

# Strojno učenje Osnove teorije SU

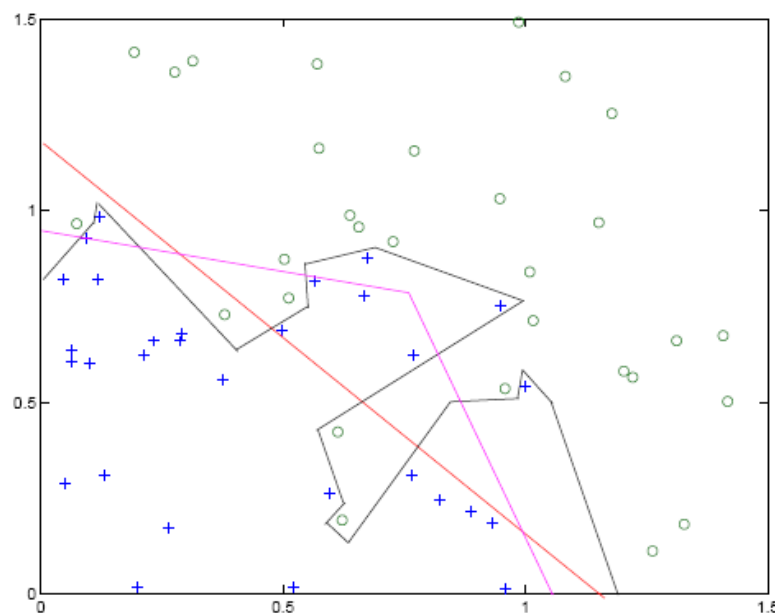
Tomislav Šmuc

PMF, Zagreb, 2020

# 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
- ESLR, Hastie, Tibshirani, Friedman (ch 7)

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.

## Computational Learning Theory (COLT- Computational Learning Theory)

- daje kvantitativne granice na pitanja vezana uz učenje na osnovu primjera

...u zavisnosti o svojstvima problema, algoritma

- ❑ Veličini i kompleksnosti prostora hipoteza/modela
- ❑ Točnosti do koje želimo aproksimirati ciljni koncept
- ❑ Vjerojatnosti da će algoritam naučiti uspješnu hipotezu
- ❑ Broju primjera potrebnih za određenu točnost
- ❑ Načinu kako su primjeri prezentirani “učeniku/algoritmu”

## 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  $\langle \mathbf{x}, ? \rangle$  za koje učitelj daje vrijednosti  $f(\mathbf{x})$
- ☐ učitelj (koji zna vrijednosti ciljne funkcije  $f$ ) daje primjere  $\langle \mathbf{x}, f(\mathbf{x}) \rangle$  učeniku
- ☐ primjeri dolaze prema nekom slučajnom redoslijedu  $\langle \mathbf{x}, ? \rangle$  (okolina, priroda), a na njih učitelj daje vrijednosti  $f(\mathbf{x})$

## Induktivno učenje

Tražimo  $h$  takav da vrijedi :

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

- $h$  – je hipoteza(model) koja reprezentira/aproksimira ciljni koncept  $c(=f(\mathbf{x}))$ , u idealnom slučaju –  $h(\mathbf{x})=f(\mathbf{x})$ .  $B$  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 (učenje pod nadzorom)

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

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

$\mathbf{x}$  – ulazni vektor atributa/varijabli:

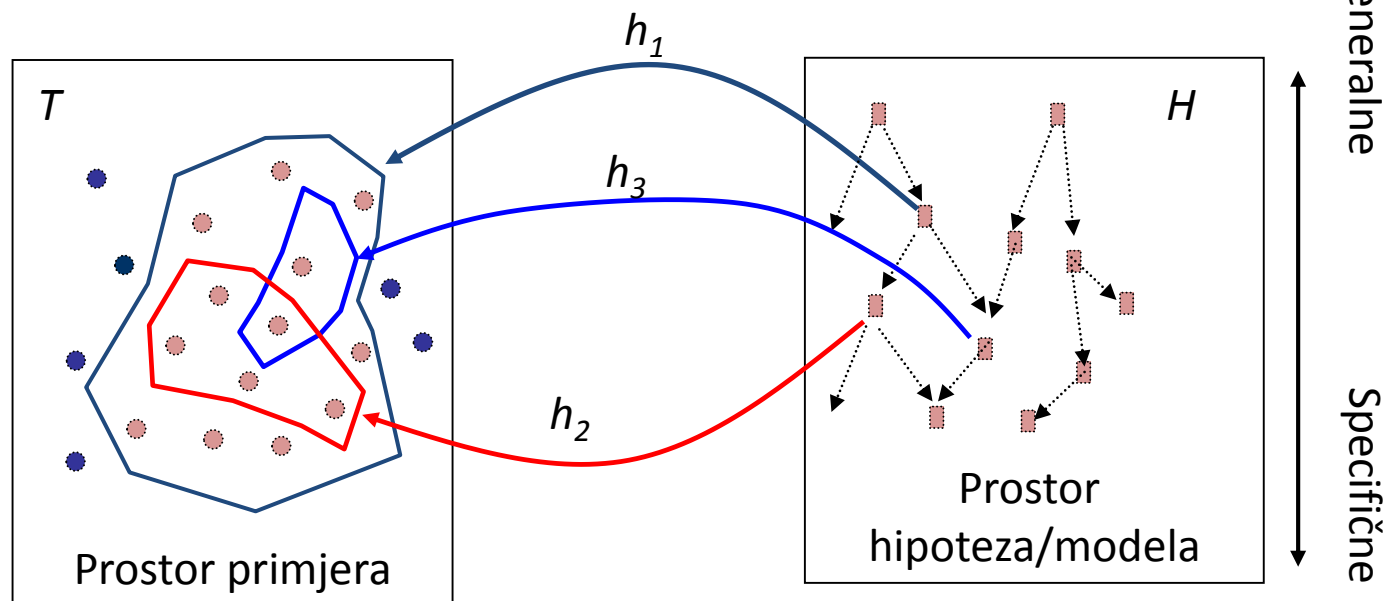
Primjer: predviđanje dana pogodnih za igranje tenisa

$(x_1 \text{ (Prognoza)}, x_2 \text{ (Vlažnost)}, x_3 \text{ (Vjetar)}, \dots)$

$y$  – ciljna varijabla *igrati\_tenis*  $\{0,1\}$

- $H$  – prostor hipoteza/modela – npr.konjunkcija uvjeta na vrijednosti atributa/varijable  $\mathbf{x}$  (npr.  $\mathbf{x}_1 = \langle \text{sunčano, vlažno/vjetrovito/\#} \rangle$ )
- Ciljni koncept  $c: \mathbf{X} \rightarrow y$
- $\Delta$  stvarna i kompletna distribucija svih primjera  $\mathbf{X}$   
( $T \subseteq \Delta$ )

## Prostor primjera i prostor hipoteza



$x_1 = \langle \text{oblačno, vlažno, vjetrovito, obaveze} \rangle$

$x_2 = \langle \text{sunčano, suho, vjetrovito, bez_obaveza} \rangle$

$h_1 = \langle \text{oblačno, \#, \#, \#} \rangle$

$h_2 = \langle \text{oblačno, suho, \#, \#} \rangle$

$h_3 = \langle \text{oblačno, suho, \#, obaveze} \rangle$



## Dva viđenja greške pri induktivnom učenju

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

- mjeri kako često  $h(x) \neq c(x)$  na skupu primjera iz  $T$

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

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

- mjeri kako često  $h(x) \neq c(x)$  na bilo kojem skupu slučajno odabranih primjera na osnovu distribucije  $\Delta$

$$e_{\Delta}(h) \equiv P_{x \in \Delta}[h(x) \neq c(x)]$$

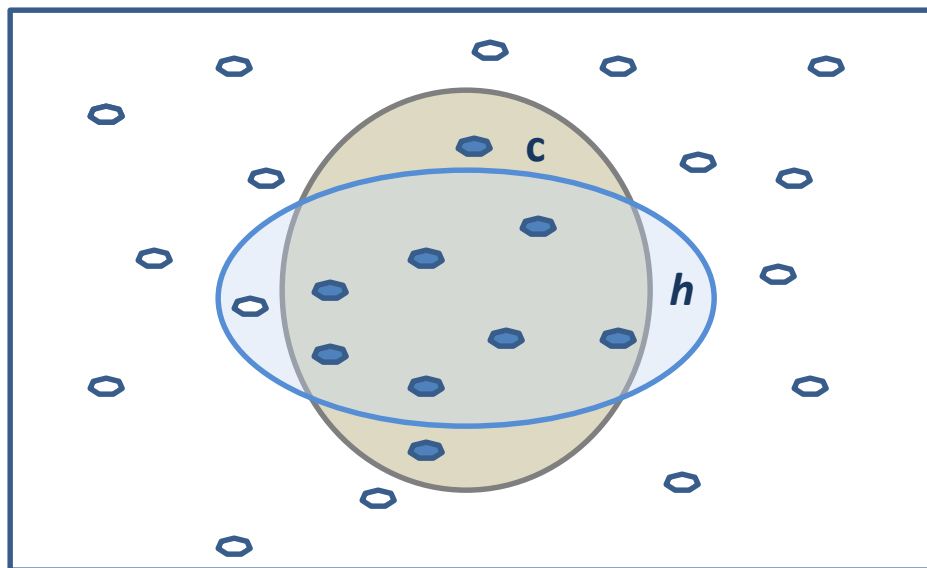
Skup primjera

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  $X$  veličine  $n$ , 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  $\Delta$  preko primjera  $X$ , konstantu  $\varepsilon$  takvu da vrijedi  $0 < \varepsilon < \frac{1}{2}$ , i vjerojatnost  $\delta$  ( $0 < \delta < \frac{1}{2}$ ), algoritam  $L$  s vjerojatnošću najmanje  $(1 - \delta)$  vraća hipotezu  $h \in H$  takvu da za nju vrijedi  $e_{\Delta}(h) \leq \varepsilon$ , 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 – **P**robably **A**pproximately **C**orrect (1988, Haussler)

PAC Learning princip – jednostavnim riječima:

- Ukoliko je neka **hipoteza izrazito pogrešna** - tada će to biti **vidljivo već na malom podskupu primjera** (velika greška), 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:

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

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

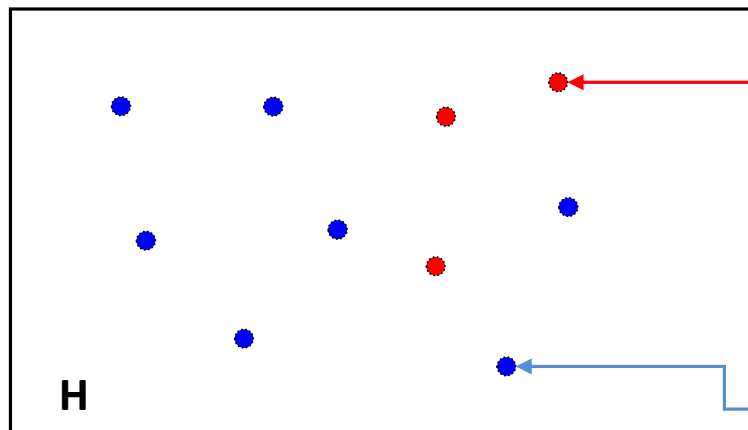
$$VS_{H,T} \equiv \{h \in H \mid \text{Konzistentna}(h, T)\}$$

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  **$T$**

# $\varepsilon$ - iscrpivost/kompletnost $VS_{H,T}$ ( $\varepsilon$ - exhausted $VS_{H,T}$ )

Prostor hipoteza  $H$



$VS_{H,T}$

$$e_T(h) = 0$$

$$e_{\Delta}(h) > 0 \text{ (stvarna greška)}$$

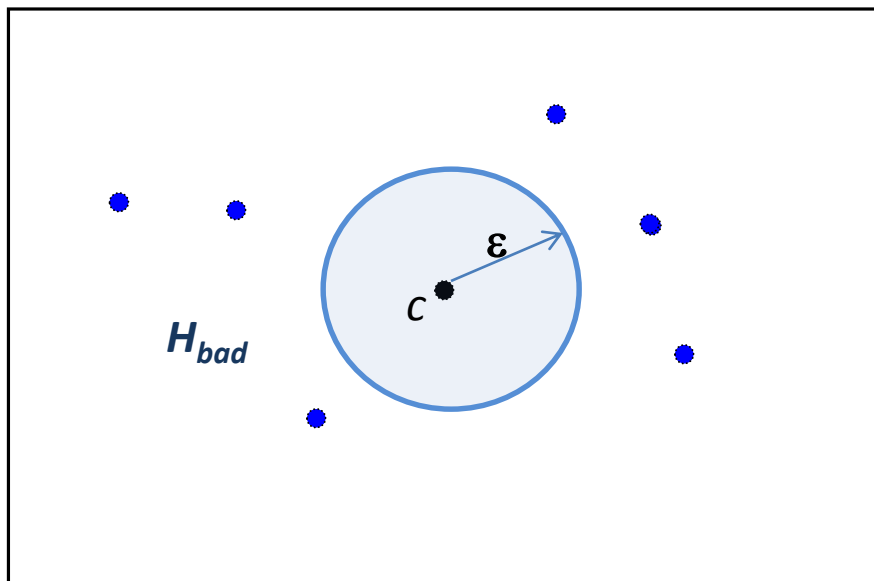
$\neg VS_{H,T}$

$$e_T(h) > 0$$

$$e_{\Delta}(h) > 0 \text{ (stvarna greška)}$$

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

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

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  $\Delta$  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 primjera  $x_i$  iz  $\Delta$  vrijedi:

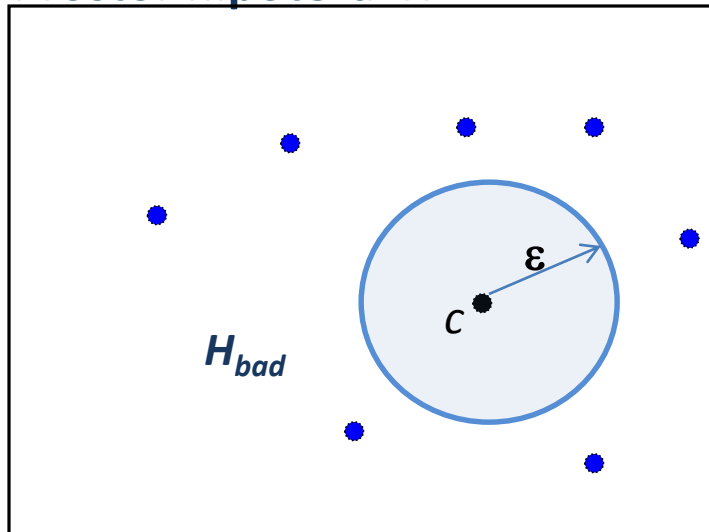
$$p(e_{\Delta}(h_{bad}) > \varepsilon, h_{bad}(x_i) = f(x_i), \forall x_i \in \{x_1, \dots, x_N\}) \leq (1-\varepsilon)^N$$

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

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



## Prostor hipoteza $H$



Želimo da ova vjerojatnost bude što manja:

$$p(e_{\Delta}(h) > \varepsilon \mid e_{\{x_1, \dots, x_N\}}(h) = 0) \leq |H|(1 - \varepsilon)^N$$

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

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

$$p(e_{\Delta}(h) > \varepsilon \mid e_{\{x_1, \dots, x_N\}}(h) = 0) \leq |H|e^{-\varepsilon N}$$

ovu ćemo vjerojatnost  $\delta$  postići nakon učenja na  $N$  primjera :

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

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

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

- Veličina  $|\mathbf{H}|$  najviše utječe na  $N$
- $|\mathbf{H}|$  zavisi o “jeziku” kojim je opisan prostor primjera
- Za prostor primjera s  $n$  binarnih varijabli opisan konjunkcijama Boole-ovih literala ( npr.  $h_i = \langle \text{oblačno}, \text{vlažno}, \#, \# \rangle$ )
- $|\mathbf{H}| = ?$

## Učenje konjunkcija Boole-ovih literala

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

npr. prognoza - (vedro/oblačno/#).

Tada vrijedi za  $n$  varijabli:

$$|H| = 3^n$$

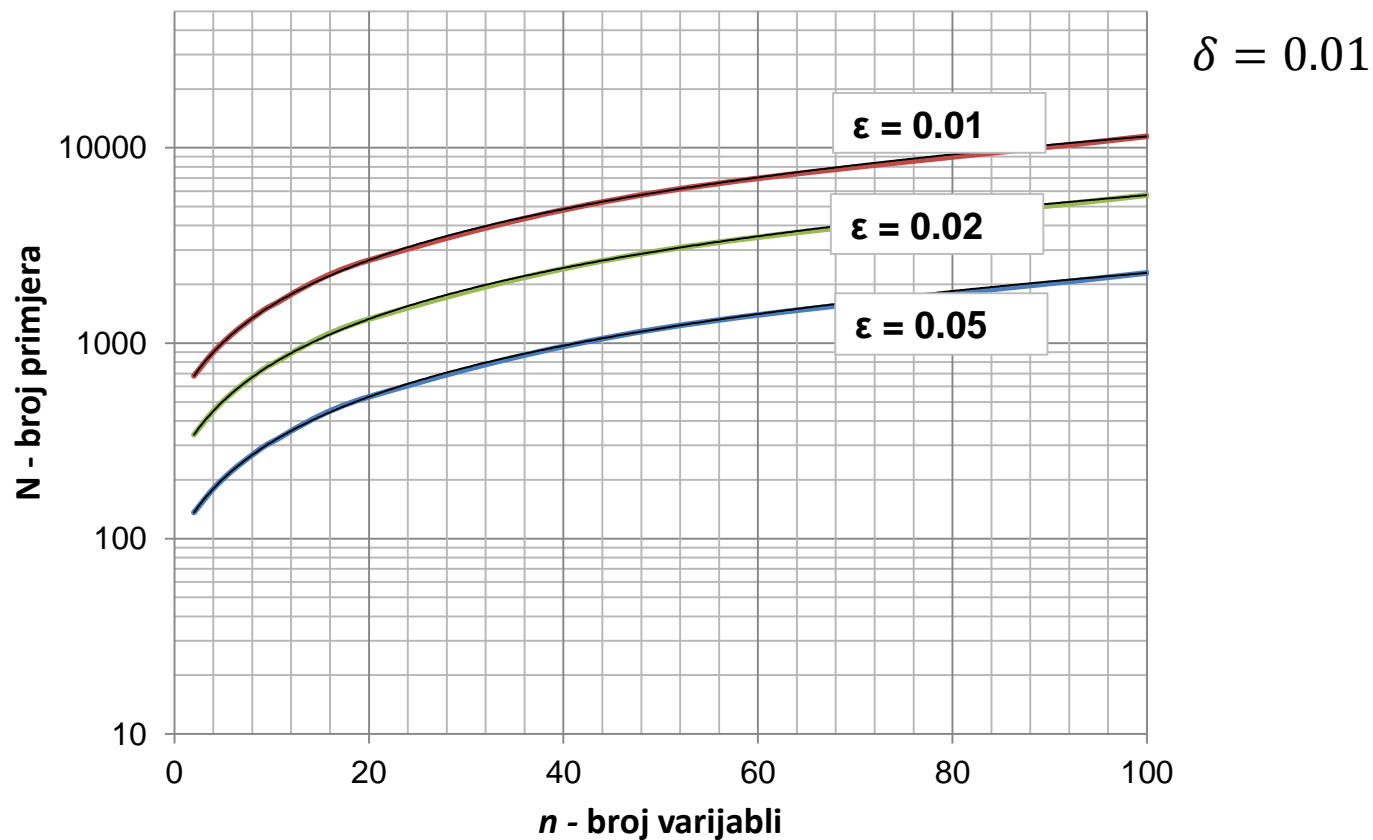
pa je broj potrebnih primjera:

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

Za  $\varepsilon = 0.05$ ;  $\delta = 0.01$ ;  $n = 10$

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

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



## PAC učenje - kad vrijedi $c \in H$

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

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 (**disjunkcije, negacije, konjunkcije**) ?

$$|C| = ?$$

$$|C| = 2^{3^n} \quad !!$$

$$\text{želimo li } c \in H \Rightarrow N \geq \frac{1}{\varepsilon} \left( \ln \frac{1}{\delta} + 3^n \ln 2 \right)$$

## PAC učenje - kad vrijedi $c \in H$

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

Kako ograničiti broj potrebnih primjera ?

### Osnovna dilema:

- a) 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 ?
- ❑ U tom slučaju - možemo samo garantirati (za bilo koju  $h$ !):

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

- ❑ moramo garantirati da ovo vrijedi za svaku iz  $H$  - pa onda i za najbolju!

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

- ❑ ako gornju vjerojatnost proglasimo  $\delta$  možemo opet doći do toga koliko nam promjera treba:

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

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

**Napomena:  $\varepsilon$  je sada samo razlika između  $e_{\Delta}(h)$  i  $e_T(h)$**

## PAC učenje - sažetak

□ Konačni prostor  $H$ , i  $c \in H$ , vrijedi:

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

pa možemo garantirati da je za točnost  $\varepsilon$  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

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

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

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



## PAC učenje - problem određivanja broja primjera kad $|H| \rightarrow \infty$

### Osnovna dilema:

- a) Napraviti restrikcije na prostor hipoteza  $H$  (restrikcija jezika, reprezentabilnih funkcija) = konačni  $H \Rightarrow$  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**

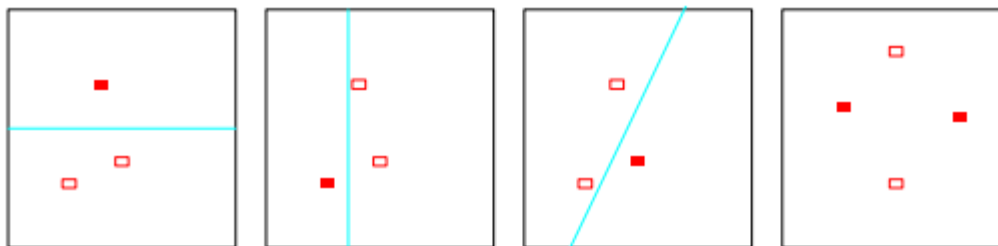
(Intuitivno: Kompleksniji skup funkcija  $H$  može **particionirati** veći skup  $S$ !)

## Definicija (Vapnik-Chervonenkis dimenzija skupa hipoteza $H$ - $VC(H)$ )

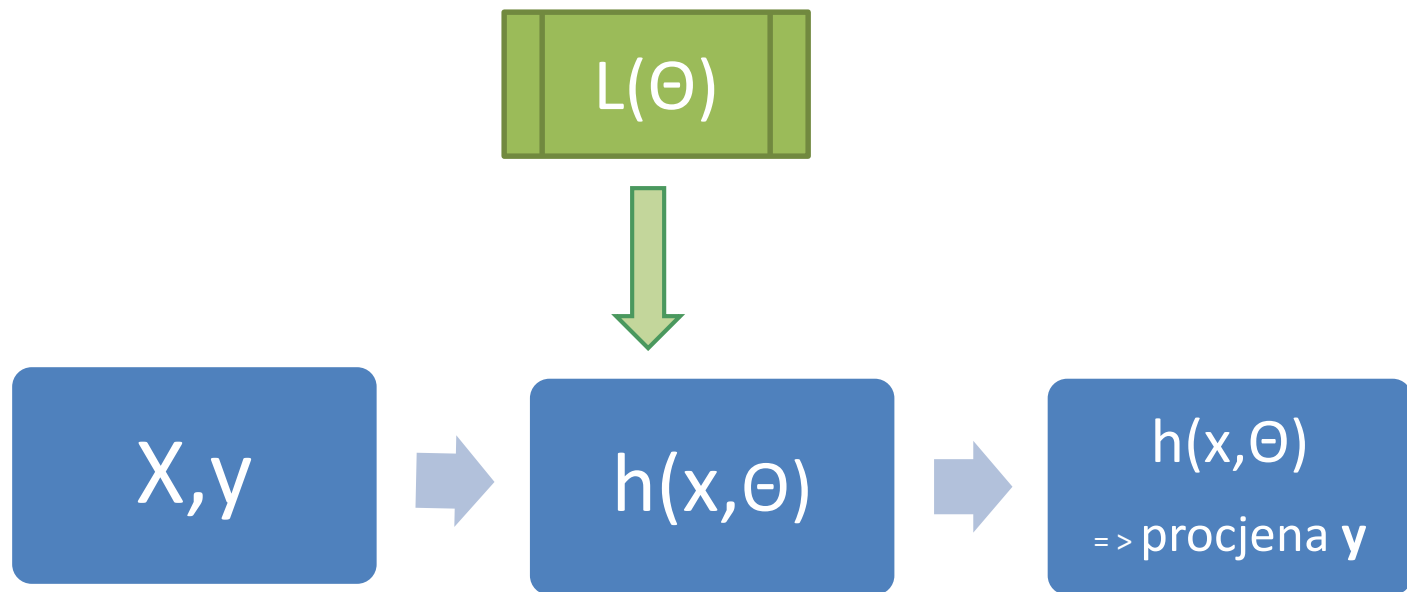
$VC(H)$  prostora hipoteza  $H$ , definiranog preko prostora primjera  $X$  je **veličina najvećeg podskupa od  $X$**  koji je moguće particionirati/rastaviti korištenjem  $H$ . Ako je pomoću  $H$  moguće rastaviti po volji velike podskupove  $X$ , tada vrijedi  $VC(H) \equiv \infty$ .

( $h \in H$  mogu generirati bilo koju klasifikaciju na skupu primjera  $S$ )

Primjer (ESL, ch7): pravci i primjeri ( $X$ ) u ravnin,  $VC=??$




$h(x, \Theta)$  predstavlja neku funkciju koju možemo naučiti algoritmom  $L$



H kao proizvod algoritma strojnog učenja....

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

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

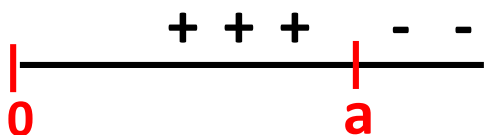


Postoji  $2^n$  takvih dihotomija, s različitim kombinacijama  $y \in \{+1, -1\}$

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

Polu-intervali:

$h(x, \Theta) \in H$  ;  $H$  - intervali tipa  $[0, a)$ , za neki realni  $a > 0$



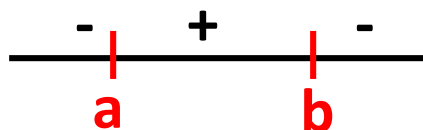
$VC(h(x, \Theta)) = ?$

$|H| = ?$

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

### Intervali

$h(x, \Theta) \in H$  ;  $H$  - intervali na realnoj osi -  $[a, b]$  |  $b > a$



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

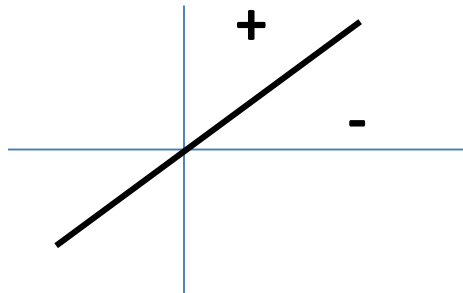
$|H| = ?$

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

### Poluprostori - 1

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

$$h(x, \Theta) = \text{sign}(x \cdot \Theta)$$



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

$$|H| = ?$$

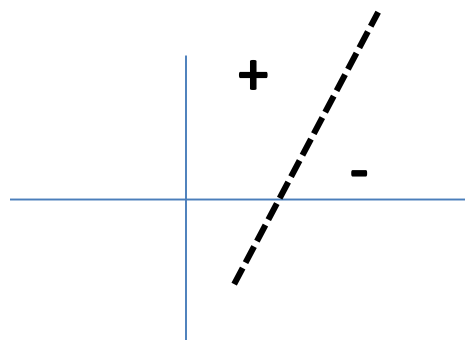


## Primjeri $h(\mathbf{x}, \Theta)$ i određivanja $VC(h)$

### Poluprostori - 2

$$h(\mathbf{x}, \Theta) \in H$$

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



$$VC(h(\mathbf{x}, \Theta)) = ?$$

$$|H| = ?$$

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

$$\underline{VC(\text{Poluintervali}) = 1}$$

$$\underline{VC(\text{Intervali}) = 2}$$

$$\underline{VC(\text{Poluprostori } 1) = 2}$$

(**podskup veličine 2** se ne može rastaviti)

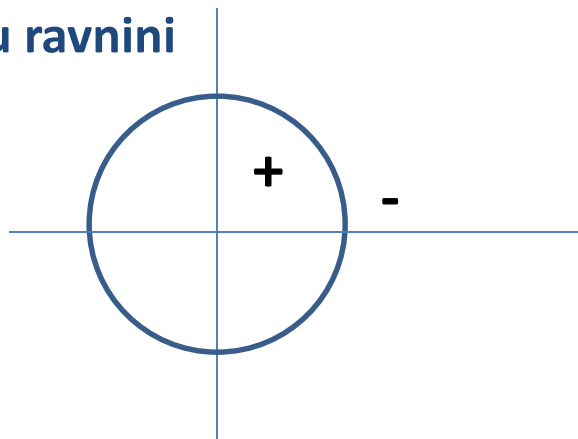
(**podskup veličine 3** se ne može rastaviti)

(**podskup veličine 3** se ne može rastaviti)

## Drugi primjeri $h(\mathbf{x}, \Theta)$ , $VC(h)$ - za vježbu ?

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

$$h(\mathbf{x}, \Theta) = \text{sign}(\Theta_1 \cdot x_1 - \Theta_2)$$



.....

- $VC(\mathbf{w}^* \mathbf{x} - \mathbf{b})$  – gdje je  $\mathbf{x}$  –  $n$  dimenzionalni prostor
- $VC(\sin(x))$
- $VC(\text{stabla odlučivanja})$
- $VC(\text{perceptron})$
- $VC(\text{neuralne mreže})$

---

Određivanje VC dimenzije je zahtjevno za kompleksnije prostora hipoteza

**Gornja granica broja primjera**

**PAC učenje – uz korištenje  $VC(H)$   $\Rightarrow$  Structural Risk Minimization (SRM)**

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

**Osim toga Vapnik je pokazao da s vjerojatnošću  $(1-\eta)$  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 algoritama („strojeva” )











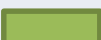


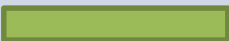
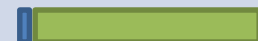
– koji uče hipoteze iz prostora  $H_i$  (funkcije) različitih  $VC(H_i)$  tako da vrijedi:

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

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 | $H_i$ | $e_T$   | $\sqrt{VC(H) \dots}$  | $\sim e_{test}$   | 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

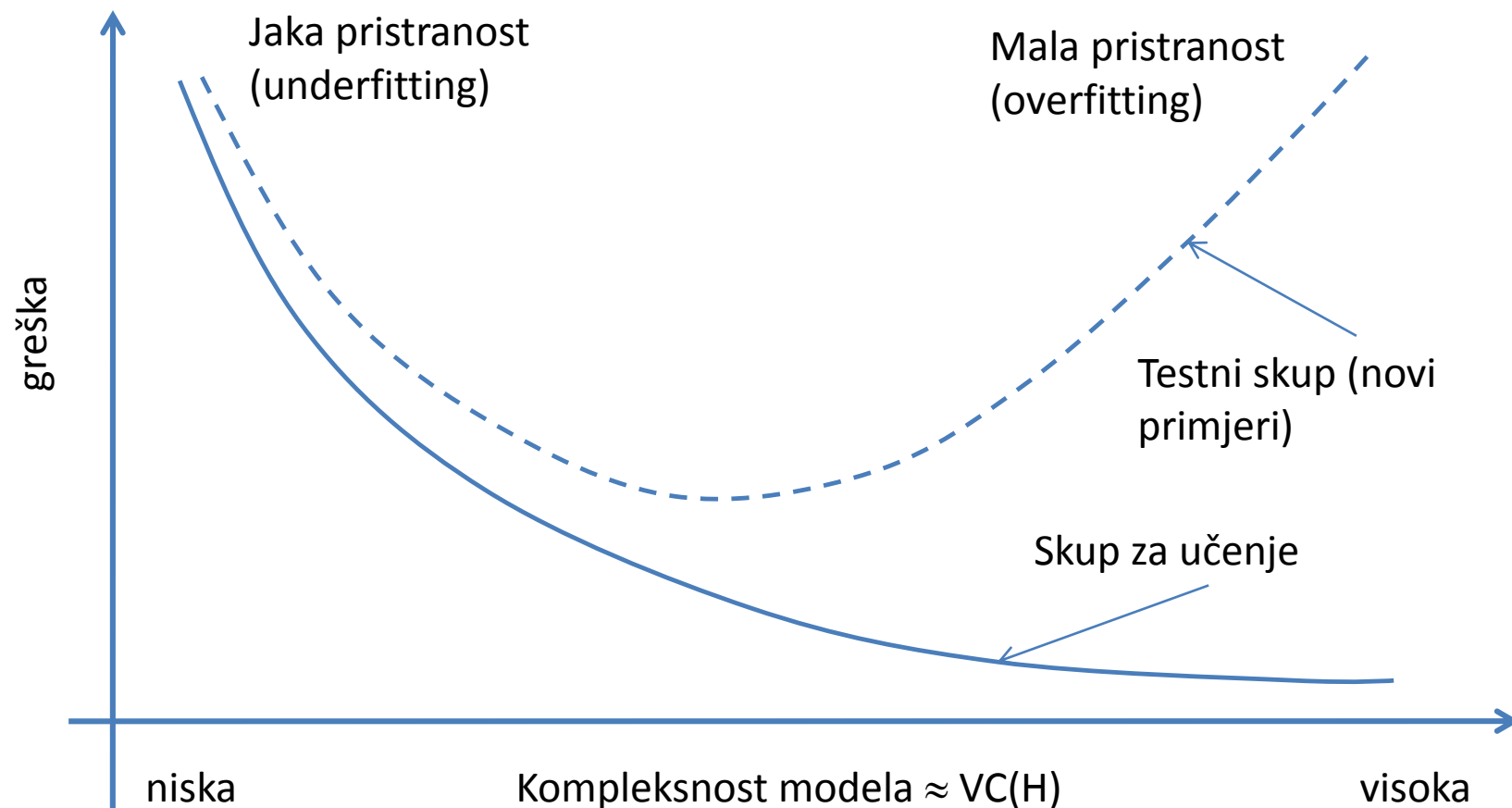
Trebalo bi zapamtiti:

- Shattering
  - definiciju VC-dimenzije
  - neke od primjera  $H$  i  $VC(H)$
  - SRM i čemu služi
- 
- $VC(H) = \infty \Rightarrow$  drugi pristupi ocjeni greške
    - Efektivni VC
    - unakrsna validacija

## SRM i sposobnost generalizacije algoritma

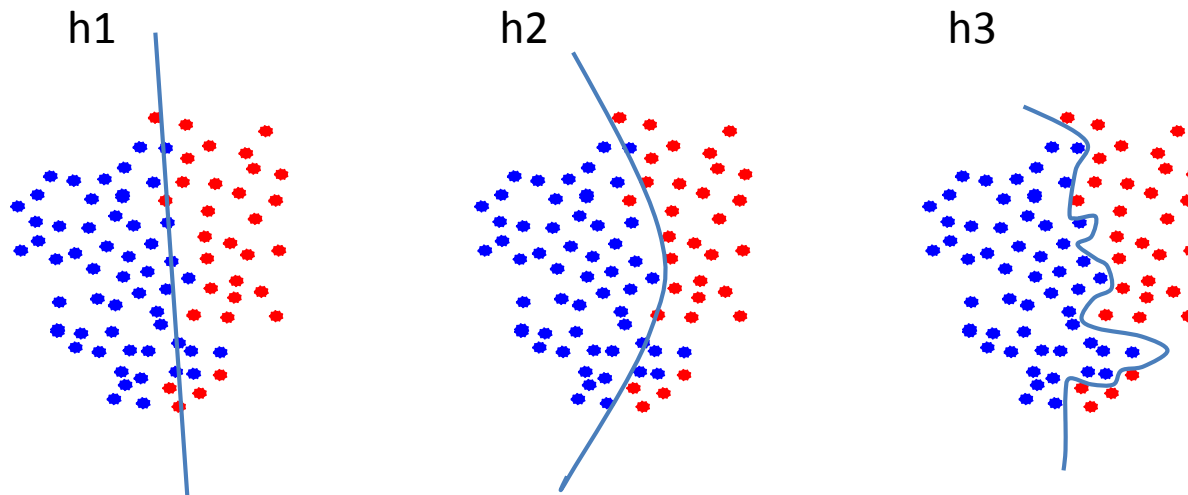
- **generalizacijom algoritma** možemo nazvati kvalitetu predikcije na novim (testnim) podacima
- naš osnovni cilj je dobiti dobra svojstva generalizacije naučenim modelom podataka
- kad je **model pre-kompleksan** za neki skup podataka – može se desiti da uči ili memorira dijelove “šuma” ili grešaka, pored stvarne strukture podataka – **pretreniranje (en. overfitting, high 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, high model bias)**

## Tipično ponašanje

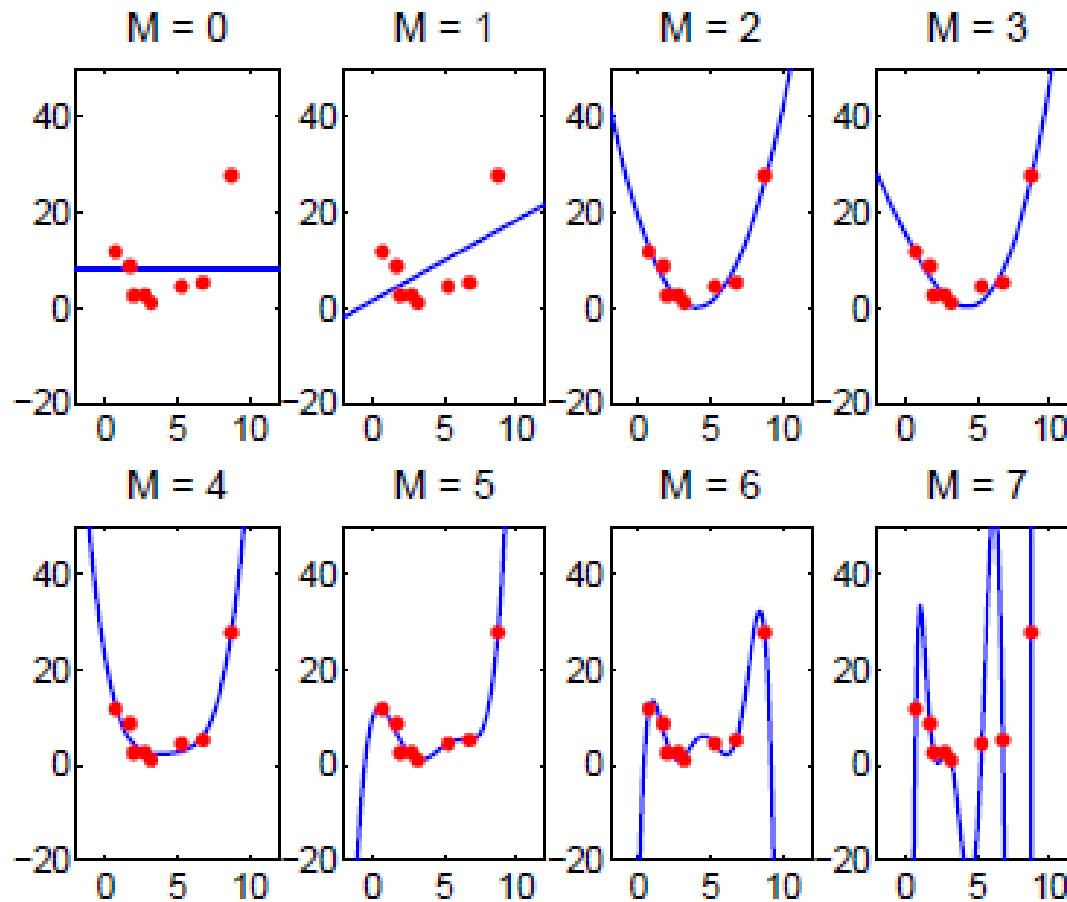




## Primjer - klasifikacija



## Primjer - regresija



## Slijedeća predavanja

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