotéza je konzistentá s $\overline{s^*}$, výstupom je táto hypotéza.

Ako sme uviedli, PAC vlastnosť algoritmu L zaručí, že L^* je úspešný s pravdepodobnosťou viac ako $\frac{1}{2}$. Nakoniec je zrejmé, že ak čas behu algoritmu L je polynomiálny v ϵ^{-1}, potom čas behu L^* je polynomiálny v $m^* = \epsilon^{-1}$.
□

Veta 5 nám umožňuje rozšíriť zložitosť výsledkov pre problém konzistencie, ako bolo dokázané vyššie, na PAC učenie. Pripomeňme, že oba problémy - problém rozhodnutia o konzistencii a problém hľadania konzistentnej hypotézy sú NP-ťažké v niektorých prípadoch, takých ako hypotézový priestor $C^k = \bigcup C_n^k$. Veta hovorí, že ak by sme mohli PAC naučiť C_n^k v polynomiálnom čase vzhľadom na ϵ^{-1} a n, potom by sme mohli nájsť konzistentnú hypotézu použitím náhodného algoritmu s časom behu polynomiálnym v m a n. V jazyku teórie zložitosti by to mohlo znamenať, že uvedený problém je v triede RP, čo je trieda problémov, ktoré môžu byť riešené v "pravdepodobne polynomiálnom čase". Teraz sa predpokladá, že RP neobsahuje žiadny NP-ťažký problém - teda, že RP≠NP, čo niektorí považujú za podobne ťažké ako NP=P. Takže keď toto akceptujeme, z toho vyplýva, že neexistuje žiadny PAC polynomiálny algoritmus pre odstupňovaný priestor C^k, ak $k \geq 2$.

Vyššie uvedená diskusia ukazuje, že pre $k \geq 2$, C^k nie je efektívne PAC naučiteľný vzhľadom na veľkosť príkladu. Avšak pre ľub. n, C_n^k je obsiahnuté v $D_{n,k}$, priestore disjunkcií jednočlenov s najviac k literálmi. Valiantov učiaci algoritmus pre $D_k = \bigcup D_{n,k}$ popísaný vyššie je konzistentný algoritmus s časom behu $R_L(m,n) = O(m \cdot n^k)$, polynóm v n aj m. Preto pre ľub. tréningovú vzorku $\overline{s}$ pre hypotézu v C_n^k tento algoritmus bude produkovať v polynomiálnom čase hypotézu v $D_{n,k}$ konzistentnú s $\overline{s}$. V ďalších poznámkach predchádzajúcej kapitoly sme definovali, čo znamená učiaci algoritmus L pre odstupňovaný priestor $\bigcup C_n$ iným odstupňovaným priestorom $\bigcup H_n$; k danej tréningovej vzorke $\overline{s}$ pre hypotézu z C_n L vráti hypotézu $L(s) \in H_n$. Použitím tejto terminológie, štandardný učiaci algoritmus pre D_k je PAC učiaci algoritmus pre $(C^k, D_{n,k})$, efektívny vzhľadom na veľkosť príkladov. Teda v protiklade k negatívnemu výsledku vyššie, C^k je efektívne naučiteľný pomocou "**väčšieho priestoru**". Nie je tu žiadny spor. Zhruba povedané, je ťažké nájsť formulu pre konzistentnú hypotézu v C_n^k, pretože tento priestor je príliš "ohraničený". Daná je včšia flexibilita v práci v "bohatšom" priestore $D_{n,k}$, v ktorom algoritmus môže vyjadriť svoje hypotézy vo výrazoch $D_{n,k}$ formúl; môže byť dosiahnuté rýchlejšie učenie. Výsledok **o nenaučiteľnosti** je preto uvažovaný aj v zmysle závislosť od reprezentácie.

7.3 Veľkosť reprezentácie

Už sme sa zmienili o tom, že výstup realistického učiaceho algoritmu nie je abstraktná funkcia ale skôr reprezentácia tejto funkcie cez formulu alebo stav stroja. Pretože booleovská funkcia, ktorá môže byť reprezentovaná krátkou booleovskou formulou je zrejme "jednoduchšia" než taká, ktorá vyžaduje dlhšiu formulu, môžeme očakávať, že je ju ťažšie naučiť než krátku formulu.

Rámec potrebný pre starostlivú diskusiu o takých veciach je poskytnutý dojmom reprezentácie $\Omega \to H$ ako bolo uvedené vyššie. Množina Ω môže byť myslená ako množina formúl alebo množina stavov stroja tak, že pre každé $\omega \in \Omega$ existuje odpovedajúca hypotéza h_ω. V nasledujúcich niekoľkých sekciách uvedieme ako reprezentácia hypotéz ovplyvňuje čas behu učiacich algoritmov.

Aby sme to mohli urobiť, potrebujeme nejakú mieru pre "veľkosť" reprezentácie hypotézy. Samozrejme, že neexistuje žiadna absolútna miera a tak musíme skonštruovať jednu, ktorá sa zdá byť rozumná pre skúmané problémy. Booleovský prípad je najpriamočiarejší.

Štandardná metóda reprezentujúca bool. funkciu pomocou formúl bola popísaná v 2.3. Formálne používame abecedu $4 + 2n$ symbolov: $(\ ) \wedge \vee\ u_1\ \overline{u_1}\ \ldots u_n\ \overline{u_n}$, ktoré sú skombinované podľa určitých pravidiel. Táto abeceda môže byť zakódovaná použitím $3 + \lceil \log n \rceil$ bitov pre každý symbol ako je možné

ukázať v nasledujúcej tabuľke:

Symbol	Kód
(	110 0000...0
)	101 000...0
$\vee$	101 000...0
$\wedge$	111 000...0
u_1	001 000...01
$\overline{u_1}$	000 000...01
u_2	$\underbrace{001}\ \underbrace{000\ldots10}$
	$\vdots$

Idea je v tom, že prvé 3 bity sú použité na kódovanie povahy symbolu a zvyšných $\lceil \log_2 n \rceil$ bitov je použitých na reprezentáciu symbolov u_i, $\overline{u_i}$. Nech ω je správna formula, ktorá je dosiahnutá z abecedy uvedenej vyššie. Ak ω má $\overline{s}$ symbolov, potom môže byť zakódovaná pomocou $s \cdot (3 + \lceil \log_2 n \rceil)$ bitov a tak môžeme definovať veľkosť ω

$$\|\omega\| = s \cdot (3 + \lceil \log_2 n \rceil).$$

Napríklad, ak $n = 3$, $\omega = (u_1 \wedge u_2) \vee u_3$ (7 symbolov)

$$\|\omega\| = 7 \cdot (3 + \lceil \log_2 3 \rceil) = 35.$$

Ako sme už uviedli, výstup učiaceho algoritmu nie je abstraktná funkcia alebo hypotéza, ale reprezentácia hypotézy pomocou formuly alebo stroja. Z tohto hľadiska je rozumné porovnať veľkosť takého výstupus veľkosťou vstupu do algoritmu, čo je vlastne tréningová vzorka označených príkladov.

Príklad 1 *Predpokladajme, že máme 20 príkladov 30 bitových do učiaceho algoritmu pre monočleny. Celkový počet bitov vstupu je* $20 \cdot (30 + 1) = 620$. *Výstupom je hypotéza-monočlen, ktorý môže byť akceptovaný pomocou 30 literálov a 29 konjunkcií. Použitím vyššie uvedenej kódovacej schémy dostávame počet bitov výstupu*

$$(30 + 29) \cdot (3 + \lceil \log_2 30 \rceil) = 59 \cdot 8 = 472$$

Pretože toto číslo je menšie než počet bitov na vstupe, je celkom rozumné hovoriť, že výstup je (v nejakom zmysle) kompresovaná forma vstupu. □

Tento príklad ilustruje, že môžeme očakávať, že učiaci algoritmus dáva na výstup reprezentáciu ω hypotézy h_ω takú, že h_ω nie je len rozšírením trénovacej vzorky, ale ω je kompresovaným tvarom vstupu. To je v zmysle, že ω obsahuje tak veľa informácie ako bolo vo vstupnej trén. vzorke, definuje rozšírenú funkciu a vyžaduje menej bitov než trén. vzorka. V ďalšom uvidíme, že ak učiaci algoritmus L dá na výstupe reprezentáciu hypotézy, ktorá nie je príliš dlhá a je signifikantne kompresovaná vzhľadom na vstup, potom L má určité pravdepodobnostné aproximatívne vlastnosti.

7.4 Hľadanie najmenšej konzistentnej hypotézy

Predpokladajme,že je daná tréningová vzorka pre monočlen. Štandardný učiaci algoritmus skonštruuje v polynomiálnom čase hypotézu konzistentnú s tréningovou vzorkou. Avšak môžeme očakávať viac a pýtať sa na najmenší monočlen konzistentný so vzorkou. V tomto kontexte môžeme ignorovať symboly konjunkcie v mnočlene a budeme uvažovať aproximáciu $k \log n$ pre dĺžku monočlena tvoreného k literálmi a definovaného na $\{0,1\}^n$. Teda v ľubovoľnej danej podmnožine M_n najmenší monočlen je taký, ktorý má najmenší počet literálov. Ukážeme, že problém nájdenia najmenšieho monočlena konzistentného s tréningovou vzorkou je NP-ťažký problém.

Náš cieľ bude dosiahnutý pomocou známeho NP-úplného problému. Predpokladajme, že U je konečná množina a $\mathbf{S}$ je konečný systém podmnožín množiny U, ktorého zjednotenie pokrýva celú množinu U. Hovoríme, že podsystém $\mathbf{S}'$ systému $\mathbf{S}$ je *podpokrytie*, ak zjednotenie množín v $\mathbf{S}'$ pokrýva celú množinu U. Nasledujúci problém je jeden z prvých problémov, o ktorom bolo dokázané, že je NP-úplný (Karp, 1972).

PODPOKRYTIE
Inštancia: Dvojica $(U, \mathbf{S})$ podľa vyššie uvedenej definície a kladné celé číslo $k \leq |\mathbf{S}|$

Otázka: Existuje podpokrytie pokrytia $\mathbf{S}$ obsahujúce najviac k množín?

Je treba poznamenať, že veľkosť inštancie závisí od $|U| = u$ aj od $|\mathbf{S}| = n$. V skutočnosti môžeme popísať $(U, \mathbf{S})$ pomocou matice veľkosti $u \times n$, v ktorej ku každému prvku (riadok) je vyjadrená príslušnosť k množine v systéme $\mathbf{S}$. Hodnota $u.n$ môže byť považovaná za veľkosť inštancie $(U, \mathbf{S})$ a toto je parameter, ktorého sa týka otázka polynomiálneho algoritmu.

Predchádzajúci problém je zapísaný v existenčnom tvare. Je možné sformulovať optimalizačný problém v nasledujúcom tvare:
PODPOKRYTIE
Inštancia: Dvojica $(U, \mathbf{S})$ podľa vyššie uvedenej definície a kladné celé číslo $k \leq |\mathbf{S}|$
Otázka: Aká je veľkosť minimálneho podpokrytia pre $(U, \mathbf{S})$?

Je zrejmé, keď máme odpoveď na druhý problém čas behu je polynomiálny v $u.n$, tak je zodpovedaný aj prvý problém v polynomiálnom čase.
Poznámka: urobené na cvičeniach.

7.5 OCCAM algoritmy

Nech $\Omega \to H$ je reprezentácia booleovských funkcií. Nech $\|\omega\|$ je miera veľkosti reprezentácie definovanej pre každé $\omega \in \Omega$. Pre každé $r \geq 1$ definujeme

$$\Omega_r = \{\omega \in \Omega | \quad \|\omega\| = r\}$$

a nech H_r označuje podmnožinu H obsahujúcu tie hypotézy h_ω, ktorých minimálna reprezentácia má veľkosť r.

Hovoríme, že také hypotézy majú veľkosť reprezentácie r.

Potom H môže byť odstupňované pomocou veľkosti reprezentácie $H = \bigcup H_r$. Učiaci algoritmus L pre H má vstup - tréningovú vzorku pre nejakú cieľovú funkciu $t \in H$. Predpokladajme, že $t \in H_r$; inak povedané najmenšia reprezentácia t má veľkosť r. Výstup z L bude špecifikovaný reprezentáciou $\omega \in \Omega_q$. Potrebujeme uvažovať vzťah medzi q a r. Na základe výsledkov predchádzajúceho odseku môže byť ťažké nájsť najmenšiu možnú hodnotu q, ale môžeme požadovať nájdenie nie najkratšej možnej reprezentáciu, ale dostatočne krátku. Táto idea je presnejšie vyjadrená v nasledujúcej definícii Blumera (1987).

Definícia 7.5.1 *Hovoríme, že učiaci algoritmus L pre H je **Occam** vzhľadom na reprezentáciu $\Omega \to H$, ak*

- *L je konzistentný*
- *k danej tréningovej vzorke $\overline{s}$ dĺžky m pre cieľovú hypotézu $t \in H_r$ výstupná hypotéza $L(s) = h_\omega$ je taká, že $\|\omega\| \leq m^\alpha \cdot r^\beta$, kde $0 < \alpha < 1$ a $\beta \geq 1$ sú konštanty.*

Hranica pre $\|\omega\|$ hovorí, že výstup je komprimovaný vzhľadom na dĺžku tréningovej vzorky a rastie len polynomilne s veľkosťou minimálnej reprezentácie dĺžky cieľovej hypotézy. Podmienka $\alpha < 1$ znamená, že výstup je vlastne komprimovaný tvar vstupu; ak povolíme $\alpha = 1$, potom výstup by bol porovnateľný s veľkosťou tréningovej vzorky, ktorej bitová dĺžka je lineárna v m a žiadna signifikantná komprimácia by nebola dosiahnutá. Nasledujúca veta ukazuje, že výstup krátkej reprezentácie v tomto zmysle, postačí pre PAC učenie.

Aby sme sformulovali túto vetu, vráťme sa k našej originálnej definícii učiaceho algoritmu s konceptovým a hypotézovým priestorom ako rôznymi. Tento rozdiel je tu použiteľný, pretože sme sa zaujímali o hypotézy v H_r použitím plného zdroja H.

Veta 6 *Nech H je priestor booleovských funkcií s reprezentáciou $\Omega \to H$, nech $H = \bigcup H_r$ je odstupňovaný veľkosťou reprezentácie. Ak L je Occam učiaci algoritmus vzhľadom na danú reprezentáciu, potom pre každé r, L existuje PAC učiaci algoritmus pre (H_r, H) so zložitosťou vzorky $m_L(H_r, \delta, \epsilon)$ polynomiálnou v r, δ^* a ϵ^{-1}.*

Dôkaz: Predpokladajme, že sú dané δ, ϵ, μ a $t \in H_r$. Pre každé dané m nech $L(m,t)$ označuje množinu hypotéz $h \in H$ takú, že h je výstup $L(\overline{s})$ algoritmu L pre nejakú tréningovú vzorku $\overline{s}$ dĺžky m a cieľový koncept t. Inak povedané, $L(m,t)$ je efektívny hypotézový priestor pre t. Podľa 2. podmienky Occam algoritmu členmi $L(m,t)$ sú hypotézy h_ω, pre ktoré ω má najviac $M = \lfloor m^\alpha \cdot r^\beta \rfloor$ bitov, a celkový počet takých ω je najviac 2^{M+1}. Odtiaľ

$$|L(m,t)| \leq 2^{m^\alpha \cdot r^\beta + 1}.$$

Poznamenajme, že hranica závisí len od r nie od t samotnej; inak povedané, platí uniformne pre všetky $t \in H_r$. Teraz zopakujeme argument daný v predchádzajúcom odseku. Pravdepodobnosť, že ľubovoľná daná ϵ-zlá hypotéza z H súhlasí s t na tréningovej vzorke dĺžky m je $(1-\epsilon)^m$. Pretože L je konzistentný, jeho výstupné hypotézy súhlasia s tréningovouou vzorkou a teda pravdepodobnosť, že výstupná hypotéza je ϵ-zlá je najviac

$$|L(m,t)| \cdot (1-\epsilon)^m \leq 2^{m^\alpha \cdot r^\beta + 1}(1-\epsilon)^m.$$

Zostáva dokázať, že toto môže byť $< \delta$, ak vezmeme dostatočne veľké m a že m je polynóm v r, δ a ϵ^{-1}. Použitím nerovnosti $(1-\epsilon)^m < e^{-\epsilon \cdot m}$ a preusporiadaním dostávame, že $\epsilon \cdot m \geq A \cdot m^\alpha + B$, kde $A = r^\beta \cdot \ln 2$ a $B = \ln(\frac{2}{\delta})$.
Pretože $\alpha > 1$, podmienka platí ak

$$m^{1-\alpha} \geq (A+B)/\epsilon, t.j. \quad m \geq m_0 = \left\lceil \left(\frac{A+B}{\epsilon} \right)^{1/(1-\alpha)} \right\rceil.$$

Inak povedané, výraz pre m_0 je horná hranica pre zložitosť vzorky. Zrejme m_0 je polynóm v r, pretože A je $O(r^3)$ a tak m_0 je $O(r^{\beta/(1-\alpha)})$; ďalej m_0 je tiež polynóm v δ^* a ϵ^{-1}.
□

Zdôrazňujeme znovu podmienku $\alpha < 1$; je zrejmé, že podmienka $\epsilon \cdot m > A \cdot m^\alpha + B$ môže byť splnená, ak $\alpha = 1$.

Existuje bezprostredný dôsledok tejto vety, že ak čas behu Occam algoritmu je polynomiálny v m, potom čas behu PAC učiaceho algoritmu je polynomiálny v r, δ^* a ϵ^{-1}. Inak povedané, Occam algoritmus L pre H PAC učí každý H_r podľa H a urobí to efektívne vzhľadom na veľkosť reprezentácie a δ a ϵ. Poznamenajme, že nemusí nutne platiť, že H samotný je PAC naučiteľný aj keď by to mohlo tak byť, ak existuje horná hranica na veľkosť reprezentácie hypotézy v H.

7.6 Príklady Occam algoritmov

Predpokladajme, že je daný systém $\mathbf{S} = \{S_1, S_2, \ldots, S_n\}$ konečných množín $U = \bigcup_{i=1}^{n} S_i$. Chceme nájsť najmenšie podpokrytie $(U, \mathbf{S})$; t. j. najmenší podsystém $\overline{\mathbf{S}}$, ktorého zjednotením je U. Videli sme, že tento problém je NP-ťažký. Neznamená to, že neexistujú efektívne prostriedky na dosiahnutie aproximatívneho riešenia pre tento problém. Existuje jednoduchá intuitívna metóda na nájdenie aproximatívneho riešenia, založená na "greedy" metóde, ktorá sa zdá byť veľmi účinnou. Najprv vyberieme množinu S_{j_1}, ktorá obsahuje najväčší počet prvkov z U a odstránime ju z $\overline{s}$. Potom vyberieme S_{j_2}, ktorá obsahuje najväčší počet zvyšných prvkov, atď. Pokračujeme týmto spôsobom, v každom kroku vyberieme množinu, ktorá obsahuje najväčší počet zvyšných prvkov.

Greedy algoritmus pre minimálne pokrytie

```
 set X=U;
while X≠ ∅ do
begin choose S_j such that |S_j ∩ X| is maximal;
set X=X-S_j;
end;
```

Pretože $\overline{\mathbf{S}}$ pokrýva U, proces musí skončiť s podpokrytím $S' = \{S_{j_1}, S_{j_2}, \ldots, S_{j_k}\}$. Samozrejme, veľkosť k výsledného podpokrytia nebude vo všeobecnosti najmenším možným podpokrytím, ale bolo ukázané (Nigmatullin (1969), Yohnsson (1974)), že platí nasledujúci vzťah: $k \leq l \cdot (\ln |U| + 1)$, kde l je veľkosť minimálneho podpokrytia.

Toto poskytuje dobrú hornú hranicu pre **pomer výkonnosti** $\frac{k}{l}$ a v tomto zmysle greedy algoritmus je dobrá aproximácia pre problém.

Čas behu závisí od $u = |U|$ a $n = |S|$, $\quad u \leq n, \quad k \leq \min(u, n) \quad \Rightarrow k \leq n \cdot (\ln u + 1)$.

Každý výberový krok obsahuje nájdenie maxima z najviac n celých čísel a vymazanie najviac u prvkov z každej z najviac n množín. Počet operácií $O(u \cdot n)$. Celkový čas behu je $O(u \cdot n \cdot \min(u, n))$.

Greedy metóda môže byť použitá na odvodenie algoritmov pre určité triedy booleovských formúl. Podľa Hausslera (1988) ukážeme technicky ako greedy algoritmus pre pokrytie môže byť použitý na priestor M_n-monočlenov, ukážeme, že výsledný učiaci algoritmus je Occam.

Počiatočná hypotéza je jednočlen bez literálov, identicky 1-ková funkcia. V každom kroku je pridaný 1 literál do priebežnej konjunkcie literálov podľa pravidla založeného na greedy algoritme pre pokrytie.Budeme hovoriť, že literál λ *eliminuje* negatívny príklad x, ak $\langle\lambda\rangle(x) = 0$. Vezmeme prvky, ktoré budú pokryté ako množinu záporných príkladov v danej tréningovej vzorke a pokrývajúce množiny ako množiny záporných príkladov eliminovaných literálmi určitého druhu. V každom stave vyberieme literál, ktorý eliminuje najväčší počet záporných príkladov zo vzorky, pridáme tento literál do formuly a vymažeme príklady, ktoré eliminuje.

Prečo toto pracuje?
Nech $\overline{s}$ je tréningová vzorka pre monočlen a E nech je množina príkladov v $\overline{s}$ takých, že $E = E^+ \cup E^-$, Pre ľubovoľný literál λ položíme $S_\lambda = \{x \in E^- | \quad \langle\lambda\rangle(x) = 0\}$
Nakoniec, nech

$$\Lambda = \{\lambda | \quad \langle\lambda\rangle(x) = 1 \quad \forall x \in E^+\}$$

Lema 1 *Systém množín* $\mathbf{S} = \{S_\lambda | \quad \lambda \in \Lambda\}$ *pokrýva* E^-.

Dôkaz: Pretože $\overline{s}$ je tréningová vzorka pre monočleny, vieme, že existuje monočlen $t = \langle\lambda_1 \wedge \ldots \wedge \lambda_l\rangle$ taký, že pre $x \in E, t(x)$ je 1 alebo 0 podľa toho či x je v E^+ alebo v E^-. Toto implikuje, že $\lambda_1, \ldots, \lambda_l$ všetky patria do Λ. A tiež, že pre ľubovoľné $x \in E^-$ aspoň jeden z literálov λ_j vyskytujúcich sa v t je taký, že $\langle\lambda_j\rangle(x) = 0$. Inak povedané $x \in S_{\lambda_j} \in \mathbf{S}$.
□

Lema 2 *Ak* $\mathbf{S}' = \{S_{\lambda_1}, \ldots, S_{\lambda_k}\}$ *je ľubovoľné podpokrytie* (E^-, S)*, potom monočlen* $h = \langle\lambda_1 \wedge \ldots \wedge \lambda_k\rangle$ *je konzistentný s* $\overline{s}$.

Dôkaz: Predpokladajme, že $x \in E^+$. Pretože $\lambda_1, \ldots, \lambda_k$ sú členmi Λ, na x dávajú hodnotu 1, teda $h(x) = 1$. Ak $x \in E^-$, potom pretože S' je podpokrytie, existuje j, $1 \leq j \leq k$ také, že $x \in S_{\lambda_j}$. Teda $\langle\lambda_j\rangle(x) = 0$, a teda $h(x) = 0$.
□

Tieto lemy ukazujú, že greedy algoritmus pre problém pokrytia môže byť transformovaný na algoritmus pre hľadanie monočlenov konzistentných s danou tréningovou vzorkou.

Aby sme zreteľne videli, že je to Occam algoritmus, uvažujme jeho chovanie sa na tréningovej vzorke pre monočlen t, ktorého najmenšia reprezentácia je pomocou formuly, ktorá obsahuje l literálov. Minimálna reprezentácia t je veľkosti $r = \lceil l \cdot \log n \rceil$. Výsledok práce greedy algoritmu pre problém pokrytia implikuje, že počet k literálov vo výstupnej formule je taký, že $k \leq l \cdot (\ln |E^-| + 1)$.Preto veľkosť výstupnej formuly splňuje

$$\|\omega\| = \lceil k \cdot \log n \rceil \leq \lceil l \cdot (\ln |E^-| + 1) \cdot \ln n \rceil \leq r \cdot (\ln |E|^- + 1)$$

Platí, že $|E|^- \leq m$.
$\|\omega\| \leq r \cdot (\ln m + 1)$, čo triviálne implikuje Occam kompresnú podmienku.
$\|\omega\| \leq m^\alpha \cdot r^\beta \quad \alpha = \frac{1}{2}, \quad \beta = 1.$
□

Greedy algoritmus sa líši evidentne od štandardného algoritmu pre monočleny. Namiesto začiatku s identicky nulovou funkciou (konjunkcia $2n$ literálov) a nasledujúcim vymazaním literalov použitím kladných príkladov greedy algoritmus štartuje s funkciou identicky rovnou 1 (prázdna konjunkcia literálov) a potom pridáva literály použiím záporných príkladov. Teda, pokým štandardný algoritmus je bezpamäťový on-line algoritmus, greedy algoritmus určite taký nie je. Avšak greedy algoritmus ako Occam algoritmus má dôležitú výhodu v tom, že jeho výstupom sú konzistentné hypotézy, ktoré sú relatívne jednoduché.

7.7 Epac učenie

Predpokladajme, že $H = \bigcup H_n$ je hypotézový veľkosťou príkladov odstupňovaný priestor a $\Omega \to H$ je reprezentácia pre H. Potom by sme mohli odstupňovať každé H_n pomocou veľkosti reprezentácie takto $H_n = \bigcup H_{n,r}$, kde $H_{n,r}$ pozostáva z tých hypotéz H_n, ktoré majú minimálnu veľkosť reprezentácie r. Teda

$$H = \bigcup_n \bigcup_r H_{n,r}$$

je *dvojite odstupňovaný*.

Nech L je učiaci algoritmus pre H v obvyklom zmysle, že $L(\overline{s})$ je v H_n vždy, keď $\overline{s}$ je tréningová vzorka pre hypotézy v H_n. Hovoríme, že L je *efektívny PAC* alebo *ePAC* ak (Valiant, 1991)

- čas behu $R_L(m, n)$ je polynomiálny v m aj v n;
- zložitosť vzorky $m_L(H_{n,r}, \delta, \epsilon)$ je polynomiálna v n, r, δ^* a ϵ^{-1}.

Teda $ePAC$ učiaci algoritmus zaručuje, že vydá pravdepodobnostne aproximovaný správny výstup s časom behu polynomiálnym v n, r, δ^* a ϵ^{-1}.

Jeden spôsob na zaručenie druhej podmienky je použitie nejakej verzie Occam podmienok. V tomto kontexte hovoríme, že L je Occam ak podmienky stanovené v definícii pre Occam algoritmus platia pre každé H_n s konštantami α a β nezávislými od n. Potom máme nasledujúci výsledok.

Veta 7 *Predpokladajme, že hypotézový priestor je $H = \bigcup H_{n,r}$ ako bolo uvedené vyššie a L je Occam algoritmus pre učenie $H_{n,r}$ pomocou H_n s polynomiálnym časom behu $R_L(m, n)$. Potom L je $ePAC$.*

Dôkaz: Z dôkazu vety vyššie uvedenej máme hornú hranicu

$$m_0(H_r, \delta, \epsilon) = \lceil (\frac{A+B}{\epsilon})^{1/(1-\alpha)} \rceil$$

pre zložitosť vzorky algoritmu L na $H_{r,n}$, kde $A = r^\beta \ln 2$ a $B = \ln(2/\delta)$. Ako sme už poznamenali, toto je polynóm v r, δ^* a ϵ^{-1}. Pretože α a β sú nezávislé od n, aj $m_0(H_r, \delta, \epsilon)$ je nezávislé od n. Výsledok vyplýva z toho, že horná hranica na čas behu algoritmu L v PAC učení $H_{n,r}$ je

$$R_L(m_0(H_r, \delta, \epsilon), n)$$

čo je polynóm v n, r, δ^* a ϵ^{-1}.
□

Príklad: Odstupňovaný hypotézový priestor $M = \bigcup M_n$ monočlenov môže byť odstupňovaný dvojite ako $M = M_{n,r}$, kde $M_{n,r}$ pozostáva z tých monočlenov n premenných, ktoré majú veľkosť reprezentácie r. V predchádzajúcom odseku bol popísaný algoritmus pre učenie $M_{n,r}$ pomocou M_n založený na greedy metóde a ukázali sme, ze má Occam vlastnosť pri $\alpha = 1/2$ a $\beta = 1$. Čas behu $R_L(m, n)$ je $O(m.n.min(m, n))$, čo je určite polynóm v m a n. Z toho nám vyplýva, že greedy algoritmus pre M je $ePAC$.
□

7.8 Ďalšie poznámky

Ako sme už uviedli, fakt, že C^k nie je efektívne naučiteľný vzhľadom na veľkosť príkladov, je výsledok, ktorý závisí od reprezentácie. Keby výstupné hypotézy mohli byť reprezentované iným spôsobom než konjunkcie najviac k klauzúl, tak generovanie pravdepodobnostne aproximatívne korektných hypotéz by mohlo byť jednoduchšie. Kearns a Valiant (1989) v tomto smere dosiahli veľmi silný výsledok založený na kryptografických predpokladoch.

7.9 Úlohy

1. Ukážte, že $C_n^k \subseteq D_{n,k}$ pre všetky k a n a že inklúzia je striktná pre niektoré hodnoty n a k.

2. Sformulujte problém pokrytia množiny, ktorý zodpovedá nájdeniu najkratšieho monočlena konzistentného s nasledujúcimi príkladmi.

$$E^+ = \{1110011, 1111011, 1011001, 1011011, 1110001\};$$

$$E^- = \{1010100, 0111011, 0001111, 1001010, 0101111, 1100000\}.$$

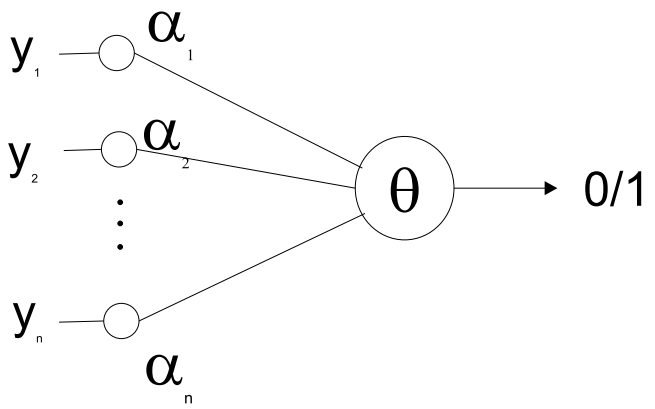
Riešte problém pokrytia a napíšte najkratší monočlen.

Kapitola 8

VC Dimenzia

8.1 Rastová funkcia

Vapnik, Chervonenkis - 1971
Najvýznamnejší pojem pre teóriu výpočtového učenia.



Obrázok 8.1: Perceptrón

Perceptrón
Pre stav $\omega = (\alpha_1, \alpha_2, \ldots, \alpha_n\Theta)$ funkcia $h_\Omega \in H$ z $X = \mathbb{R}^n$ do $\{0,1\}$ je daná

$$h_\omega(y) = \begin{cases} 1, & \text{ak } \sum_{i=1}^{n} \alpha_i y_i \geq \Theta \\ 0, & \text{inak} \end{cases}$$

$\omega \vdash h_\omega$ nie je injekcia;
pre ľubovoľné λ, stav λ_ω definuje tú istú funkciu.

$\Pi_H(x)$ =počet klasifikácií podľa H, t.j. počet rôznych vektorov tvaru

$$(h(x_1), \ldots, h(x_m))$$

kde h prebieha všetky hypotézy v H.

H môže byť nekonečný $\ldots H|E_x, \quad E_x = \{x_1 \ldots x_m\}$ je konečný a má kardinalitu $\Pi_H(x)$.

$$\Pi_H(x) \leq 2^m$$

Definícia 8.1.1 ***Rastová funkcia*** *je*

$$\Pi_H(m) = \max\{\Pi_H(x) : \quad x \in X^m\}$$

Definícia 8.1.2 *Hovoríme, že vzorka x dĺžky m* ***je rozbitá podľa H*** *alebo* ***H rozbíja x****, ak počet možných klasifikácií podľa H je 2^m, t.j. H poskytuje všetky možné klasifikácie.*

- Ak príklady v x nie sú rôzne, tak x nemôže byť rozbitá.
- Ak príklady v x sú rôzne, x je rozbitá podľa $H \Leftrightarrow$ ak pre ľubovoľné $S \subseteq E_x$ existuje nejaká hypotéza $h \in H$ taká, že pre ľubovoľné $1 \leq i \leq m$ platí

$$h(x_i) = 1 \Leftrightarrow x_i \in S.$$

  S je potom podmnožina sústreďujúca kladné príklady.

Definícia 8.1.3 ***VC dimenzia*** *H je maximálna dĺžka vzorky, ktorú H rozbíja. Ak neexistuje, hovoríme, že VC je ∞.*

$$VCdim(H) = \max\{m : \quad \Pi_H(m) = 2^m\}$$

Príklady:

1. $X = \mathbb{R}$, $H \subseteq$ lúče; $x = (x_1, \ldots, x_m)$ je tréningová vzorka, $x_1 < x_2 < \ldots < x_m$.
   Pre $\Theta \in \mathbb{R}$, $\quad r_\Theta = 1 \quad \Leftrightarrow \quad x_i \geq \Theta$.
   Množina klasifikovaných vektorov $\ldots m + 1$.
   $(0 \ldots 00)$, $(0 \ldots 01)$, $\ldots (1 \ldots 11)$
   ľubovoľná vzorka s rôznymi príkladmi má jeden z týchto klas. vektorov (spermutovaný). $\Rightarrow$ $\Pi_H(m) = m + 1$
   V prípade rovnakých príkladov počet klasifikácií je menší.

2. r_Θ, H je priestor lúčov. Nech je daná tréningová vzorka (x_1, x_2) dĺžky 2, $x_1 < x_2$.
   $\Rightarrow$ neexistuje lúč taký, že $h(x_1) = 1$ a $h(x_2) = 0$, pretože by muselo platiť $x2 < \Theta \leq x_1$.
   $\Rightarrow$ H nerozbíja žiadnu vzorku dĺžky 2. H rozbíja ľubovoľnú vzorku dĺžky 1 $\Rightarrow$ $VCdim(H) = 1$.

3. Nech $X = \mathbb{R}^2$. H je hypotézový priestor perceptrónu P_2, $x = (x_1, x_2, x_3)$ sú tri nekolineárne rôzne body. X je rozbitá pomocou $H \Leftrightarrow$ ak pre ľubovoľnú podmnožinu $S \subseteq E_x = \{x_1, x_2, x_3\}$ platí, že S a E_x sú separovateľné.
   $\Rightarrow$ $VCdim(H) \geq 3$ (P_2 rozbíja 3 nekolineárne body.)
   Rovnosť dokážeme tým, že nájdeme vzorku, ktorej príklady sa nedajú separovať $\Rightarrow$ existuje vzorka dĺžky 4, ktorú P_2 nerozbíja.

Tvrdenie 2 *Ak H je konečný hypotézový priestor, tak $VCdim(H) \leq \ln |H|$.*

Dôkaz:
Počet klasifikácií podľa konečného H vzorky ľubovoľnej dĺžky je najviac počet rôznych hypotéz v H
$\Rightarrow$ $\Pi_H(m) \leq |H|$
VCdim je najväčšie d, pre ktoré $\Pi_H(d) = 2^d$.

$$2^d = \Pi_H(d) \leq |H| \quad \Rightarrow \quad d = \Pi_H(d) \leq \ln |H|.$$

□
Príklad: $VCdim(M_n) \ldots |M_n| = 3^n$, M_n sú monočleny.
$VCdim(M_n) \leq (\ln 3) + n$ je horná hranica. čo dolná hranica? Pokúsime sa dokázať, že je rovná n.
Tvrdíme, že M_n rozbíja každú vzorku dĺžky n:
$(e_1, e_2, \ldots, e_n)$, kde $e_i = (0 \ldots 010 \ldots 0)$, kde 1 je na i-tom mieste, $1 \leq i \leq n$.
Predpokladajme, že $(q_1, \ldots, q_n) = q \in \{0, 1\}^n$. Ukážeme, že existuje $h \in M_n$ také, že $h(e_i) = q_i$ pre $1 \leq i \leq n$.
Ak $q = (1 \ldots 1)$ vezmeme za h prázdnu hypotézu.
Ak $q = (0 \ldots 00 \ldots 01)$ tak h vytvoríme ako konjunkciu negácií literálov, pre ktoré $q_j = 0$.
$\Rightarrow$ $VCdim(M_n) \geq n$ pre ľubovoľné n.

8.2 VC dimenzia reálnych perceptrónov

Veta 8 *Pre ľubovoľné n nech P_n je reálny perceptrón s n vstupmi. Potom*

$$VCdim(P_n) = n + 1.$$

Dôkaz: P_n je v stave $\omega = (\alpha_1 \alpha_2 \dots \alpha_n \Theta)$
h_ω - funkcia, ktorú perceptrón počíta

$$h_\omega(y) = 1 \quad \Leftrightarrow \quad \alpha_1 y_1 + \dots + \alpha_n y_n \geq \Theta.$$

Označme

$$l_\omega^+ = \{y \in \mathbb{R}^n \ : \quad \sum_{i=1}^{n} \alpha_i y_i \geq \Theta\}$$

$$l_\omega^- = \{y \in \mathbb{R}^n \ : \quad \sum_{i=1}^{n} \alpha_i y_i < \Theta\}$$

$$l_\omega = \{y \in \mathbb{R}^n \ : \quad \sum_{i=1}^{n} \alpha_i y_i = \Theta\}$$

$C \subseteq \mathbb{R}^n$ je **konvexná**, ak pre ľubovoľné $x, y \in C$ a ľubovoľné reálne číslo λ, $0 \leq \lambda \leq 1$, bod $\lambda x + (1 - \lambda) y \in C$.

Prienik ľubovoľných 2 konvexných množín v $\mathbb{R}^n$ je konvexná množina. Pre ľubovoľnú množinu bodov $S \subseteq \mathbb{R}^n$ existuje najmenšia konvexná množina obsahujúca S;

$conv(S)$ ... konvexný obal S je prienik všetkých konvexných množín obsahujúcich S;

Pripomeňme Radonovu vetu:

Veta 9 *Nech n je kladné celé číslo, E je ľubovoľná množina n+2 bodov z $\mathbb{R}^n$. Potom existuje $\emptyset \neq S \subseteq E$ taká, že*

$$conv(S) \cap conv(E \setminus S) \neq \emptyset.$$

Nech $x = \underbrace{(x_1 \dots x_{n+2})}_{rozne}$ je ľubovoľná vzorka príkladov z $\mathbb{R}^n$ dĺžky n+2.

E_x je množina príkladov v x ... $|E_x| = n + 2$. Podľa Radonovej vety existuje $S \neq \emptyset$, $S \subseteq E_x$

$$conv(S) \cap conv(E_x \setminus S) \neq \emptyset.$$

Predpokladajme, že existuje h_{omega} v P_n také, že S je množina pozitívnych príkladov h_ω v E_x.

$$\Rightarrow S \subseteq l_\omega^+, \ E_x \setminus S \subseteq l_\omega^-$$

Pretože uzavretý polpriestor l_ω^+ a otvorený l_ω^- sú disjunktné a sú konvexné v $\mathbb{R}^n$ $\Rightarrow$ $conv(S) \subseteq l_\omega^+$, $conv(E_x \setminus S) \subseteq l_\omega^-$
$conv(S) \cap conv(E \setminus S) \subseteq conv(S) \cap conv(E_x \setminus S) = \emptyset$

$\Rightarrow$ neexistuje taká h_ω, a preto x nie je rozbitá p_n.
$\Rightarrow$ žiadna vzorka dĺžky n+2 nie je rozbitá pomocou p_n.

$\Rightarrow VCdim(P_n) \leq n + 1$.

Opačná nerovnosť: $o \in \mathbb{R}^n$, $o = (0, 0 \dots 0)$, $e_i = (0 \dots 10 \dots 0)$, $1 \leq i \leq n$. Ukážeme, že P_n rozbíja $x = (o, e_1, \dots e_n)$ dĺžky $n + 1$.

Nech $S \subseteq E_x = \{o, e_1 \dots e_n\}$. Nech

$$\alpha i = \begin{cases} 1, & \text{ak } e_i \in S \\ -1, & \text{ak } e_i \notin S \end{cases}$$

$$\Theta = \begin{cases} -\frac{1}{2}, & \text{ak } o \in S \\ +\frac{1}{2}, & \text{ak } o \notin S \end{cases}$$

Priamou verifikáciou, ak $\omega = (\alpha_1 \dots \alpha_n \Theta)$ je stav P_n, potom množina pozitívnych príkladov h_ω je práve $S \Rightarrow VCdim(P_n) \geq n + 1$. □

8.3 Sauerova lema

Rastová funkcia - miera počtu rôznych klasifikácií vzorky dĺžky m na pozitívne a negatívne príklady podľa H, pokiaľ $VCdim\ H$ je max. hodnota m, pre ktorú platí $\Pi_H(m) = 2^m$.

Veta 10 ***(Sauerova lema)****: Nech $d \geq 0$ a $m \geq 1$ sú celé kladné čísla, nech H je hypotézový priestor s $VCdim(H) = d$. Potom*

$$\Pi_H(m) \leq \underbrace{1 + \binom{m}{1} + \binom{m}{2} + \ldots + \binom{m}{d}}_{\Phi(d,m)}.$$

Binomické koeficienty spĺňajú

$$\binom{a}{b} = \binom{a-1}{b} + \binom{a-1}{b-1}$$

Zaveďme funkciu:

$$\Phi(0, m) = 1 \quad (m \geq 1);$$

$$\Phi(d, 1) = 2 \quad (d \geq 1)$$

$$\Phi(d, m) = \Phi(d, m-1) + \Phi(d-1, m-1), \quad (d \geq 1\, m \geq 2)$$

Dôkaz: Ak $VCdim(H) = d = 0$, potom príklad x-ľub, $h(x)$ je rovnaké (buď 0 alebo 1) pre ľub. hypotézu $h \in H$. $\Rightarrow$ $\Pi_H(x) = 1$ pre ľub. vzorku dĺžky m $\Rightarrow \Pi_H(m) = 1 = \Phi(0, m)$ $\Rightarrow$ veta platí pre $d = 0$.
Ak $m = 1$ a $d \geq 1$ $\Rightarrow$ $\Pi_H(1) \leq 2 = \Phi(d, 1)$

Indukciou na $d + m$:

- Prípad $d + m = 2$ je dokázaný.
- Predpokladajme, že veta platí pre $d + m \leq k$, kde $k \geq 2$ a nech H je hypotézový priestor s $VCdim = d$, x je trénovacia vzorka dĺžky m, kde $d + m = k + 1$.
  Prípady $(d, m) = (0, k + 1)$ a $(d, m) = (k, 1)$ sú už dokázané.
  Nech $d \geq 1$, $m \geq 2$, x obsahuje rôzne príklady, E je množina príkladov v x, $H_E = H|E$ je obmedzenie hypotéz H na E.
  $\Rightarrow$ H_E je konečný a $\Pi_H(x) = |H_E|$.
  Potrebujeme ukázať, že $|H_E| \leq \Phi(d, m)$.
  Nech $F = E \setminus \{x_m\}$, $H_F = H|F$.
  Dve rôzne hypotézy $h, g \in H_E$ pri obmedzení na F dávajú tú istú hypotézu z H_F práve vtedy, keď sa zhodujú na F a nezhodujú na x_m.
  H_* je množina všetkých hypotéz, ktoré vzniknú takto:
  Ak $h^* \in H_*$, tak sú možné obe rozšírenia h^* na funkciu na E ... hypotézu z H_E. h^* je rozšírenie
  $\Rightarrow$ $|H_E| = |H_F| + |H_*|$.
  Nech $x' = (x_1 \ldots x_{m-1})$. Potom

$$|H_F| = \Pi_H(x') \leq \Pi_H(m-1) \leq \Phi(d, m-1)$$

  pretože $d + (m-1) \leq k$.
  Tvrdíme, že $VCdim(H_*)$ je najviac $d - 1$. Ak by $VCdim(H_*) = d \Rightarrow h^*$ rozbíja nejakú vzorku $z = (z_1 \ldots z_d)$ dĺžky d príkladov z F.
  Pre

$$h^* \in H_* \begin{cases} h_1 \in H_E \ldots & h_1(x_m) = 0 \\ h_2 \in H_E \ldots & h_2(x_m) = 1 \end{cases}$$

  $\Rightarrow$ H_E a teda H rozbíja vzorku $(z_1 \ldots z_d x_m)$ dĺžky $d + 1$, čo je v spore s $VCdim(H) \leq d$.
  $\Rightarrow$ $VCdim(H_*) \leq d - 1$.

  Použitím indukčnej hypotézy

$$|H_*| = \Pi_{H_*}(x'0) \leq \Pi_{H_*}(m-1) \leq \Phi(d-1, m-1)$$

  pretože $(d-1) + (m-1) \leq k$. Kombináciou oboch výsledkov dostaneme

$$\Pi_H(x) = |H_E| = |H_F| + |H_*| \leq \Phi(d, m-1) + \Phi(d-1, m-1) = \Phi(d, m).$$

□

Tvrdenie 3 *Pre všetky* $m \geq d \geq 1$ *platí* $\Phi(d,m) < \left(\frac{e \cdot m}{d}\right)^d$.

Tvrdenie 4 *Nech* H *je ľubovoľný hypotézový priestor obsahujúci aspoň 2 hypotézy a definovaný na konečnom príkladovom priestore* X*, potom* $VCdim(H) > \frac{\ln|H|}{1-\ln|X|}$.

8.4 Úlohy k VC-dim

1. Ukážte, že ak $X = \mathrm{R}$ a H je množina všetkých uzavretých intervalov, tak $\Pi_H(m) = 1+m+\frac{1}{2}m(m-1)$.
2. Popíšte expl. hypotézový priestor P_1 a ukážte, že $VCdim(P_1) = 2$.
3. Ukážte, že ak H je hypotézový priestor reálneho perceptrónu P_2, tak $\Pi_H(4) = 14$.
4. Nech H má konečnú $VCdim$. Pre $h \in H$ definujme $\overline{h}$

   $$\overline{h} = 1 \quad \Leftrightarrow \quad h(x) = 0$$

   a nech komplement H je priestor $\{\overline{h} : \quad h \in H\}$. Dokážte, že majú oba priestory rovnakú VC dimenziu.
5. Dokážte
   (a) $\Phi(d,m) = \Phi(d,m-1) + \Phi(d-1,m-1), \quad d \geq 1, m \geq 2$
   (b) $\Phi(d,m) \leq m^d, \quad m \geq d > 1$.
6. Monočlen je monotónny, ak neobsahuje žiadne negované literály. Dokážte, že priestor monotónnych monočlenov definovaný na $\{0,1\}^*$ má VC dimenziu práve n.
7. Hypotézový priestor H je lineárne usporiadaný, ak má aspoň 2 hypotézy a ak pre ľubovoľné dve $h, g \in H$ platí
   buď $h(x) = 1 \Rightarrow g(x) = 1$
   alebo $g(x) = 1 \Rightarrow h(x) = 1$
   Dokážte, že ak H je lineárne usporiadaný, $VCdim(H) = 1$.
8. Nech G_n je množina hypotéz z P_n, pre ktoré nulový vektor o je negatívny príklad. Predpokladajme, že vzorka $x = (x_1 \ldots x_m)$ je rozbitá pomocou G_n. Prečo sa žiadne x_i nesmie rovnať 0? Dokážte, že vzorka $(x_1 \ldots x_m, 0)$ je rozbitá pomocou P_n. Dokážte, že $VCdim(G_n) = n$.

Kapitola 9

Učenie a VC dimenzia

9.1 Úvod

Ukážeme, že pre ľub. hypotézový priestor H nutnou a postačujúcou podmienkou pre jeho potenciálnu naučiteľnosť je konečná VC dimenzia.

Máme potenciálne naučiteľné hypotézové priestory práve tie, ktoré majú konečnú VC dimenziu?

9.2 VC dimenzia a potenciálna naučiteľnosť

Označenie: $\overline{s} = (\overline{x}, \overline{b})$ pre $\overline{s} = ((x_1, b_1) \dots (x_m, b_m)) \in (X \times \{0,1\})^m$ Ak t je cieľový koncept a $\overline{s}$ je tréningová vzorka pre t, potom $\overline{s}$ označíme $(\overline{x}, t(\overline{x}))$.

Toto zdôrazňuje fakt, že keď $\overline{s} \in S(m,t)$ sú dané len hodnoty t na prvkoch tréningovej vzorky $\overline{x}$.

$H[\overline{x}, t]$ podmnožinu H, ktorá je v súlade s $\overline{s}$ $H[(\overline{x}, t(\overline{x}))]$

Pozorovaná chyba hypotézy $h \in H$ na $\overline{s}$

$$er_{\overline{s}}(h) = \frac{1}{m} |\{i : h(x_i) \neq b_i\}|$$

$H[\overline{s}] = \{h \in H| \quad h(x_i) = t(x_i), \quad 1 \leq i \leq m\}, H[\overline{s}]$ je množina hypotéz, ktoré majú nulovú chybu na $\overline{s}$.

Ak $\overline{s} = (\overline{x}, t(\overline{x}))$, potom $er_{\overline{s}}(h)$ sa nazýva pozorovaná chyba h na $\overline{x}$ vzhľadom na t $er_{\overline{s}}(h,t)$, alebo $er_{\overline{s}}(h)$, ak t je zrejmé.

Veta 11 *Ak hypotézový priestor má nekonečnú VC dimenziu, tak nie je potenciálne naučiteľný.*

Dôkaz:

Nech H má nekonečnú VC dimenziu, t.j. pre ľub. $m > 0$ existuje vzorka $\overline{z}$ dĺžky 2m, ktorá je rozbitá podľa H.

Nech $E = E_{\overline{z}}$ je množina príkladov v tejto vzorke a definujme rozdelenie pravdepodobnosti μ na X

$$\mu(x) = \begin{cases} \frac{1}{2m} & \text{pre } x \in E \\ 0 & \text{inak} \end{cases}$$

Vzhľadom na μ každý príklad v E má rovnakú pravdepodobnosť prezentácie a ostatné príklady majú pravdepodobnosť 0.

$$\mu^m \dots E^m \qquad \text{a 0 inde}$$

Teda s pravdepodobnosťou 1 náhodne vybratá vzorka $\overline{s}$ dĺžky m je vzorkou príkladov z E.

Nech $\overline{s} = (\overline{x}, t(\overline{x})) \in S(m,t)$. S pravdepodobosťou 1 (vzhľadom μ^m) máme $x_i \in E$ pre $1 \leq i \leq m$. Pretože $\overline{z}$ je rozbitá podľa H, ex. hypotéza $h \in H$ taká ,že

$$h(x_i) = t(x_i) \text{ pre } 1 \leq i \leq m \qquad \text{a}$$

$$h(x) \neq t(x) \text{ pre vsetky ostatne } x \in E$$

Z toho vyplýva, že $h \in H[\overline{s}]$: h má chybu aspoň $\frac{1}{2}$ vzhľadom na t.

Ukázali sme, že pre ľub. $m > 0$, ľub. t ex. pravdep. r oz. μ na X také, že udalosť $H[\overline{s}] \bigcap B_{\frac{1}{2}} = \emptyset$ má pravdepodobnosť 0.

Teda neex. žiadne $m_0 > 0$, $m_0 = m_0(\frac{1}{2}, \frac{1}{2})$ pre ktoré by sme mohli tvrdiť, že pre všetky $m > m_0 \mu^m \{\overline{s} \in S(m,t) | H[s] \cap B_{\frac{1}{2}} = \emptyset\} > \frac{1}{2}$.

$\Rightarrow$ H nie je potenciálne naučiteľné. $\square$

Príklad: Pre každú $A \subseteq R$ definujeme charakteristickú funkciu

$$\chi_A : \chi_A(y) = \begin{cases} 1 & \text{ak } y \in A \\ 0 & \text{inak} \end{cases}$$

U-systém všetkých podmnožín R takých, ktoré sú konečnými zjednoteniami uzavretých intervalov

$J = \{\chi_A : A \in U\}$ - priestor zjedn. intervalov

VCdim(J) je nekonečná!

$\overline{x} = (x_1 \dots x_m)$ je trén. vzorka príkladov z R, E_x je odp. množ. príkladov $S \subseteq E_x$; k S je možné skonštruovať $A \in U$ tak, že $S \subseteq A$ a $(E_{\overline{x}} \mathbf{S}) \bigcap A = \emptyset$ takto:

Pre každé $x_i \in S$ nech A_i je inerval, ktorý obsahuje x_i ale neobsahuje žiadny iný prvok $x \in E_{\overline{x}}$.

Nech $A = \bigcup A_i$, A je konečné zjednotenie

$$\chi_A = 1 \qquad na \qquad S$$

$$\chi_A = 0 \qquad na \qquad E_{\overline{x}} - S$$

Inak povedané, J rozbíja $\overline{x}$.

Pretože tento argument je použiteľný a platí len pre konečné vzorky $\Rightarrow$ VC dim(J) je nekonečná.

<u>*Poznamka*</u> :

J je obsiahnutý v priestore všetkých uzavretých podm. R (char. funkcií).

Čo by sa dalo povedať o VC dim tohto všeobec. priestoru?

Jeho VC dim je nekonečná.

H-hypotézový priestor def. na príkl. priestore X

t-cieľový koncept v H

μ - pravdepodobnostné rozdelenie na X

ϵ - ľub. r.č. $0 \leq \epsilon \leq 1$

Nech t, μ, ϵ - fixné, ale ľubovoľné

Definujme

$$Q_m^\epsilon = \{x \in X^m : H[x,t] \cap B_\epsilon \neq \emptyset\}$$

Pravdepodobnosť výberu trén. vzorky, pre ktorú existuje konzist., ale ϵ-zlá hypotéza je

$$\mu^m = \{\overline{s} \in S(m,t) : H[\overline{s}] \cap B_\epsilon \neq \emptyset\}$$

čo je vlastne $\mu^m(Q_m^\epsilon)$.

Teda ukázať, že H je potenciálne naučiteľný znamená nájsť hornú hranicu $f(m,\epsilon)$ pre $\mu^m(Q_m^\epsilon)$, ktorá je nezávislá od t a μ a ktorá ide k 0, keď $m \to \infty$

Potom

$$\forall(0 < \delta < 1)\exists(m_0)\forall(m \geq m_0) \qquad f(m,\epsilon) < \delta$$

$m_0 = m_0(\delta, \epsilon)$, m_0 je tiež hornou hranicou pre veľkosť trén. vzorky.

Tvrdenie: Nech H je hypotézový priestor definovaný na príkladovom priestore X, t, μ, ϵ sú ľub. ale pevne dané. Potom

$$\mu^m\{\overline{s} \in S(m,t) : H[\overline{s}] \cap B_\epsilon \neq \emptyset\} < \underbrace{2\Pi_H(2m)2^{\frac{-\epsilon m}{2}}}_{f(m,\epsilon)}$$

pre vš. $m \geq \frac{8}{\epsilon}$.

$$\Pi_H(m) \leq 1 + \binom{m}{1} + \binom{m}{2} + \ldots + \binom{m}{d} \qquad \text{polynóm}$$

$$lim_{n\to\infty} 2\Pi_H(2m)2^{\frac{-\epsilon m}{2}} \to 0 \qquad ?$$

Kľúčový výsledok: VC dim implikuje potenciálnu naučiteľnosť.

Veta 12 *Ak má hypotézový priestor konečnú VC dimenziu, potom je potenciálne naučietľný.*

D: 4 fázy

- Ohraničiť $\mu^m(Q_m^\epsilon)$ pravdepodobnosťou určitej podmnožiny R_m^ϵ v X^{2m}
- Použitím skupini akcií ohraničiť pravdepodobnosť R_m^ϵ pomocou konečných výrazov
- Vyjadriť túto hranicu pomocou Π_H - komb. argumentami
- Použiť argument posl. lemy na vyjadrenie výsledku, že $\mu^m(Q_m^\epsilon) \to 0$, keď $m \to \infty$

1.fáza:
$\overline{x}, \overline{y} \in X^m$, $\overline{xy} \in X^{2m}$ - je trén. vzorka dĺžky 2m
$R_m^\epsilon = \{\overline{xy} \in X^{2m} : \exists h \in B_\epsilon$ pre ktorú $er_{\overline{x}}(h) = 0$ a $er_{\overline{y}}(h) > \frac{\epsilon}{2}\}$
Lema 1: Pre vš. $m \leq \frac{8}{\epsilon}$ platí

$$\mu^m(Q_m^\epsilon) \geq 2\mu^{2m}(R_m^\epsilon).$$

D:
$\chi_Q \ldots$ charakteristická funkcia Q_m^ϵ:
$\chi_Q(\overline{x}) = 1$, ak $\overline{x} \in Q_m^\epsilon$
$\chi_Q(\overline{x}) = 0$, inak.
$\chi_R(\overline{xy}) = \chi_Q(\overline{x})\psi_{\overline{x}}(\overline{y})$, kde

$$\psi_{\overline{x}}(\overline{y}) = \begin{cases} 1 & \text{ak } \exists h \in H[\overline{x}] \cap B_\epsilon \text{ s } er_{\overline{y}}(h) > \frac{\epsilon}{2} \\ 0 & \text{inak} \end{cases}$$

$$\mu^{2m}(R_m^\epsilon) = \int \chi_R(\overline{xy} = \int (\chi_Q(\overline{x} \int \psi_{\overline{x}}(\overline{y}))$$

$\int$ sú cez celé relevantné priestory;
Vnútorný integrál je pravdepodobnosť, že pri danom $\overline{x}$ existuje $h \in B_\epsilon$, ktoré je konzistentné s $\overline{x}$ a splňuje $er_{\overline{y}}(h) > \frac{\epsilon}{2}$.
Tento integrál určite nie je menej než pravdep., že partikulárne $h \in B_\epsilon$ konzist. s $\overline{x}$ splňuje $er_{\overline{y}}(H) > \frac{\epsilon}{2}$.

$$?? \Rightarrow \mu^{2m}(R_m^\epsilon) \geq \int \frac{1}{2}\chi_Q(\overline{x}) = \frac{1}{2}\mu^m(Q_m^\epsilon)$$

Chernovova hranica (bin. rozdelenie)
$0 \leq p \leq 1$, LE(p,m,s)-pravdepodobnosť najviac s úspechov v m nezávislých pokusoch, z ktorých každý má pravdepodob. p.

$$\text{LE(p,m,(1-}\beta\text{)mp)} \leq e^{\frac{-\beta^2 mp}{2}} \qquad \text{pre ľub.} \qquad 0 \leq \beta \leq 1.$$

Nech $h \in B_\epsilon$, $er_\mu(h) = \epsilon_h > \epsilon$. Pre $\overline{y} \in X^m$, $mer_{overliney}(h)$ označuje počet komponentov, na ktorých t a h sa líšia = je to bin. rozdelená náh. prem.

$$\mu^m\{\overline{y} : er_{\overline{y}}(h) \leq \frac{\epsilon}{2}\} = \mu^m\{\overline{y} : mer_{\overline{y}}(h) \leq \frac{\epsilon}{2}m\}$$

$$\leq \mu^m\{\overline{y} : mer_{\overline{y}}(h) \leq \frac{\epsilon_h}{2}m\} = LE(\epsilon_h, m, (1 - \frac{1}{2})m\epsilon_h)$$

$$\leq exp(\frac{-\epsilon_h m}{8}) < exp(\frac{-\epsilon m}{8})$$

??
2.fáza:
Nech $i \in \{1, \ldots m\}$
τ_i je permutácia, ktorá vymení i a m+i tý prvok;

Nech G_m je grupa generovaná permutáciami $\tau_i\{1 \leq i \leq m\}$ $|G_m| = 2^m$

Lema 2: Pre dané $\overline{z} \in X^{2m}$ nech $\Gamma(\overline{z})$ označuje počet $\sigma \in G_m$, pre ktoré $\sigma\overline{z} \in R_m^\epsilon$. Potom $|G_m|\mu^{2m}(R_m^\epsilon) \leq max\Gamma(\overline{z})$ kde max je cez vš. $\overline{z} \in X^{2m}$.

3.fáza:

Pre ľub. $h \in B_\epsilon$ nech

$$R_m^\epsilon(h) = \{\overline{xy} \in X^{2m} : er_{\overline{x}}(h) = 0 \qquad \text{a} \qquad er_{\overline{y}}(h) > \frac{\epsilon}{2}\}$$

Pre $\overline{z} \in X^{2m}$, $\Gamma(h, \overline{z})$ počet $\sigma \in G_m$, ktoré transformujú $\overline{z}$ na vektor v $R_m^\epsilon(h)$.

Lema 3: $m > 0$, $h \in B_\epsilon$. Potom $\Gamma(h, \overline{z}) < 2^{m(1-\frac{\epsilon}{2})}$ pre vš. $\overline{z} \in X^{2m}$.

Lema 4: Pre ľub. $m > 0$ $\mu^{2m}(R_m^\epsilon < \Pi_H(2m)2^{\frac{-\epsilon m}{2}}$

4. fáza: - kombinácia predch. výsledkov

9.3 Zložitosť vzorky konzistentných algoritmov

Fakt: **Ak hypotézový priestor H má konečnú *VC* dimenziu, tak je potencionálne naučiteľný.**

Inak povedané:

K ľubovoľným parametrom δ - dôveryhodnosť, ϵ - presnosť $(0 < \delta, \epsilon < 1)$, existuje vzorka dĺžky $m_0 = m_0(H, \delta, \epsilon)$ taká, že

$$m \geq m_0 \Longrightarrow \mu^m\{\mathbf{s} \in S(m, t); H[\mathbf{s}] \cap B_\epsilon = 0\} > 1 - \delta \tag{9.1}$$

pre ľubovoľné pravdepodobnostné rozdelenie μ na X ľubovoľný cieľový koncept $t \in H$.

Teda z toho vyplýva, že

- ľubovoľný konzistentný učiaci algoritmus L pre H je *pac* a ďalej
- ľubovoľné $\mu_0(H, \delta, \epsilon)$, pre ktoré platí vyššie uvedená podmienka, je horná hranica na zložitosť vzorky $m_L(H, \delta, \epsilon)$.

Ukážeme, že existuje presnejší výraz pre m_0, a teda pre hornú hranicu na zložitosť vzorky ľubovoľného konzistentného učiaceho algoritmu L pre H.

Pripomeňme, že bolo dokázané, že ak H je konečný hypotézový priestor a L je konzistentný učiaci algoritmus pre H, tak L je *pac* a platí

$$m_L(H, \delta, \epsilon) \leq \left\lceil \frac{1}{\epsilon} ln(\frac{|H|}{\delta}) \right\rceil \tag{9.2}$$

Horná hranica, ktorú odvodíme, závisí od VC dimenzie H a nie od mohutnosti H.

Veta: Predpokladajme, že H je hypotézový priestor s konečnou VC dimenziou $d \geq 1$ a platí $0 < \delta, \epsilon < 1$. Nech

$$m_0 = m_0(H, \delta, \epsilon) = \left\lceil \frac{4}{\epsilon}(d.lg(\frac{12}{\epsilon}) + lg(\frac{2}{\delta})) \right\rceil \tag{9.3}$$

Potom pre ľubovoľné $m \geq m_0$ platí

$$\mu^m\{\mathbf{s} \in S(m, t); H[\mathbf{s}] \cap B_\epsilon \neq \emptyset\} < \delta. \tag{9.4}$$

Dôsledok: Predpokladajme,že hypotézový priestor H má VC dimenziu $d \geq 1$. Potom ľubovoľný konzistentný učiaci algoritmus L pre H je *pac* so zložitosťou vzorky

$$m_L(H, \delta, \epsilon) \leq \left\lceil \frac{4}{\epsilon}(d.lg(\frac{12}{\epsilon}) + lg(\frac{2}{\delta})) \right\rceil \tag{9.5}$$

Toto už je sľúbený výsledok.

Príklad 1: Nech H je priestor lúčov. Jeho VC dimenzia je $VCdim(H) = 1$, teda ak L je ľubovoľný konzistentný učiaci algoritmus pre priestor lúčov, máme

$$m_L(H, \delta, \epsilon) \leq \left\lceil \frac{4}{\epsilon}(d.lg(\frac{12}{\epsilon}) + lg(\frac{2}{\delta})) \right\rceil \tag{9.6}$$

My sme priamo dokázali, že

$$m_0 = \left\lceil \frac{1}{\epsilon} ln(\frac{1}{\delta}) \right\rceil. \tag{9.7}$$

Vidíme, že hranica vypočítaná priamo je lepšia než vyplývajúca z VC dimenzie. Avšak priame argumenty sú mnohokrát ťažké a v tomto prípade je zrejmé, že ak δ a ϵ sú toho istého radu, tak tieto hranice sa líšia len o konštantu.

Príklad 2: Reálny perceptrón P_n má VC dimenziu $n + 1$. Predpokladajme, že pre ľubovoľnú tréningovú vzorku pre hypotézu v P_n máme nájsť stav ω perceptrónu taký, že h_ω je konzistentná so vzorkou. Potom keď použijeme tréningovú vzorku dĺžky

$$\left\lceil \frac{4}{\epsilon}((n+1).lg(\frac{12}{\epsilon}) + lg(\frac{2}{\delta})) \right\rceil \tag{9.8}$$

máme zaručenú približne správnu hypotézu bez ohľadu na cieľovú hypotézu a pravdepodobnostné rozdelenie príkladov.

9.4 Dolné hranice pre zložitosť vzorky

Pripomeňme, že sme dokázali, že ak priestor H má nekonečnú VC dimenziu, tak nie je potenciálne naučiteľný.

Veta: Predpokladajme, že hypotézový priestor H má VC dimenziu $d \geq 1$. Potom existuje konzistentný *pac* učiaci algoritmus L pre H taký, že pre ľubovoľné δ a ϵ zložitosť vzorky spĺňuje

$$m_L(H, \delta, \epsilon) \geq d(1 - \epsilon).$$

Uvedený výsledok je síce jednoduchý, ale je použiteľný len na konzistentné učiace algoritmy. Ďalej neposkytuje univerzálnu dolnú hranicu na zložitosť vzorky konzistentného učiaceho algoritmu; skôr sa zaoberá najhoršími možnými konzistentnými učiacimi algoritmami. Ukážeme silnejší výsledok Ehrenfeuchta et al. - 1989, ktorý poskytuje dolnú hranicu na zložitosť vzorky *ľubovoľného pac* učiaceho algoritmu pre hypotézový priestor konečnej VC dimenzie.

Veta: Pre ľubovoľný hypotézový priestor H s VC dimenziou $d \geq 1$ a pre ľubovoľný *pac* učiaci algoritmus L pre H platí

$$m_L(H, \delta, \epsilon) > \frac{d-1}{32\epsilon} \tag{9.9}$$

pre $\delta \leq \frac{1}{100}$ a $\epsilon \leq \frac{1}{8}$.

Ak by sme uvažovali, že zložitosť vzorky prekročí $(d_0 - 1)/32\epsilon$, kde d_0 je ľubovoľné kladné číslo spĺňajúce $VCdim(H) \geq d_0$, tak by sme mohli dokázať nasledujúce tvrdenie:

Tvrdenie: Ak hypotézový priestor H má konečnú VC dimenziu, tak neexistuje žiadny *pac* učiaci algoritmus pre H.

Tieto výsledky podporujú tvrdenie, že VC dimenzia je dobrá miera "expresívnej sily" hypotézového priestoru H: väčšia VC dimenzia H znamená väčšiu zložitosť vzorky pre *pac* učenie H. V skutočnosti výsledky môžu byť zovšeobecnené aj pre prípady, keď C je ľubovoľný konceptový priestor s VC dimenziou aaspoň $d_0 \geq 1$ a H je ľubovoľný hypotézový priestor (nie nutne rovný C).

Ak L je učiaci algoritmus pre (C, H), vstupom do L musí byť tréningová vzorka dĺžky väčšej než $(d_0 - 1)/32\epsilon$, aby bola zaručená presnosť $\epsilon \leq \frac{1}{8}$ s pravdepodobnosťou $1 - \delta > 99/100$. Ak C má nekonečnú VC dimenziu, tak nemôže existovať žiadny učiaci algoritmus pre (C, H), ktorý je *pac* pre ľubovoľný hypotézový priestor H.

Príklad: Ak J je priestor všetkých zjednotení intervalov, tak pretože J má nekonečnú VC dimenziu, neexistuje žiadny *pac* učiaci algoritmus pre (J, H) pre ľubovoľný hypotézový priestor H. Vyššie uvedený výsledok je veľmi silný. Ukazuje nielen to, že neexistuje žiadny konzistentný alebo efektívny *pac* učiaci algoritmus, ale tiež, že pre dané neohraničené výpočtové zdroje žiadny algoritmus nemôže *pac* naučiť J nezávisle od toho ako by boli reprezentované výstupné hypotézy. Samozrejme, tieto závery platia pre ľubovoľný priestor nekonečnej VC dimenzie, takých ako priestor všetkých uzavretých množín alebo priestor charakteristických funkcií všetkých polygonálnych oblastí v $\mathbf{R}^2$.

Iný výsledok týkajúci sa dolných hraníc zaviedol Blumer et al., 1989. Tento výsledok obsahuje δ a ϵ, ale nezávisí od VC dimenzie hypotézového priestoru. Toto sa dá použiť na netriviálne hypotézové priestory, t. j. také priestory, ktoré obsahujú viac než dve hypotézy.

Veta: Predpokladajme, že L je ľubovoľný *pac* učiaci algoritmus pre netriviálny hypotézový priestor H. Potom

$$m_L(H, \delta, \epsilon) > \frac{(1-\epsilon)}{\epsilon} ln(\frac{1}{\delta}), \tag{9.10}$$

pre ľubovoľné $0 < \delta, \epsilon < 1$.

9.5 Porovnanie hraníc zložitosti vzoriek

Ako už bolo uvedené, mnohé predchádzajúce výsledky môžu byť zovšeobecnené pre prípad, keď konceptový a hypotézový priestor sú rôzne. Ukážeme hranice pre zložitosť vzorky v týchto všeobecnejších prípadoch.

Veta: Nech C je konceptový a H je hypotézový priestor a predpokladajme, že H má konečnú VC dimenziu aspoň 1. Ak L je ľubovoľný konzistentný učiaci algoritmus pre (C, H), potom L je *pac* a pre zložitosť vzorky platí

$$m_L \leq \left\lceil \frac{4}{\epsilon}(VCdim(H) lg(\frac{12}{\epsilon}) + lg(\frac{2}{\delta})) \right\rceil \tag{9.11}$$

pre ľubovoľné δ a ϵ.

Veta: Nech C je konceptový a H je hypotézový priestor taký, že C má VC dimenziu aspoň 1. Predpokladajme, že L je ľubovoľný *pac* učiaci algoritmus pre (C, H). Potom pre zložitosť vzorky pre L platí

$$m_L > max(\frac{VCdim(C) - 1}{32\epsilon}, \frac{1}{\epsilon} ln(\frac{1}{\delta})) \tag{9.12}$$

pre všetky $\delta \leq 1/100$ a $\epsilon \leq 1/8$.

Označenie:
Píšeme $f = O(g)$, ak existuje nejaká konštanta $k > 0$ taká, že pre všetky relevantné x platí $f(x) \leq k.g(x)$.
Píšeme $f = \Omega(g)$, ak existuje nejaká konštanta $k > 0$ taká, že pre všetky relevantné x platí $f(x) \geq k.g(x)$.

Použitím toho označenia sformulujeme vzťahy pre zložitosť vzorky.

- Ak L je *pac*, potom C musí mať konečnú VC dimenziu a platí

$$m_L(C, \delta, \epsilon) = \Omega(\frac{VCdim(C)}{\epsilon} + \frac{1}{\epsilon} ln(\frac{1}{\delta})) \tag{9.13}$$

- Ak H má konečnú VC dimenziu a L je konzistentný, potom L je *pac* a platí

$$m_L(C, \delta, \epsilon) = O\Big(\frac{VCdim(C)}{\epsilon} ln(\frac{1}{\epsilon}) + \frac{1}{\epsilon} ln(\frac{1}{\delta})\Big) \tag{9.14}$$

- Ak H je konečný a L je konzistentný, potom L je *pac* a platí

$$m_L(C, \delta, \epsilon) = O\Big(\frac{1}{\epsilon} ln|H| + \frac{1}{\epsilon} ln(\frac{1}{\delta})\Big) \tag{9.15}$$

V prípade, že $C = H$, VC dimenzia d je konečná a L je konzistentný, máme dolné a horné hranice

$$m_L(H, \delta, \epsilon) = \Omega\Big(\frac{d}{\epsilon} + \frac{1}{\epsilon} ln(\frac{1}{\delta})\Big);$$

$$m_L(H, \delta, \epsilon) = \emptyset\Big(\frac{d}{\epsilon} ln(\frac{1}{\epsilon}) + \frac{1}{\epsilon} ln(\frac{1}{\delta})\Big);$$

Faktor $ln(1/\epsilon)$, ktorý odlíšuje hornú hranicu od dolnej, je nevyhnutný. Výsledky Hausslera, Littlestonea a Warmutha, 1988, ukazujú, že pre každé $d \geq 1$ existuje hypotézový priestor H_d a konzistentný učiaci algoritmus L pre H_d so zložitosťou vzorky rovnou hornej hranici. Na druhej strane je otvorený problém rozhodnúť, či pre každé d a pre každý konceptový priestor C s VC dimenziou d, existuje *nejaký* hypotézový priestor H a *nejaký* (C, H) učiaci algoritmus L, pre ktorý zložitosť vzorky má dolnú hranicu.

Cvičenia:

1. Nech H je hypotézový priestor s vlastnosťou, že pre ľubovoľné $t \in H$ a ľubovoľné $0 < \delta, \epsilon < 1$, existuje $m_0(t, \delta, \epsilon)$ také, že

$$m \geq m_0(t, \delta, \epsilon) \Longrightarrow \mu^m\{\mathbf{s} \in S(m, t); H[\mathbf{s}] \cap B_\epsilon\} > 1 - \delta$$

   pre ľubovoľné rozdelenie pravdepodobnosti μ na vstupnom priestore. Teda m_0 môže závisieť len na cieľovom koncepte. Ukážte, že H musí mať konečnú VC dimenziu a je preto potencionálne naučiteľný.

2. Dokážte, že pre ľubovoľné $c > 0$ platí

$$lnx \leq (ln(\frac{1}{c}) - 1) + cx,$$

   pre všetky $x > 0$.

3. Ukážte,že priestor charakteristických funkcií uzavretých a ohrani4ených polygónových oblastí v rovine $\mathbf{R}^2$ nie *pac* naučiteľný.

4. Prepokladajme, že H je ľubovoľný hypotézový priestor konečnej VC dimenzie $d \geq 1$ a že L je ľubovoľný konzistentný učiaci algoritmus pre H. Je dané nejaké rozdelenie pravdepodobnosti na príkladovom priestore. Akú veľkú tréningovú vzorku potrebujeme, aby s aspoň z 90% nou šancou sme získali hypotézu s chybou menšou než 5%?

5. Booleovská funkcia f sa nazýva *symetrická*, ak $f(x)$ závisí len od počtu vstupov x, ktoré sú rovné 1. Napríklad, pre ľubovoľné n koncept parity definovaný na $\{0,1\}^n$ je symetrick'a. Nech n je kladné celé číslo a nech S_n označuje množinu všetkých symetrických funkcií na $\{0,1\}^n$. Aká je VC dimenzia S_n? Uveďte dolnú a hornú hranicu zložitosti vzorky pre ľubovoľný konzistentný *pac* učiaci algoritmus pre S_n. Poznamenajme, že ľubovoľná hypotéza h v S_n môže byť reprezentovaná vektorom $(h_0, h_1, \ldots, h_n) \in \{0,1\}^{n+1}$, kde h_i je hodnota h na príkladoch majúcich presne i jedničiek. Navrhnite konzistentný učiaci algoritmus pre S_n, ktorý reprezentuje priestor týmto spôsobom.

6. Nech H, G sú hypotézové priestory definované na tom istom príkladovom priestore X. Pre hypotézy $h \in H, g \in G$, definujme $h \vee g$ takto

$$h \vee g = 1, \text{ ak } h(x) = 1 \text{ alebo } g(x) = 1; 0 \text{ inak .}$$

   Nech $H \vee G = \{h \vee g; h \in H, g \in G\}$.

   Dokážte, že

$$\Pi_{H \vee G} \leq \Pi_H(m)\Pi_G(m)$$

   pre všetky m. Definujte $H \wedge G$ obvyklým spôsobom, dokážte analogický výsledok pre tento priestor. Ak H a G sú potenciálne naučiteľné, čo je možné povedať o $H \vee G$ a $H \wedge G$?

7. Nech H je hypotézový priestor konečnej VC dimenzie $d \geq 1$ a pre $s \geq 1$ definujme $H(s)$ induktívne takto: $H(1) = H$ a $H(k) = H \vee H(k-1), k \geq 2$. Dokážte, že pre $m > d$ platí

$$\Pi_{H(s)} \leq (\frac{em}{d})^{sd}.$$

   Potom ukážte, že VC dimenzia $H(s)$ je najviac $2sdlg(3s)$.