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

Konzistentné algoritmy a naučiteľnosť

2.1 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 2.1.1 *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 $\overline{s}$.

nie na musí všetko zo vzorky

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 2.1.2 *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*

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

- existuje také množstvo príkladov, že chyba $< \epsilon$ s $1-\delta$

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

Veta 1 *Ak H je potenciálne naučiteľný a L je konzistentný učiaci algoritmus pre H, potom L je PAC.*

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

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

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

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

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

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

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

□

2.2 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 2 *Ľubovoľný konečný hypotézový priestor H je potenciálne naučiteľný.*

Dôkaz: Predpokladajme, že H je konečný hypotézový priestor a δ, ϵ, 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 $m \geq m_0 = \left[\frac{1}{\epsilon} \cdot \ln \frac{|H|}{\delta}\right]$, 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. δ, ϵ, t, μ 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

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

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.

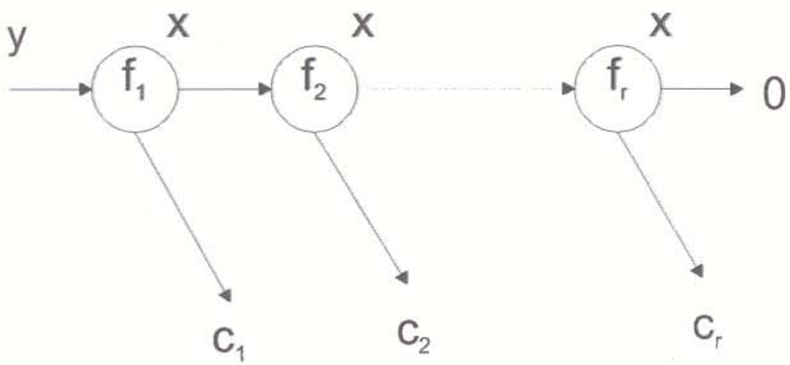
V tomto bode by sa mal čitateľ zadiviť, prečo je to tak. Vytvorili sme komplikovanú podmienku, aby sme dokázali, že je vždy splnená v konečnom prípade, ktorý je najdôležitejší v praxi. Ale praktické predpoklady zavádzajú dodatočné ohraničenie, že počet príkladov by mal byť "ovládateľný" a teda nie je 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 $m_0 = \left[\frac{2^n}{\epsilon} \ln \frac{2}{\delta}\right]$.

2.3 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 2.3.1 *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 2.1: 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` $f(y) = c_2$
$\vdots$
`else if` $f_r(y) = 1$ `then set` $f(y) = c_r$
`else set` $f(y) = 0$

Formálnejšie môžeme definovať:

Definícia 2.3.2 *$DL(K)$ 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*

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

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.

2.4 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), $1 \le i \le m$. 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 $\{g_1, g_2, \dots g_p\}$ 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

$$\langle\rangle\ \langle u_1\rangle\ \langle u_1u_2\rangle\ \langle u_1u_3\rangle\ \langle\overline{u_1}u_4\rangle$$

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* $b_1 \neq b_4$
$\langle u_1u_2\rangle$ *je splnené len pre* x_3 *a* $b_3=0$, *teda vyberieme ako prvý term* $(\langle u_1u_2\rangle, 0)$ *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.

2.5 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 džky, 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.

2.6 Cvičenia:

1. Dokážte, že priestor $H = \{r_\Theta | \quad \Theta \in R\}$ lúčov je potenciálne naučiteľný.
2. Ukážte, že pre hypotézový priestor $D_{n,k}$ $(n \geq k > 1)$ 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.