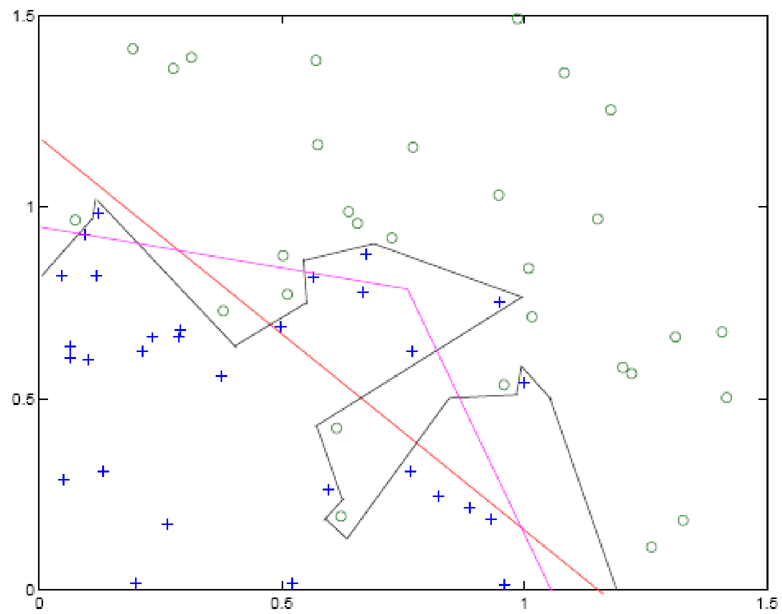
Strojno učenje
Osnove teorije SU

Tomislav Šmuc

Teorija (računalnog) strojnog učenja (TSU)

COmputational Learning Theory (COLT)

Nothing is more practical than a good theory (V. Vapnik)



Literatura:

- Machine learning, T. Mitchel (ch. 7)
- The Nature of Statistical Learning Theory, V. Vapnik

Bousquet, Boucheron, Lugosi:
Introduction to Statistical Learning Theory,
Advanced Lectures on Machine Learning
Lecture Notes in Artificial Intelligence 3176, 169-207.

C.J.C. Burges:
A tutorial on support vector machines for pattern recognition.
Data Mining and Knowledge Discovery, 2(2):955-974, 1998.

(COLT) daje kvantitativne granice na pitanja vezana uz učenje na osnovu primjera

- u zavisnosti o svojstvima problema

  - ❑ Veličini i kompleksnosti prostora hipoteza/modela
  - ❑ Točnosti do koje želimo aproksimirati ciljni koncept
  - ❑ Vjerojatnosti da će algoritam naučiti uspješnu hipotezu
  - ❑ Načinu kako su primjeri prezentirani “učeniku”
    - ❑ Slučajno, od strane tutora, na “traženje učenika” ...

Koliko primjera nam treba, da bi naučili neki koncept ?

Zavisi i o načinu-redoslijedu prezentiranja primjera !

Barem 3 različita načina prezentiranja primjera:

- učenik postavlja primjere <***x*,?** > za koje učitelj daje vrijednosti *f(**x**)*
- učitelj (koji zna vrijednosti ciljne funkcije *f*) daje primjere <***x**,f(**x**)*> učeniku
- primjeri dolaze prema nekom slučajnom redoslijedu <***x*,?** > (okolina, priroda), a na njih učitelj daje vrijednosti f(**x**)

Induktivno učenje

Tražimo h takav da vrijedi :

$$(\forall \langle x_i, f(x_i) \rangle \in T) \quad (B \wedge h \wedge x_i) \vdash f(x_i)$$

- *h* – je hipoteza(model) koja reprezentira/aproksimira ciljni koncept *c(=f(**x**)),* u idealnom slučaju – *h(**x**)=f(**x**). B* predstavlja neko prethodno znanje o problemu.
- ono što u najboljem slučaju možemo garantirati učenjem nekim algoritmom strojnog učenja jest da naučena hipoteza *h* dobro aproksimira ciljni koncept *c* nad skupom primjera za učenje *T*
- ***Osnovna hipoteza induktivnog učenja:***

*Bilo koja hipoteza koja dobro aproksimira ciljni koncept na **dovoljno velikom skupu primjera dostupnih za učenje**, isto će tako dobro aproksimirati ciljni koncept i **na novim, još nedostupnim primjerima.***

Induktivno učenje – učenje na osnovu skupa primjera

Skup primjera za učenje (en. Training set) $\subseteq \Delta$

$$T = \{\langle \mathbf{x}_1, y_1 \rangle, \langle \mathbf{x}_2, y_2 \rangle, \ldots \langle \mathbf{x}_n, y_n \rangle\}$$

x **–** ulazni vektor atributa/varijabli:

Primjer s igranjem danima za igranje tenisa

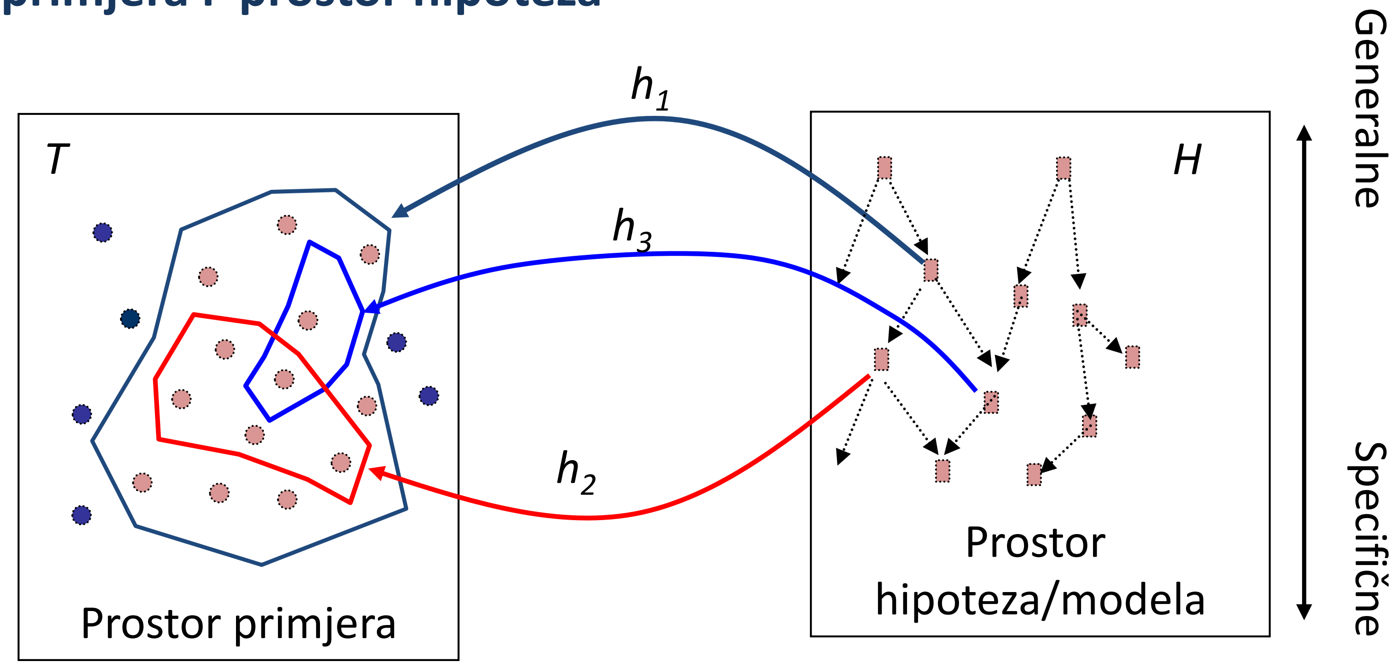
(x_1 (Prognoza), x_2 (Vlažnost), x_3 (Vjetar),....)

y – ciljna varijabla *Igrati_tenis {0,1}*

- *H* – prostor hipoteza/modela – konjunkcija uvjeta na vrijednosti atribute/varijable **x** (npr. x_1 = sunčano/oblačno/kiša/#)
- Ciljni koncept *c*: **X**→y
- *Δ* stvarna i kompletna distribucija svih primjera **X**

(T $\subseteq \Delta$)

Prostor primjera i prostor hipoteza



x_1 = <oblačno, vlažno, vjetrovito, obaveze>

x_2 = <sunčano, suho, vjetrovito, bez_obaveza>

h_1 = <oblačno, #, #, #>

h_2 = <oblačno, suho, #, # >

h_3 = <oblačno, suho, #, obaveze >

Dva viđenja greške pri induktivnom učenju

Greška na skupu za učenje hipoteze ***h*** s obzirom na ciljani koncept *c*

- mjeri kako često *h(**x**)≠ c(**x**)* na skupu primjera iz *T*

$$e_T(h) \equiv \underset{x \in T}{P}\left[h(x) \neq c(x)\right] \equiv \frac{\sum_{x \in T} \delta(h(x) \neq c(x))}{|T|}$$

Stvarna greška hipoteze ***h*** s obzirom na ciljani koncept ***c***

- mjeri kako često *h(**x**)≠ c(**x**)* na bilo kojem skupu slučajno odabranih primjera na osnovu distribucije *Δ*

$$e_\Delta(h) \equiv \underset{x \in \Delta}{P}\left[h(x) \neq c(x)\right]$$

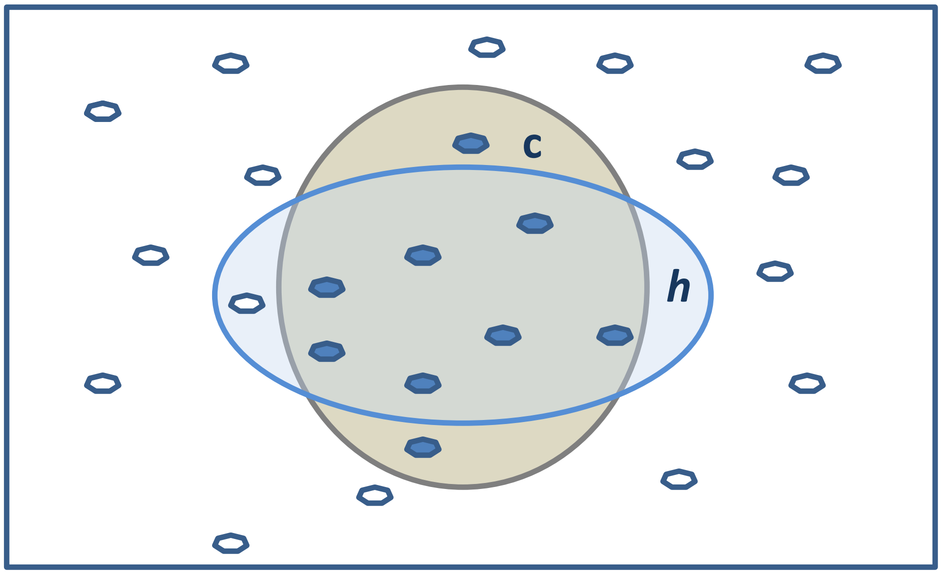
Skup primjera

Distribucija vjerojatnosti

Dva viđenja greške pri induktivnom učenju

Možemo li predvidjeti stvarnu grešku na osnovu izmjerene greške na dostupnom skupu primjera ?

Prostor primjera $X_{P(X)} = \Delta$



PAC Learning – Probably Approximately Correct (1988, Haussler)

Pretpostavimo da imamo klasu mogućih ciljnih koncepata *C* definiranu preko skupa primjera *X* duljine *n*, te algoritam učenja *L* koji koristi prostor hipoteza *H.*

Definicija:

Koncept iz *C* je moguće naučiti u ***PAC smislu*** od strane algoritma *L* korištenjem prostora hipoteza *H,* ako za sve $c \in C$, te distribuciju Δ preko primjera *X,* konstantu ε takvu da vrijedi $0 < \varepsilon < ½$, i vjerojatnost δ ($0 < \delta < ½$), algoritam *L* s vjerojatnošću najmanje ($1 - \delta$) vraća hipotezu $h \in H$ takvu da za nju vrijedi $e_{\Delta}(h) \le \varepsilon$, u vremenu koje je polinomijalno s obzirom na $1/\varepsilon$, $1/\delta$, n *i* $|C|$.

(svodi se na to da:

L treba samo polinomijalni broj primjera za učenje, te da je i vrijeme procesiranja po primjeru polinomijalno !!)

PAC Learning – Probably Approximately Correct (1988, Haussler)

PAC Learning princip – jednostavnim riječima:

- Ukoliko je neka hipoteza izrazito pogrešna - tada će to biti vidljivo već na malom podskupu primjera (preko velike pogreške), s velikom vjerojatnošću;
- I obratno - za bilo koju hipotezu konzistentnu s dovoljno velikim brojem primjera malo je vjerojatno da je izrazito pogrešna

- t.j. **vjerojatno je približno točna (PAC)**

Podprostor konzistentnih hipoteza (en. Version space)

- Hipoteza/model ***h je konzistentna*** sa skupom primjera za učenje *T* nekog ciljnog koncepta $c(\mathbf{x})$ samo ako vrijedi:

$$Konzistentna(h,T) \equiv (\forall (\mathbf{x},c(\mathbf{x})) \in T)\ h(\mathbf{x}) = c(\mathbf{x})$$

- **Version space – $VS_{H,T}$** definicija :

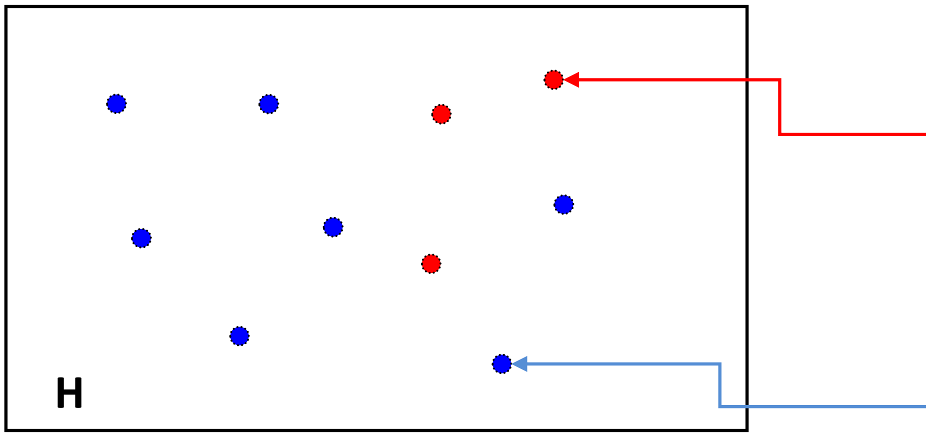
$$VS_{H,T} \equiv \{h \in H \mid Konzistentna(h,T)\}$$

Pretpostavlja se da je $c(\mathbf{x}) \in H$!!

$VS_{H,T}$ u odnosu na prostor hipoteza ***H*** i skup primjera predstavlja podskup hipoteza konzistentnih u odnosu na sve primjere skupa ***T***

ε - iscrpivost/kompletnost $VS_{H,T}$ (ε - exhausted $VS_{H,T}$)

Prostor hipoteza H



$VS_{H,T}$

$e_T(h) = 0$

$e_\Delta(h) > 0$ *(stvarna greška)*

$\neg VS_{H,T}$

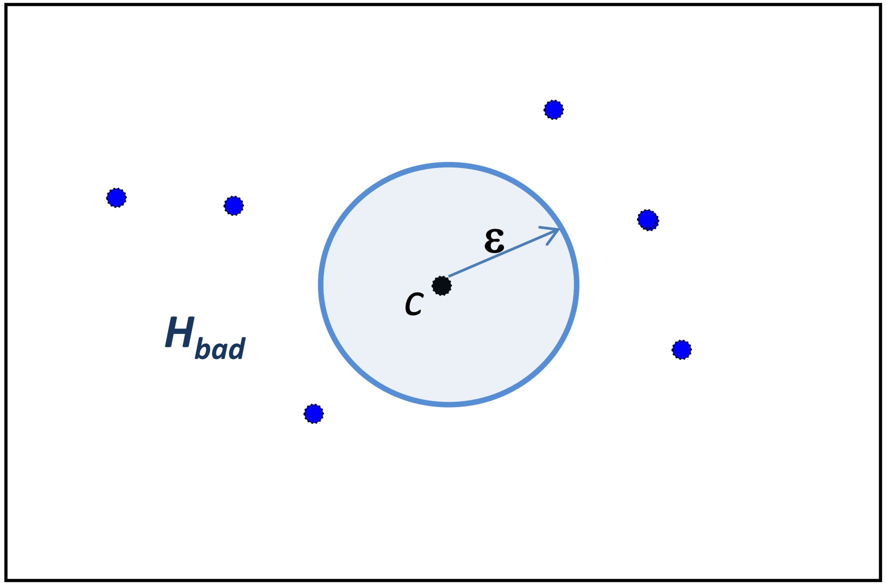
$e_T(h) > 0$

$e_\Delta(h) > 0$ *(stvarna greška)*

$VS_{H,T}$ je <u>ε - iscrpiv/kompletan</u> (s obzirom na *c* i *T*) ako za svaku hipotezu/model u $VS_{H,T}$ vrijedi da ima stvarnu grešku manju od ε:

$$(\forall h \in VS_{H,T})\ e_\Delta(h) < \varepsilon$$

Prostor hipoteza H



$(\forall h \in VS_{H,T})\ e_{\Delta}(h) < \varepsilon$

-za svaku takvu *h* kažemo da je **približno točna (PAC)**

Cilj nam je pokazati za koliko primjera to vrijedi za čitav $VS_{H,T}$!

Prostor hipoteza H

Za svaku $h_{bad} \in H_{bad}$

$$e_{\Delta}(h_{bad}) > \varepsilon$$

Za neki novi primjer x_i iz Δ predikcija neke h_{bad} je točna s vjerojatnosti :

$$p\,(e_{\Delta}(h_{bad}) > \varepsilon\,,\ h_{bad}(x_i) = f(x_i)) \leq (1-\varepsilon)$$

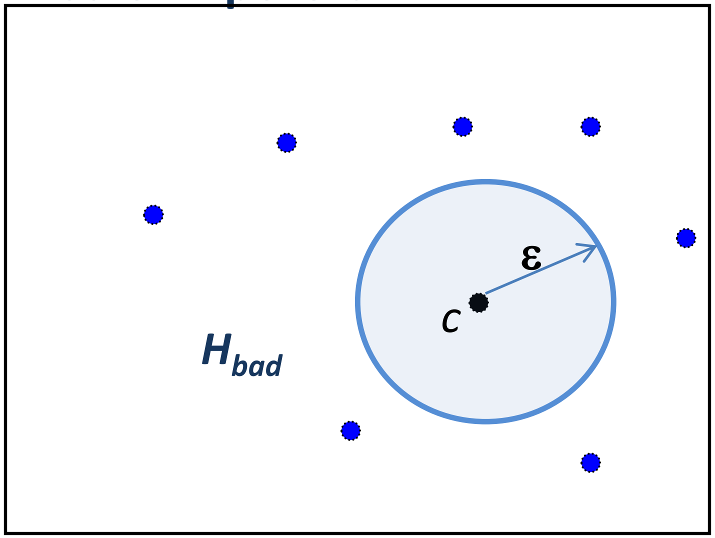
Za N novih primjer x_i iz Δ *vrijedi:*

$$p\,(e_{\Delta}(h_{bad}) > \varepsilon,\ h_{bad}(x_i) = f(x_i),\ \forall x_i \in \{x_1,\ldots,x_N\}) \leq (1-\varepsilon)^N$$

Vjerojatnost da ćemo nabasati na barem jednu hipotezu h za koju vrijedi:

$$p\,(e_{\Delta}(h) > \varepsilon,\ h(x_i) = f(x_i),\ \forall x_i \in \{x_1,\ldots,x_N\}) \leq |H_{bad}|(1-\varepsilon)^N \leq |H|(1-\varepsilon)^N$$

Prostor hipoteza H



Želimo da ova vjerojatnost bude što manja:

$$p(e_{\Delta}(h) > \varepsilon \mid e_{\{x1,\ldots xN\}}(h) = 0) \leq |H|(1-\varepsilon)^N$$

Dakle $|H|(1-\varepsilon)^N \leq \delta$; gdje je δ po volji mali

S obzirom da vrijedi: $1-\varepsilon \leq e^{-\varepsilon}$

$$p(e_{\Delta}(h) > \varepsilon \mid e_{\{x1,\ldots xN\}}(h) = 0) \leq |H| e^{-\varepsilon N}$$

ovu ćemo vjerojatnost δ postići nakon učenja na N primjera :

$$N \geq \frac{1}{\varepsilon}(\ln\frac{1}{\delta} + \ln|\mathbf{H}|)$$

$$N \geq \frac{1}{\varepsilon}(\ln\frac{1}{\delta} + \ln|\mathbf{H}|)$$

Ako algoritam vrati hipotezu koja je točna na *N* primjera, tada s vjerojatnošću od najmanje (*1-*δ) možemo očekivati da je njena stvarna greška najviše ε *!*

- Brojem *N* zapravo označavamo kompleksnost prostora hipoteza

- Veličina |**H**| najviše utječe na *N*

-|**H**| zavisi o "jeziku" kojim je opisan prostor primjera

- Za prostor primjera s *n* varijabli opisan konjunkcijama Boole-ovih literala (npr. h_i = *<oblačno, vlažno, #, #>*)

- |**H**| = ?

Učenje konjunkcija Boole-ovih literala

- Pretpostavimo *H* i neka je stvarni koncept c $\in$ *H*. *H* je opisan konjunkcijama uvjeta na *n* Boole-ovih varijabli (=Boole-ovi literali). (U našem primjeru sa igranjem tenisa svaka varijabla ima 3 moguće vrijednosti).

  Tada vrijedi

  $|H|=3^n$

  pa je broj potrebnih primjera:

$$N \geq \frac{1}{\varepsilon}(\ln\frac{1}{\delta} + n\ln 3)$$

Za $\varepsilon=0.05$; $\delta=0.01$

$$N \geq \frac{1}{0.05}(\ln\frac{1}{0.01} + 10\ln 3) = 312 \text{ primjera}$$

PAC učenje - kad vrijedi c $\in$ *H*

$$N \geq \frac{1}{\varepsilon}(\ln\frac{1}{\delta} + \ln|\mathbf{H}|)$$

Što ako želimo moći naučiti bilo koji ciljni koncept (c $\in$ C) koji možemo zamisliti nad skupom primjera opisanim uvjetima na *n* Boole-ovih varijabli ?

|H| =?

$$|H| = 2^{3^n}$$

!!

želimo li c $\in$ *H* **=>** $N \geq \frac{1}{\varepsilon}(\ln\frac{1}{\delta} + 3^n \ln 2)$

PAC učenje - kad vrijedi c $\in H$

$|H| = 2^{3^n}$ želimo li c $\in H$ **=>** $N \geq \frac{1}{\varepsilon}(\ln \frac{1}{\delta} + 3^n \ln 2)$

Kako ograničiti broj potrebnih primjera ?

Osnovna dilema:

a) Kako napraviti restrikcije na prostor hipoteza (restrikcija jezika, reprezentabilnih funkcija)
b) Dovoljno kompleksan prostor, ali inzistirati da konzistentne hipoteze budu što jednostavnije

PAC učenje - kad ne vrijedi c $\in H$ (agnostičko učenje)

- Možemo probati naučiti *h* koja radi najmanje grešaka na *T* !
- Koliko nam tada treba primjera ? Kako mjerimo stvarnu grešku ?
  - Ustvari možemo samo garantirati (za bilo koju *h*!):

$$P\left[e_{\Delta}(h) > e_T(h) + \varepsilon\right] \leq e^{-2N\varepsilon^2}$$

  - moramo garantirati da ovo vrijedi za svaku h iz H - pa onda i za najbolju*!*

$$P\left[(\exists h \in H) \mid e_{\Delta}(h) > e_T(h) + \varepsilon\right] \leq |H| e^{-2N\varepsilon^2}$$

  - ako gornju vjerojatnost proglasimo δ možemo opet postaviti pitanje koliko nam promjera treba:

$$N \geq \frac{1}{2\varepsilon^2}\left(\ln\frac{1}{\delta} + \ln|\mathbf{H}|\right)$$

Ovo nije garancija da će algoritam naći najbolju *h* !

Napomena: ε je sada samo razlika između $e_{\Delta}(h)$ i $e_T(h)$
- tj. mjera overfitting-a!

PAC učenje - sažetak

- Konačni prostor *H*, i $c \in H$, vrijedi:

$$P\left[\exists h \in H \mid (e_{\Delta}(h) > \varepsilon) \wedge (e_T(h) = 0)\right] \leq \mid H \mid e^{-\varepsilon \cdot N}$$

pa možemo garantirati da je za točnost ε uz vjerojatnost $P \leq \delta$

dovoljno $N \geq \frac{1}{\varepsilon}(\ln\frac{1}{\delta} + \ln|\mathbf{H}|)$ primjera

- Konačni prostor *H* i vjerojatno $c \notin H$ vrijedi

$$P\left[(\exists h \in H) \mid e_{\Delta}(h) > e_T(h) + \varepsilon\right] \leq |H| e^{-2N\varepsilon^2}$$

pa možemo garantirati da je za točnost ε uz vjerojatnost $P \leq \delta$

dovoljno $N \geq \frac{1}{2\varepsilon^2}(\ln\frac{1}{\delta} + \ln|\mathbf{H}|)$ primjera

PAC učenje - problem kad |H| → ∞

Osnovna dilema:

a) Napraviti restrikcije na prostor hipoteza H (restrikcija jezika, reprezentabilnih funkcija) = konačni H => PAC bounds
b) Dovoljno kompleksan prostor H, ali inzistirati da konzistentne hipoteze budu što jednostavnije = beskonačni H ????

Vapnik-Chervonenkis dimenzija !

Shattering => Sposobnost particioniranja skupa primjera od strane nekog prostora hipoteza H
(en. shattering; hr.drobljenje, razbijanje)

Definicija. Skup primjera $S=\{\mathbf{x}_i\}_{i=1,N}$ moguće je ***particionirati/rastaviti*** skupom hipoteza *H*, onda i samo onda ako za svaku ***dihotomiju*** skupa *S*, postoji $h \in H$ koja je konzistentna s takvom dihotomijom.

Dihotomija. Particija skupa primjera *S* u npr. pozitivne i negativne primjere.
Npr. za $N = 3$: postoji 2^3 mogućih dihotomija.

Skup *S* je moguće ***particionirati/rastaviti*** skupom hipoteza *H*, ako za svaku particiju primjera iz *S* u pozitivne i negativne primjere, postoji hipoteza/funkcija *h* iz *H* koja daje upravo iste oznake primjerima

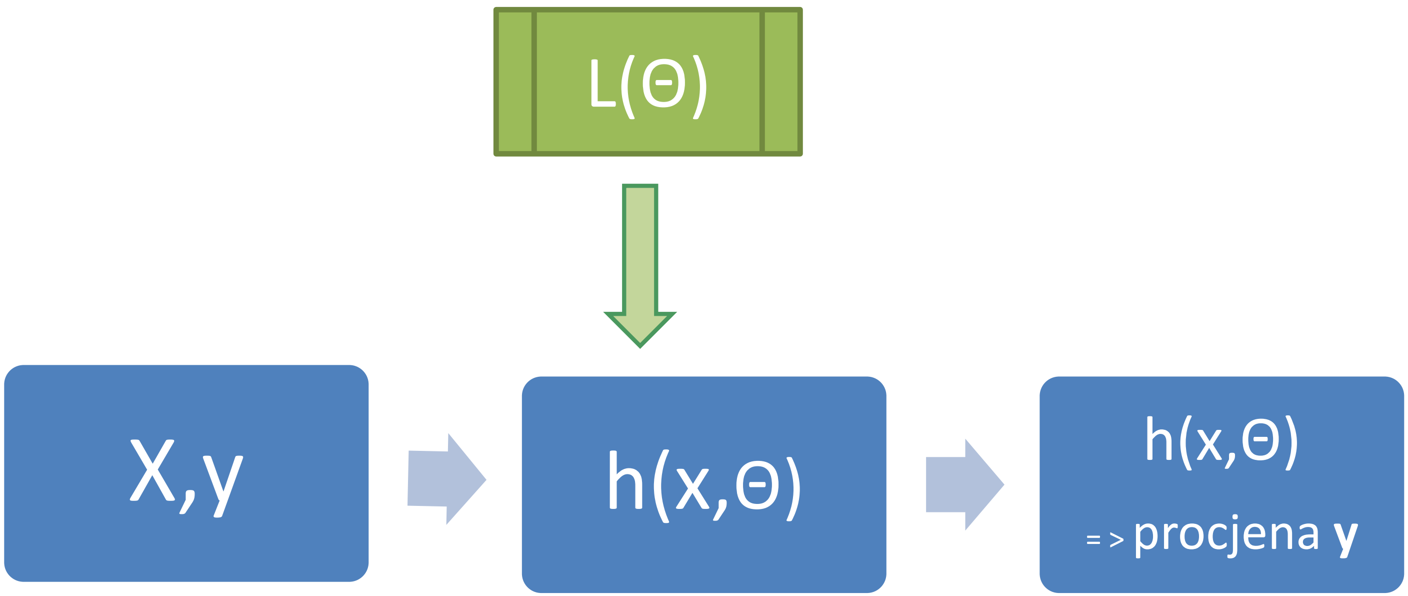
(Intuitivno: Kompleksniji skup funkcija *H* može ***particionirati*** veći skup S!)

Definicija (Vapnik-Chervonenkis dimenzija skupa hipoteza *H* - *VC(H)*)

VC(H) prostora hipoteza H, definiranog preko prostora primjera X **je veličina najvećeg podskupa od X** koji je moguće particionirati/rastaviti korištenjem H. Ako je pomoću H moguće rastaviti po volji velike podskupove X, tada vrijedi VC(H)≡ ∞.

(*h* Є *H* mogu generirati bilo koju klasifikaciju na skupu primjera *S*)

h(x,Θ) **predstavlja neku funkciju koju možemo naučiti algoritmom** ***L***



Malo drugačiji pogled....

Algoritam *L* kojom može particionirati skup točaka S onda ako:

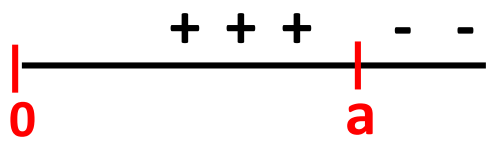
Za svaku moguću dihotomiju u obliku (x_1,y_1), (x_2,y_2),...(x_n,y_n) ..... može naučiti/proizvesti funkciju $h(x,\Theta)$ koja apsolutno točno razdvaja primjere tog skupa (ima grešku =0!)

Postoji 2^n takvih dihotomija, s različitim kombinacijama $y \in (+1,-1)$

Primjeri h(x,Θ) i određivanja VC(h)

Polu-intervali:

h(x, Θ) ∈ H ; H - intervali tipa [0,a), za neki realni a>0



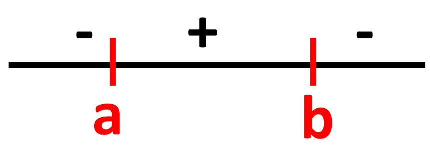
VC(h(x, Θ)) = ?

|H|= ?

Primjeri h(x,Θ) i određivanja VC(h)

Intervali

h(x, Θ) ∈ H ; H - intervali na realnoj osi - [a,b] | b>a



VC(h(x, Θ)) = ?

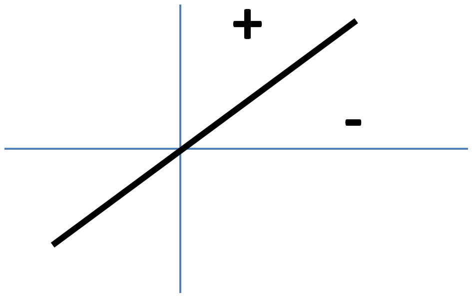
|H|= ?

Primjeri $h(x,\Theta)$ i određivanja VC(h)

Poluprostori - 1

$h(x, \Theta) \in H$

$h(\mathbf{x}, \boldsymbol{\Theta}) = \mathrm{sign}(x \cdot \Theta)$



$VC(h(x, \Theta)) = ?$

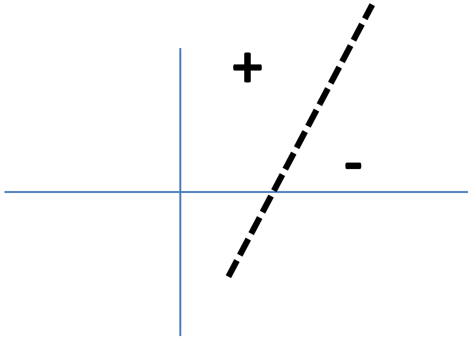
$|H| = ?$

Primjeri h(x,Θ) i određivanja VC(h)

Poluprostori - 2

$h(x, \Theta) \in H$

$h(\mathbf{x}, \mathbf{\Theta}) = \text{sign}(x \cdot \Theta_1 + \Theta_2)$



$VC(h(x, \Theta)) = ?$

$|H| = ?$

VC dimenzija prostora hipoteza H na prostoru primjera **X** je veličina najvećeg konačnog podskupa **x** koji se može rastaviti hipotezama iz **H**

(makar to bio samo jedan podskup te veličine!)

Ako postoji podskup veličine **d** koji se može rastaviti, tada vrijedi - VC(H) >=d

Ako ne postoji podskup veličine **d** koji se može rastaviti, tada vrijedi - VC(H) < d

VC(Poluintervali) = 1 (podskup veličine **2** se ne može rastaviti)

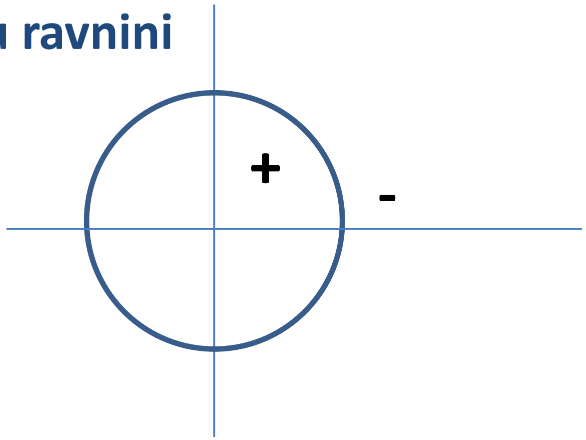
VC(Intervali) = 2 (podskup veličine **3** se ne može rastaviti)

VC(Poluprostori 1) = 2 (podskup veličine **3** se ne može rastaviti)

Drugi primjeri $h(x,\Theta)$, VC(h) - za vježbu ?

$h(x, \Theta) \in H$; – kružnice u ravnini

$h(\mathbf{x}, \boldsymbol{\Theta})=\text{sign}(\Theta_1 \cdot x \cdot x - \Theta_2)$



....

- VC (**w*x -b**) – gdje je **x** – n dimenzionalni prostor
- VC(sin(x))
- VC(stabla odlučivanja)
- VC(perceptron)
- VC(neuralne mreže)

Gornja granica broja primjera

PAC učenje – uz korištenje VC(H) => SRM

$$N \geq \frac{1}{\varepsilon}\left(4\log_2\left(2/\delta\right) + 8VC(H)\log_2(13/\varepsilon)\right)$$

Osim toga Vapnik je pokazao da s vjerojatnošću (1-η) vrijedi

$$e_\Delta \approx e_{test} \leq e_T + \sqrt{\frac{VC(H)(\log(2N/VC(H))+1)-\log(\eta/4)}{N}}$$

Dakle – imamo procjenu greške na novim primjerima na osnovu greške na skupu za učenje i VC (H) !

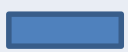
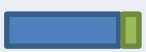
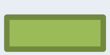
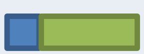
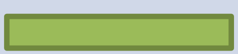
Structural Risk Minimization (Vapnik)

Pretpostavimo da imamo na izbor niz “strojeva”
– koji uče hipoteze iz prostora H_i (funkcije) različitih $VC(H_i)$ tako da vrijedi:

$VC(H_1) \leq VC(H_2) \leq VC(H_3) \leq VC(H_4) \leq \ldots \leq VC(H_N)$

Koji ćemo od “strojeva” - algoritama koristiti ?
- Treniramo svaki od strojeva i mjerimo e_T ... i procjenjujemo e_{test} na osnovu:

$$e_\Delta \approx e_{test} \leq e_T + \sqrt{\frac{VC(H)(\log(2N/VC(H))+1)-\log(\eta/4)}{N}}$$

rbr	H_i	e_T	√ (VC(H)....	~e_{test}	Rang
1	H_1				4
2	H_2				1
3	H_3				1
4	H_4				1
5	H_5				5

VC-dimenzija + SRM – sažetak

- VC-dimenzija je mjera informacija o aproksimacijskoj snazi nekog "stroja za učenje" – algoritma izražena kroz ekspresivnost prostora hipoteza (funkcija) koji taj algoritam koristi;

- SRM: odabir algoritma koji ima minimalnu procjenu greške na novim primjerima

Trebalo bi zapamtiti:

- Shattering
- definiciju VC-dimenzije
- neke od primjera H i VC(H)
- SRM i čemu služi

SRM i sposobnost generalizacije algoritma

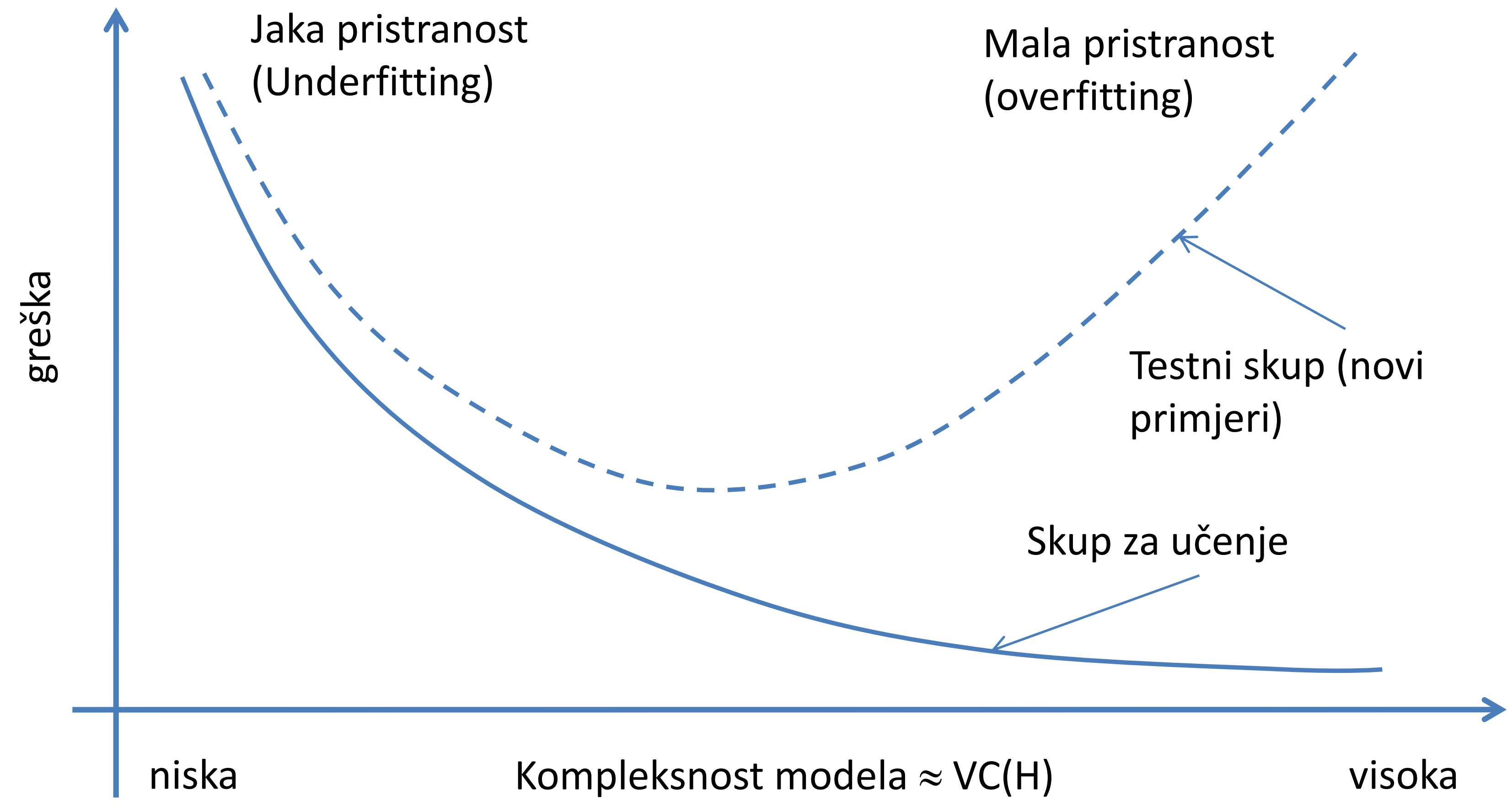
- **generalizacijom algoritma** možemo nazvati kvalitetu predikcije na novim (testnim) podacima

- cilj je dobiti **dobra svojstva generalizacije** naučenim modelom podataka:

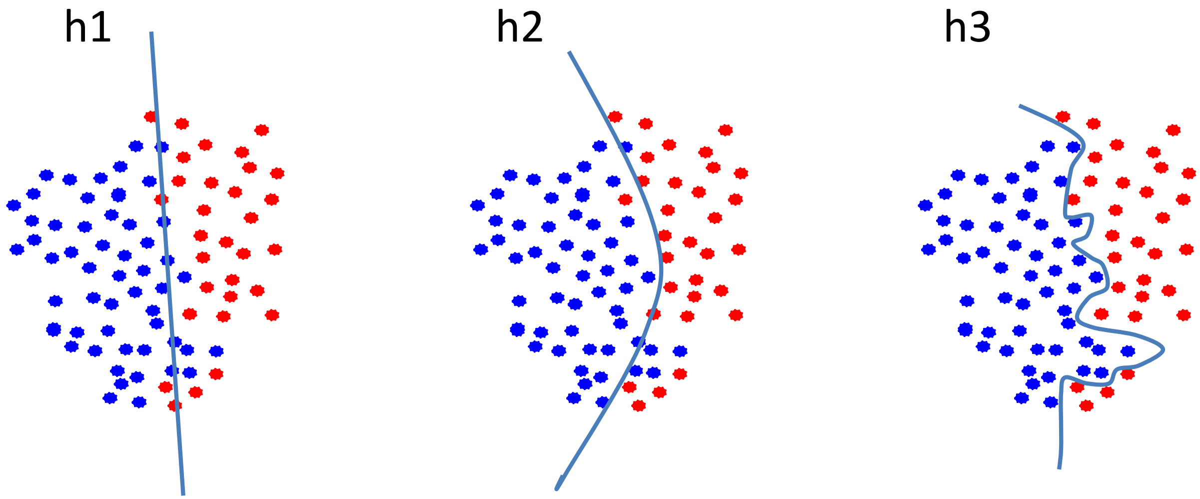
  - kad je **model pre-kompleksan** za neki skup podataka – može se desiti da uči ili memorira dijelove "šuma" ili grešaka, pored stvarne strukture podataka – **pretreniranje (overfitting**, model variance)

  - Nasuprot tome kad je naš **model nedovoljno kompleksan**, tada ne može naučiti (dobro aproksimirati) stvarnu strukturu podataka, bez obzira na njihovu količinu –**pristranost modela (en. underfitting**, model **bias**)

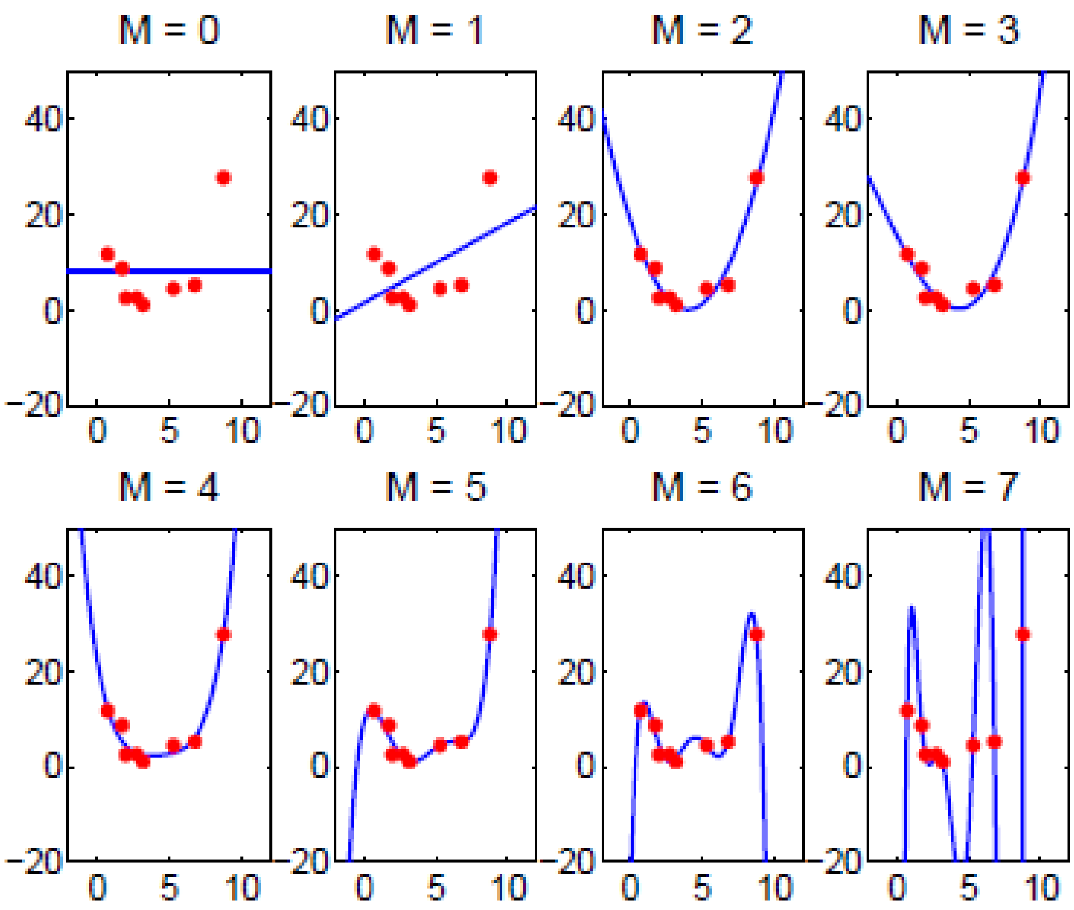
Tipično ponašanje



Primjer - klasifikacija



Primjer - regresija



Nastavak priče:

- struktura i klasifikacija algoritama strojnog učenja
- tehnike evaluacije modela, bias & variance;
- metrike u ocjenjivanju modela