



# Teoria systemów uczących się i wymiar Vapnika-Chervonenkisa

*Hung Son Nguyen*

Wydział Matematyki, Informatyki i Mechaniki  
Uniwersytet Warszawski  
email: `son@mimuw.edu.pl`

19.04.2016

# Plan wykładu

---



- 1 Wstęp do komputerowego uczenia się pojęć
- 2 Model PAC (probably approximately correct)
- 3 Wyuczalność klasy pojęć
- 4 Przykład modelu PAC i niesprzecznego algorytmu
- 5 Wymiar Vapnika Chervonenkisa (VC dimension)
- 6 Podstawowe twierdzenia teorii uczenia się
- 7 Appendix: „Nie ma nic za darmo” czyli “Non Free Lunch Theorem”



- Np. Pokazać, że dla każdego  $n \in \mathbb{N}$  zachodzi

$$\Psi(n) : \boxed{1^2 + 2^2 + \dots + n^2 = \frac{n(n+1)(2n+1)}{6}}$$

- Indukcja pełna:

$$\Psi(1) \quad \text{oraz} \quad \forall_{n \geq 1} [\Psi(n) \implies \Psi(n+1)]$$

- Indukcja niepełna: czy wystarczy sprawdzić, np.

$$\Psi(1), \Psi(2), \Psi(3), \Psi(4)?$$



## Podjęcie indukcyjne:

Wnioskowanie na podstawie skończonego zbioru obserwacji

**Jakie prawa rządzą procesem indukcyjnego uczenia się pojęć?**

Szukamy teorii obejmującej zagadnienia:

- Szansy na skuteczne wyuczanie się pojęć;
- Niezbędnej liczby przykładów treningowych;
- Złożoności przestrzeni hipotez;
- Jakości aproksymacji;
- Metod reprezentacji danych treningowych;



## ■ Niech

- $\mathcal{X}$  – (skończony lub nieskończony) zbiór obiektów;
- $\mathbb{C}$  – klasa pojęć w  $\mathcal{X}$ , tj.  $\mathbb{C} = \{f : \mathcal{X} \rightarrow \{0, 1\}\}$
- $c \in \mathbb{C}$  – pojęcie docelowe lub funkcja celu;

## ■ Dane są

- skończona próbka etykietowanych obiektów:

$$D = \{\langle x_1, c(x_1) \rangle, \dots, \langle x_m, c(x_m) \rangle\} \in \mathcal{S}(m, c)$$

gdzie  $x_1, \dots, x_m \in \mathcal{X}$ .

- przestrzeń hipotez  $\mathbb{H} = \{h : \mathcal{X} \rightarrow \{0, 1\}\}$ ;

## ■ Szukana

- hipoteza  $h \in \mathbb{H}$  będąca dobrą aproksymacją pojęcia  $c$ .

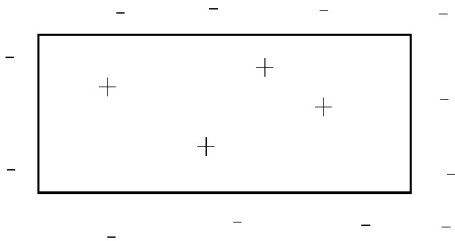
## ■ Wymagane

- dobra jakość aproksymacji
- szybki czas wyuczania.

# Przykład



- Pojęcie: "człowieka o średniej budowie ciała".
- Dane – czyli osoby – są reprezentowane przez ich wagę(*kg*) i wzrost(*cm*) i są etykietowane przez + i -.
- Dodatkowa wiedza: szukane pojęcie można wyrazić za pomocą PROSTOKĄTA



## Uczenie prostokąta

- $\mathcal{X} = \mathbb{R}^2$ ;
- $\mathbb{C} = \mathbb{H} =$  zbiór prostokątów;
- Przykład zbioru treningowego  
((84, 184), +),  
((70, 170), +),  
((75, 163), -),  
((80, 180), +),  
((81, 195), -),  
((63, 191), -),  
((77, 187), -),  
((68, 168), +)  
■ ((79, 183, ?))



- **Uczenie półosi (lub dyskretyzacji):**

$$\mathcal{X} = \mathbb{R}; \quad \mathbb{C} = \mathbb{H} = \{[\lambda, \infty) : \lambda \in \mathbb{R}\}$$

- **Uczenie hiperpłaszczyzny:**

$$\mathcal{X} = \mathbb{R}^n; \quad \mathbb{H} = \{f_{w_0, w_1, \dots, w_n} : \mathbb{R}^n \rightarrow \{0, 1\}\}$$

gdzie  $f_{w_0, \dots, w_n}(x_1, \dots, x_n) = \text{sgn}(w_0 + w_1x_1 + \dots + w_nx_n)$ .

- **Uczenie jednomianów Boolowskich:**

$$\mathcal{X} = \{0, 1\}^n; \quad c : \{0, 1\}^n \rightarrow \{0, 1\};$$

$\mathbb{H} = \mathbf{M}_n =$  zbiór jednomianów Boolowskich o  $n$  zmiennych.

## Błąd rzeczywisty

- $\Omega = (\mathcal{X}, \mu)$  – przestrzeń probabilistyczna na  $\mathcal{X}$ ;
- Błąd hipotezy  $h \in \mathbb{H}$  względem funkcji celu  $c$ :

$$er_{\Omega}(h, c) = er_{\Omega}^c(h) = \mu(\mathcal{X}_{h \neq c})$$

gdzie  $\mathcal{X}_{h \neq c} = \{x \in \mathcal{X} : h(x) \neq c(x)\}$ .

**Statystyka:** Jeśli przykłady z  $D$  są wybrane zgodnie z miarą prawdopodobieństwa  $\mu$  w sposób niezależny oraz  $|D| \geq 30$ , to

- $er_{\Omega}^c(h) \approx er_D^c(h) = \frac{|D \cap \mathcal{X}_{h \neq c}|}{|D|}$ ,
- z prawdopodobieństwem  $(1 - \varepsilon)$

$$|er_{\Omega}^c - er_D^c| \leq s_{\frac{\varepsilon}{2}} \cdot \sqrt{\frac{er_D^c(1 - er_D^c)}{|D|}}$$



# Plan wykładu

---



- 1 Wstęp do komputerowego uczenia się pojęć
- 2 Model PAC (probably approximately correct)
- 3 Wyuczalność klasy pojęć
- 4 Przykład modelu PAC i niesprzecznego algorytmu
- 5 Wymiar Vapnika Chervonenkisa (VC dimension)
- 6 Podstawowe twierdzenia teorii uczenia się
- 7 Appendix: „Nie ma nic za darmo” czyli “Non Free Lunch Theorem”

# Model uczenia się PAC

Idea modelu PAC (Probably Approximately Correct):

Określenie warunków, przy których uczeń (algorytm uczenia się) z „dużym prawdopodobieństwem” znajdzie „dobrą hipotezę” na podstawie danych  $D$ .

PAC-owy uczeń

Niech  $\mathcal{L}$  będzie algorytmem uczenia się, jeśli

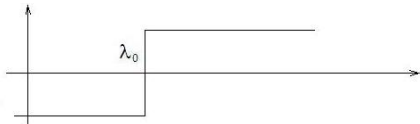
dla każdego  $0 < \varepsilon, \delta < 1$ , istnieje liczba  $m_0 = m_0(\varepsilon, \delta)$  taka, że dla dowolnego pojęcia  $c \in \mathbb{C}$ , dla dowolnego rozkładu  $\Omega$  na  $X$  i dla  $m > m_0$  mamy

$$\mu^m \{D \in \mathcal{S}(m, c) : er_{\Omega}(\mathcal{L}(D)) < \varepsilon\} > 1 - \delta$$

Wówczas mówimy w skrócie, że  $\mathcal{L}$  jest **PAC** dla klasy  $\mathbb{C}$  (“prawdopodobnie aproksymacyjnie poprawny”).

$\varepsilon$  = dopuszczalny poziom błędów;  $(1 - \delta)$  = poziom zaufania.

- $\mathbb{H} = \mathbb{C} = \{f_\lambda : \mathfrak{R} \rightarrow \{0, 1\} : f_\lambda(x) = 1 \Leftrightarrow x \geq \lambda\}$
- $c = f_{\lambda_0}$



- znaleźć  $\lambda_0$  na podstawie losowo wygenerowanych przykładów  $D = \{\langle x_1, f_{\lambda_0}(x_1) \rangle, \dots, \langle x_m, f_{\lambda_0}(x_m) \rangle\}$

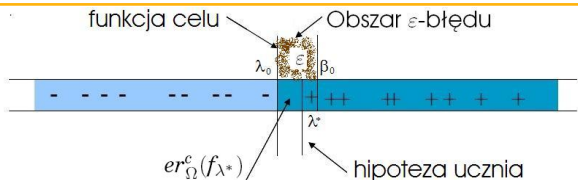
### Algorytm:

- 1 Set  $\lambda^* := \min_{i \in \{1, \dots, m\}} \{x_i : f_{\lambda_0}(x_i) = 1\}$ ;
- 2  $\mathcal{L}(D) := f_{\lambda^*}$ ;

### Twierdzenie:

Powyższy algorytm jest PAC

# Dowód



- $er_{\Omega}^c(f_{\lambda^*}) = \mu([\lambda_0, \lambda^*])$ .
- Niech  $\beta_0 = \sup\{\beta : \mu([\lambda_0, \beta]) < \epsilon\}$ .
- Wówczas  $er_{\Omega}^c(f_{\lambda^*}) \geq \epsilon \Leftrightarrow \forall x_i \in D : x_i \notin [\lambda_0, \beta_0]$ ;
- Stąd

$$\mu^m \{(x_1, \dots, x_m) : \forall x_i \in D : x_i \notin [\lambda_0, \beta_0]\} \leq (1 - \epsilon)^m$$
$$\mu^m \{D \in \mathcal{S}(m, f_{\lambda_0}) : er_{\Omega}(f_{\lambda^*}) \leq \epsilon\} \geq 1 - (1 - \epsilon)^m$$

- Aby to prawdopodobieństwo było  $> 1 - \delta$ , wystarczy przyjąć  $m \geq m_0 = \left\lceil \frac{1}{\epsilon} \ln \frac{1}{\delta} \right\rceil$



- Niech  $\Omega$  będzie rozkładem dyskretnym zdefiniowanym przez  $\mu_1 = \mu(x_1), \dots, \mu_n = \mu(x_n)$  – dla pewnych  $x_1, \dots, x_n \in X$  – takich, że  $\mu_1 + \dots + \mu_n = 1$ . Niech  $\varepsilon_{\min} = \min_i \mu_i$ .
- Jeśli  $\mathcal{L}$  jest PAC, i jeśli  $\varepsilon \leq \varepsilon_{\min}$  to warunek  $er_{\Omega}^c(\mathbf{L}(\mathbf{D})) < \varepsilon$  jest równoważny z  $er_{\Omega}^c(\mathbf{L}(\mathbf{D})) = 0$ . Stąd dla każdego  $\delta$ , istnieje  $m_0 = m_0(\varepsilon_{\min}, \delta)$  taka, że dla dowolnego  $c \in \mathbb{C}$  i  $\Omega$

$$m > m_0 \Rightarrow \mu^m \{ \mathbf{D} \in \mathcal{S}(m, t) \mid er_{\Omega}(\mathbf{L}(\mathbf{D})) = 0 \} > 1 - \delta$$

- Wówczas mówimy, że prawdopodobnie  $\mathcal{L}$  jest dokładnym algorytmem (*jest PEC – probably exactly correct*)

# Plan wykładu

---



- 1 Wstęp do komputerowego uczenia się pojęć
- 2 Model PAC (probably approximately correct)
- 3 Wyuczalność klasy pojęć
- 4 Przykład modelu PAC i niesprzecznego algorytmu
- 5 Wymiar Vapnika Chervonenkisa (VC dimension)
- 6 Podstawowe twierdzenia teorii uczenia się
- 7 Appendix: „Nie ma nic za darmo” czyli “Non Free Lunch Theorem”

# Potencjalna wyuczalność

- Niech  $D = \{\langle x_1, c(x_1) \rangle, \dots, \langle x_m, c(x_m) \rangle\}$  i niech

$$\mathbb{H}^c(D) = \{h \in \mathbb{H} : h|D = c|D\}$$

zbiór hipotez zgodnych z  $c$  na próbce  $D$ .

- $\mathbb{B}_\varepsilon^c = \{h \in \mathbb{H} : er_\Omega(h) \geq \varepsilon\}$  – zbiór  $\varepsilon$ -złych hipotez

## Definicja: Potencjalna wyuczalność

Mówimy, że  $\mathbb{C}$  jest potencjalnie wyuczalna za pomocą  $\mathbb{H}$ , jeśli dla każdego rozkładu  $\Omega$  na  $\mathcal{X}$  i dowolnego pojęcia  $c \in \mathbb{C}$  oraz dla dowolnych  $0 < \varepsilon, \delta < 1$  istnieje  $m_0 = m_0(\varepsilon, \delta)$  takie, że

$$m \geq m_0 \Rightarrow \mu^m \{D \in \mathcal{S}(m, c) : \mathbb{H}^c(D) \cap \mathbb{B}_\varepsilon^c = \emptyset\} > 1 - \delta$$

# Potencjalna wyuczalność

Algorytm  $\mathcal{L}$  nazywamy **niesprzecznym** jeśli  $\mathcal{L}(D) \in \mathbb{H}^c(D)$  dla każdego zbioru  $D$ .

## Twierdzenie

W przestrzeni potencjalnie wyuczalnej, każdy wzorowy uczeń (niespreczny algorytm) jest PAC-owy.

## Twierdzenie (Haussler, 1988)

Jeśli  $\mathbb{C} = \mathbb{H}$  i  $|\mathbb{C}| < \infty$ , to  $\mathbb{C}$  jest potencjalnie wyuczalna.

Dowód: Niech  $h \in \mathbb{B}_\varepsilon$  (tzn.  $er_\Omega(h) \geq \varepsilon$ ). Wówczas

$$\mu^m \{D \in \mathcal{S}(m, c) : er_D(h) = 0\} \leq (1 - \varepsilon)^m$$

$$\Rightarrow \mu^m \{D : \mathbb{H}^c(D) \cap \mathbb{B}_\varepsilon \neq \emptyset\} \leq |\mathbb{B}_\varepsilon| (1 - \varepsilon)^m \leq |\mathbb{H}| (1 - \varepsilon)^m$$

Aby  $|\mathbb{H}| (1 - \varepsilon)^m < \delta$  wystarczy wybrać  $m \geq m_0 = \left\lceil \frac{1}{\varepsilon} \ln \frac{|\mathbb{H}|}{\delta} \right\rceil$





- 1 Initialize the hypothesis to the conjunction of all  $2n$  literals.

$$h = x_1 \bar{x}_1 x_2 \bar{x}_2 \dots x_n \bar{x}_n$$

- 2 For each training instance
  - if it is a positive example:  
remove  $x_i$  from  $h$  if  $x_i = 0$  and remove  $\bar{x}_i$  from  $h$  if  $x_i = 1$ ;
  - if it is a negative example:  
do nothing;
- 3 Output the remaining hypothesis.



```
set h = disjunction of all monomials of length at most k ;  
for  $i := 1$  to  $m$  do  
    if  $b_i = 0$  and  $h(x_i) = 1$   
        then delete monomials  $\mu$  for which  $\mu(x_i) = 1$ ;  
 $L(\mathbf{s}) := h$ 
```


# Plan wykładu

---



- 1 Wstęp do komputerowego uczenia się pojęć
- 2 Model PAC (probably approximately correct)
- 3 Wyuczalność klasy pojęć
- 4 Przykład modelu PAC i niesprzecznego algorytmu
- 5 Wymiar Vapnika Chervonenkisa (VC dimension)
- 6 Podstawowe twierdzenia teorii uczenia się
- 7 Appendix: „Nie ma nic za darmo” czyli “Non Free Lunch Theorem”

# Przykład: Uczenie jednomianów Boolowskich



Funkcje	000	001	010	011	100	101	110	111
0	0	0	0	0	0	0	0	0
x1	0	0	0	0	1	1	1	1
x2	0	0	1	1	0	0	1	1
x3	0	1	0	1	0	1	0	1
-x1	1	1	1	1	0	0	0	0
-x2	1	1	0	0	1	1	0	0
-x3	1	0	1	0	1	0	1	0
x1x2	0	0	0	0	0	0	1	1
	1	1	0	0	0	0	0	0
	0	0	0	0	1	1	0	0
	0	0	1	1	0	0	0	0
x1x3	0	0	0	0	0	1	0	1
	0	0	0	0	1	0	1	0
	0	1	0	1	0	0	0	0
	1	0	1	0	0	0	0	0
x2x3	0	0	0	1	0	0	0	1
	0	0	1	0	0	0	1	0
	0	1	0	0	0	1	0	0
	1	0	0	0	1	0	0	0
x1x2x3	0	0	0	0	0	0	0	1
	0	0	0	0	0	0	1	0
	0	0	0	0	0	1	0	0
	0	0	0	0	1	0	0	0
	0	0	0	1	0	0	0	0
	0	0	1	0	0	0	0	0
	0	1	0	0	0	0	0	0
	1	0	0	0	0	0	0	0

- $\mathbb{X} = \{0, 1\}^3$ ;
- $\mathbb{M}_3 =$  zbiór jednomianów
- **Gracz 1 (nauczyciel):**  
Wybrał jeden z jednomianów, np.  $c = \bar{x}_2$ .
- **Gracz 2 (uczeń):**  
Dostaje próbkę przykładów, np.

$D$	$dec$
011	0
100	1
101	1

i musi podać funkcję  $h$  podobną do  $c$ .

## Algorytm $\mathcal{L}$

Dane

$D$	$dec$
011	0
100	1
101	1

- $h_0 = x_1x_2x_3\bar{x}_1\bar{x}_2\bar{x}_3$
- $h_1 = x_1x_2x_3\bar{x}_1\bar{x}_2x_3$
- $h_2 = x_1x_2x_3\bar{x}_1x_2\bar{x}_3$
- $h_3 = x_1x_2x_3\bar{x}_1x_2x_3$   
 $h = \mathcal{L}(D) = x_1\bar{x}_2$

- Jak dobry jest algorytm  $\mathcal{L}$ ?
- Sprawdźmy:

	$c = \bar{x}_2$	$h = x_1\bar{x}_2$	$h = c$
000	1	0	-
001	1	0	-
010	0	0	+
011	0	0	+
100	1	1	+
101	1	1	+
110	0	0	+
111	0	0	+

- $er_c(\mathcal{L}(D)) = er_c(h) = 0,25 = 25\%$

# Przykład: Uczenie jednomianów Boolowskich



Funkcje	000	001	010	011	100	101	110	111
0	0	0	0	0	0	0	0	0
x1	0	0	0	0	1	1	1	1
x2	0	0	1	1	0	0	1	1
x3	0	1	0	1	0	1	0	1
-x1	1	1	1	1	0	0	0	0
-x2	1	1	0	0	1	1	0	0
-x3	1	0	1	0	1	0	1	0
x1x2	0	0	0	0	0	0	1	1
	1	1	0	0	0	0	0	0
	0	0	0	0	1	1	0	0
	0	0	1	1	0	0	0	0
x1x3	0	0	0	0	0	1	0	1
	0	0	0	0	1	0	1	0
	0	1	0	1	0	0	0	0
	1	0	1	0	0	0	0	0
x2x3	0	0	0	1	0	0	0	1
	0	0	1	0	0	0	1	0
	0	1	0	0	0	1	0	0
	1	0	0	0	1	0	0	0
x1x2x3	0	0	0	0	0	0	0	1
	0	0	0	0	0	0	1	0
	0	0	0	0	0	1	0	0
	0	0	0	0	1	0	0	0
	0	0	0	1	0	0	0	0
	0	0	1	0	0	0	0	0
	0	1	0	0	0	0	0	0
	1	0	0	0	0	0	0	0

- Dla próbki

$$D_1 = \{(010, 0), (011, 0), (110, 0)\}$$

mamy  $\mathcal{L}(D_1) = 0$ , czyli  
 $er_c(\mathcal{L}(D_1)) = 0,5 = 50\%$

- Dla próbki

$$D_2 = \{(000, 1), (011, 0), (010, 0)\}$$

mamy  $\mathcal{L}(D_2) = \bar{x}_1\bar{x}_2\bar{x}_3$ , czyli  
 $er_c(\mathcal{L}(D_2)) = 0,375 = 37,5\%$


- Dla próbki

$$D_3 = \{(001, 1), (111, 0), (100, 1)\}$$

mamy  $\mathcal{L}(D_3) = \bar{x}_2$ , czyli  
 $er_c(\mathcal{L}(D_3)) = 0$

# Przykład: Uczenie jednomianów Boolowskich

Dla różnych próbek,  
różne są błędy dla  $\mathcal{L}$ : Liczba takich próbek



Próbka	$er_c(\mathcal{L}(\cdot))$
$D_3$	0
...	...
$D$	0.25
...	...
$D_2$	0.375
...	...
$D_1$	0.5
...	...

$$|\mathcal{S}(3)| = \binom{8}{3} = 56$$

w tym:

- $\binom{4}{3} = 4$  próbek z trzema "1";
  - $\binom{4}{1} \binom{4}{2} = 24$  próbek z dwiema "1";
  - $\binom{4}{2} \binom{4}{1} = 24$  próbek z jedną "1";
  - $\binom{4}{3} = 4$  próbek bez "1";
- 
- $P(\{D \in \mathcal{S}(3) : er_c(\mathcal{L}(D)) < 0.4\}) = 52/56 = 92,9\%$
  - $P(\{D \in \mathcal{S}(3) : er_c(\mathcal{L}(D)) < 0.3\}) = 28/56 = 0.5 = 50\%$
  - $P(\{D \in \mathcal{S}(3) : er_c(\mathcal{L}(D)) = 0\}) = 21,4\%$

Dla  $m = 4$ :

- Liczba próbek:

$$|\mathcal{S}(4)| = \binom{8}{4} = 70$$

w tym:

- $\binom{4}{4} + \binom{4}{3} \binom{4}{1} = 17$  próbek z co najmniej trzema "1";
- $\binom{4}{2} \binom{4}{2} = 36 = 12 + 24$  próbek z dwiema "1";
- $\binom{4}{3} \binom{4}{1} = 16$  próbek z jedną "1";
- $\binom{4}{4} = 1$  próbki bez "1";
- $P(\{D \in \mathcal{S}(4) : er_c(\mathcal{L}(D)) < 0.4\}) = 69/70 = 98,6\%$
- $P(\{D \in \mathcal{S}(4) : er_c(\mathcal{L}(D)) < 0.3\}) = 53/70 = 75,7\%$
- $P(\{D \in \mathcal{S}(4) : er_c(\mathcal{L}(D)) = 0\}) = 29/70 = 41,4\%$



# Plan wykładu

---

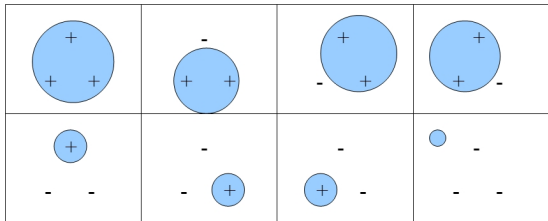


- 1 Wstęp do komputerowego uczenia się pojęć
- 2 Model PAC (probably approximately correct)
- 3 Wyuczalność klasy pojęć
- 4 Przykład modelu PAC i niesprzecznego algorytmu
- 5 Wymiar Vapnika Chervonenkisa (VC dimension)
- 6 Podstawowe twierdzenia teorii uczenia się
- 7 Appendix: „Nie ma nic za darmo” czyli “Non Free Lunch Theorem”

- Niech  $\vec{\mathbf{x}} = \langle x_1, \dots, x_m \rangle \in \mathcal{X}^m$ . Niech

$$\Pi_{\mathbb{H}}(\vec{\mathbf{x}}) = |\{\langle h(x_1), \dots, h(x_m) \rangle \in \{0, 1\}^m : h \in \mathbb{H}\}|$$

- $\Pi_{\mathbb{H}}(\vec{\mathbf{x}})$  jest liczbą podziałów zbioru elementów  $\vec{\mathbf{x}}$  wyznaczonych przez  $\mathbb{H}$ . Mamy  $\Pi_{\mathbb{H}}(\vec{\mathbf{x}}) \leq 2^m$ .
- Gdy  $\Pi_{\mathbb{H}}(\vec{\mathbf{x}}) = 2^m$ , mówimy, że  $\mathbb{H}$  **rozbija**  $\mathbf{x}$ .



- Niech  $\Pi_{\mathbb{H}}(m) = \max_{\vec{\mathbf{x}} \in \mathcal{X}^m} \Pi_{\mathbb{H}}(\vec{\mathbf{x}})$
- Na przykład:** W przypadku klasy pojęć "półosi" postaci  $[\alpha, \infty)$  mamy  $\Pi_{\mathbb{H}}(m) = m + 1$ .

## Uwagi:

- Jeśli  $\Pi_{\mathbb{H}}(m) = 2^m$ , to istnieje pewien zbiór o mocy  $m$  taki, że  $\mathbb{H}$  może definiować każdy jego podzbiór ( $\mathbb{H}$  rozbija ten zbiór).
- Maksymalna wartość  $m$ , dla której  $\Pi_{\mathbb{H}}(m) = 2^m$  można uważać za siłę wyrażalności przestrzeni  $\mathbb{H}$

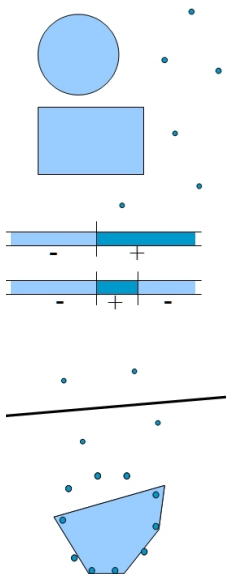
### Definicja: wymiar $VCdim$

Wymiarem Vapnika-Chervonenkisa przestrzeni hipotez  $\mathbb{H}$  nazywamy liczbę

$$VCdim(\mathbb{H}) = \max\{m : \Pi_{\mathbb{H}}(m) = 2^m\}$$

gdzie maksimum wynosi  $\infty$  jeśli ten zbiór jest nieograniczony.

# Przykłady wymiarów $VCdim$



- $H = \{\text{okręgi ...}\} \implies VC(H) = 3$
- $H = \{\text{prostokąty ...}\} \implies VC(H) = 4$
- $H = \{\text{funkcje progowe ...}\} \implies$   
 $VC(H) = 1$  jeśli "+" są zawsze po  
prawej stronie;  
 $VC(H) = 2$  jeśli "+" mogą być po obu  
stronach
- $H = \{\text{przedziały ...}\} \implies$   
 $VC(H) = 2$  jeśli "+" są zawsze w środku  
 $VC(H) = 3$  jeśli w środku mogą być  
zarówno "+" i "-"
- $H = \{\text{półpłaszczyzny w } \mathbb{R}^2 \dots\}$   
 $\implies VC(H) = 3$
- czy istnieje  $H$  dla której  $VC(H) = \infty$ ?

## Twierdzenie

Dla każdej liczby naturalnej  $n$ , niech  $P_n$  będzie perceptronem o  $n$  wejściach rzeczywistych. Wówczas

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

## Dowód:

- $VCdim(P_n) \leq n + 1$ : Wynika z Twierdzenia Radona: Dla dowolnego zbioru  $E$  zawierającego  $n + 2$  punktów w przestrzeni  $\mathbb{R}^n$  istnieje niepusty podzbiór  $S \subset E$  taki, że

$$\text{conv}(S) \cap \text{conv}(E \setminus S) \neq \emptyset$$

- $VCdim(P_n) \geq n + 1$ : Wystarczy wybrać  $\mathbf{x} = \{0, e_1, \dots, e_n\}$  i pokazać, że każdy jego podzbiór jest definiowany przez jakiś perceptron.

## Twierdzenie

1 Jeśli  $|\mathbb{H}| < \infty$  to  $VCdim(\mathbb{H}) \leq \log |\mathbb{H}|$ .

2 (**Lemat Sauer'a**) Jeśli  $VCdim(\mathbb{H}) = d \geq 0$  i  $m \geq 1$ , to

$$\Pi_{\mathbb{H}}(m) \leq 1 + \binom{m}{1} + \dots + \binom{m}{d} = \Phi(d, m)$$

3 Wniosek:  $\Phi(d, m) \leq \left(\frac{em}{d}\right)^d \Rightarrow \Pi_{\mathbb{H}}(m) \leq \left(\frac{em}{d}\right)^d$

4 Jeśli  $|\mathcal{X}| < \infty$ ,  $\mathbb{H} \subset 2^{\mathcal{X}}$  oraz  $\mathbb{H} > 1$

$$|\mathcal{X}| < \infty \implies VCdim(\mathbb{H}) > \frac{\ln |\mathbb{H}|}{1 + \ln |\mathcal{X}|}$$

# Plan wykładu

---



- 1 Wstęp do komputerowego uczenia się pojęć
- 2 Model PAC (probably approximately correct)
- 3 Wyuczalność klasy pojęć
- 4 Przykład modelu PAC i niesprzecznego algorytmu
- 5 Wymiar Vapnika Chervonenkisa (VC dimension)
- 6** Podstawowe twierdzenia teorii uczenia się
- 7 Appendix: „Nie ma nic za darmo” czyli “Non Free Lunch Theorem”



**Twierdzenie:** (Warunek konieczny)

Jeśli przestrzeń hipotez ma *nieskończony* wymiar  $VCdim$  to *nie jest potencjalnie wyuczalna*.

**Twierdzenie:** (fundamentalne)

Jeśli przestrzeń hipotez ma skończony wymiar  $VC$ , to jest ona potencjalnie wyuczalna.





1 Definiujemy

$$Q_m^\varepsilon = \{D \in \mathcal{S}(m, c) : H^c[D] \cap B_\varepsilon \neq \emptyset\}$$

2 Szukamy górnego ograniczenia  $f(m, \varepsilon)$  dla  $\mu^m(Q_m^\varepsilon)$ , które powinno

- być niezależne od  $c \in \mathbb{C}$  i  $\mu$  (rozkład).
- dążyć do 0 przy  $m \rightarrow \infty$

3 **Twierdzenie** Niech  $\mathbb{H}$  będzie przestrzenią hipotez określonych na  $X$ . Dla dowolnych  $c, \mu, \varepsilon$  (ale ustalonych) mamy

$$\mu^m(Q_m^\varepsilon) < 2\Pi_{\mathbb{H}}(2m)2^{-\varepsilon m/2}$$

o ile  $m \geq 8/\varepsilon$ .

4 Korzystamy z lematu Sauer'a, aby pokazać, że  $\mu^m(Q_m^\varepsilon) < \delta$  dla dostatecznie dużych  $m$ .



- Z Fundamentalnego Twierdzenia wynika, że jeśli  $VCdim(\mathbb{H}) < \infty$ , to dla danych  $\delta$  i  $\varepsilon$ , istnieje  $m_0 = m_0(\mathbb{H}, \delta, \varepsilon)$  takie, że

$$m \geq m_0 \Rightarrow \mu^m \{D \in \mathcal{S}(m, c) : \mathbb{H}[D] \cap \mathbb{B}_\varepsilon = \emptyset\} > 1 - \delta$$

- Wówczas każdy niesprzeczny algorytm  $\mathcal{L}$  jest PAC oraz wymagana liczba przykładów  $m_{\mathcal{L}}(\mathbb{H}, \delta, \varepsilon)$  dla  $\mathcal{L}$  jest ograniczona z góry przez  $m_0(\mathbb{H}, \delta, \varepsilon)$ .

- Dla skończonych przestrzeni hipotez  $\mathbb{H}$  mamy

$$m_L(\mathbb{H}, \delta, \varepsilon) \leq \left\lceil \frac{1}{\varepsilon} \ln \frac{|\mathbb{H}|}{\delta} \right\rceil = \left\lceil \frac{1}{\varepsilon} (\ln |\mathbb{H}| + \ln(1/\delta)) \right\rceil$$

- **Twierdzenie** Niech  $VCdim(\mathbb{H}) = d \geq 1$ . Wówczas każdy algorytm niesprzeczny  $\mathcal{L}$  jest PAC oraz wymagana liczba przykładów dla  $\mathcal{L}$  wynosi

$$m_L(\mathbb{H}, \delta, \varepsilon) \leq \left\lceil \frac{4}{\varepsilon} \left( d \log \frac{12}{\varepsilon} + \log \frac{2}{\delta} \right) \right\rceil$$

- Dolne ograniczenia:

- $m_L(\mathbb{H}, \delta, \varepsilon) \geq d(1 - \varepsilon)$
- Jeśli  $\delta \leq 1/100$  i  $\varepsilon \leq 1/8$ , to  $m_L(\mathbb{H}, \delta, \varepsilon) > \frac{d-1}{32\varepsilon}$
- $m_L(\mathbb{H}, \delta, \varepsilon) > \frac{1-\varepsilon}{\varepsilon} \ln \frac{1}{\delta}$



## 1. Wyuczalność

Kiedy każdy „wzorowy uczeń” będzie PAC-owy?

## 2. Liczba przykładów

Ile przykładów musi mieć uczeń, by się nauczyć?

## Skończoność wymiaru $VCdim()$

1  $VCdim(\mathbb{C}) = d < \infty \Leftrightarrow \mathbb{C}$  jest wyuczalna;

2 Wówczas  $L(\frac{1}{\varepsilon}, \frac{1}{\delta}, d) < m(\varepsilon, \delta) < U(\frac{1}{\varepsilon}, \frac{1}{\delta}, d)$

## 3. Ocena ucznia



$$R(\alpha) = \min_{\alpha \in A} \int Q_{\Omega}^c(h_{\alpha}) d\mu$$

na podstawie  $N$  losowych przykładów

$$R(\alpha_N) = \min_{\alpha_i \in D} \frac{1}{N} \sum_{i=1}^N Q^c(h_{\alpha_i})$$

Kiedy i jak szybko  $R(\alpha_N) \rightarrow R(\alpha)$ ?

## Skończoność wymiaru $VCdim()$

3 Dla algorytmów typu ERM,  $R(\alpha_N) \rightarrow R(\alpha)$  szybko.

$$Prob \{ R(\alpha) - R(\alpha_N) > \varepsilon \} < e^{-2c\varepsilon^2 N}$$

# Plan wykładu

---



- 1 Wstęp do komputerowego uczenia się pojęć
- 2 Model PAC (probably approximately correct)
- 3 Wyuczalność klasy pojęć
- 4 Przykład modelu PAC i niesprzecznego algorytmu
- 5 Wymiar Vapnika Chervonenkisa (VC dimension)
- 6 Podstawowe twierdzenia teorii uczenia się
- 7 Appendix: „Nie ma nic za darmo” czyli “Non Free Lunch Theorem”

# O co chodzi w NFL?



- Znaleźć optimum nieznannej funkcji  $f : S \rightarrow W$  ( $f \in \mathcal{F}$ ), gdzie  $S, W$  są skończonymi zbiorami.
- Działanie algorytmu przeszukiwania  $\mathcal{A}$  dla funkcji  $f$  jest identyfikowany z wektorem:

$$V_{\mathcal{A}}(f, t) = \langle (s_1, f(s_1)), (s_2, f(s_2)), \dots, (s_t, f(s_t)) \rangle$$

- Ocena algorytmu:  $M : \{V_{\mathcal{A}}(f, t) | \mathcal{A}, f, t\} \rightarrow \mathbb{R}$ ;  
Np.  $M(V_{\mathcal{A}}(f, t)) = \min\{i | f(s_i) = f_{\max}\}$

- **Warunek NFL:** Dla dowolnej funkcji  $M$ , i dla dowolnych algorytmów  $\mathcal{A}, \mathcal{A}'$

$$\sum_{f \in \mathcal{F}} M(V_{\mathcal{A}}(f, |S|)) = \sum_{f \in \mathcal{F}} M(V_{\mathcal{A}'}(f, |S|))$$

- **$\mathcal{F}$  jest zamknięta wzg. permutacji:** dla dowolnej funkcji  $f \in \mathcal{F}$  i dowolnej permutacji  $\sigma \in \text{Perm}(S)$  mamy  $\sigma f \in \mathcal{F}$

### Twierdzenie o NFL

- zachodzi równoważność

$NFL \Leftrightarrow \mathcal{F}$  jest zamknięta wzg. permutacji

- Prawdopodobieństwo wylosowania niepustej klasy funkcji zamkniętej wzg. permutacji wynosi:

$$\frac{2^{\binom{|S|+|W|-1}{|S|}} - 1}{2^{|S||W|} - 1}$$



# The No Free Lunch Theorem for learning



- Algorytm  $\mathcal{L}$  dobrze się uczy pojęcia  $c$  jeśli  $er_{\Omega}^c$  jest mały.
- Niech  $\mathbb{P}(X) = \{c : X \rightarrow \{0, 1\}\}$ .  
Czy można stwierdzić wiedzieć, że  $\mathcal{L}_1$  uczy się wszystkich pojęć z  $\mathbb{P}(X)$  lepiej od  $\mathcal{L}_2$ ?
- “No Free Lunch theorem” (Wolpert, Schaffer) w wersji problemów uczenia się głosi, że:
  - Żaden algorytm nie może być najlepszy w uczeniu wszystkich pojęć.
  - Każdy algorytm jest najlepszy dla takiej samej liczby pojęć
  - Ale interesuje nas tylko pewna klasa problemów czyli klasa pojęć  $C \subset \mathbb{P}(X)$
  - Wniosek: Należy znaleźć odp. algorytm do każdego problemu.