

# **Systemy uczące się**

## wykład 2

# Drzewa klasyfikacyjne - uzupełnienie

Jerzy Stefanowski  
Instytut Informatyki PP  
2021

Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI-TECH)  
projekt finansowany z środków Programu Operacyjnego Polska Cyfrowa  
POPC.03.02.00-00-0001/20



**Fundusze  
Europejskie**  
Polska Cyfrowa



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz  
Rozwoju Regionalnego



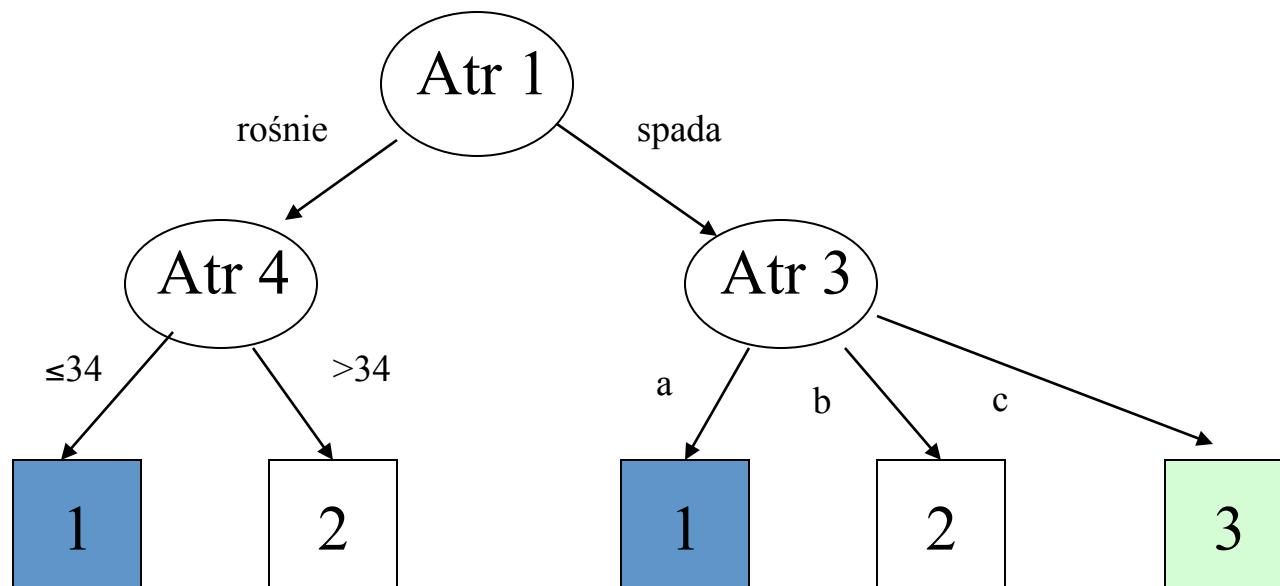
# Plan wykładu

1. Drzewa decyzyjne
2. Algorytm ID3, entropia informacji
  - Uwzględnianie danych niedoskonałych
  - Dyskretyzacja atrybutów liczbowych
3. Inne rozszerzenia → C4.5
4. Przeuczenie klasyfikatorów i tzw. upraszczanie budowy drzew
5. Podsumowanie

# Co to jest drzewo decyzyjne?

Jest to struktura grafu skierowanego z góry na dół:

- Węzły reprezentują pytanie o wartości cech
- Z węzłów wychodzą gałęzie które reprezentują wynik pytania
- Liście reprezentują klasy decyzyjne



# Drzewa - zagadnienia

W miarę dojrzała metodologia, wiele implementacji, liczne zastosowania

Podstawowe problemy:

Jak je tworzyć automatycznie?

Algorytmy

Kryterium wyboru w węzle

Przeuczenie (dobra ilustracja – wielkość drzewa)

Tzw. redukcja drzewa (ang. pruning)

# Metody indukcji drzew decyzyjnych

- Podejście obejmuje dwa etapy:
  - **Konstrukcja drzева (rekurencyjna procedura)**
    - Na początku wszystkie przykłady w węźle.
    - Rekurencyjnie dziel przykłady w oparciu o wybrane testy na wartościach atrybutu (kryterium wyboru najlepszego atrybutu).
    - Zatrzymaj gdy wszystkie przykłady „w gałęzi” należą do jednej klasy
  - Upraszczanie drzева - „Tree pruning”
    - Usuwanie poddrzew, które mogą prowadzić do błędnych decyzji podczas klasyfikacji przypadków testowych.
  - Przykłady algorytmów: ID3, C4.5, CART,...

# Przykład budowy DT – Quinlan „play golf”

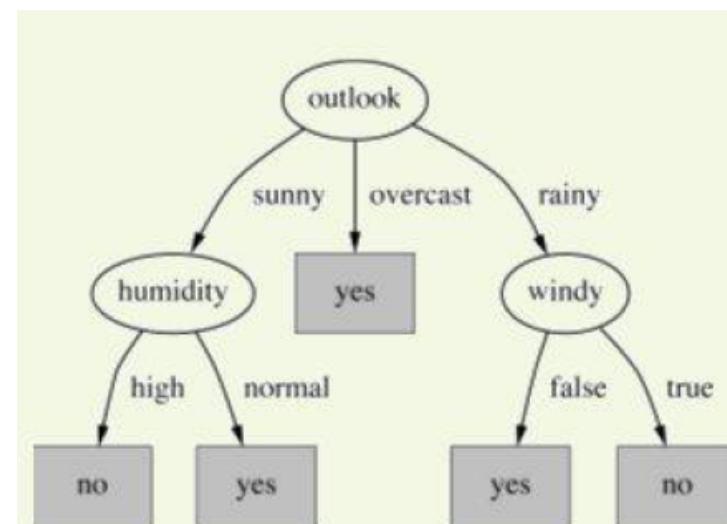
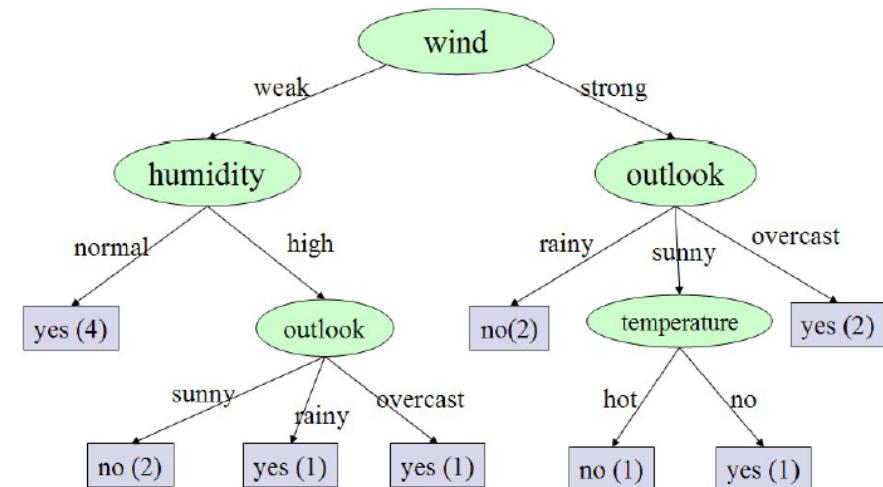
Outlook	Temperature	Humidity	Windy	Play?
sunny	hot	high	false	No
sunny	hot	high	true	No
overcast	hot	high	false	Yes
rain	mild	high	false	Yes
rain	cool	normal	false	Yes
rain	cool	normal	true	No
overcast	cool	normal	true	Yes
sunny	mild	high	false	No
sunny	cool	normal	false	Yes
rain	mild	normal	false	Yes
sunny	mild	normal	true	Yes
overcast	mild	high	true	Yes
overcast	hot	normal	false	Yes
rain	mild	high	true	No

*Uproszczona  
Tabela danych*

# Poszukiwanie dobrych drzew

## Play or not (Quinlan)

x	outlook	Temperature	humidity	wind	play(x)
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cold	normal	weak	yes
6	rain	cold	normal	strong	no
7	overcast	cold	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cold	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no



# Ogólny schemat

## TDIDT - Top Down Iterative Decision Tree

```
function DT( $E$ : zbiór przykładów) returns drzewo;  
     $T' :=$  buduj_drzewo( $E$ );  
     $T :=$  obetnij_drzewo( $T'$ );  
    return  $T$ ;
```

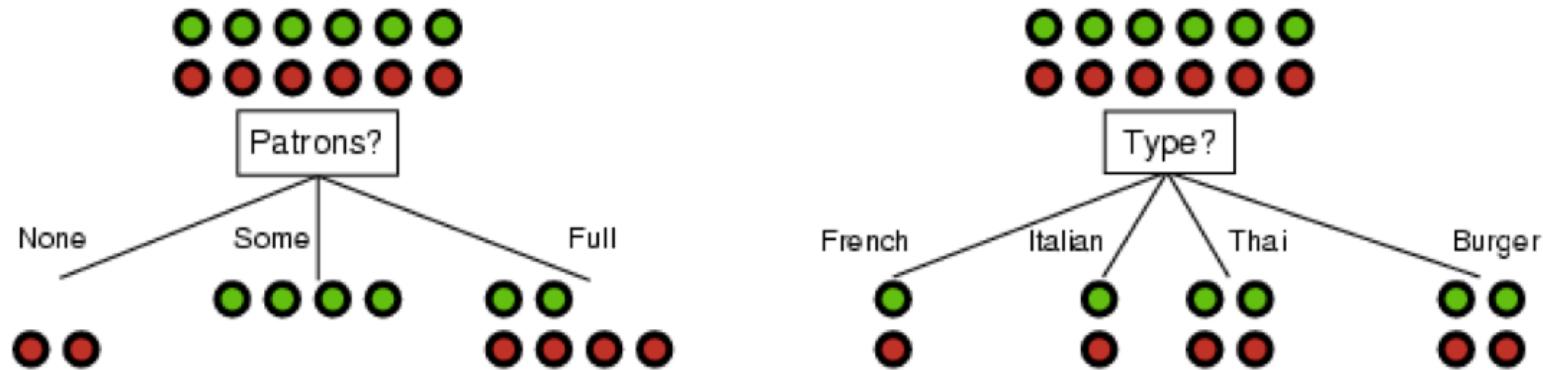
```
function buduj_drzewo( $E$ : zbiór przyk.) returns drzewo;  
     $T :=$  generuj_tests_atr_A( $E$ );  
     $t :=$  najlepszy_test( $T, E$ );  
     $P :=$  podział  $E$  indukowany przez  $t$ ,  
    if kryterium_stopu( $E, P$ )  
    then return liść(info( $E$ ))  
    else  
        for all  $E_j$  in  $P$ :  $t_j :=$  buduj_drzewo( $E_j$ );  
        return węzeł( $t, \{(j, t_j)\}$ );
```

# Intuicja wyboru atrybutu

Przykład decyzji o wyborze restauracji [Russell, Norvig]

Split condition -

Dobry atrybut powinien podzielić zbiór przykładów S na podzbiory S<sub>1</sub>, S<sub>2</sub>,..., które są możliwie jednoznaczne (purity) wskazać klasy decyzyjne – poszukiwanie możliwie najprostszego drzewa zgodnego z przykładami uczącymi



Which split is more informative: *Patrons?* or *Type?*

# Entropia (C. Shannon)

- Entropia (zawartość informacyjna, *information content*): miara oceniająca zbiór przykładów pod kątem ‘czystości’ (jednolitości przynależności do klas decyzyjnych)
- Dla dwóch klas decyzyjnych (pozytywna, negatywna) –  $p$  liczba przykładów pozytywnych,  $n$  - negatywnych:

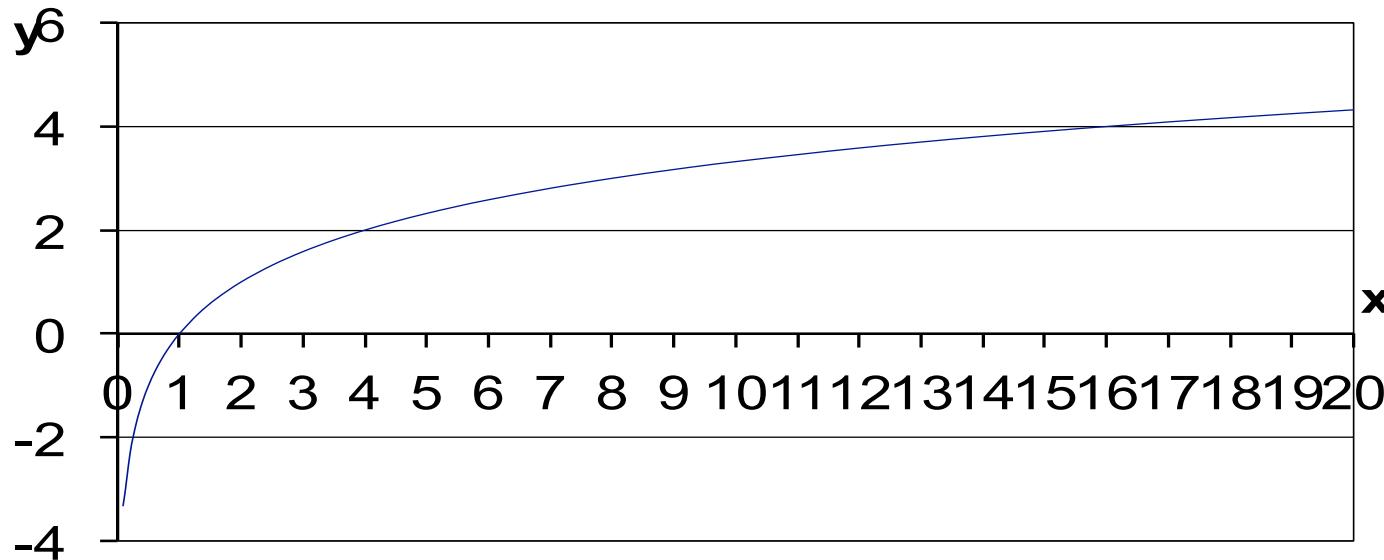
$$I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n}$$

- Wersja ogólna wieloklasowa  $p_i$  – prawdopodobieństwo, że przykład należy do i-tej klasy:

$$I = -\sum_{i=1}^K p_i \cdot (\log_2 p_i)$$

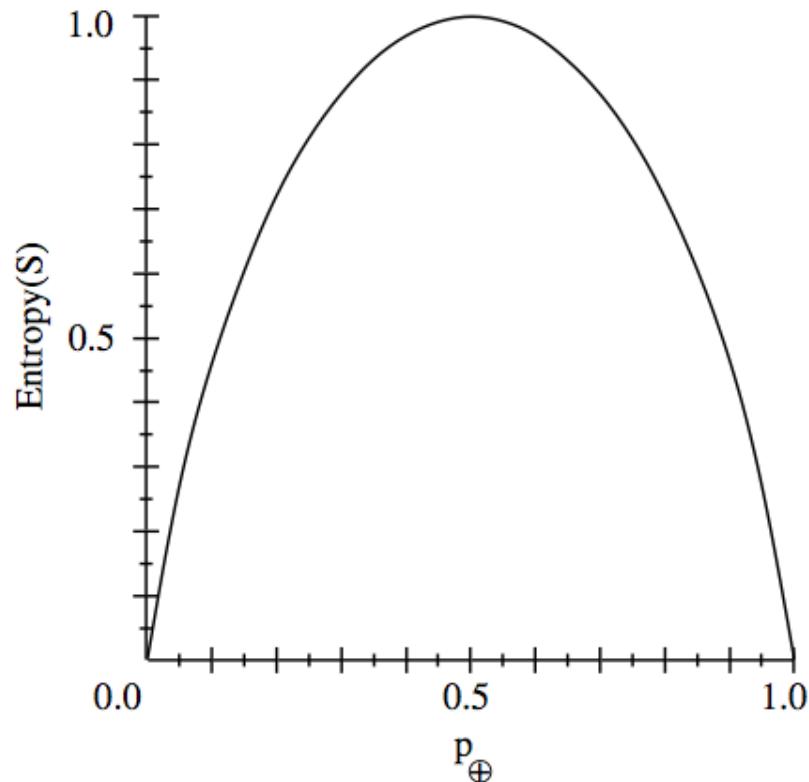
# Przypomnienie logarytmów

- Funkcja log.  $y = \log_a x$
- $a$  – podstawa logarytmu  $x = a^y$
- Rozważmy funkcję logarytmiczną dla  $a = 2$  (tj.  $\log_2 x$ )



$x$	$1/8$	$1/4$	$1/2$	$1$	$2$	$4$	$8$
$y$	-3	-2	-1	0	1	2	3

# Entropia – interpretacja i własności mat.



Analiza binarnej entropii dla dwóch klas

# Entropia dla przykładu golf

Nie oceniamy podziału atrybutem, tylko rozkład wartości klas decyzyjnych

Dwie klasy : yes and no

Z 14 przykładów 9 etykietowanych jako yes, reszta jako no

$$p_{yes} = -\left(\frac{9}{14}\right) \log_2 \left(\frac{9}{14}\right) = 0.41$$

$$p_{no} = -\left(\frac{5}{14}\right) \log_2 \left(\frac{5}{14}\right) = 0.53$$

$$E(S) = p_{yes} + p_{no} = 0.94$$

Outlook	Temp.	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes

Outlook	Temp.	Humidity	Windy	play
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

# Zysk informacji - Information gain

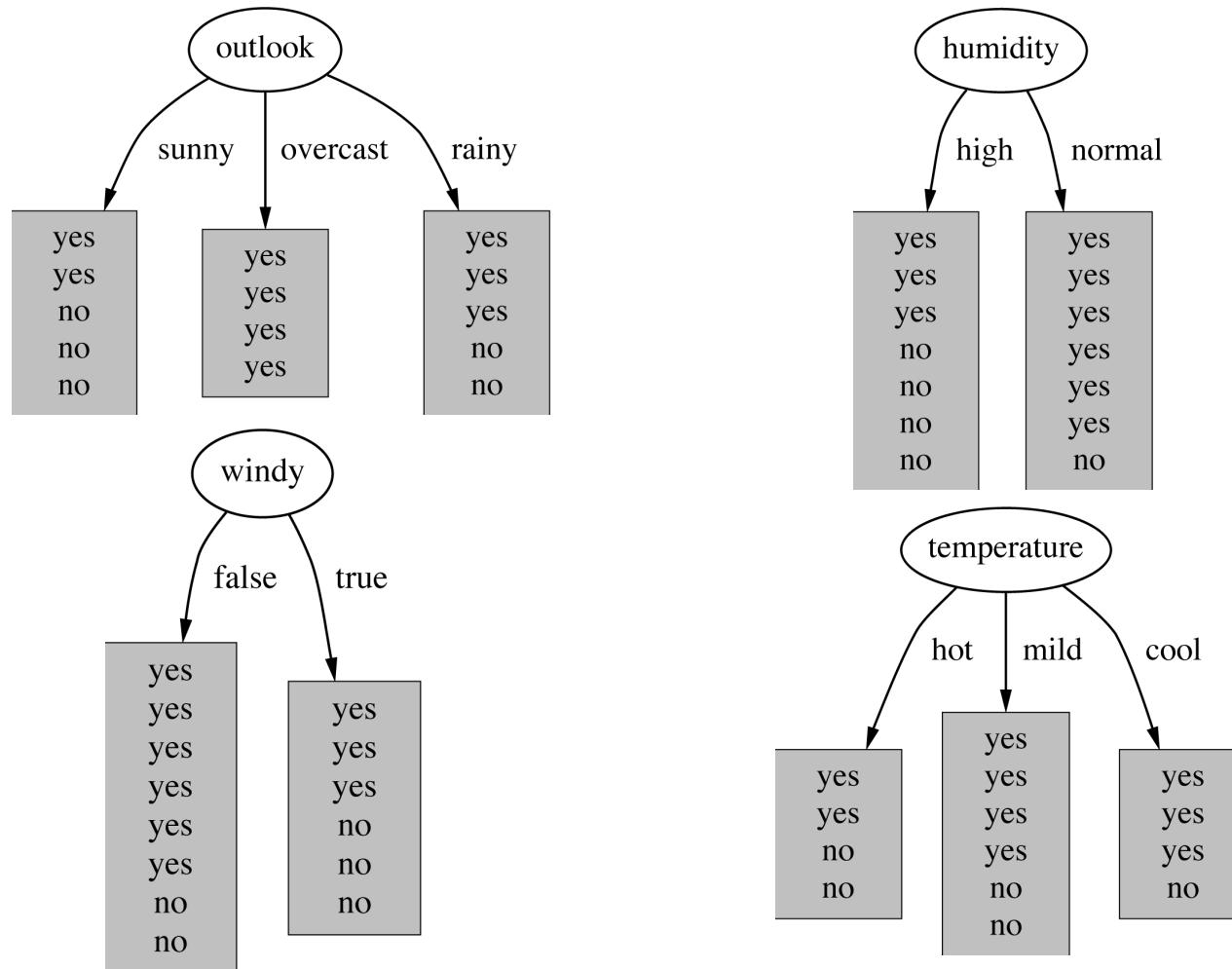
- Entropia warunkowa: entropia po podziale zbioru przykładów przy pomocy atrybutu A (założymy że A przyjmuje  $v$  możliwych wartości):

$$Entropia\ Warunkowa(A) = \sum_{i=1}^v \frac{p_i + n_i}{p + n} I\left(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i}\right)$$

- Zysk informacyjny (*Information Gain*): redukcja entropii przy wykorzystaniu danego atrybutu:

$$IG(A) = I - Entropia\ Warunkowa(A)$$

# Który atrybut należy wybrać?



# Przykład oceny atrybutu “Outlook”

- “Outlook” = “Sunny”:

$$\text{info}([2,3]) = \text{entropy}(2/5,3/5) = -2/5\log(2/5) - 3/5\log(3/5) = 0.971$$

- “Outlook” = “Overcast”:

$$\text{info}([4,0]) = \text{entropy}(1,0) = -1\log(1) - 0\log(0) = 0$$



*Uwaga:  $\log(0)$  jest nieskończone lecz  $0 * \log(0)$  dąży do zera*

- “Outlook” = “Rainy”:

$$\text{info}([3,2]) = \text{entropy}(3/5,2/5) = -3/5\log(3/5) - 2/5\log(2/5) = 0.971$$

- Entropia warunkowa dla podziału wartościami atrybutu

$$\begin{aligned}\text{info}([3,2],[4,0],[3,2]) &= (5/14) \times 0.971 + (4/14) \times 0 + (5/14) \times 0.971 \\ &= 0.693\end{aligned}$$

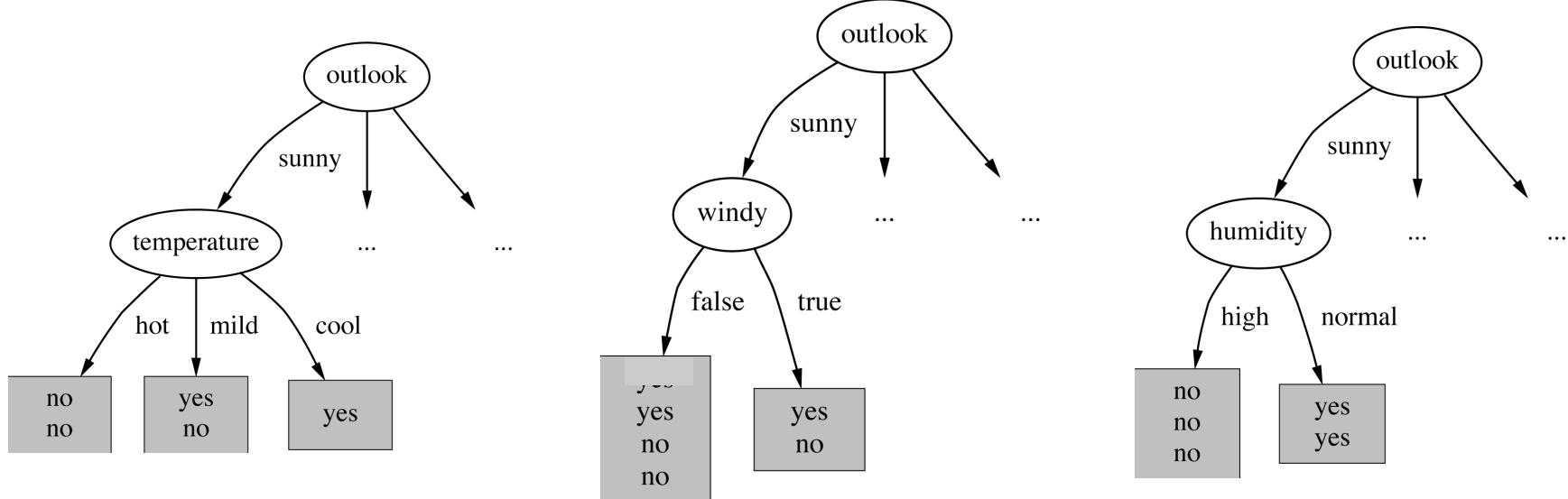
# Obliczanie zysku informacyjnego miary entropii

- Zysk informacji -> Information gain:  
(information before split) – (information after split)

$$\text{gain("Outlook")} = \text{info}([9,5]) - \text{info}([2,3], [4,0], [3,2]) = 0.940 - 0.693 \\ = 0.247$$

- Ostateczne wartości zysku
  - Gain(„Temperature”)=0.029
  - Gain(„Humidity”)=0.152
  - Gain(„Windy”)=0.048
  - Co wybieramy?

# Rozbuduj drzewo

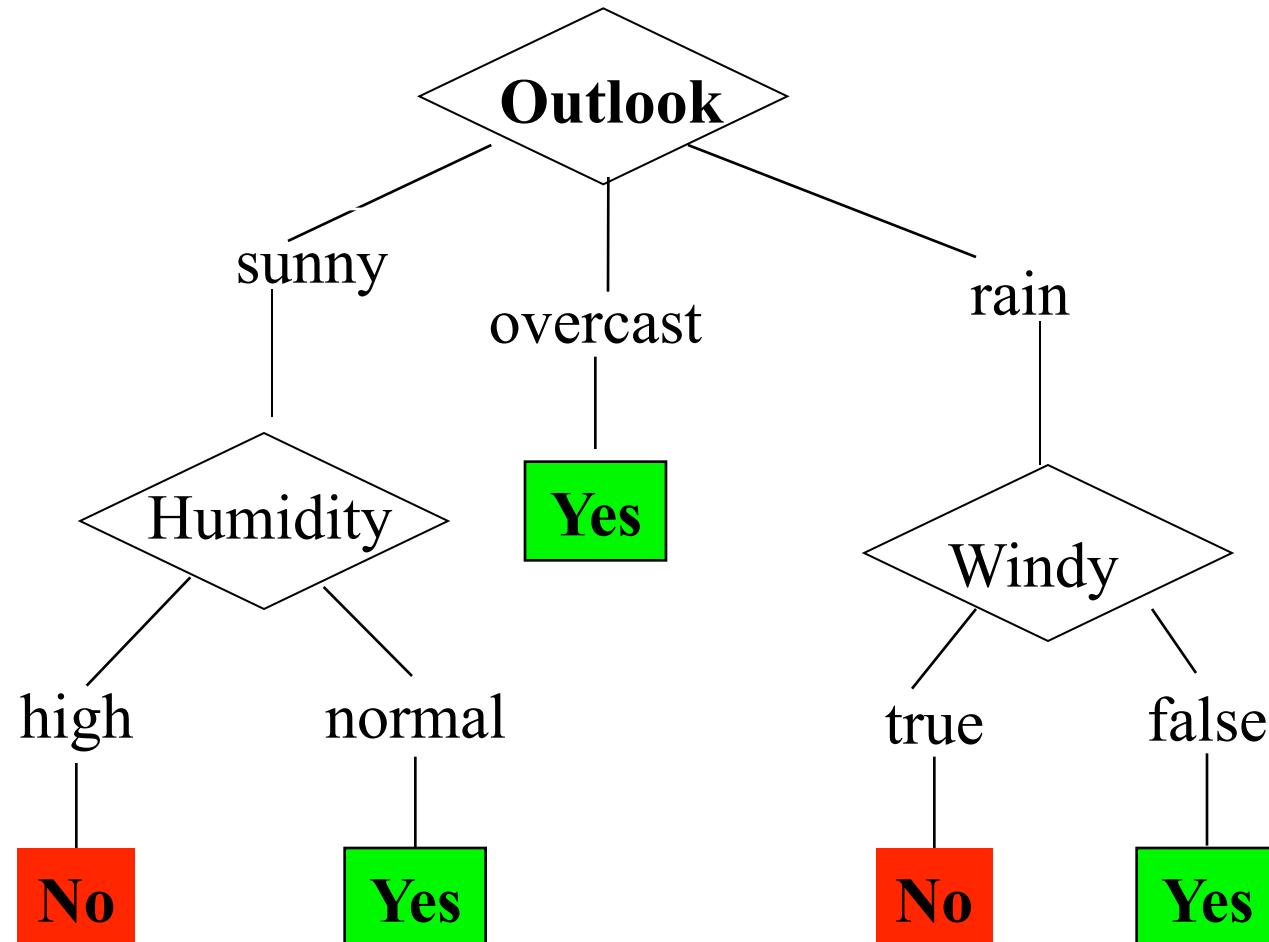


gain("Humidity") = 0.971

gain("Temperature") = 0.571

gain("Windy") = 0.020

# Ostateczne drzewo

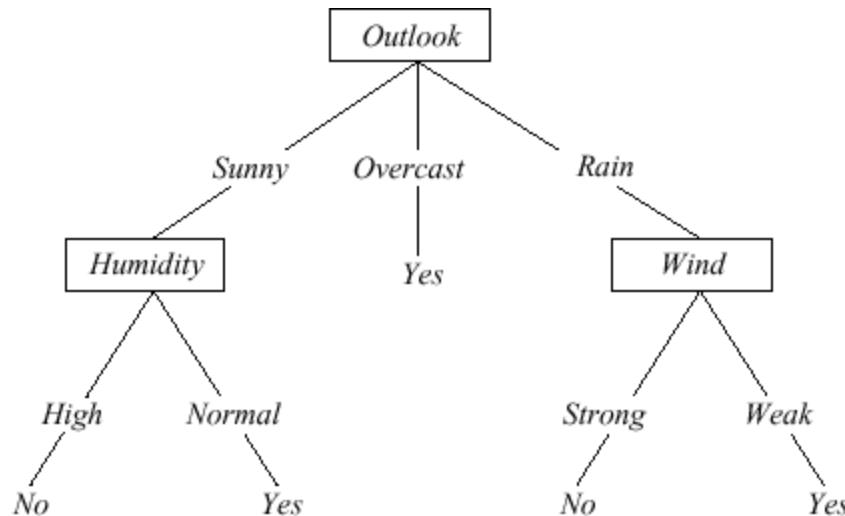


# Wykorzystanie drzewa

- Bezpośrednio:
  - sprawdzaj wartości atrybutu nowego przykładu zaczynając od korzenia do liści
- Pośrednio:
  - zamień strukturę drzewa na zbiór reguł decyzyjnych (upraszczając nadmiarowe warunki)
  - reguły uważa się za czytelniejszą reprezentację

# DT => reguły

Zamień DT na reguły i uprość: łatwo ocenić, które reguły można usunąć i optymalizować pozostałe.



IF (*Outlook* = *Sunny*)  $\wedge$  (*Humidity* = *High*) THEN *PlayTennis* = *No*

IF (*Outlook* = *Sunny*)  $\wedge$  (*Humidity* = *Normal*) THEN *PlayTennis* = *Yes*

# Inne kryteria podziału

Indeks Gini-ego (CART, alg. dla tzw. datamining)

$$Gini = 1 - \sum_{i=1}^K p_i^2$$

Także warunkowa postać po wyborze podziału A

# Silnie wielowartościowe atrybuty

- Problematyczne : atrybuty o relatywnie większej dziedzinie niż inne
- Podzbiory mało liczne po podziale mogą być „czystsze”
  - ⇒ Preferencja miary entropi / zysku informacyjnego
  - ⇒ Słabe własności generalizujące
- Wykorzystanie miary gain ratio lub binaryzacja drzewa
- Gain-ratio (Quinlan)

$$GainRatio(S, A) = \frac{Gain(S, A)}{IntrinsicInfo(S, A)}.$$

$$IntrinsicInfo(S, A) \equiv -\sum \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}.$$

# Binary Tree – budowa drzew binarnych

- Drzewa binarne mogą być skuteczniejsze w klasyfikacji nowych faktów
- Podział binarny w węźle drzeca:
  - Atrybuty liczbowe A, reprezentacja w postaci  $\text{value}(A) < x$  gdzie x jest wartością z dziedziny A.
  - Atrybuty nieliczbowe A, warunek w postaci  $\text{value}(A) \in X$  gdzie  $X \subset \text{domain}(A)$

# Binary tree (Quinlan's C4.5 output)

Pruned decision tree:

```
A9 = t:
  A15 > 22B : + (106.0/3.8)
  A15 <- 22B :
    A14 <- 102 :
      A4 in {l,t} : + (0.0)
      A4 = u:
        A6 in {c,d,cc,i,k,m,q,w,x,e,aa} : + (46.4/3.1)
        A6 in {j,ff} : - (2.0/1.0)
        A6 = r : + (0.0)
      A4 = y:
        A6 in {c,i,aa,ff} : - (7.0/3.4)
        A6 in {d,j,w,x} : + (4.0/1.2)
        A6 in {cc,k,m,r,q,e} : + (0.0)
    A14 > 102 :
      A6 in {j,r} : + (0.0)
      A6 in {c,d,k,m,e,aa,ff} :
        A14 <- 132 : - (4.1/1.2)
        A14 > 132 :
          A3 <- 1.625 :
            A14 <- 292 : - (13.0/1.3)
            A14 > 292 :
              | A13 = g : + (2.0/1.0)
              | A13 = s : - (6.0/2.3)
              | A13 = p : - (0.0)
          A3 > 1.625 :
            A6 in {k,m} : + (5.0/1.2)
            A6 = ff : + (0.0)
            A6 in {c,d,e,aa} :
              | A2 <- 32.08 : + (9.5/4.1)
              | A2 > 32.08 : - (8.0/3.5)
            A6 in {cc,i,q,w,x} :
              | A8 <- 10.75 : + (36.0/9.3)
              | A8 > 10.75 : - (2.0/1.0)
A9 = f:
  A4 in {u,y} : - (237.0/17.3)
  A4 = l : + (2.0/1.0)
  A4 = t : - (0.0)
```

- Crx (Credit Data) UCI ML Repository
- źródło własne

# Binaryzacja atrybutu ilościowego

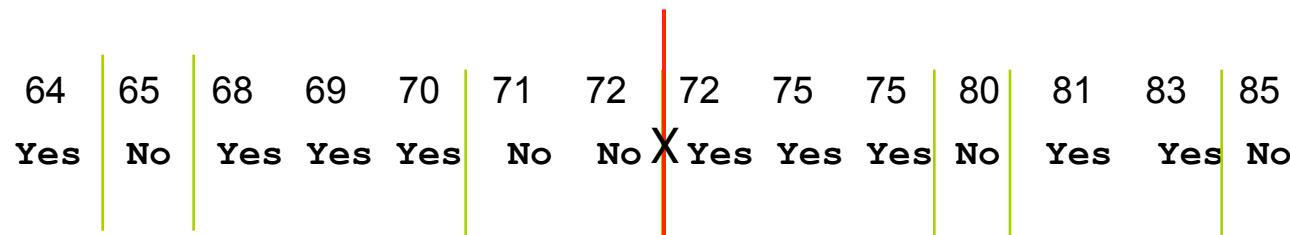
- Punkt podziału - Split dla atr. temperature :

64	65	68	69	70	71	72	72	75	75	80	81	83	85
Yes	No	Yes	Yes	Yes	No	No	Yes	Yes	Yes	No	Yes	Yes	No

- Np.  $\text{temperature} < 71.5$ : yes/4, no/2  
 $\text{temperature} \geq 71.5$ : yes/5, no/3
- $\text{Info}([4,2],[5,3])$   
 $= 6/14 \text{ info}([4,2]) + 8/14 \text{ info}([5,3])$   
 $= 0.939$
- Wstaw próg między istniejące przykłady
- Efektywne obliczeniowo

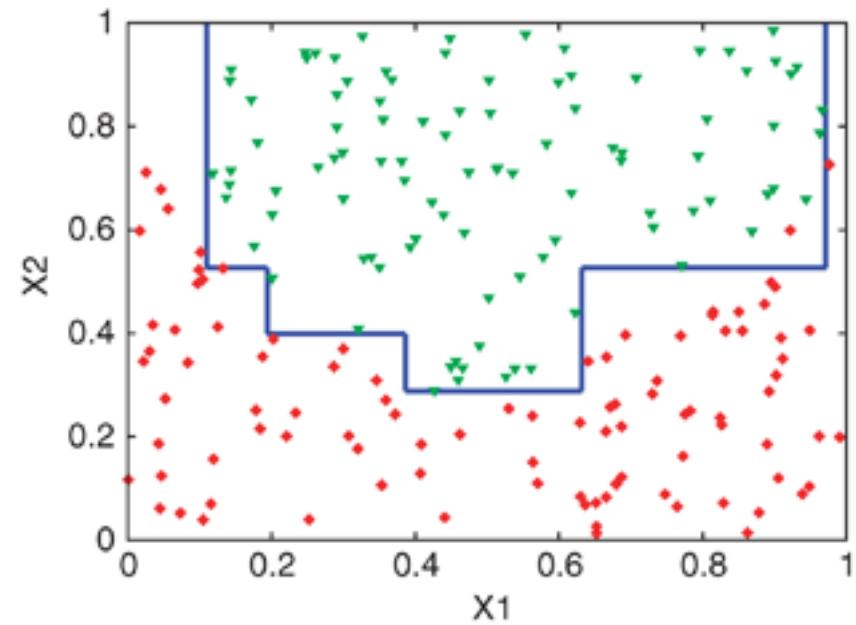
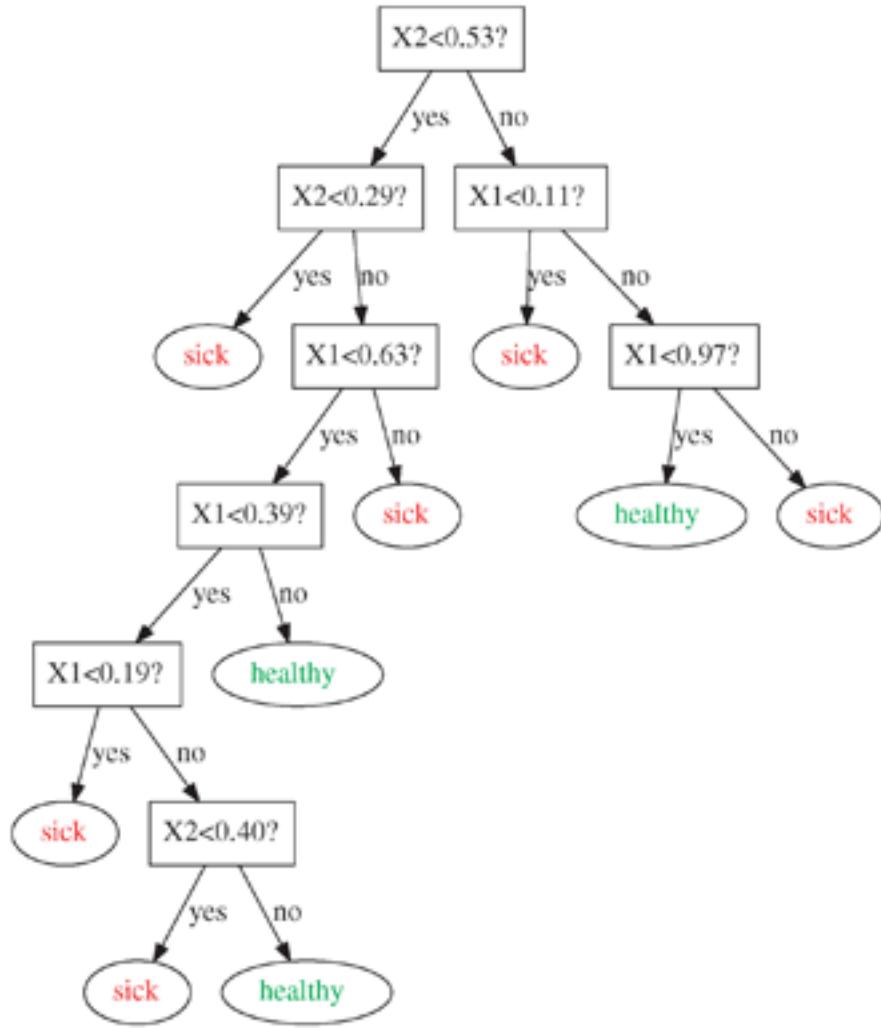
# Szybsze obliczenia

- Własności mat. entropii(Fayyad & Irani, 1992)



Potencjalne punkty cięcia

# Przykład medyczny



# Inne wyzwanie ...

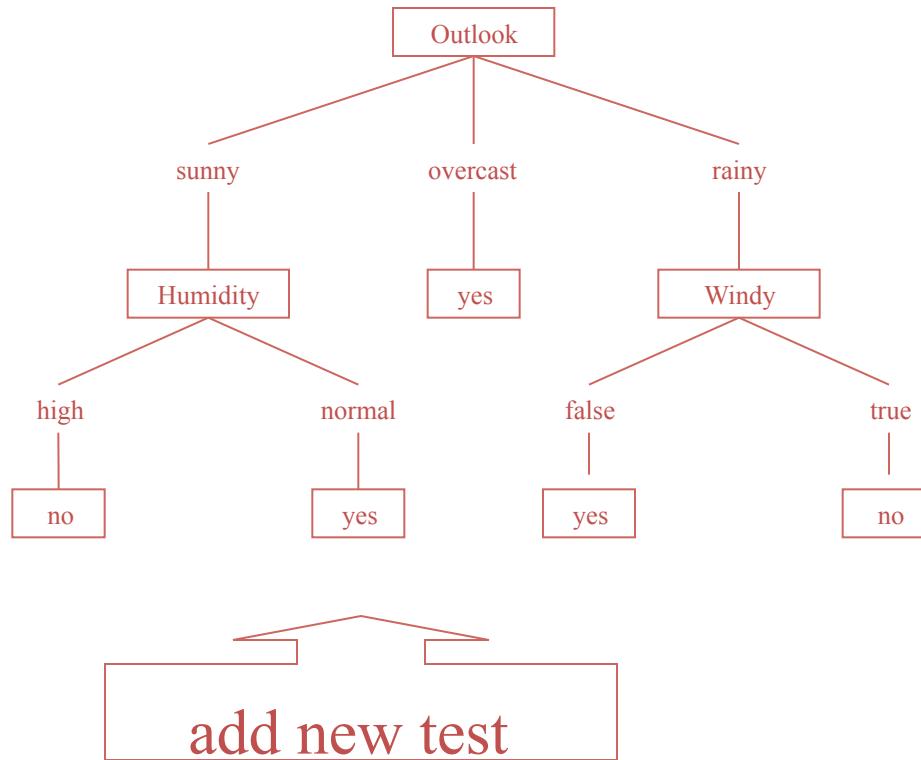
Outlook	Temperature	Humidity	Windy	Play?
sunny	hot	high	false	No
sunny	hot	high	true	No
overcast	hot	high	false	Yes
rain	mild	high	false	Yes
rain	cool	normal	false	Yes
rain	cool	normal	true	No
overcast	cool	normal	true	Yes
sunny	mild	high	false	No
<b>sunny</b>	<b>cool</b>	<b>normal</b>	<b>false</b>	<b>Yes</b>
rain	mild	normal	false	Yes
sunny	mild	normal	true	Yes
overcast	mild	high	true	Yes
overcast	hot	normal	false	Yes
rain	mild	high	true	No

*Co będzie dla sprzącznych przykładów*

# Sprzeczne opisy przykładów

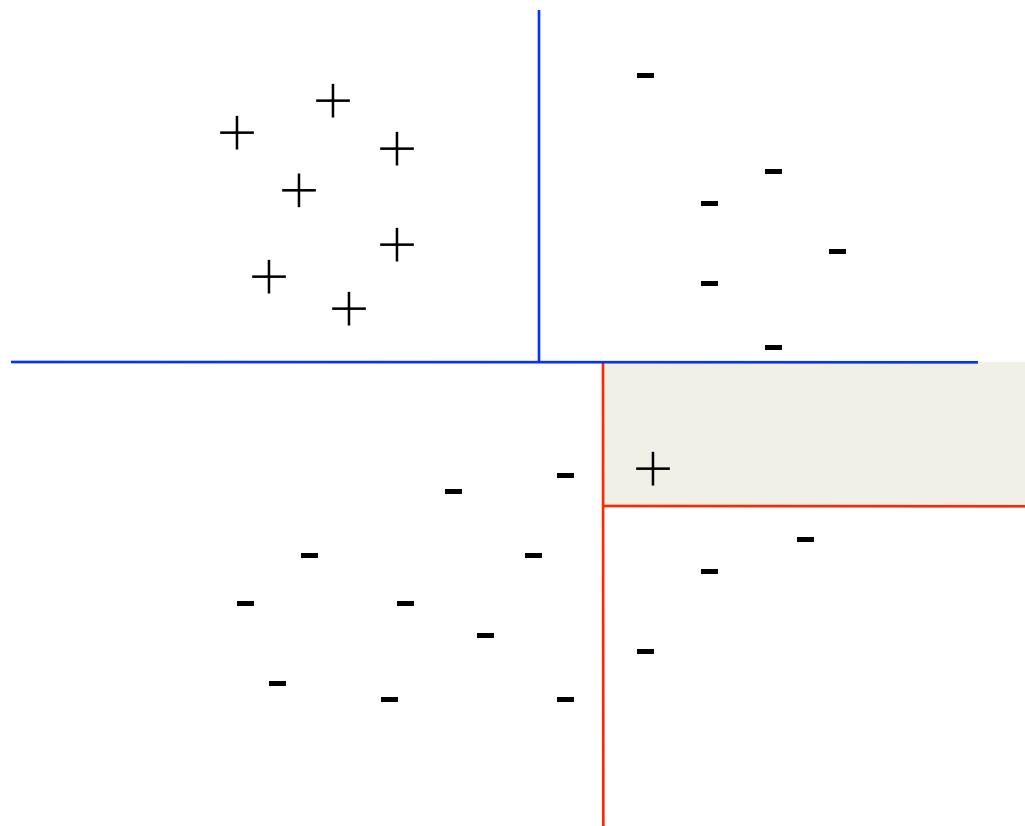
**Nowy sprzeczny przykład:**

Outlook = *Sunny*; Temperature = *Cool*; Humidity = *Normal*; Wind = *False*; PlayTennis = *No*



# Inne przyczyny przeuczenia drzewa

- **Nietypowe przykłady** – prowadzą do rozbudowanych poddrzew z małą liczbą przykładów wspierających liście



# Przeuczenie klasyfikatora (Overfitting)

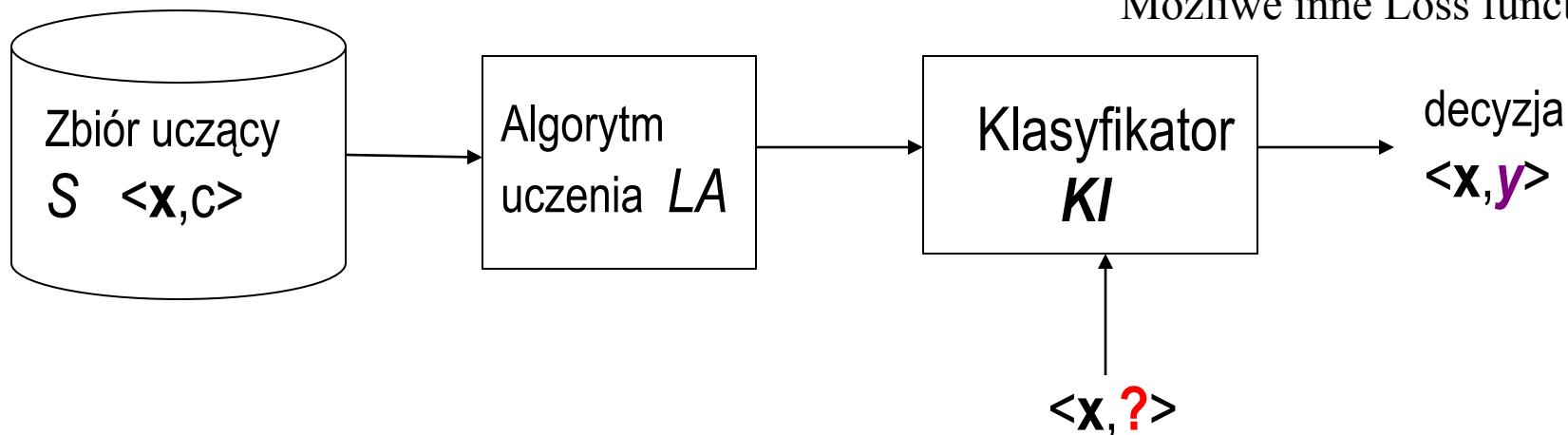
- Dobry klasyfikator (drzewo) musi nie tylko wystarczająco spójnie odwzorowywać dane uczące, lecz także trafnie klasyfikować nowe dane (niewidziane w trakcie procesu uczenia się).
- Innymi słowami – klasyfikator musi mieć niski błąd uczący, lecz przede wszystkim niski błąd uogólnienia na nowe dane (testowe)
  - Błąd treningowy vs. błąd testowy
- Nadmiernie rozbudowane drzewo, dopasowane do trudnych przykładów uczących traci zdolności uogólniania.

# Predykcja nowych faktów - klasyfikatory

Predykcja klasyfikacji nowych obiektów (zbiór testowy) Miara oceny, np:  
→ Cross validation **trafność klasyfikowania**

$$\eta = \frac{N_c}{N_t}$$

Mozliwe inne Loss functions



Przykłady  $S = \{\langle \mathbf{x}_1, c_1 \rangle, \langle \mathbf{x}_2, c_2 \rangle, \dots, \langle \mathbf{x}_n, c_n \rangle\}$

$\mathbf{x}_i = \langle x_{i1}, x_{i2}, \dots, x_{im} \rangle$  opisywane przez  $m$  atrybutów

Atrybuty różnego typu

$c_i$  – etykieta jednej z klas  $\{C_1, \dots, C_K\}$

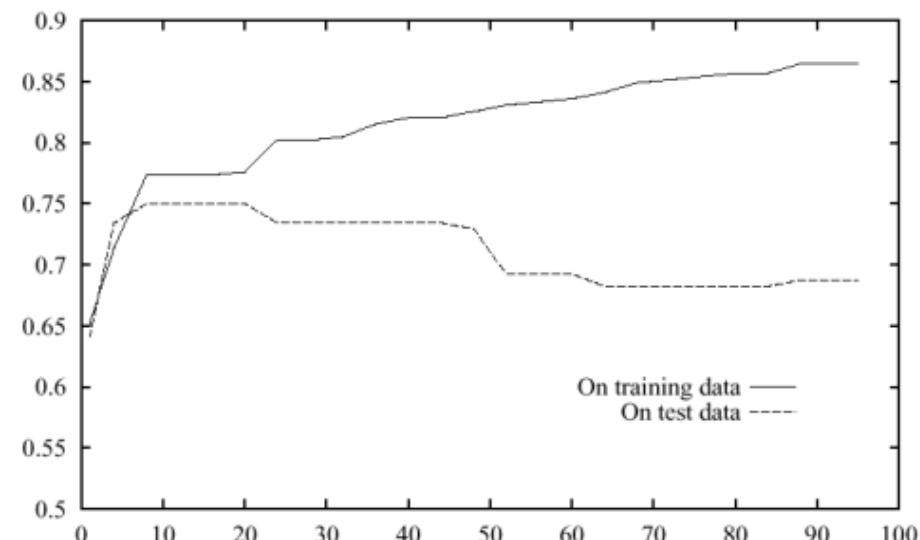
Nowe instancje →  $Kl(\mathbf{x})$   
albo

Przykłady testowe  
→  $\langle \mathbf{x}_j, ? \rangle$  / + znamy poprawną etykietę klasy  $C_{ji}$

$Kl(\mathbf{x}) ? C_{ji}$  [Loss function]

# Overfitting the Data – nadmierne dopasowanie do danych uczących

- Podstawowy algorytm ID3 → Rozbuduj gałąź drzewa do pełnego rozróżnienia przykładów
  - Sensowe na spójnych przykładów i celów dokładnego opisu
- Rzeczywiste dane (niespójne, szum informacyjny) oraz cel klasyfikowania przykładów
  - Drzewa mają tendencje do przeuczenia / nadmiernego dopasowania do specyficznych przykładów *overfit the learning examples*
  - Occam razor – zasada brzytwy Occama (z konkurencyjnych drzew wybierz prostsze; ma lepsze właściwości generalizacyjne)



# Brzytwa Ockhama

Czemu preferować prostsze drzewa?

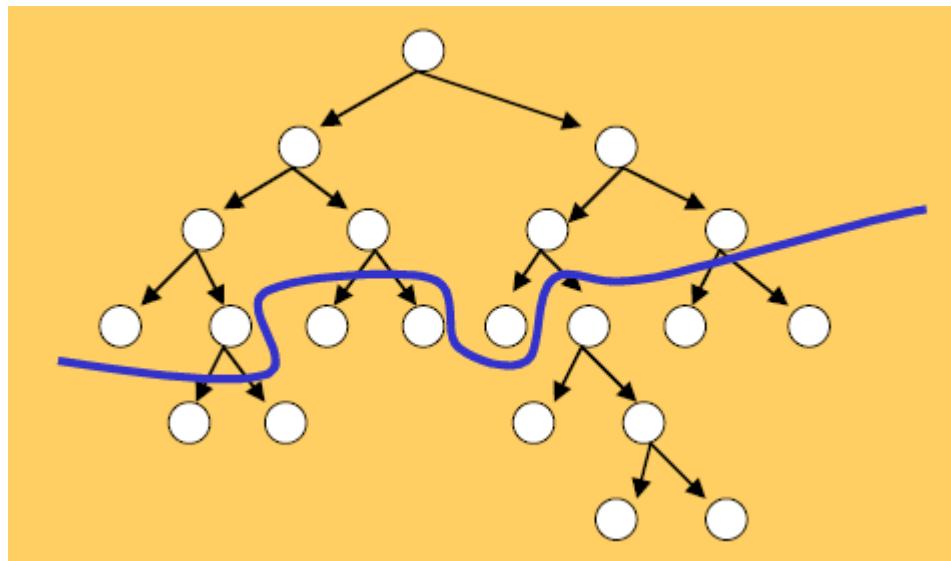
1. Mało prostych hipotez, więc mała szansa, że przypadkiem pasują do danych.
2. Proste drzewa nie powinny zbytnio dopasować się do danych.
3. Przetrenowanie modelu dla zbyt złożonych drzew, zła generalizacja.

Ale:

1. Dla małych zbiorów o wielu atrybutach można tworzyć wiele prostych opisów danych.

# Tree pruning – upraszczanie drzewa

- Mechanizm „walki” z przeuczeniem
- Po uproszczeniu struktury drzewa może wzrosnąć trafność na przykładach testowych!



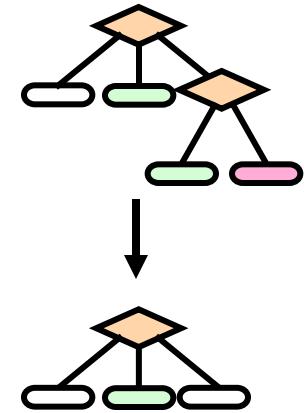
# Unikanie przetrenowania

Jak uniknąć przetrenowania i radzić sobie z szumem?

1. Zakończ rozwijanie węzła jeśli jest zbyt mało danych by wiarygodnie dokonać podziału.
  2. Zakończ jeśli czystość węzłów (dominacja jednej klasy) jest większa od zadanego progu – pre- pruning  
 $DT \Rightarrow$  drzewo prawd. klas.
  3. Utwórz drzewo a potem je przytnij (post - pruning)
- 
1. Przycinaj korzystając z wyników dla k-cv lub dla zbioru walidacyjnego.
  2. Korzystaj z MDL (Minimum Description Length):  
 $\text{Min Rozmiar(Drzewa)} + \text{Rozmiar(Drzewa(Błędów))}$
  3. Oceniaj podziały zaglądając poziom (lub więcej) w głąb.

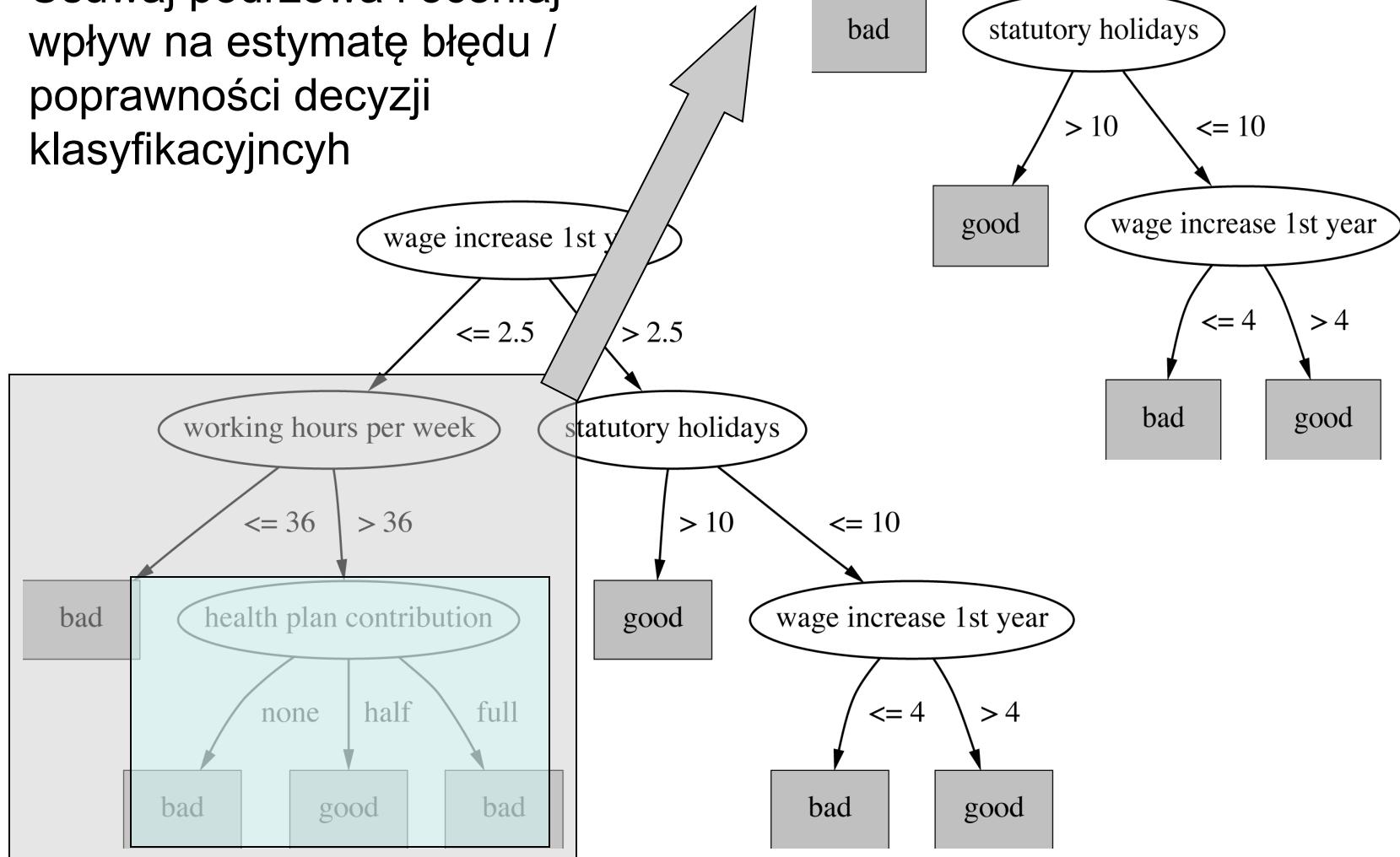
# Reduced-Error Pruning

- Post-Pruning, Cross-Validation Approach
- Split Data into Training and Validation Sets
- Function  $\text{Prune}(T, \text{node})$ 
  - Remove the subtree rooted at  $\text{node}$
  - Make  $\text{node}$  a leaf (with majority label of associated examples)
- Algorithm *Reduced-Error-Pruning* ( $D$ )
  - Partition  $D$  into  $D_{train}$  (training / “growing”),  $D_{validation}$  (validation / “pruning”)
  - Build complete tree  $T$  using *ID3* on  $D_{train}$
  - UNTIL accuracy on  $D_{validation}$  decreases DO
    - FOR each non-leaf node  $\text{candidate}$  in  $T$ 
      - $\text{Temp}[\text{candidate}] \leftarrow \text{Prune } (T, \text{candidate})$
      - $\text{Accuracy}[\text{candidate}] \leftarrow \text{Test } (\text{Temp}[\text{candidate}], D_{validation})$
    - $T \leftarrow T' \in \text{Temp}$  with best value of  $\text{Accuracy}$  (best increase; *greedy*)
  - RETURN (pruned)  $T$



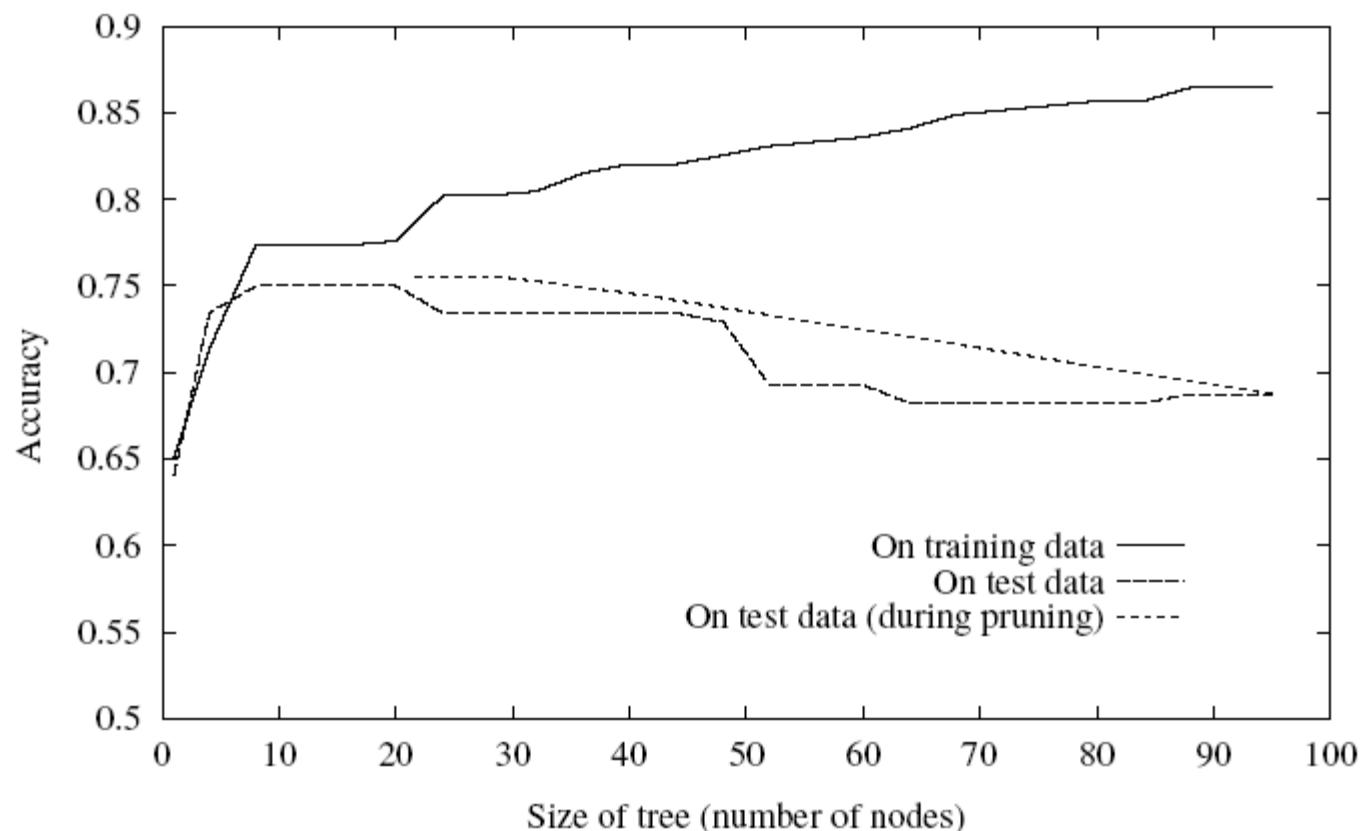
# Przykład redukcji

- Tzw. *post-pruning*
- Usuwaj podrzewa i oceniaj wpływ na estymatę błędu / poprawności decyzji klasyfikacyjnych

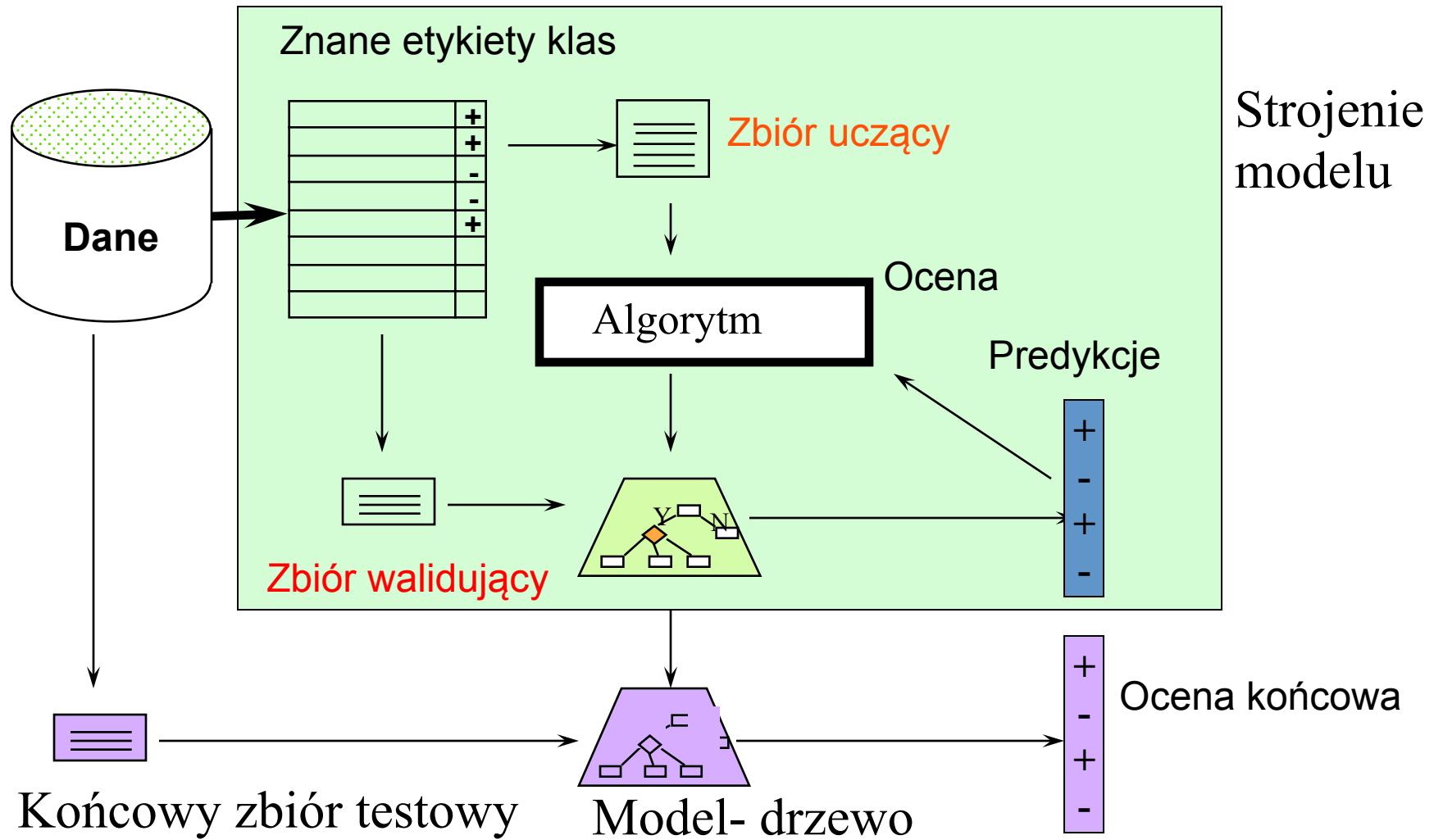


# Reduced post-pruning

Zbiór przykładów testowych – nie można wykorzystywać w trakcie uczenia, potrzebny tzw. zbiór przykładów walidacyjnych



# Trzy rodzaje danych: treningowe, walidacyjne, testowe

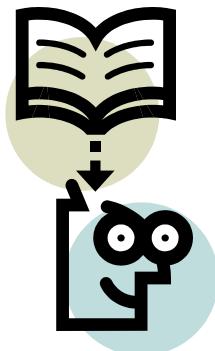


# Przykłady zastosowań

- Wiele przykładów analizy podejmowania decyzji o diagnozowaniu chorób, także terapii, oraz farmacja i budowa związków - leków:
- Przykładowe omówienia:
  - I.Kononenko, I.Bratko, M.Kukar: Application of Machine Learning to Medical Diagnosis. w: Michalski R.S., Bratko I, Kubat M. (red.), Machine learning and data mining, John Wiley & Sons, 1998, s. 389-408.
  - Langley, P., Simon, H. A., Fielded applications of machine learning, w: Michalski R.S., Bratko I, Kubat M. (red.), Machine learning and data mining, John Wiley & Sons, 1998 , s. 113-129.
- Spójrz także na dodatkowe materiały – na podanym linku w ekursy

# Trochę książek

- Uczenie maszynowe i sieci neuronowe.  
Krawiec K., Stefanowski J., Wydawnictwo  
Politechniki Poznańskiej, Poznań, 2003  
(kolejne wydanie 2004)
- Systemy uczące się. Cichosz P., WNT,  
Warszawa, 2000
- Statystyczne systemy uczące się. Koronacki  
J., Ćwik J. WNT Warszawa 2008
- 



# **Pytanie i komentarze?**

Dalszy kontakt:

[jerzy.stefanowski@cs.put.poznan.pl](mailto:jerzy.stefanowski@cs.put.poznan.pl)

<http://www.cs.put.poznan.pl/jstefanowski/>



**Fundusze  
Europejskie**  
Polska Cyfrowa



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz  
Rozwoju Regionalnego



# **Systemy uczące się**

# **Metoda wektorów wspierających**

## **wykład 2-3**

Jerzy Stefanowski

Instytut Informatyki PP + PAN

2021 / update 2024

Poprzednia wersja przygotowana dla

Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI-TECH) projekt finansowany z środków Programu Operacyjnego Polska Cyfrowa POPC.03.02.00-00-0001/20



**Fundusze  
Europejskie**  
Polska Cyfrowa



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz  
Rozwoju Regionalnego



# Plan wykładu

1. Liniowa separowalność w statystycznej klasyfikacji
2. Podejścia klasyczne – Fisher Liniowa Analiz Dyskryminacyjna
3. LDA = sformułowanie probabilistyczne
4. Podstawy matematyczne metody SVM
5. Sformułowanie zadania optymalizacji

Kolejny wykład

1. Uogólnienie SVM (nie w pełni separowalne liniowo)
2. Funkcje jądrowe (tzw. kernel functions)
3. SVM dla danych z nieliniowymi granicami
4. Podsumowanie
5. Gdzie szukać więcej?

# Przypomnienie

**Poprzedni wykład:**

Klasyfikatory liniowe  $g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$

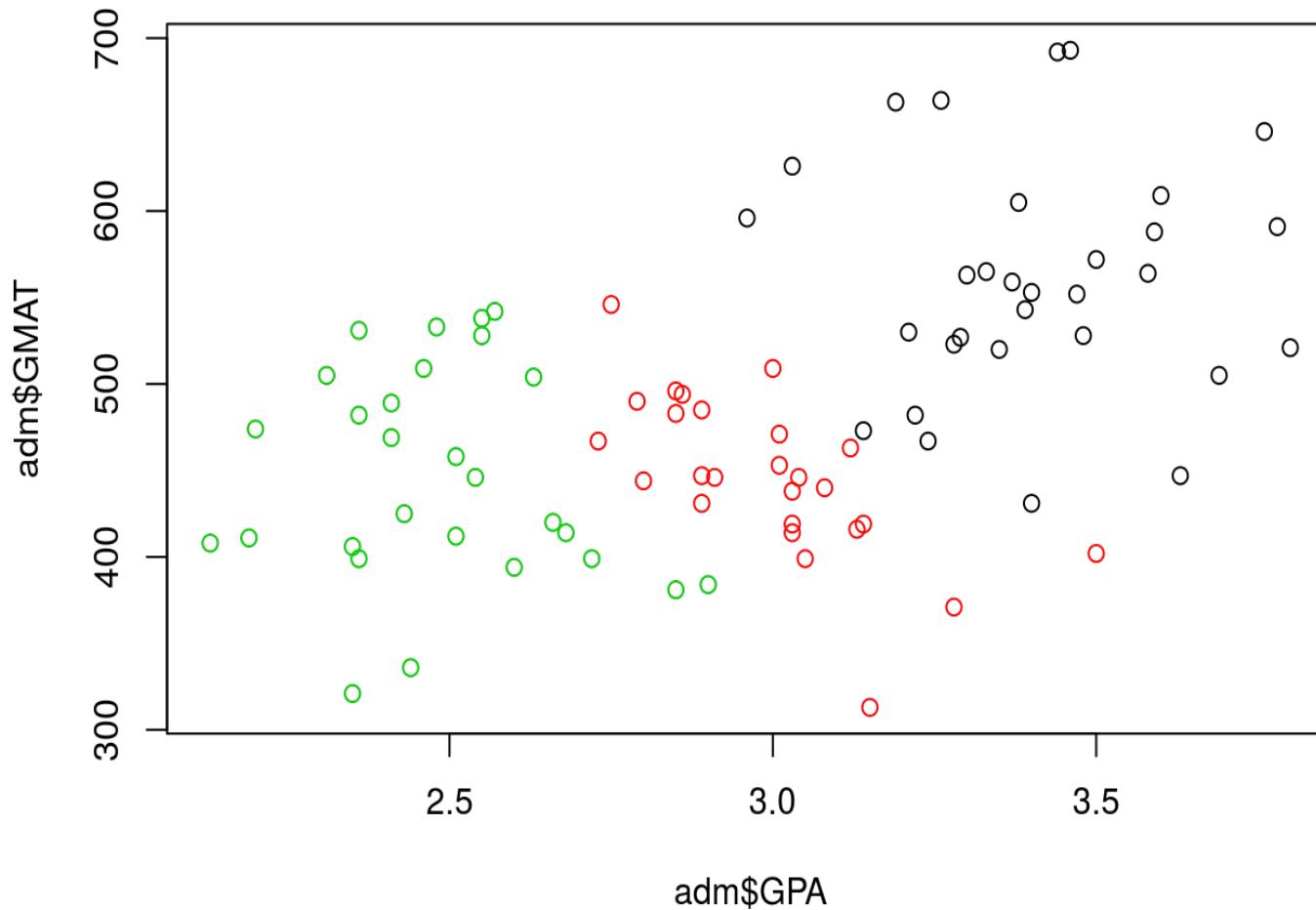
W wielowymiarowej przestrzeni danych  $S$  znajdują się przykłady  $\mathbf{x}$  stanowiące próbę uczącą  $D$ , należące do dwóch klas

$$D = \{(\mathbf{x}_i, c_i) \mid x_i \in R^p, c_i \in \{1, -1\}\}_{i=1}^N$$

Szukamy klasyfikatora pozwalającego na podział całej przestrzeni  $S$  na dwa rozłączne obszary odpowiadające klasom  $\{1, -1\}$  oraz pozwalającego jak najlepiej klasyfikować nowe obiekty  $x$  do klas

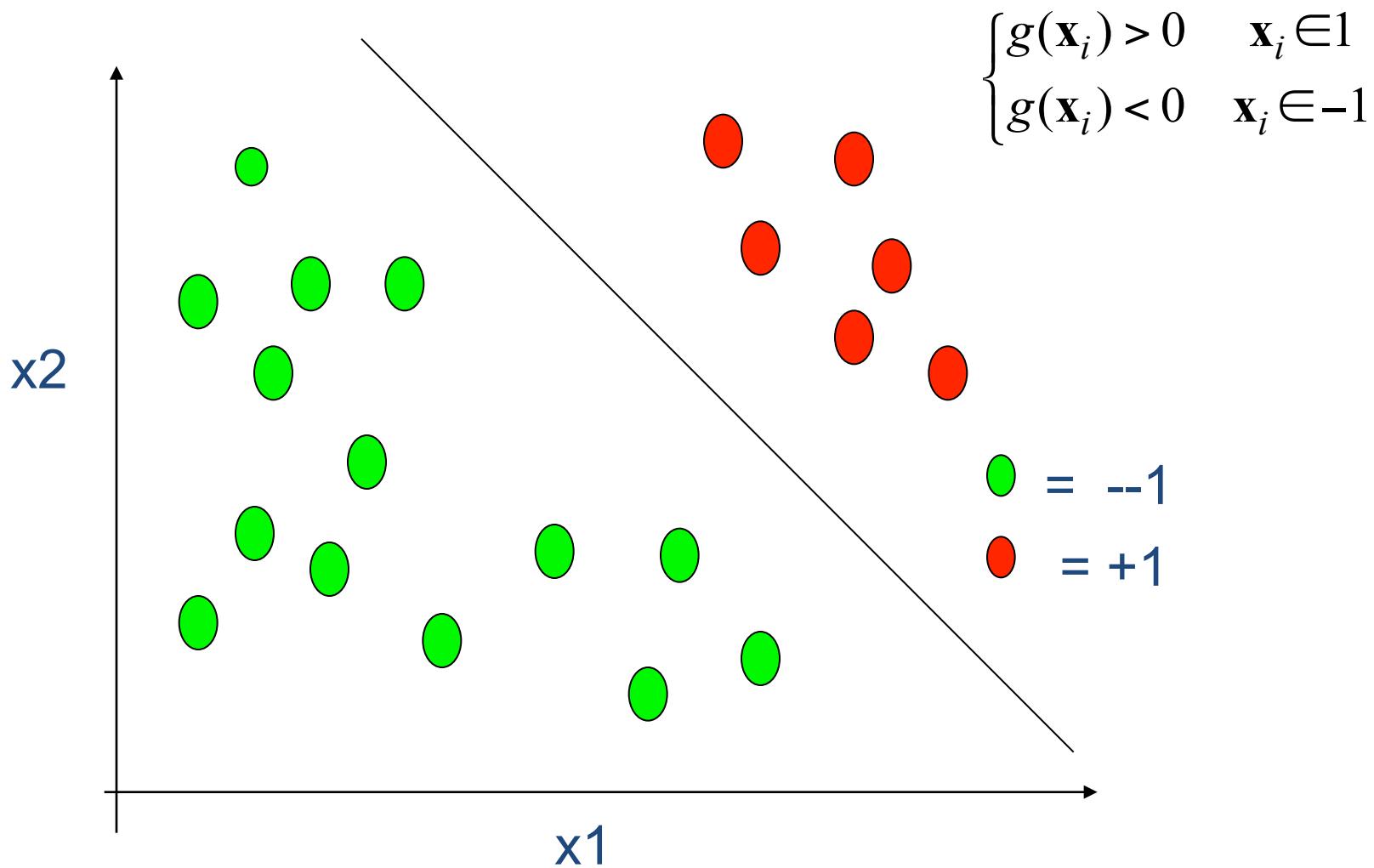
Podejście opiera się na znalezieniu tzw. granicy decyzyjnej między klasami  $\rightarrow$  funkcja  $g(x)$

# Przykład – dokumentacja R nt. Analizy dyskryminacyjnej



- Przyjęcie do amerykańskich uczelni biznesowych – wybrane atrybuty wskaźnikowe GPA oraz GMAT oraz trzy klasy kandydatów (admit, notadmit, and borderline).
  - '<http://www.biz.uiowa.edu/faculty/jledolter/DataMining/admission.csv>'

# Separowalność liniowa



# Sformułowanie problemu Fisher LDA

Cel

- Maksymalizuj odległość zrzutowanych średnich klas
- Minimalizuj wariancje wewnętrz klasową
- Odległość między rzutami średnich  
$$(\mathbf{w}^T \bar{\mathbf{x}}_1 - \mathbf{w}^T \bar{\mathbf{x}}_2)^2$$
- Fisher założył, że obie klasy mają taką samą macierz kowariancji  $S=S_1+S_2$ . Dlatego wskaźnik zmienności wewnętrzgrupowej (wspólnej dla obu klas) zdefiniowany jest jako:

$$S_W = \frac{1}{n-2} \sum_{k=1}^2 (n_k - 1) S_k$$

- Pamiętaj, że po rzutowaniu mamy  $\mathbf{w}^T S_W \mathbf{w}$

# Sformułowanie problemu Fiszerowskiej LDA

- W celu maksymalizacji odległości rzutów średnich klas i minimalizacji wariancji wewnętrzklasowej należy poszukiwać wektora w który maksymalizuje następujące wyrażenie:

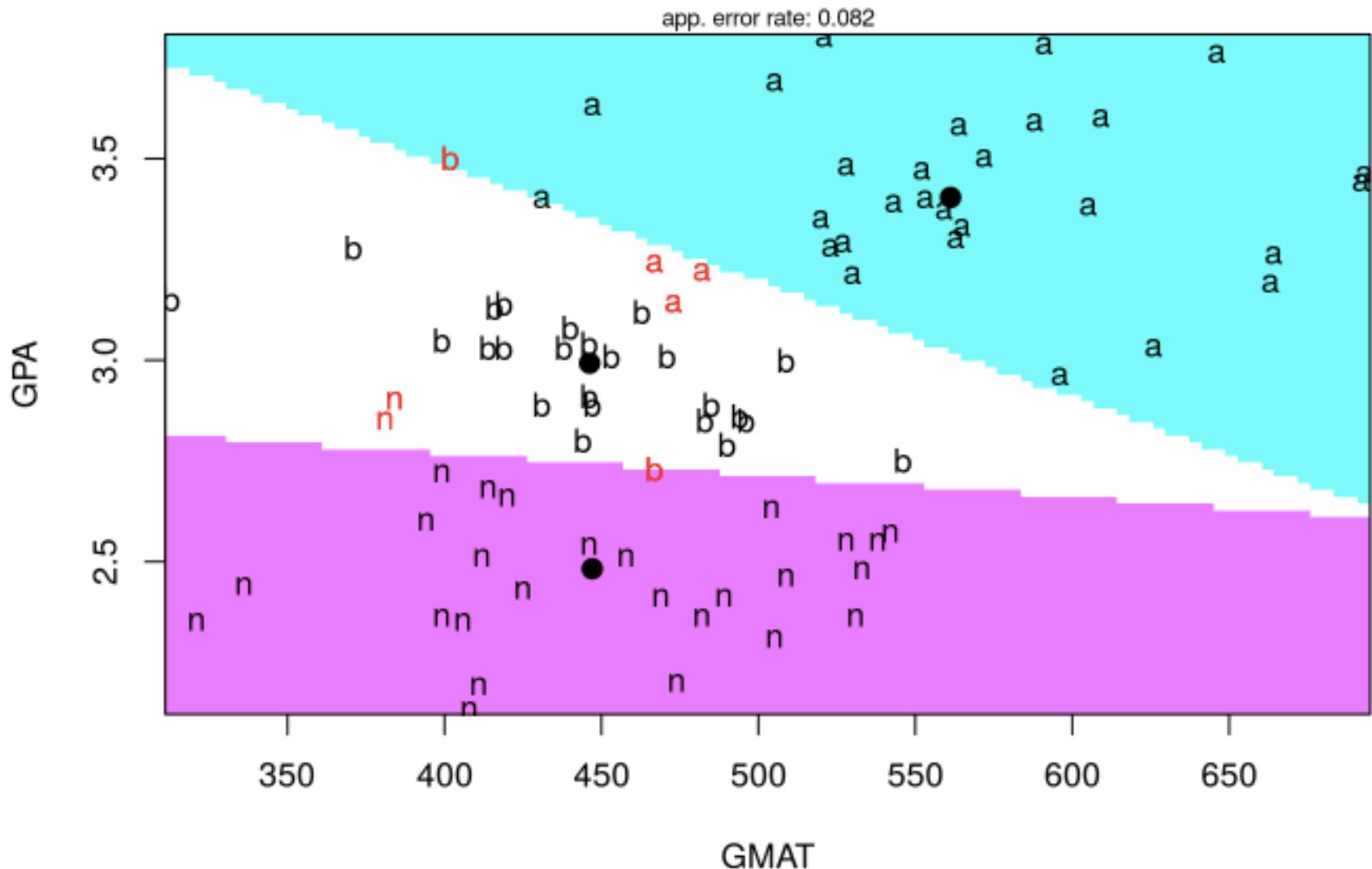
$$J(\mathbf{w}) = \frac{(\mathbf{w}^T \bar{\mathbf{x}}_1 - \mathbf{w}^T \bar{\mathbf{x}}_2)^2}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}$$

- $\mathbf{S}_w$  – macierz kowariancji
- Ostatecznie liniowa funkcja dyskryminacyjna Fishera:

$$y = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^T S_W^{-1} [\mathbf{x} - \frac{1}{2}(\bar{\mathbf{x}}_1 + \bar{\mathbf{x}}_2)]$$

- Dalej: uogólnienia na wiele klas;
- Quadratic Linear Disciminant Analysis
- Sformułowanie Bayesowskie – patrz wykłady na przedmiocie obieralnym inżynierskim

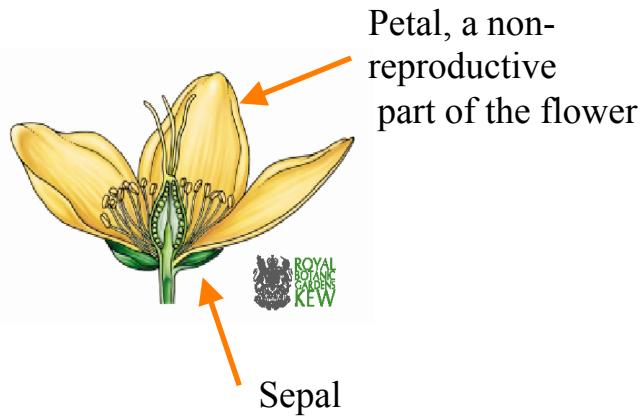
# Dane o przyjęciach do uczelni biznesowych



# Przykład – dane IRIS

## R. Fisher iris data set

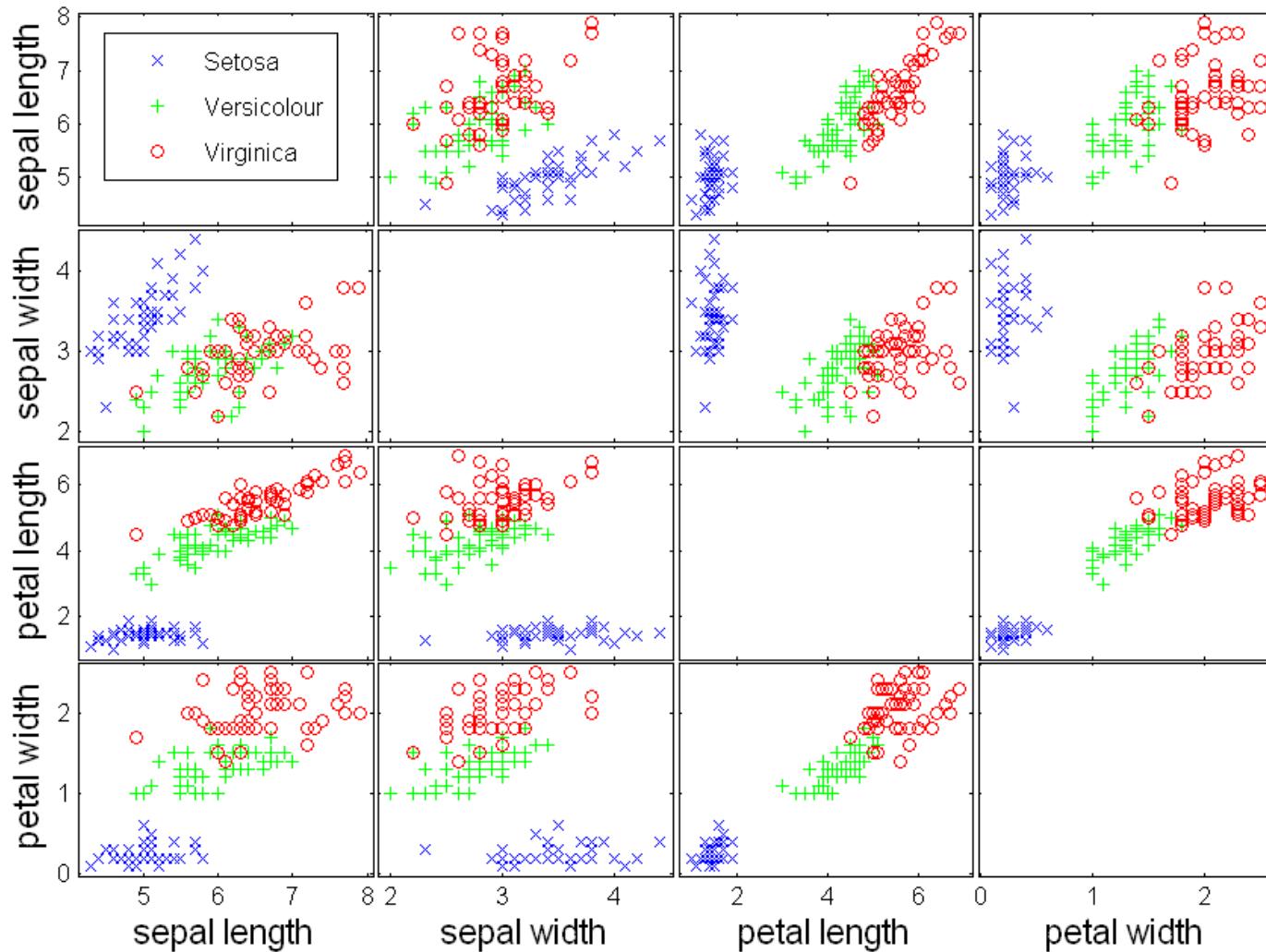
- 150 observations of 4 variables (length, width of petal and sepal)



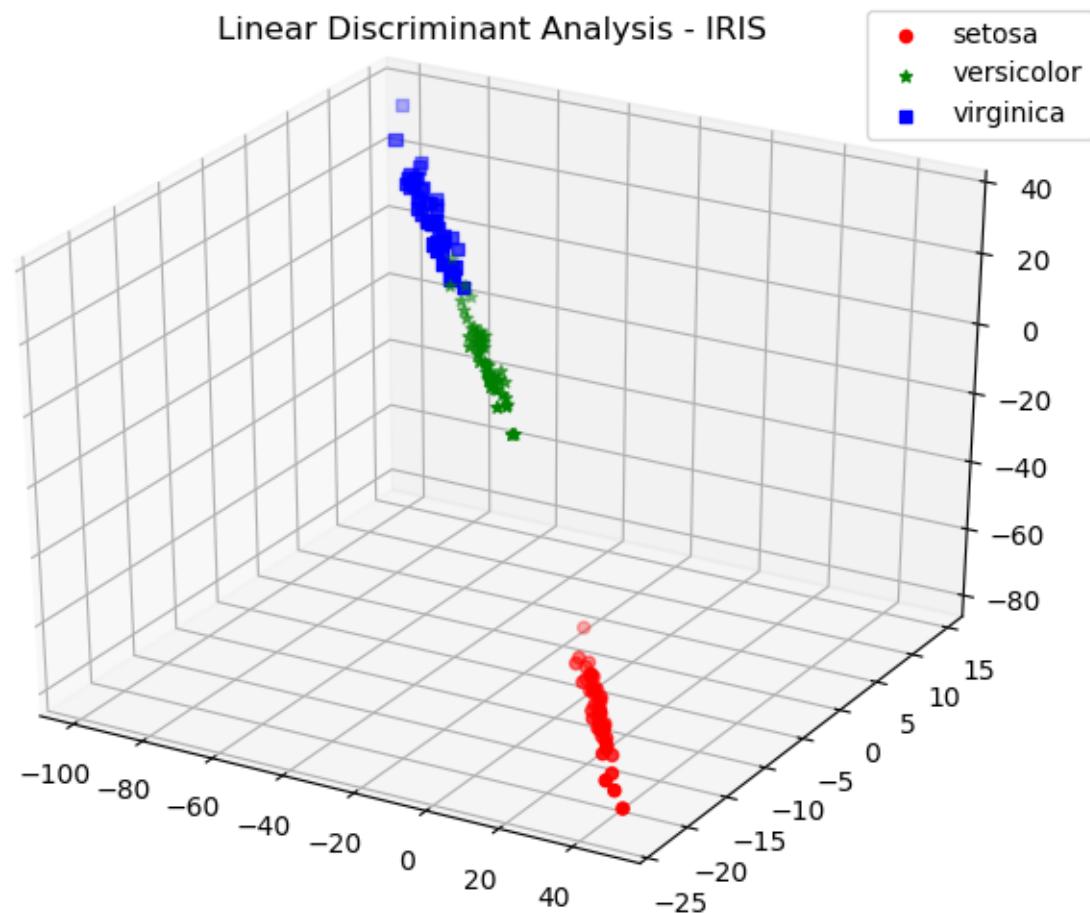
4	150			
sepal_length				
sepal_width				
petal_length				
petal_width				
4.3	7.9	5		
2.0	4.4	5		
1.0	6.9	5		
0.1	2.5	5		
5.1	3.5	1.4	0.2	
4.9	3	1.4	0.2	
4.7	3.2	1.3	0.2	
4.6	3.1	1.5	0.2	
5	3.6	1.4	0.2	
5.4	3.9	1.7	0.4	
4.6	3.4	1.4	0.3	
5	3.4	1.5	0.2	
4.4	2.9	1.4	0.2	
4.9	3.1	1.5	0.1	
5.4	3.7	1.5	0.2	
4.8	3.4	1.6	0.2	
4.8	3	1.4	0.1	
4.3	3	1.1	0.1	
5.8	4	1.2	0.2	
5.7	4	1.5	0.4	

Challenge in visualization is to design the visualization to match the analytical task

# Scatter Plot Array of Iris Attributes

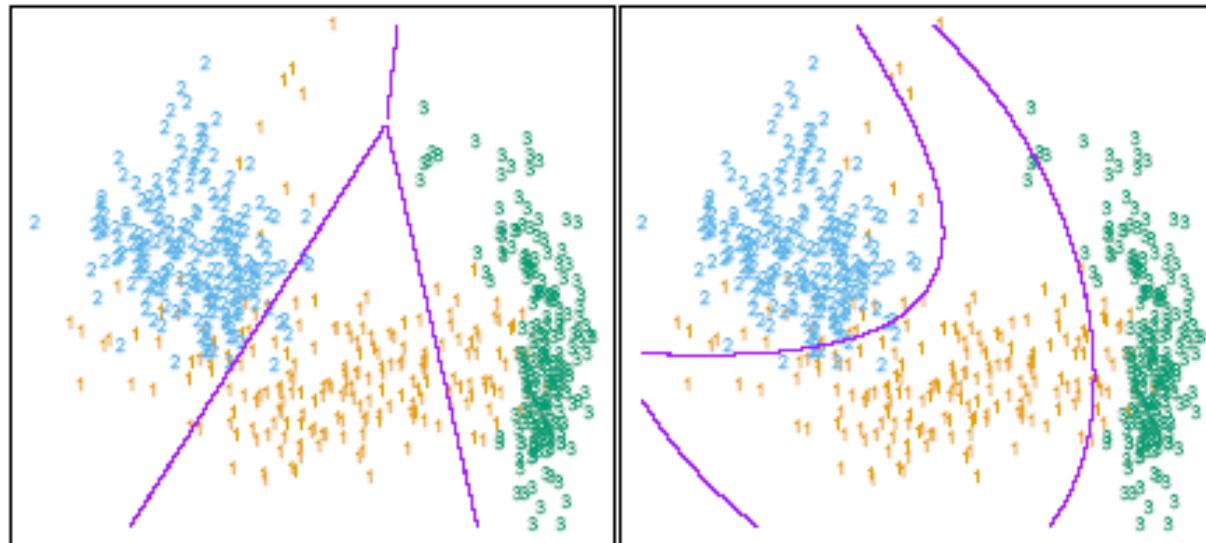


### Linear Discriminant Analysis - IRIS



# Porównanie rozwiązań LDA i QDA

Wybrany zbiór danych (za Hastie et al. Elements of Statistical Learning)



**FIGURE 4.1.** The left plot shows some data from three classes, with linear decision boundaries found by linear discriminant analysis. The right plot shows quadratic decision boundaries. These were obtained by finding linear boundaries in the five-dimensional space  $X_1, X_2, X_1X_2, X_1^2, X_2^2$ . Linear inequalities in this space are quadratic inequalities in the original space.

# Wymogi stosowania modeli AD

- Zmienne wyrażone na skalach liczbowych
  - Specjalne podejścia dla zmiennych jakościowych  
(binaryzacja, model lokacyjny,...)
- Zmienne mają wielowymiarowy rozkład normalnych
- Macierze kowariancji dla poszczególnych klas są równe → jeśli nie, to bardziej złożone funkcje kwadratowe dyskryminujące.
- Problem doboru właściwych zmiennych.

# Selekcja zmiennych

- W funkcji dyskryminującej uwzględniaj zmienne o dobrych właściwościach dyskryminujących
- Przykład kryterium jakości dyskryminacji:

$$\lambda = \frac{|S_w|}{|S_W + S_B|}$$

gdzie macierz zmienności wewnętrzklasowej

$$S_W = \frac{1}{n-k} \sum_{j=1}^k \sum_{i \in C_j} (x_i - \bar{x}_j)(x_i - \bar{x}_j)^T$$

a macierz zmienności międzyklasowej

$$S_B = \frac{1}{k-1} \sum_{j=1}^k n_j (\bar{x}_j - \bar{x})(\bar{x}_j - \bar{x})^T$$

# Więcej

Przeczytaj literaturę

- T.Hastie, R.Tibshirani, J.Friedman: The Elements of Statistical Learning. Springer (zwłaszcza rozdz. 4) → poszukaj wersji elektronicznej pdf
- J.Koronacki, J.Ćwik: Statystyczne systemy uczące się (rozdz. 1 oraz o FDA w rozdz. 6)
- M.Krzyśko, W.Wołyński, T.Górecki,M.Skorzybut: Systemy uczące się. + wcześniejsze prace M.Krzyśko o analizie dyskryminacyjnej
- **Angielska Wikipedia „Linear discriminant analysis”**
- McLachlan, G. J. (2004). Discriminant Analysis and Statistical Pattern Recognition. Wiley.
- Duda, R. O.; Hart, P. E.; Stork, D. H. (2000). Pattern Classification (2nd ed.). Wiley

# Separowalność liniowa - co dalej?

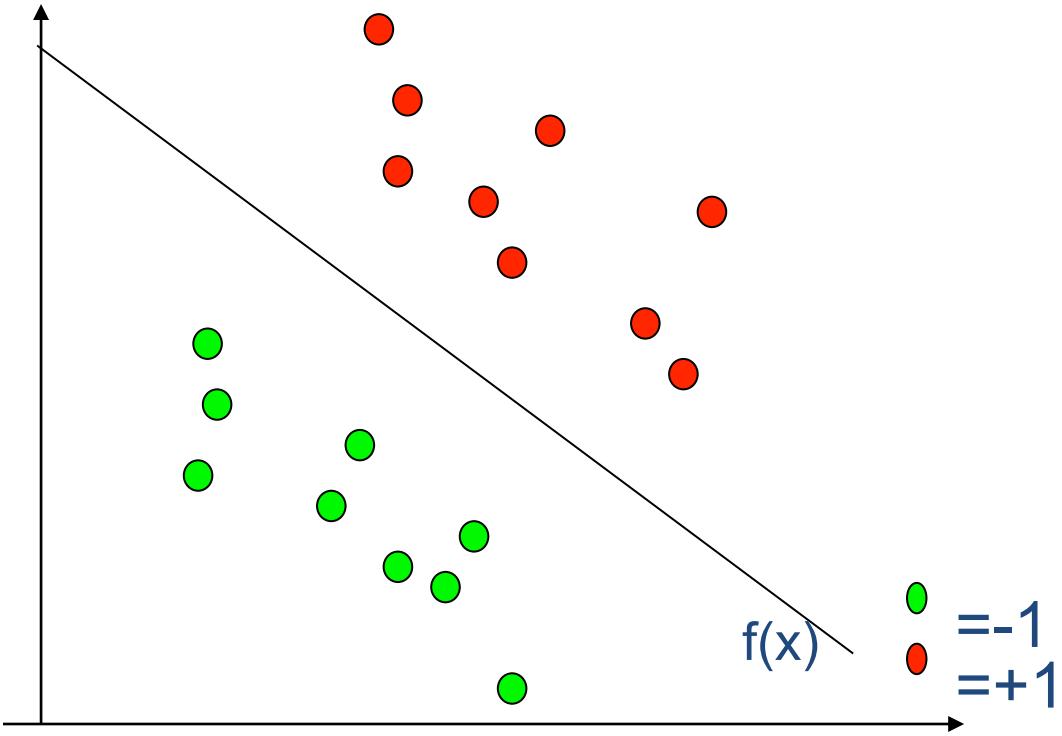
- Wiele rozwiązań – potencjalnie nieskoczenie
- Niektóre granice decyzyjne mogą być lepsze niż inne, z uwagi na pozycje wobec rozkładu przykładów z klas.
- Przejdźmy w stronę metody SVM (V.Vapnik) – podejście dyskryminacyjne (nie probabilistyczne) – specjalne podejście optymalizacji wektora wag modelu liniowego granicy decyzyjnej

# SVM kontekst

Nowe spojrzenie na tworzenie klasyfikatorów liniowych

- Przed latami 80tymi
  - Popularność modeli liniowych (często w algorytmach, także analizy teoretyczne)
- Lata 80te
  - Początek popularności ANN, drzew decyzyjnych, innych metod heurystycznych (brak optymalności)
- Od lat 90tych
  - Nowe algorytmy oparte na teorii systemów uczących
  - Optymalność przy pewnych założeniach
  - Uogólnienie na problemy nieliniowe z wykorzystaniem tzw. kernelizacji

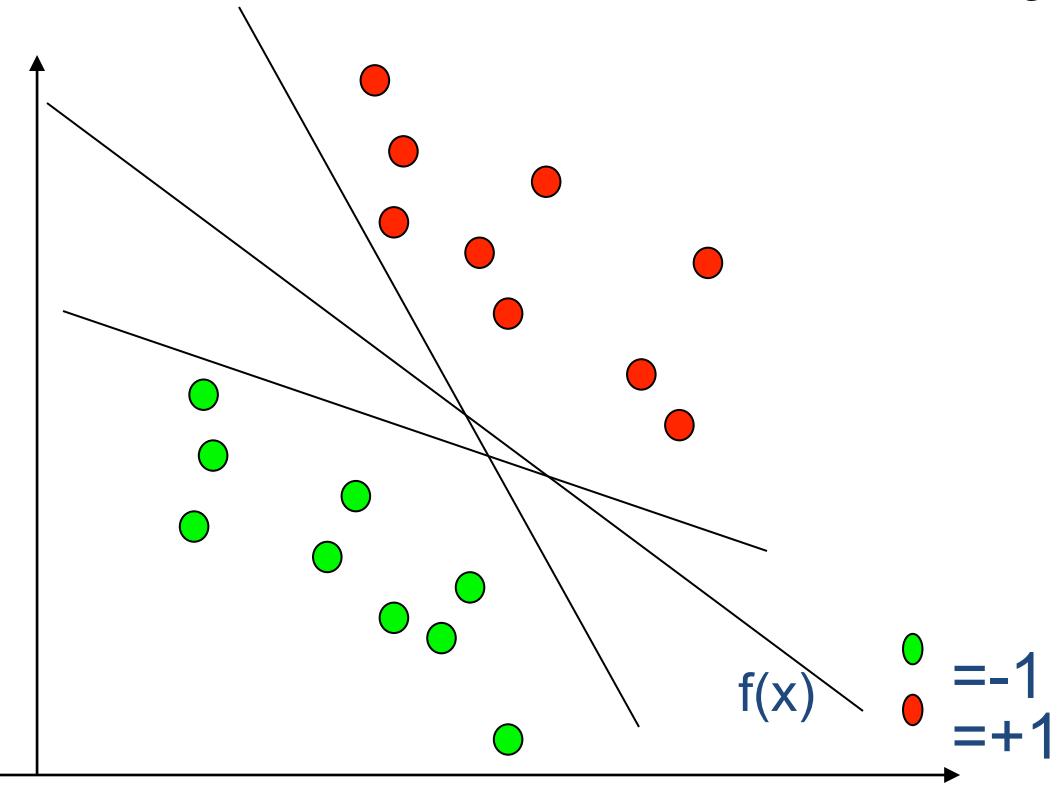
# Support Vector Machines – V.Vapnik



Znajdź liniową hiperpłaszczyznę (ang. decision boundary) oddzielającą obszary przykładów z dwóch różnych klas

Rysunek - jedno z możliwych rozwiązań, lecz pomyśl czy mogą być inne granice decyzyjne?

# Liniowa wersja SVM



Dane:  $\langle \mathbf{x}_i, y_i \rangle$ ,  $i=1, \dots, l$   
 $\mathbf{x}_i \in \mathbb{R}^d$   
 $y_i \in \{-1, +1\}$

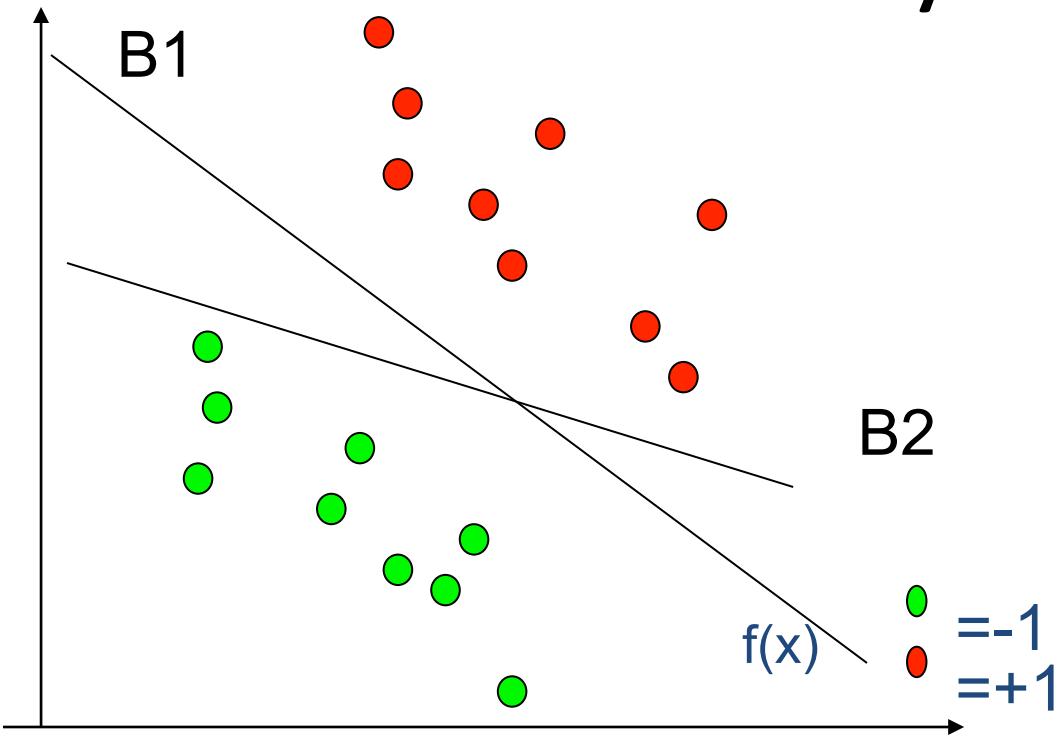
Istnienie nieskończenie wiele hiperpłaszczyzn separujących

Która wybrać?

Być może rozważyć wiele i „złożyć” je jako ważone podejście Bayesowskie? – nie jest to efektywne.

Poszukujemy jednej hiperpłaszczyzny!

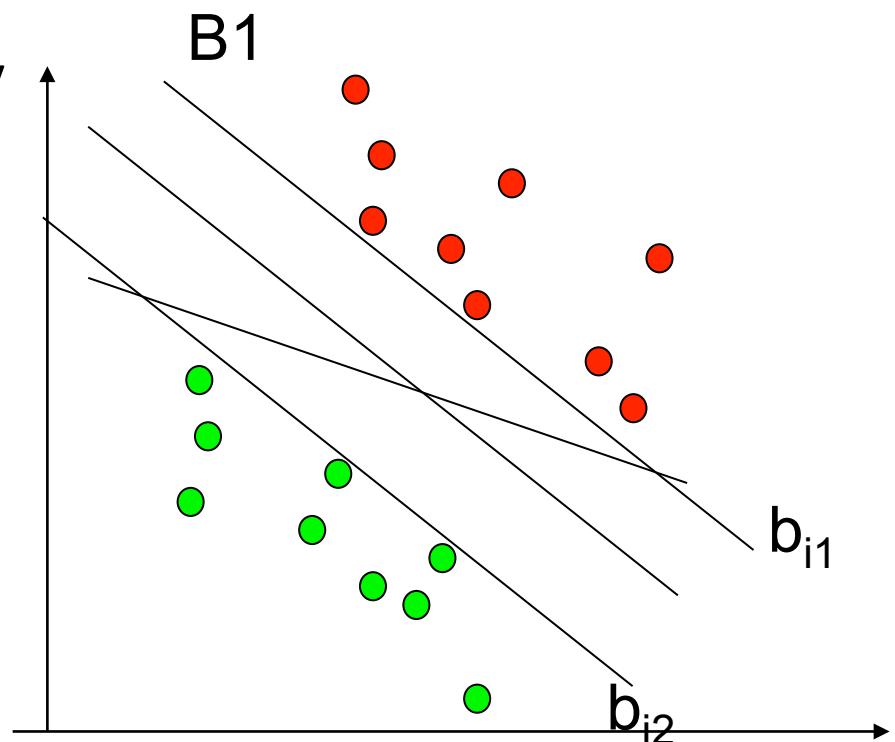
# Liniowy SVM



Któżą hiperpłaszczyznę  $B1$  czy  $B2$  – byśmy wybrali i dlaczego?

# Uwagi o marginesie w SVM

- Hiperpłaszczyzny  $b_{i1}$  i  $b_{i2}$  są otrzymywane przez równoległe przesuwanie hiperpłaszczyzny granicznej aż do pierwszych punktów z obu klas.
- Odległość między nimi – **margines klasyfikatora liniowego**
  - Odpowiednik odległości pomiędzy granicą a najbliższymi przykładami
- Jaki margines wybierać?



# Uwagi o marginesie w SVM

- Hiperpłaszczyzny  $b_{i1}$  i  $b_{i2}$  są otrzymywane przez równoległe przesuwanie hiperpłaszczyzny granicznej aż do pierwszych punktów z obu klas.
- Odległość między nimi – **margines klasyfikatora liniowego**
- Jaki margines wybierać?

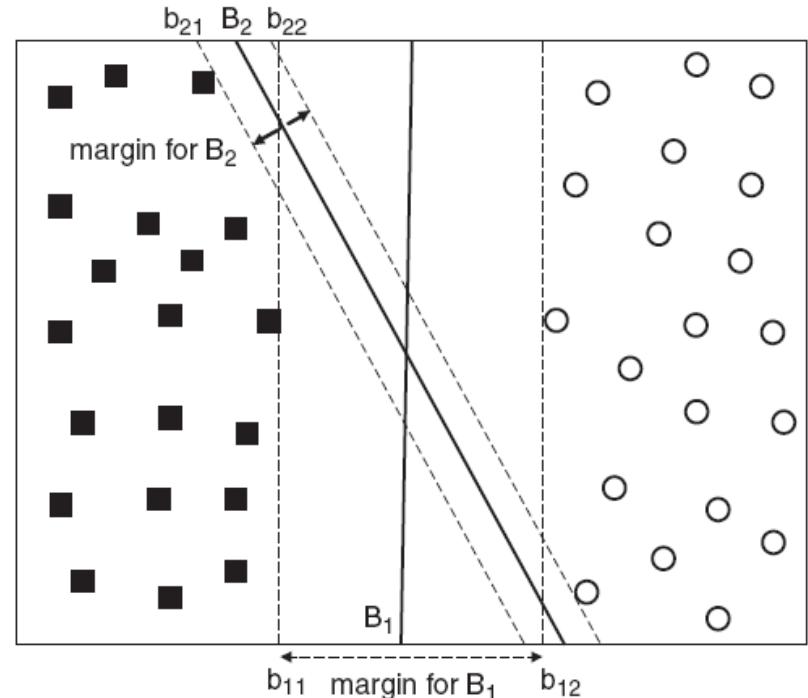


Figure 5.22. Margin of a decision boundary.

# Węższe czy szersze marginesy?

- Szerszy margines → lepsze właściwości generalizacji, mniejsza podatność na ew. przeuczenie (overfitting)
- Wąski margines – mała zmiana granicy, radykalne zmiany klasyfikacji
- Bardziej formalnie tzw. wymiar VC, teoria Vapnik–Chervonenkis
- Znajdź hiperpłaszczyznę, która maksymalizuje tzw. margines => B1 jest lepsze niż B2
- Uwaga – założenie w obszarze marginesu nie ma przykładów uczących (powinny być poza przesuniętymi hiperpłaszczyznami)

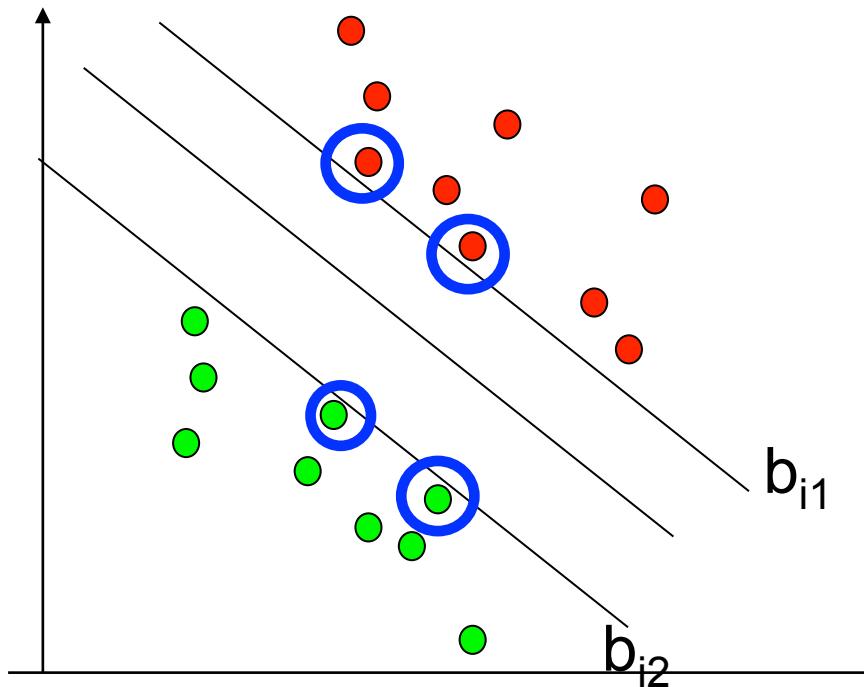
# Teoria „Structural risk minimization”

- Oszacowanie górnej granicy błędu ze względu na błąd uczący  $R_e$ , liczbę przykładów  $N$  i tzw. model complexity  $h$  z prawdopodobieństwem  $1-\eta$  „generalization error” nie przekroczy:

$$R \leq R_e + \varphi\left(\frac{h}{N}, \frac{\log(\eta)}{N}\right)$$

- Prace teoretyczne –  $h$  complexity dla modelu liniowego:
  - „Models with small margins have higher capacity - complexity because they are more flexible and can fit many training sets”
- Także „The hypothesis space with minimal **VC-dimension** according to SRM” / **teoria Vapnik–Chervonenkis**
- Reasumując modele o większej „complexity” mają gorsze oszacowanie błędu
- Dlatego wybieraj większy margines!

# Wektory nośne (ang. support vectors)



Przykłady uczące wspierające przesunięte hiperpłaszczyzny (najbliższe granicy decyzyjnej) ; Ponadto będą pełnić specjalną rolę w rozwiązaniu zadania optymalizacji parametrów modelu.

# Wektory nośne

- Punkty położone najbliżej granicy decyzyjnej
- Założenie: są najtrudniejsze do nauczenia się (dyskusyjne, ale podstawa metody SVM)
- Mają bezpośredni wpływ na znalezienie optymalnego rozwiązania i położenia hiperpłaszczyyny (iloczyn wag w i cech opisu przykładu  $x$ )
  - Inne przykłady dużo mniejszy wpływ na rozwiązanie
- Optymalny margines – przypisuje największe wagi najistotniejszym cechom dla separowalności liniowej.

# Liniowe SVM hiperpłaszczyzna graniczna

- Vapnik – poszukuj „maximal margin classifier”

$$\mathbf{w} \cdot \mathbf{x} + \mathbf{b} = 0$$

gdzie  $\mathbf{w}$  i  $\mathbf{b}$  są parametrami modelu

$$y = \begin{cases} 1 & \mathbf{w} \cdot \mathbf{x} + \mathbf{b} > 0 \\ -1 & \mathbf{w} \cdot \mathbf{x} + \mathbf{b} < 0 \end{cases}$$

- Parametry granicy wyznaczaj tak, aby maksymalne marginesy odpowiadały  $b_{i1}$  i  $b_{i2}$  i były miejscem geometrycznym punktów  $\mathbf{x}$  spełniających warunki

$$b_{i1} \quad \mathbf{w} \cdot \mathbf{x} + \mathbf{b} = 1$$

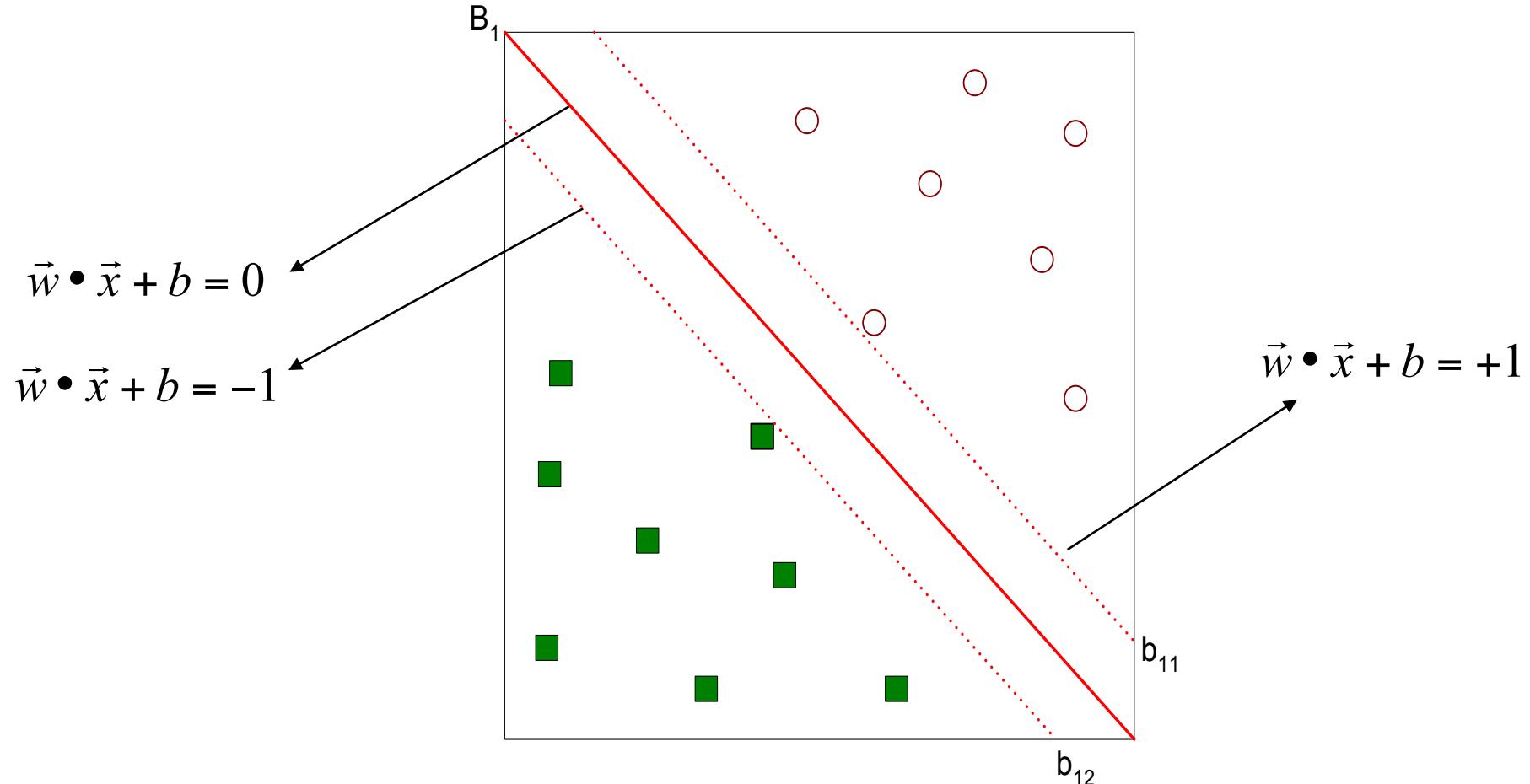
$$b_{i2} \quad \mathbf{w} \cdot \mathbf{x} + \mathbf{b} = -1$$

- **Margines** – odległość między płaszczyznami  $b_{i1}$  i  $b_{i2}$

# Komentarze

- hiperpłaszczyzny  $b_{i1}$  i  $b_{i2}$  muszą zawierać wektory nośne
- H – granica decyzyjna – „mediana” między nimi
- $b+$  najmniejsza odległość z H do najbliższych przykładów pozytywnych
- $b-$  najmniejsza odległość z H do najbliższych przykładów negatywnych
- Tylko przesunięcie wektorów nośnych może zmienić położenie hiperpłaszczyn
- Odległość między  $b_{i1}$  i  $b_{i2}$  (margines) może być przeformułowane na  $2/||\mathbf{w}|| = \text{norma wektora wag}$

# Poszukiwanie parametrów hiperpłaszczyzny



$$f(\vec{x}) = \begin{cases} 1 & \text{if } \vec{w} \cdot \vec{x} + b \geq 1 \\ -1 & \text{if } \vec{w} \cdot \vec{x} + b \leq -1 \end{cases}$$

# L-SVM ilustracja i margines

$$f(\vec{x}) = \begin{cases} 1 & \text{if } \vec{w}\vec{x} + b \geq 1 \\ -1 & \text{if } \vec{w}\vec{x} + b \leq -1 \end{cases}$$

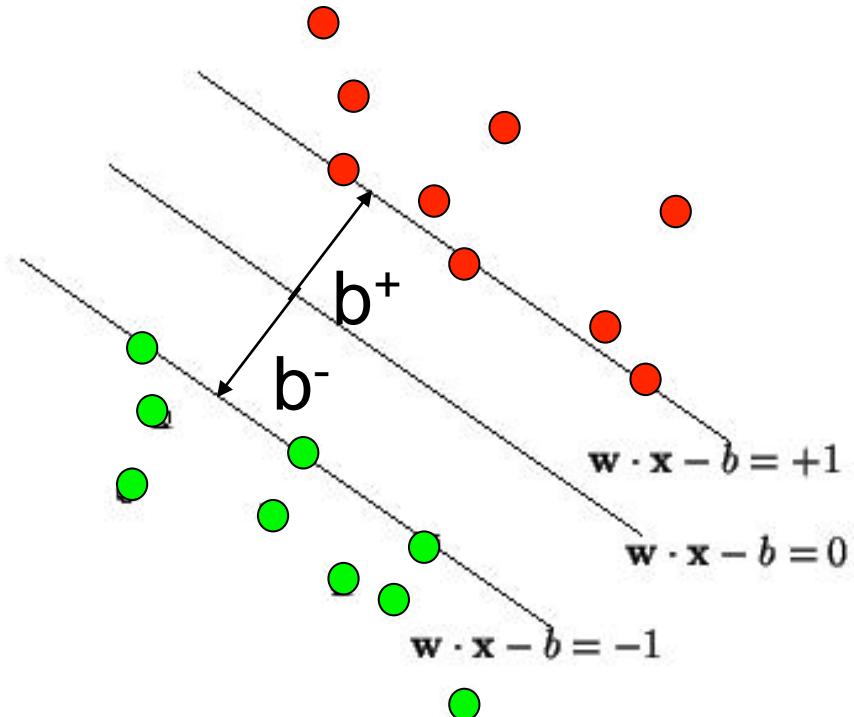
Przekształcenia marginesu:

$$\vec{w}\vec{x}_1 + b = +1 \text{ oraz } \vec{w}\vec{x}_2 + b = -1 \Rightarrow$$

$$\vec{w}(\vec{x}_1 - \vec{x}_2) + b - b = +1 - (-1)$$

$$\vec{w}(\vec{x}_1 - \vec{x}_2) = 2 \Rightarrow$$

$$\frac{\vec{w}}{\|\vec{w}\|}(\vec{x}_1 - \vec{x}_2) = \frac{2}{\|\vec{w}\|}$$



$$\text{margin} = \frac{2}{\|\vec{w}\|}$$

# Ilustracje i sposób przekształceń

$$\text{margin} = \frac{2}{\| \mathbf{w} \|} \quad \| \mathbf{w} \| \equiv \sqrt{{w_1}^2 + \dots + {w_p}^2}$$

**Cel: Maksymalizuj margines!**

$$\frac{2}{\| \mathbf{w} \|} \longrightarrow \frac{\| \mathbf{w} \|}{2} \longrightarrow \frac{\| \mathbf{w} \|^2}{2}$$

maximize      minimize      **minimize**

# L-SVM zadanie optymalizacji

Sformułowanie mat. problemu:

$$\min_{\mathbf{w}} = \frac{\|\mathbf{w}\|^2}{2}$$

- Przy warunkach ograniczających

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \quad i = 1, 2, \dots, N$$

- Jest to problem optymalizacji kwadratowej z liniowymi ogr. → uogólnione zadanie optymalizacji rozwiązywany metodą mnożników Lagrange'a (tak aby, np. nie dojść do  $\mathbf{w} \rightarrow 0$ )

# LSVM – mnożniki Lagrange'a

- Minimalizuj funkcję Lagrange'a

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^N \alpha_i (y_i (w \cdot x_i + b) - 1)$$

- parametry  $\alpha \geq 0$  mnożniki Lagrange'a
- Powinno się różniczkować  $L$  po  $w$  i  $b$  – nadal trudności w rozwiązaniu
- Przy przekształceniach wykorzystuje się ograniczenia Karush-Kuhn-Tucker na mnożniki:

$$\alpha_i \geq 0$$

$$\alpha_i [y_i (w \cdot x_i + b) - 1] = 0$$

- W konsekwencji  $\alpha_i$  są niezerowe wyłącznie dla **wektorów nośnych**  $x$ , pozostałe są zerowe
- Rozwiążanie parametrów  $w$  i  $b$  zależy wyłącznie od wektorów nośnych!

# LSVM – sformułowanie dualne

- Nadal zbyt wiele parametrów  $\mathbf{w}$ ,  $\mathbf{b}$ ,  $\alpha$  do oszacowania

- Przechodzi się na postać *dualną zadania optymalizacji*

- Maksymalizuj  $L(\alpha)$

$$\sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j)$$

- przy ograniczeniach

$$\alpha_i \geq 0, \quad \forall i \quad \sum_{i=1}^N \alpha_i y_i = 0$$

- Rozwiązanie ( $\alpha > 0$  dla  $i \in SV$ ) ;  $b$  – odpowiednio uśredniane

$$\mathbf{w} = \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i$$

- Hiperpłaszczyzna decyzyjna

$$\sum_{i=1}^N \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b = 0$$

# Rozwiążanie LSVM

- Klasyfikacja – funkcja decyzyjna

$$f(x) = \text{sign}\left(\sum_{i=1}^N \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b\right)$$

- O ostatecznej postaci hiperpłaszczyzny decydują **wyłącznie wektory nośne ( $\alpha_i > 0$ )**
- Im większa wartość  $\alpha_i$ , tym większy wpływ wektora na granicę decyzyjną
- Klasyfikacja zależy od iloczynu skalarnego nowego  $\mathbf{x}$  z wektorami nośnymi  $\mathbf{x}_i$  ze zbioru uczącego
- Pewne założenie metody – starać się zbudować klasyfikator liniowy używając **możliwie minimalną liczbę wektorów** z danych uczących (wektory nośne)

# Przykład

## Obliczmy wagi

$$w_1 = \sum_i \alpha_i y_i x_{i1} = 65.5621 \cdot 1 \cdot 0.3858 + 65.5621 \cdot (-1) \cdot 0.4871 = -6.64$$

$$w_2 = \sum_i \alpha_i y_i x_{i2} = 65.5621 \cdot 1 \cdot 0.4687 + 65.5621 \cdot (-1) \cdot 0.611 = -9.32$$

$x_1$	$x_2$	y	Lagrange Multiplier
0.3858	0.4687	1	65.5261
0.4871	0.611	-1	65.5261
0.9218	0.4103	-1	0
0.7382	0.8936	-1	0
0.1763	0.0579	1	0
0.4057	0.3529	1	0
0.9355	0.8132	-1	0
0.2146	0.0099	1	0

$$b' = 1 - (-6.64) \cdot 0.3858 - (-9.32)(0.4687) = 7.930$$

$$b'' = -1 - (-6.64) \cdot 0.4871 - (-9.32)(0.611) = 7.928$$

$$b = 0.5 \cdot (b' + b'') = 7.93$$

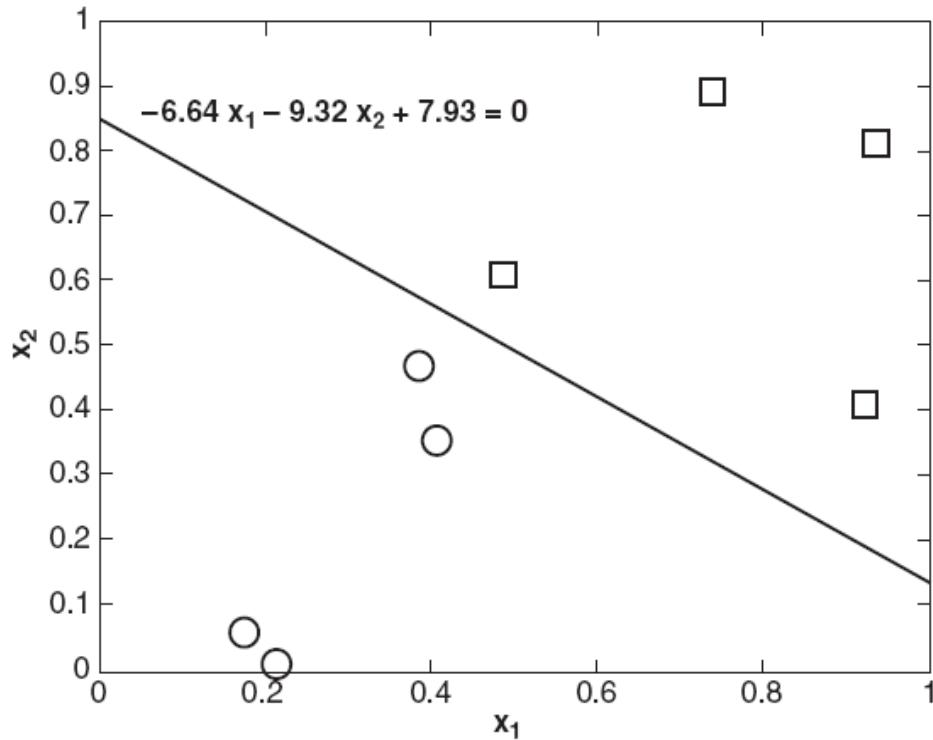
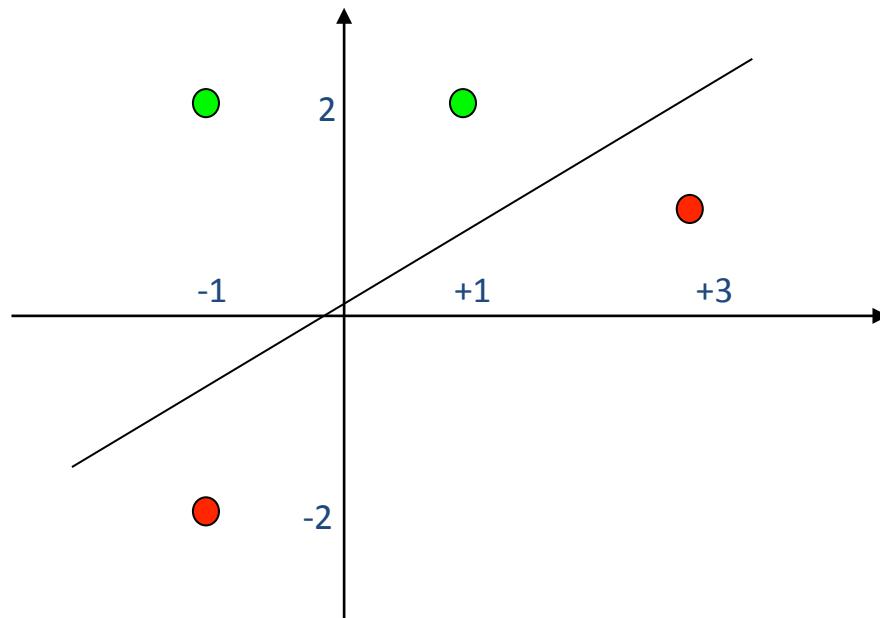


Figure 5.24. Example of a linearly separable data set.

# Inny przykład

- Rozważmy 4 dwuwymiarowe ( $a_1, a_2, y$ ) przykłady uczące:  
 $(1, 2, -1); (-1, 2, -1); (-1, -2, +1); (3, 1, +1)$
- Obliczenia współczynników Lagrange'a :  $\alpha_1=1/2$  ;  $\alpha_2 = 0$ ;  
 $\alpha_3=1/10$  ;  $\alpha_4=2/5$
- Ostateczny wektor wag  $w_1=3/5$  oraz  $w_2=-4/5$



# Parę uwag podsumowujących

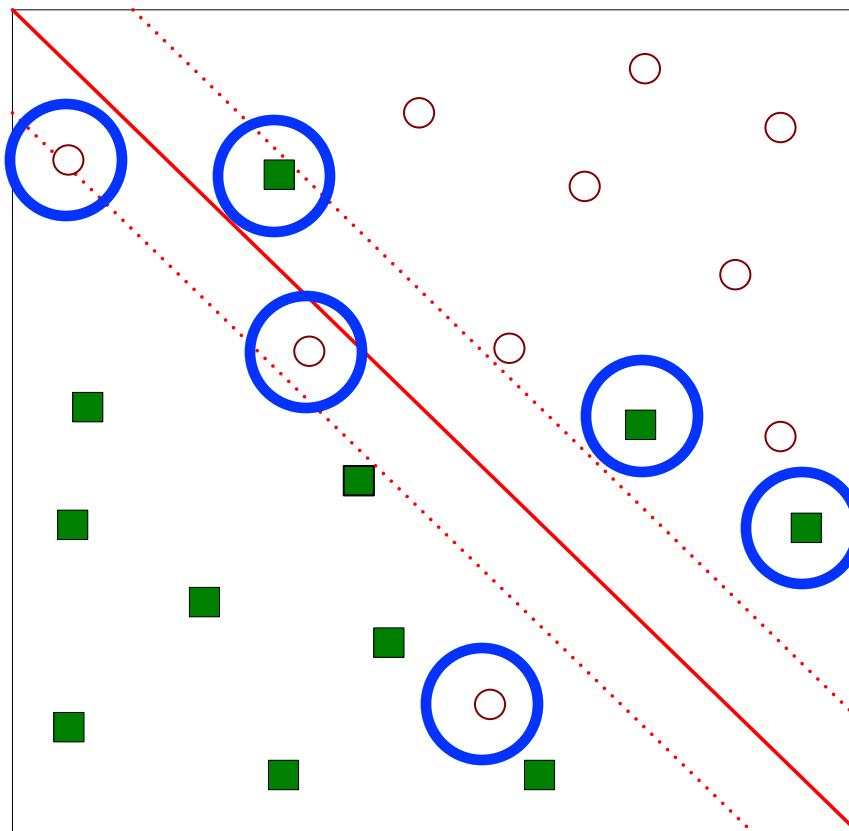
- Liniowy SVM – solidne matematycznie rozwiązanie problemu liniowo separowalnego
  - Podejście nazywane Hard margin SVM
  - Założenie o silnym znaczeniu przykładów brzegowych / inne przykłady we wnętrzu rozkładu klas nie mają takiego znaczenia (pomyśl na realnością i różnymi możliwymi rozkładami przykładów w klasach)
  - Związek z teoretycznymi pracami (np. wymiar VC)
- Użycie zmodyfikowanego sformułowanie zadania programowanie kwadratowego
  - Jedno globalne rozwiązanie
  - Algorytm optymalizacyjny
- Teoretycznie poszukuję minimum globalnego a nie lokalnego (jak podejścia heurystyczne – MLP)

# Dalsze pytania

- Liniowy SVM – praktycznie nie musi być konkurencyjny dla znanych wcześniej metod liniowych (LDA, ANN, itd.)
- Otwarte pytania na kolejny wykład:
- Jak podejść do problemów nieseparowalnych (zmienne osłabiające i przeformułowanie zadania) – tzw. soft margin SVM
- Nieliniowe, trudne rozkłady przykładów
  - Dane odwzorowane (przy pomocy funkcji jądrowych) w nową przestrzeń cech – silna przewaga nad innymi metodami
  - W nowej przestrzeni dane powinny być liniowo separowalne

# Support Vector Machines – dalej!

Co robić z LSVM gdy dane nie są w pełni liniowo separowalne?



# Odrośniki do literatury anglojęzycznej

- “Statistical Learning Theory” by Vapnik – wymaga przygotowania matematycznego.
- C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Knowledge Discovery and Data Mining*, 2(2), 1998.
  - Czytelniejsza niż powyższa książka
- R. Berwick: An Idiot’s guide to Support vector machines (SVMs)
  - łatwy wykład po ang.
- Książka “An Introduction to Support Vector Machines” by N. Cristianini and J. Shawe-Taylor
- Rozdziały w książce Hastie, Tibshirani, Friedman: Elements of statistical learning (dostępna online pdf).
- Po polsku: M.Krzyśko, T.Górecki i inni, książka pt. Systemy uczące się

# **Pytanie i komentarze?**

Dalszy kontakt:

[jerzy.stefanowski@cs.put.poznan.pl](mailto:jerzy.stefanowski@cs.put.poznan.pl)

<http://www.cs.put.poznan.pl/jstefanowski/>



**Fundusze  
Europejskie**  
Polska Cyfrowa



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz  
Rozwoju Regionalnego



# **Systemy uczące się**

# **Metoda wektorów wspierających**

## **Przekształcenie z funkcjami jądrowymi**

## **wykład 4**

Jerzy Stefanowski  
Instytut Informatyki PP

2021

Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI-TECH)  
projekt finansowany z środków Programu Operacyjnego Polska Cyfrowa  
POPC.03.02.00-00-0001/20



**Fundusze  
Europejskie**  
Polska Cyfrowa



**Rzeczpospolita  
Polska**

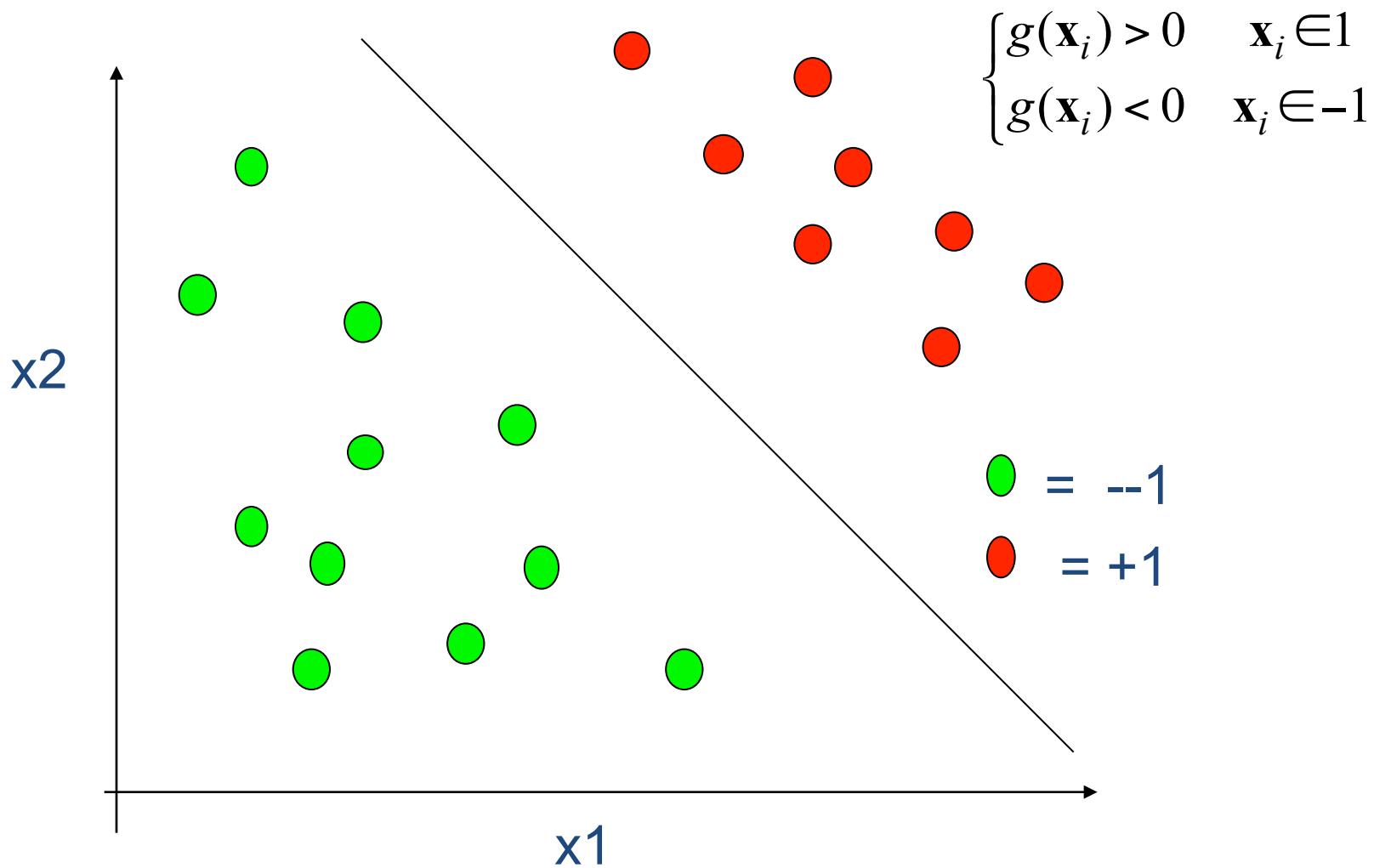
**Unia Europejska**  
Europejski Fundusz  
Rozwoju Regionalnego



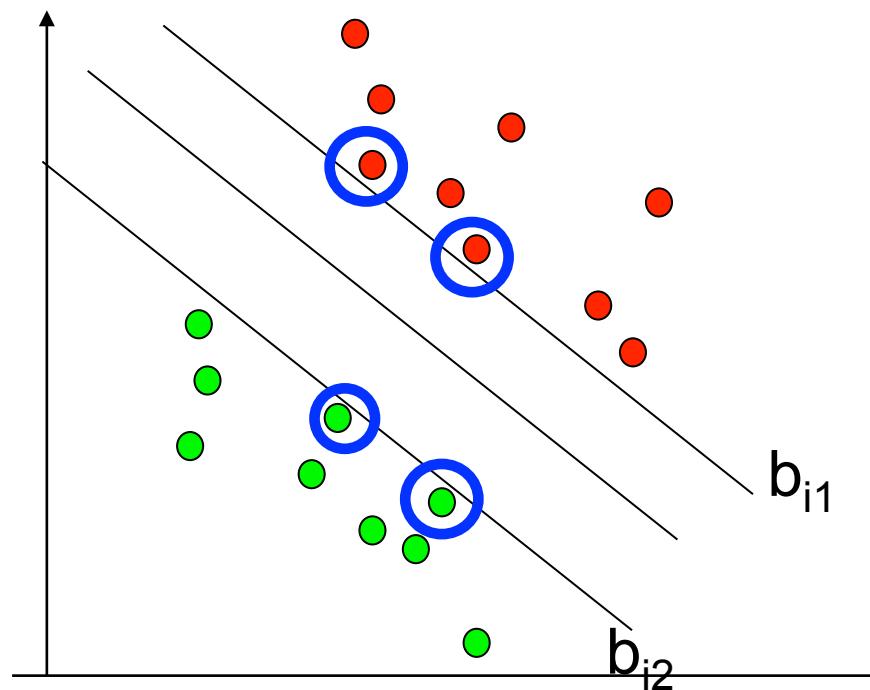
# Plan wykładu

1. Przypomnienie liniowego SVM
  1. Ogólna zasada
  2. Sformułowanie zadania optymalizacji
2. Uogólnienie SVM (dane nie w pełni separowalne liniowo)
3. Funkcje jądrowe (tzw. kernel functions)
4. SVM dla danych z nieliniowymi granicami
5. Podsumowanie
6. Inne możliwości SVM oraz kernelizacji
7. Gdzie szukać więcej?

# Poszukiwanie hiperpłaszczyzny separującej



# Wektory nośne (wspierające)



Przykłady wspierające przesunięte hiperpłaszczyzny do granic rozkładów przykładów  
(wyznaczające tzw. margines klasyfikatora liniowego)

# L-SVM zadanie optymalizacji

Sformułowanie mat. problemu:

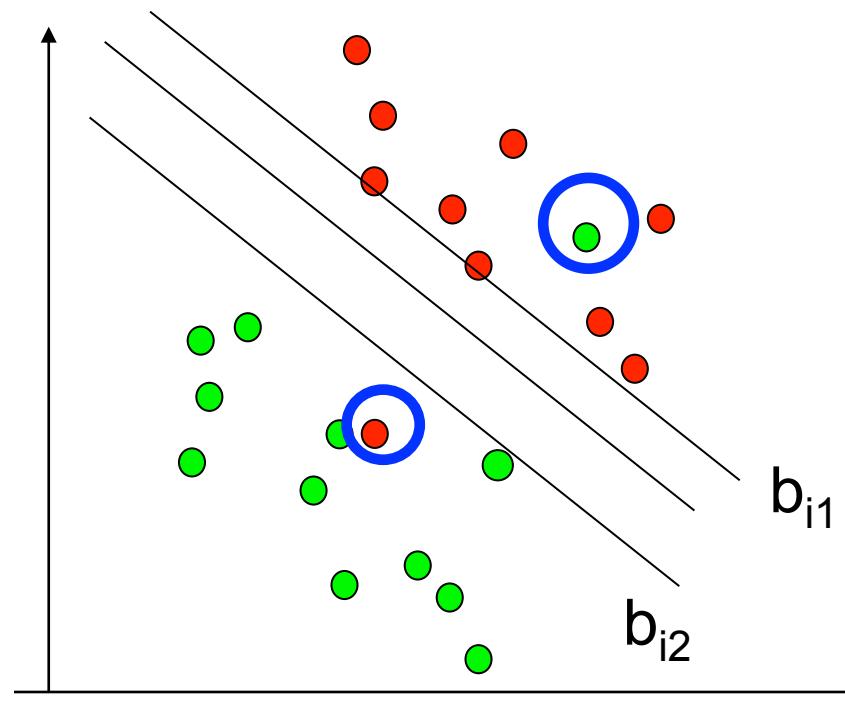
$$\min_{\mathbf{w}} = \frac{\|\mathbf{w}\|^2}{2}$$

- Przy warunkach ograniczających

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \quad i = 1, 2, \dots, N$$

- Jest to problem optymalizacji kwadratowej z liniowymi ogr.  
→ uogólnione zadanie optymalizacji rozwiązywany metodą mnożników Lagrange'a (tak aby np. nie dojść do  $\mathbf{w} \rightarrow 0$ )
- Poprzedni wykład zawiera szczegóły – przekształcenia (zadanie dualne, ...)

Dane uczące nie są liniowo separowalne  
(nakładanie się klas i przykłady położone po  
niewłaściwej stronie)



Przykłady położone po niewłaściwej stronie hiperpłaszczyzny w  
stosunku do rozkładów przykładów z danej klasy;  
Ograniczenie  $y(wx_i+b) \geq 1$  – może nie być spełnione dla niektórych  
przykładów

Zmienne osłabiające / dopełniające, tzw. slack variables – niezerowe dla przykładów położonych po niewłaściwej stronie granicy decyzyjnej

$$\begin{aligned} -in + 1 & \quad \mathbf{w} \cdot \mathbf{x} + b \geq 1 - \xi \\ +in - 1 & \quad \mathbf{w} \cdot \mathbf{x} + b \leq -1 + \xi \end{aligned}$$

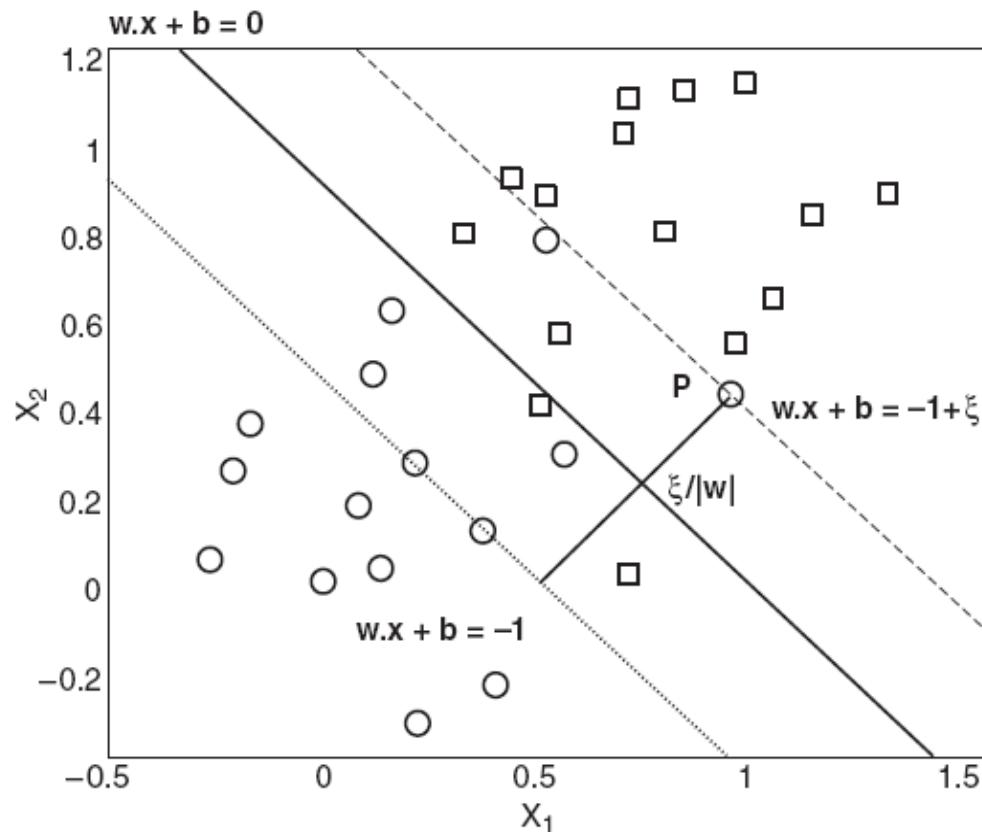


Figure 5.26. Slack variables for nonseparable data.

# Zmienne osłabiające

- Zmienne  $\xi_i \geq 0$  oceniają odstępstwa przykładu  $x_i$  od marginesu
  - Przykład  $x_i$  po niewłaściwej stronie granicy decyzyjnej i poza marginesem
  - Także przykład  $x_i$  po właściwej stronie granicy decyzyjnej lecz leży w obszarze marginesu zbyt blisko granicy
- Jeśli  $\xi_i = 0$ , to nie występują trudności z położeniem przykładu  $x_i$
- Definiuje się tzw. soft error jako sumę zmiennych  $\xi_i$  = patrz dalsze sformułowanie problemu optymalizacyjnego z dodatkowym elementem funkcji „kary”

# Zmienne osłabiające - interpretacja

- Zmienne  $\xi_i \geq 0$  dobiera się dla każdego przykładu uczącego. Jej wartość zmniejsza margines separacji (rodzaj „zwisu” punktu poza hiperpłaszczyzną nośną)
- Jeżeli  $0 \leq \xi_i \leq 1$ , to punkt danych  $(\mathbf{x}_i, d_i)$  leży wewnątrz strefy separacji, ale po właściwej stronie
- Jeżeli  $\xi_i > 1$ , punkt po niewłaściwej stronie hiperpłaszczyzny i wystąpi błąd klasyfikacji
- Modyfikacja wymagań dla wektorów nośnych

$$b_{i1} \quad \mathbf{w} \cdot \mathbf{x} + \mathbf{b} = 1 - \xi$$

$$b_{i2} \quad \mathbf{w} \cdot \mathbf{x} + \mathbf{b} = -1 + \varsigma$$

# Rola zmiennych osłabiających w optymalizacji

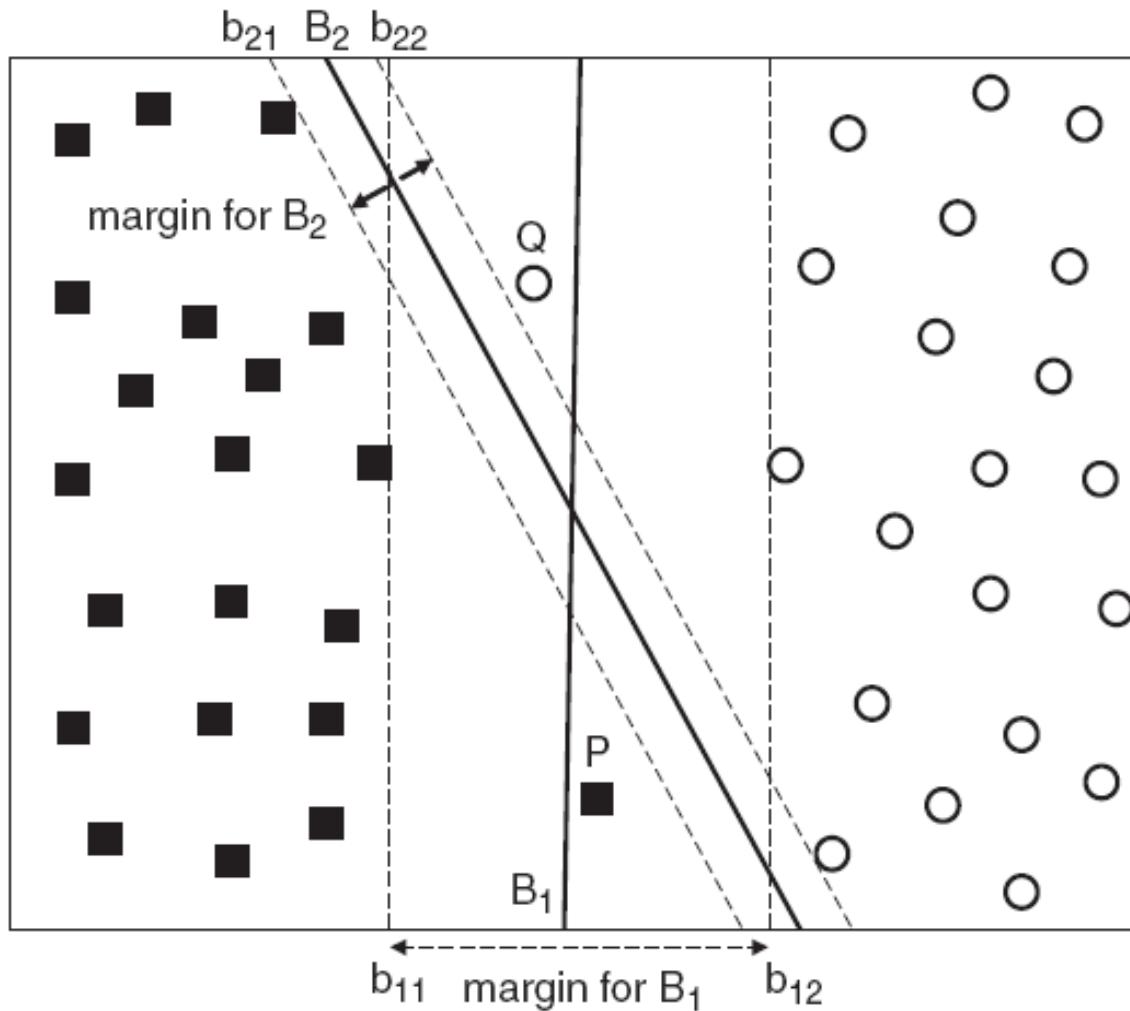


Figure 5.25. Decision boundary of SVM for the nonseparable case.

# SVM z dodatkowymi zmiennymi

- Jak przeddefiniować sformułowanie? Z dodatkowymi zmiennymi osłabiającym oraz kosztem błędu na danych uczących
  - Minimalizuj funkcję:
  - z ograniczeniami:
$$L(w) = \frac{\|\vec{w}\|^2}{2} + C \left( \sum_{i=1}^N \xi_i^k \right)$$
$$f(\vec{x}_i) = \begin{cases} 1 & \text{if } \vec{w} \cdot \vec{x}_i + b \geq 1 - \xi_i \\ -1 & \text{if } \vec{w} \cdot \vec{x}_i + b \leq -1 + \xi_i \end{cases}$$
  - Drugi czynnik odpowiada za ew. błędy klasyfikowania (górnne oszacowanie tych błędów)
  - Parametr **C** ocena straty związanej z każdym błędnie klasyfikowanym punktem dla które  $\xi > 0$
  - Przetarg „szeroki margines” to dużo błędów i odwrotnie

# Rozwiązywanie problemu - przekształcenia

Programowanie kwadratowe (QP) : trudności rozwiązania -  
przeformułuj problem

Ponownie dojdziemy do dualnego problemu:

$$\text{Max: } W(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \bullet \mathbf{x}_j)$$

przy ograniczeniach:

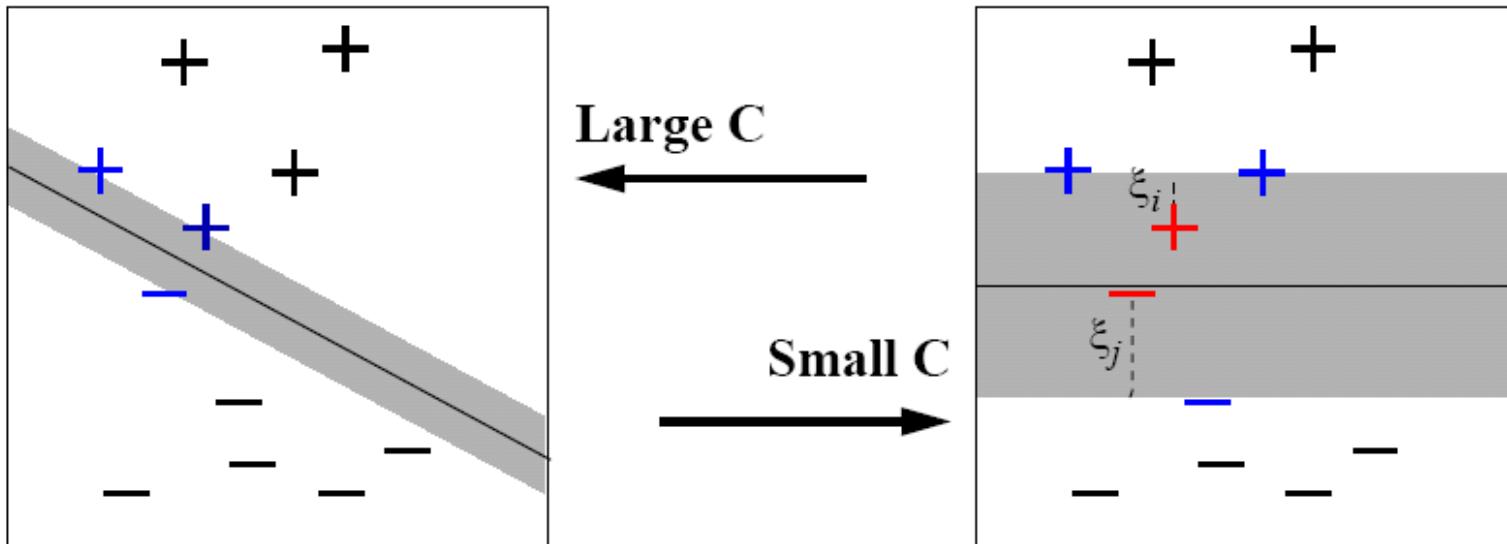
$$(1) \quad 0 \leq \alpha_i \leq C, \quad \forall i$$

$$(2) \quad \sum_{i=1}^m \alpha_i y_i = 0$$

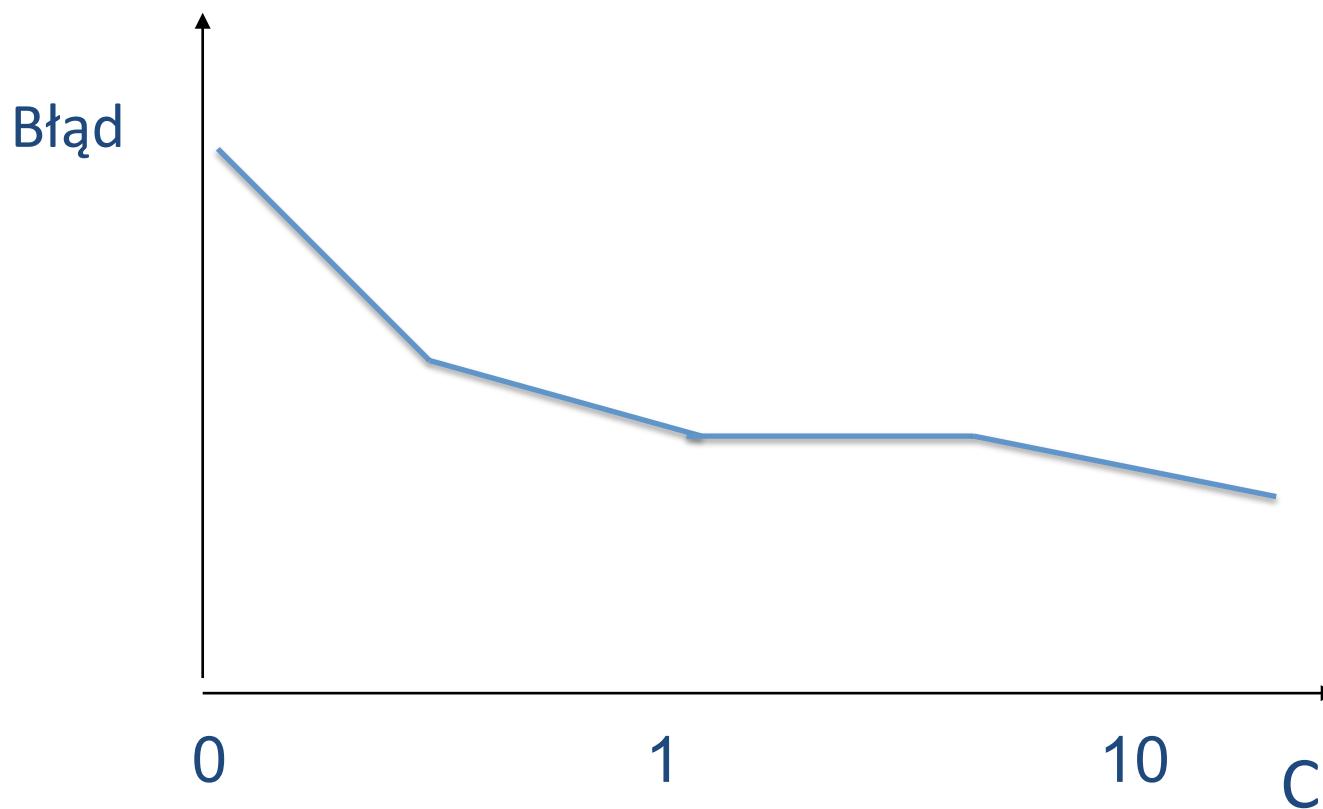
## Controlling Soft-Margin Separation

$$\begin{aligned}\textbf{Soft Margin: minimize } P(\vec{w}, b, \vec{\xi}) &= \frac{1}{2} \vec{w} \cdot \vec{w} + C \sum_{i=1}^n \xi_i \\ \text{s. t. } y_i [\vec{w} \cdot \vec{x}_i + b] &\geq 1 - \xi_i \text{ and } \xi_i \geq 0\end{aligned}$$

- $\sum \xi_i$  is an upper bound on the number of training errors.
- C is a parameter that controls trade-off between margin and error.

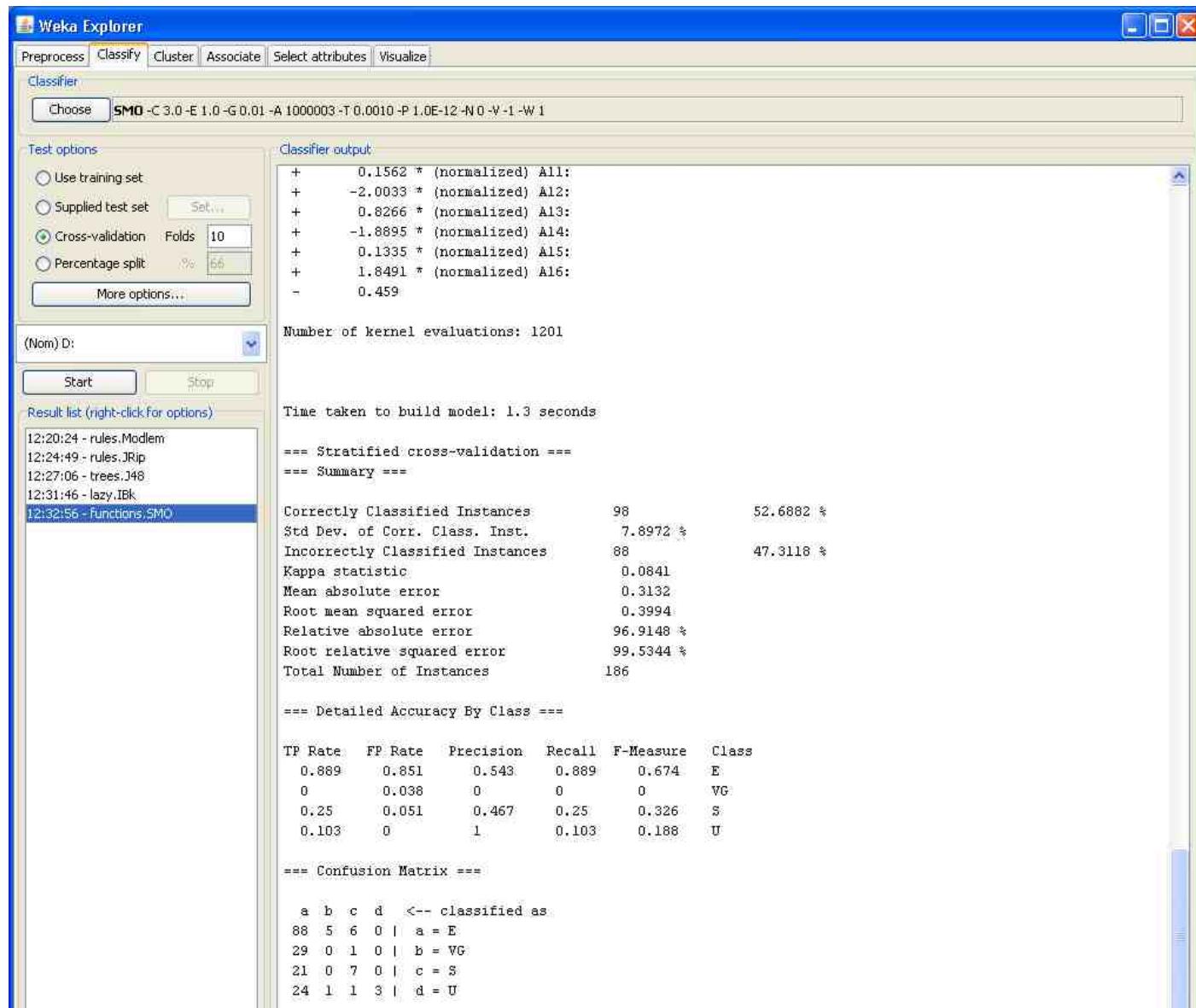


# Dosztajanie par. C



Przykład danych Reuters;  
Często domyślnie około 1, lecz dla  
trudnych danych warto podwyższyć

# Dostrajanie par. C



# Gdzie jesteśmy w wykładzie:

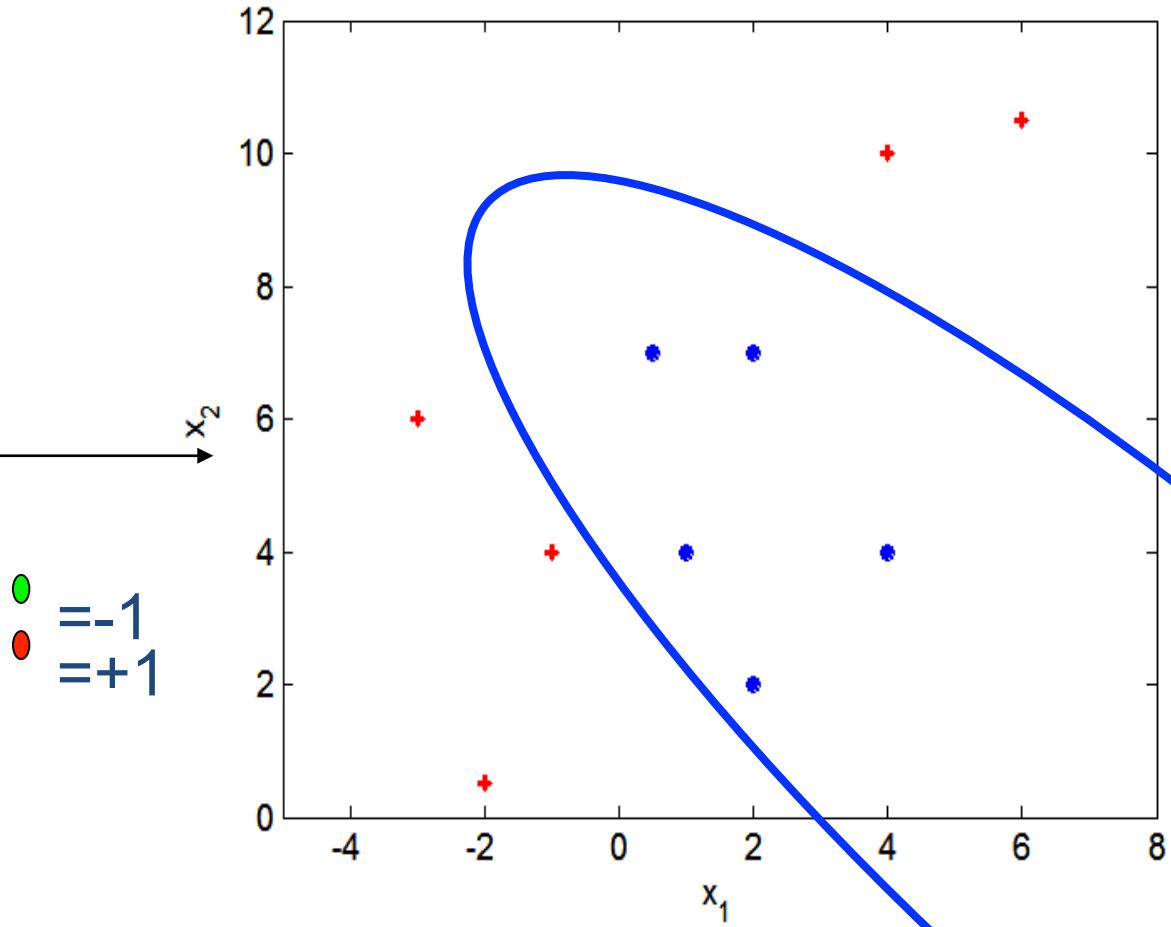
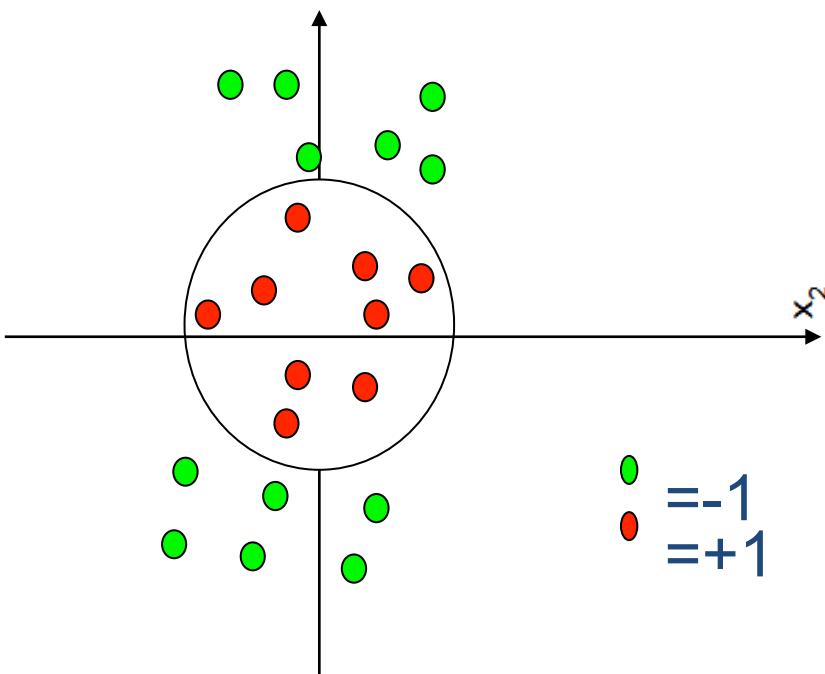
- Podstawy matematyczne SVM
- Przekształcenia do obliczalnych wersji zadania programowania matematycznego
- Rozszerzenie na tzw. overlapping nakładania się klas

Lecz rzeczywiste problemy mają złożone nieliniowe granice decyzyjne!

Jak przejść z L-SVM na N-SVM?

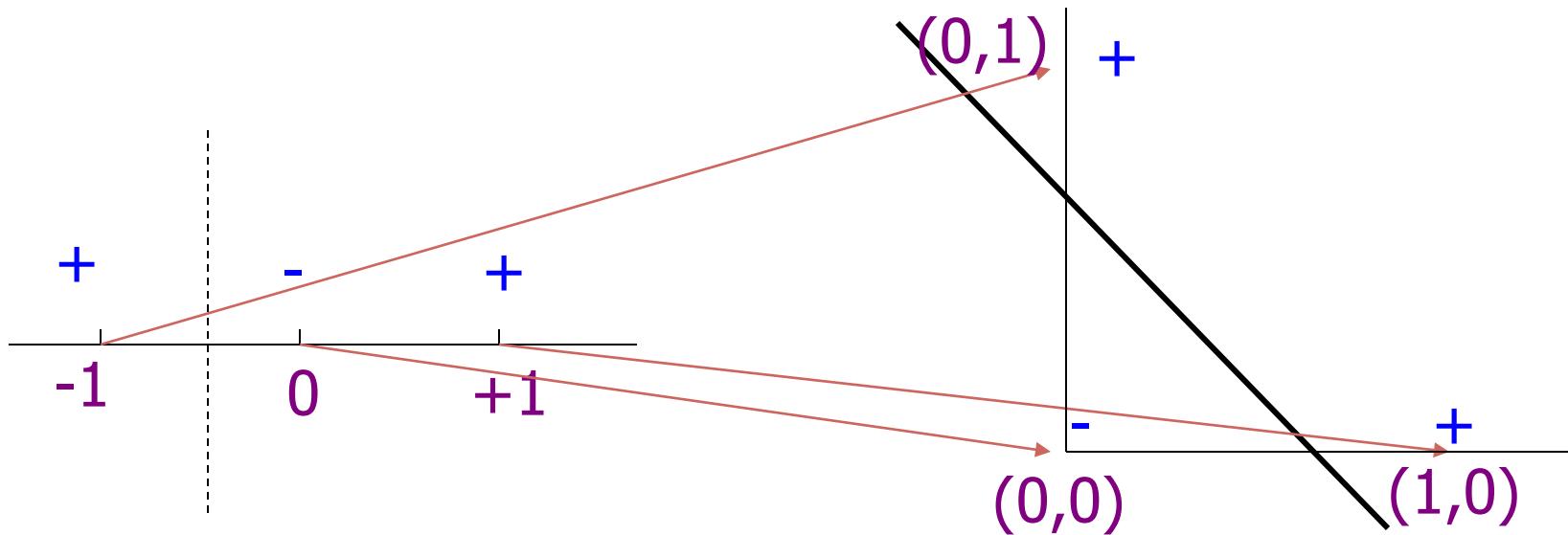
# Nieliniowy SVM

Kiedy klasy są nieliniowo separowalne oraz kształt granicy decyzyjnej jest dość złożony?



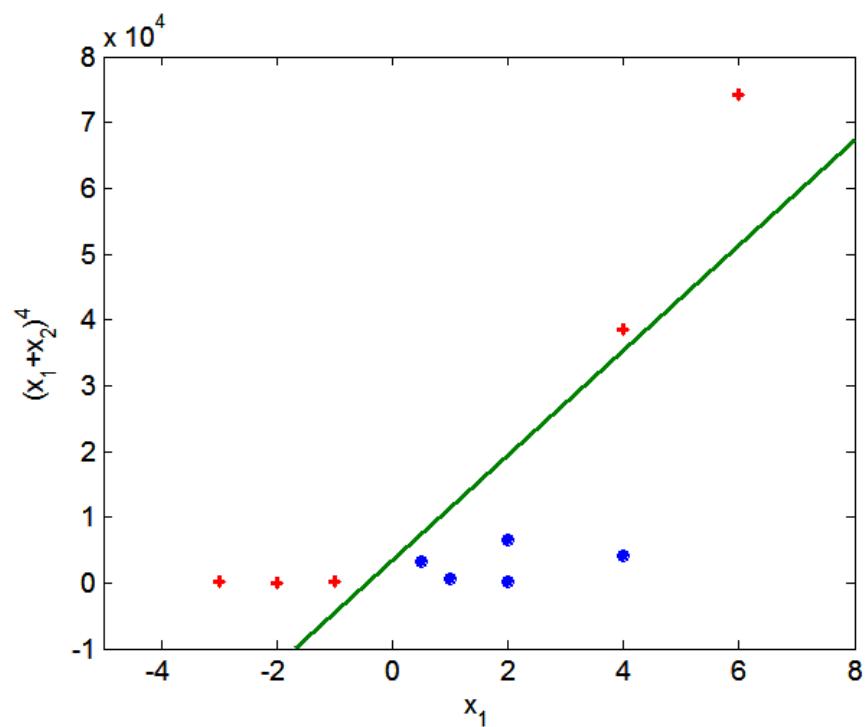
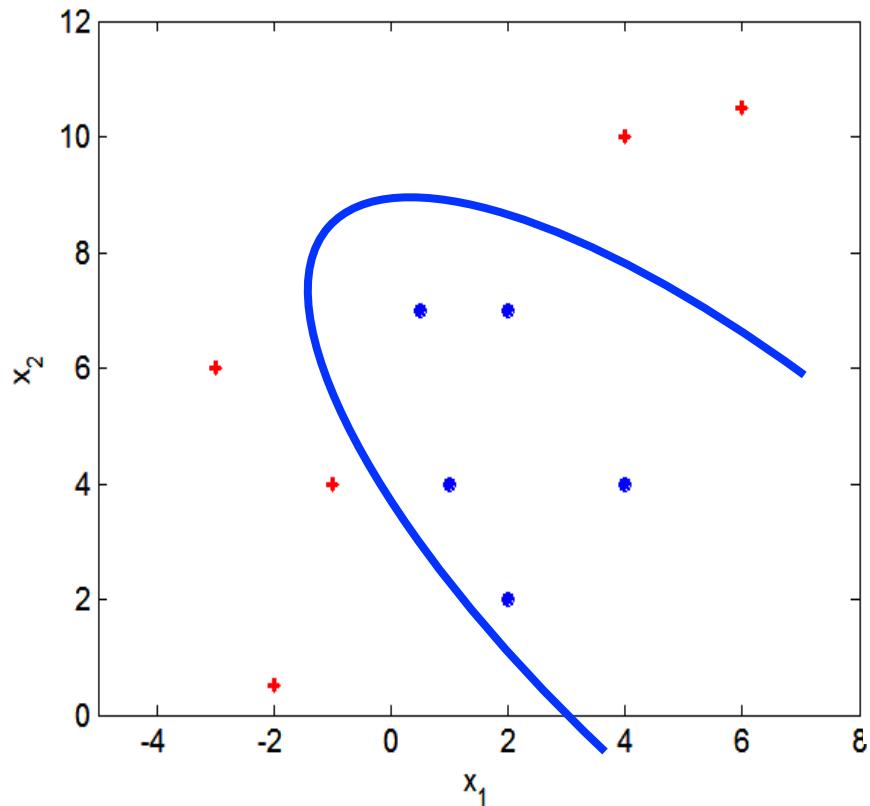
# N-SVM – Transformacje

- Przykład transformacji  $1D \rightarrow 2D$
- Projekcja danych oryginalnych  $x \in R^p$  w nową  $m > p$  wielowymiarową przestrzeń, w której z dużym prawdopodobieństwem będą separowalne liniowo (Twierdzenia matem. np. Covera)
- Przykład przekształcenia wielomianowe wyższego stopnia gdzie do zmiennych  $x$  dołącza się ich  $p$ -te potęgi oraz iloczyny mieszane zmiennych.
- W ogólności trzeba użyć większej liczbie wyżej wymiarowych przestrzeni przekształconych zmiennych



# Nonlinear Support Vector Machines

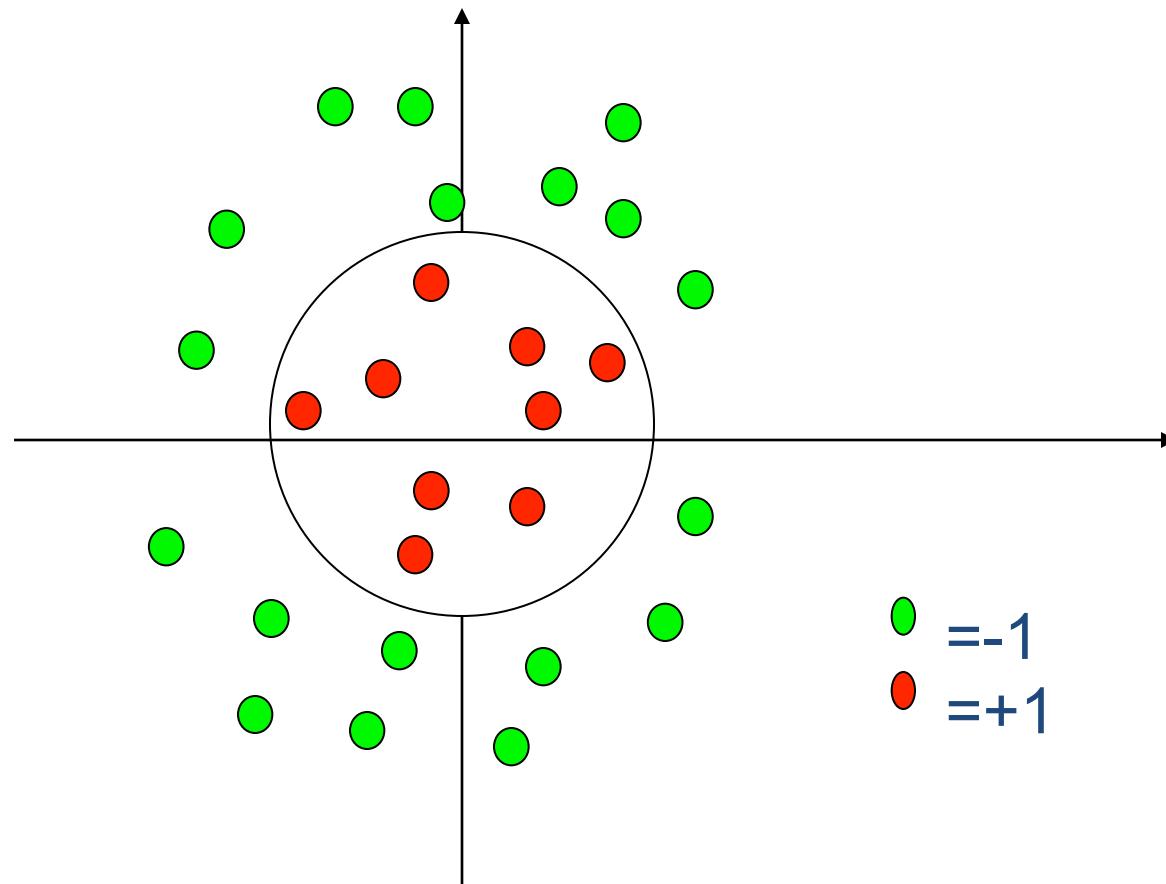
Przykład transformacji wielomianowej  $x_2 \rightarrow (x_1+x_2)^4$



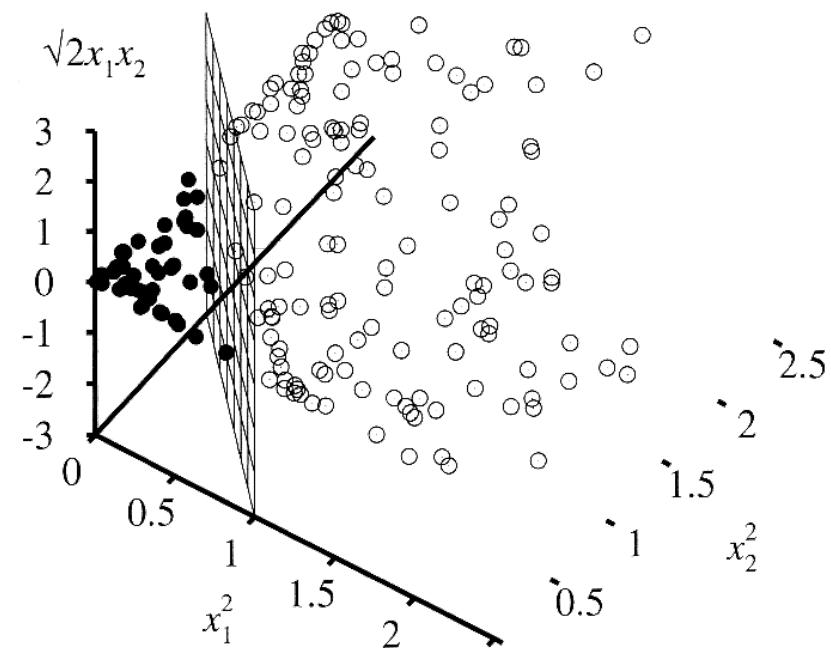
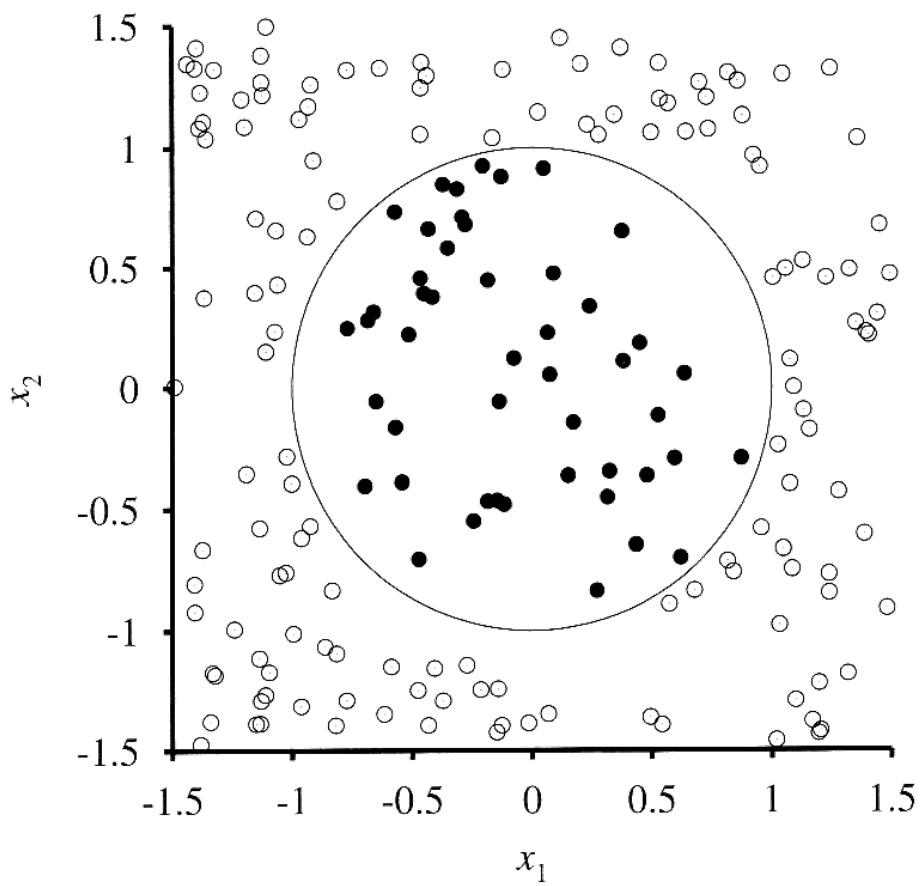
# Trochę inspiracji statystyki mat.

- W zagadnieniach regresyjnych i klasyfikacyjnych, dawno zauważono że można otrzymać bardziej efektywne rozwiązanie, jeżeli oprócz (zamiast) oryginalnego wektora danych  $x \in \mathbb{R}^d$  rozpatrzy się rozszerzony wektor  $z=\phi(x)$  zawierający  $m$  składowych, gdzie  $m \geq d$ .
- Przykładem są przekształcenia wielomianowe
  - Do oryginalnych zmiennych dodajemy ich potęgi oraz iloczyny mieszane zmiennych
  - Inne b. złożone przekształcenia
  - Własności matematyczne – spójrz do literatury
- Przykład rozkładu kołowego

# Przykład rozkładu kołowego klasy otoczonego przykładami z innej klasy



# Przykład transformacji wielomianowej



# Koło - przykład transformacji wielomianowej

- Oryginalna funkcja celu

$$y(x_1, x_2) = \begin{cases} 1 & \sqrt{(x_1 - 0.5)^2 + (x_2 - 0.5)^2} > 0.2 \\ -1 & \text{otherwise} \end{cases}$$

$$\Phi : (x_1, x_2) \rightarrow (x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, 1)$$

- **Transformacja**
- Poszukiwanie parametrów równania liniowego po transformacji

$$w_4x_1^2 + w_3x_2^2 + w_2\sqrt{2}x_1 + w_1\sqrt{2}x_2 + w_0 = 0$$

- Rozwiązanie

$$x_1^2 - x_1 + x_2^2 - x_2 = -0.46$$

# Model nieliniowy SVM

- Funkcja decyzyjna po przekształceniu  $g(\mathbf{x}) = \mathbf{w}\varphi(\mathbf{x}) + b$
- Zasady klasyfikacji
  - +1  $g(\mathbf{x}) > 0$
  - 1  $g(\mathbf{x}) < 0$
- Sformułowanie problemu nieliniowego SVM

$$\min_{\mathbf{w}} = \frac{\|\mathbf{w}\|^2}{2} \quad y_i(w \cdot \Phi(x_i) + b) \geq 1 \quad i = 1, 2, \dots, N$$

- Podobnie jak poprzednio optymalizujemy funkcje z mnożnikami Lagrange'a

$$\sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j))$$

- Funkcja klasyfikująca

$$f(\mathbf{x}) = \text{sign} \left( \sum_{i=1}^N \alpha_i y_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}) + b \right)$$

# Wyzwanie obliczeniowe

- Oblicz  $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$
- **Problem:** Trudne obliczeniowo do wykonania!
- Wiele parametrów do oszacowania - wielomian stopnia  $p$  dla  $N$  atrybutów w oryginalnej przestrzeni prowadzi do  $O(N^p)$  atrybutów w nowej rozszerzonej  $F$  przestrzeni cech
- Skorzystaj z **dot product** (iloczynu skalarnego) na wektorach wejściowych jako miary podobieństwa wektorów
- Iloczyn  $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$  może być odniesiony do podobieństwa wektorów  $\mathbf{x}_i \cdot \mathbf{x}_j$  w **transformowanej** rozszerzonej przestrzeni
- Idea "kerneli" (funkcji jądrowych)
  - Proste funkcje  $K$  dwóch argumentów wektorowych pozwalają obliczyć wartość iloczynu skalarnego w rozszerzonej przestrzeni

# Co to są funkcje jądrowe (ang. Kernel function)

- Wywodzą się z badań liniowych przestrzeni wektorowych, przestrzeni Hilberta, Banacha
- Intuicyjnie są to stosunkowo proste symetryczne  $K(\mathbf{x}_i, \mathbf{x}_j)$  zależne od odległości między  $\mathbf{x}_i$  i  $\mathbf{x}_j$  które spełniają pewne wymagania matem.

$$K(u) \geq 0, \int K(u)du = 1, \sigma_K^2 = \int uK(u)du > 0$$

# Wniosek z twierdzenia Mercera

- Jeśli  $K(\mathbf{x}, \mathbf{y})$  jest funkcją jądrową dla każdego  $\mathbf{x}, \mathbf{y} \in X (\mathbb{R})$ , to można określić przekształcenie  $\Phi: X \rightarrow F$  (przestrzeń przekształconych zmiennych) takie, że

$$K(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{y})$$

- Własność podstawą tzw. **triku kernelowego (*kernel trick*)**
- Cytat z literatury
  - „.... map the data into some other scalar product space (feature space)  $F$  by means of a nonlinear mapping like the above, and perform the linear algorithm (like decision boundary for 2 classes) in the feature space  $F$ . In many cases the mapping  $\Phi$  cannot be explicitly computed, due to the high-dimensionality of  $F$ . But this is not an obstacle, when the decision function requires the evaluation of scalar products  $\Phi(\mathbf{x}) \cdot \Phi(\mathbf{y})$ , and not the pattern  $\Phi(\mathbf{x})$  in explicit form.” [Camastra]
- **Każdy iloczyn funkcji – można zastąpić nieliniową funkcją jądrową na prostszym iloczynie wektorów (przykładów)**

# SVM: tzw. trik kernelowy

Przykład prostego przekształcenia wielomianowego

- The kernel trick:

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)^2 = (x_{i1}^2, \sqrt{2}x_{i1}x_{i2}, x_{i2}^2) \cdot (x_{j1}^2, \sqrt{2}x_{j1}x_{j2}, x_{j2}^2)$$
$$= \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$$

- zadanie optymalizacyjne

$$\sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i \cdot \mathbf{x}_j)$$

Nie musimy znać funkcji  $\Phi$ , wystarczy znać jądro (kernel)  
i można “pracować” w nowej przestrzeni.

# Typowo stosowane f. jądrowe w SVM

Normalne (Gaussowskie)	$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left\{-\frac{(\mathbf{x}_i - \mathbf{x}_j)^2}{2\sigma^2}\right\}$
Wielomianowe	$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + d)^p$
sigmoidalne	$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\kappa \mathbf{x}_i \cdot \mathbf{x}_j - \delta)$

Najpopularniejsze funkcje jądrowe – patrz też implementacje z b. złożonymi funkcjami

# Przykład obliczeń

Najprostsza funkcja wielomianowa:  $K(\mathbf{X}, \mathbf{Y}) = (1 + \mathbf{X} \cdot \mathbf{Y})^d$

Zastosujmy dla przestrzeni 2D

$$\begin{aligned} K(\mathbf{X}, \mathbf{Y}) &= (1 + X_1 Y_1 + X_2 Y_2)^2 \\ &= 1 + 2X_1 Y_1 + 2X_2 Y_2 + (X_1 Y_1)^2 + (X_2 Y_2)^2 + 2X_1 Y_1 X_2 Y_2 \end{aligned}$$

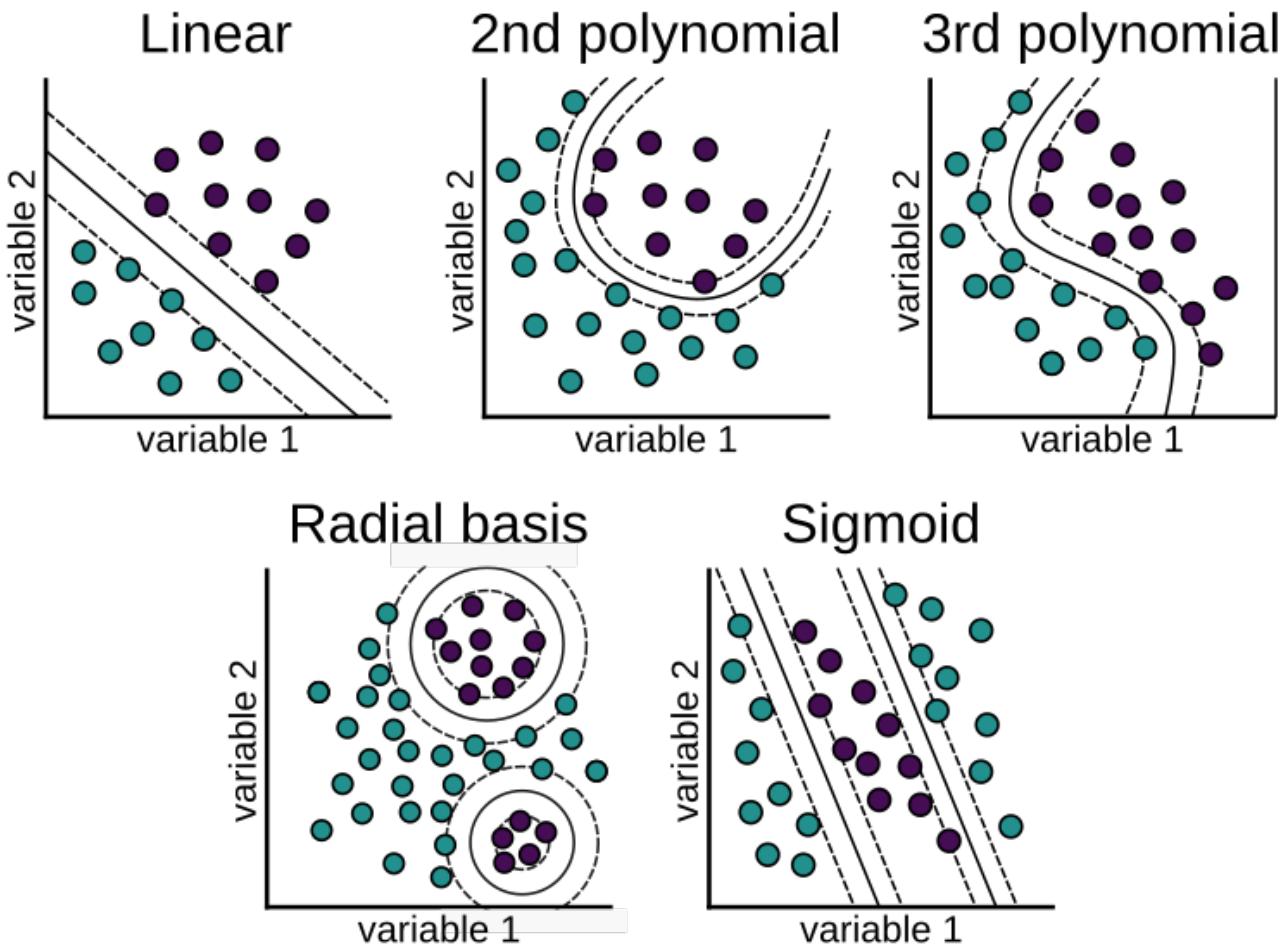
W ten sposób przenosimy się do przestrzeni 5 D

$$\mathbf{X} = (X_1, X_2) \Rightarrow \Phi(\mathbf{X}) = (1, \sqrt{2}X_1, \sqrt{2}X_2, X_1^2, X_2^2, \sqrt{2}X_1 X_2)$$

Hiperpłaszczyzna w 5D – odnaleziona liniowym SVM – odpowiada funkcji kwadratowej w 2D

Oczywiście wybór funkcji jądrowej wpływa na wynik przekształcenia

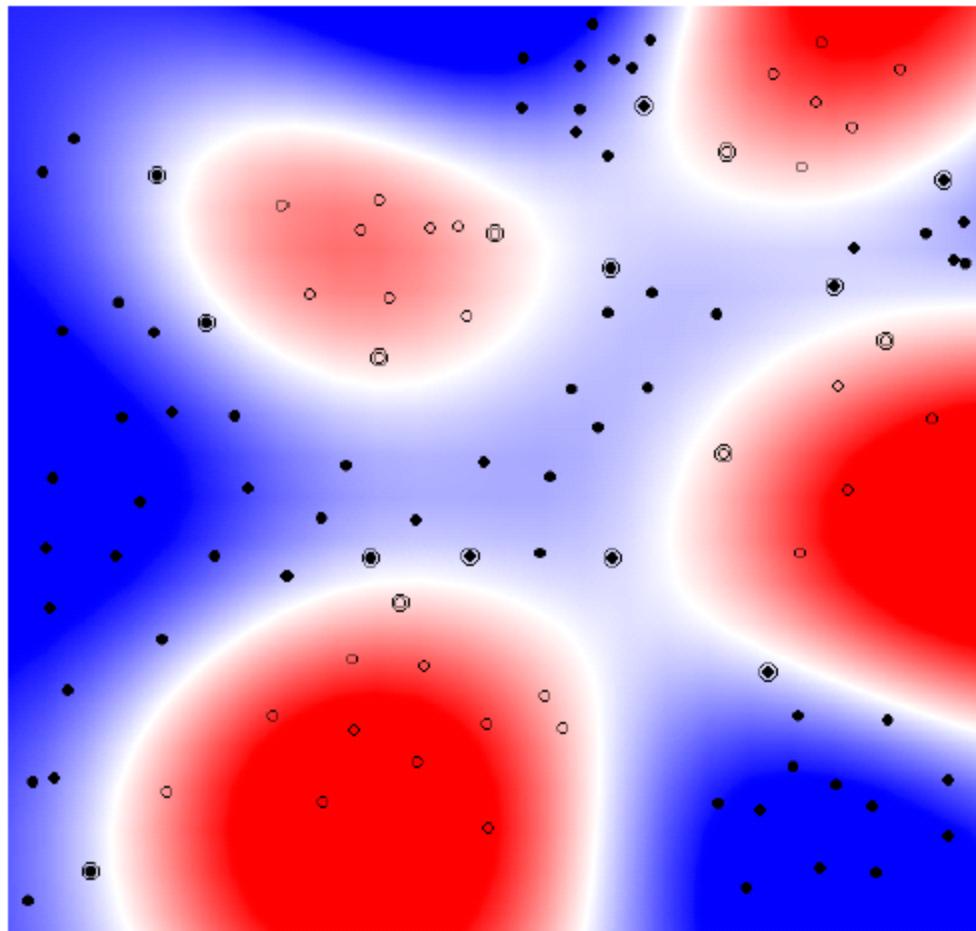
# Ilustracyjne rady



## Example: SVM with RBF-Kernel

Kernel:  $K(\vec{x}_i, \vec{x}_j) = \exp(-|\vec{x}_i - \vec{x}_j|^2 / \sigma^2)$

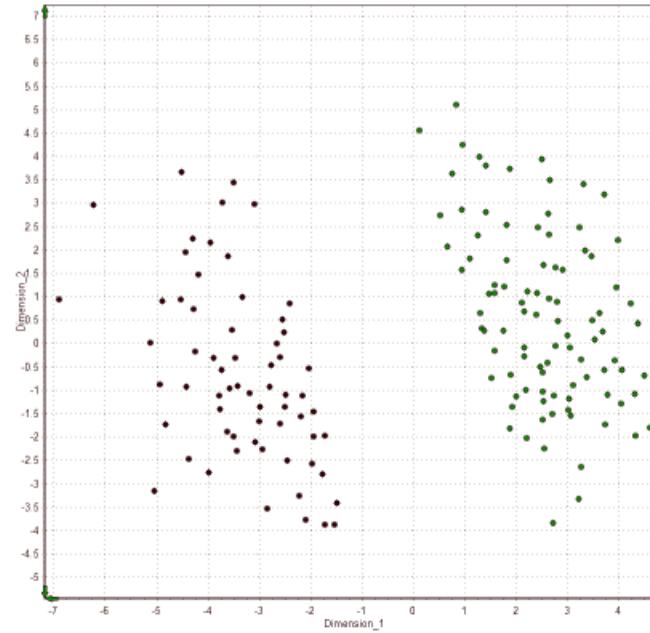
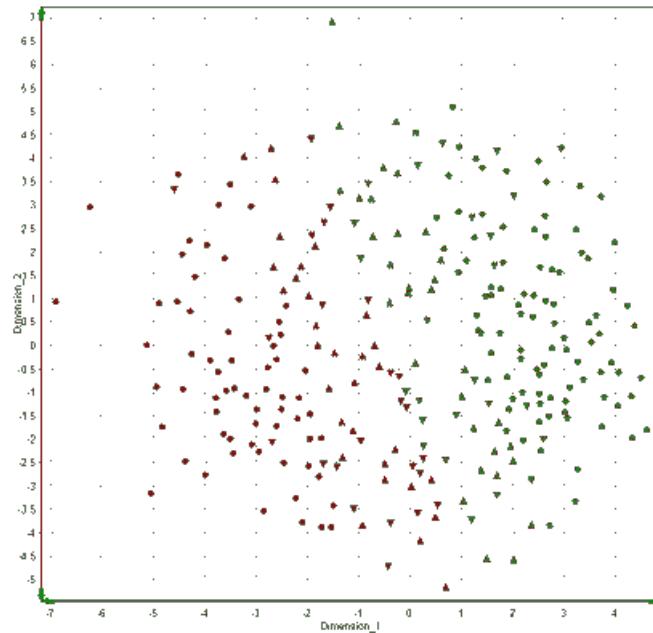
plot by Bell SVM applet



# Przykład 2: Cleveland heart data

Left: 2D MDS features, linear SVM, C=1, acc. 81.9%

Right: support vectors removed, margin is clear.



Gaussian kernel, C=10000, 10xCV, 100% train,  $79.3 \pm 7.8$  test

Gaussian kernel, C=1, 10xCV, 93.8% train,  $82.6 \pm 8.0$  test

Auto C=32 and Gaussian dispersion 0.004: about  $84.4 \pm 5.1$  on test

# Funkcja decyzyjna

- Wykorzystanie funkcji jądrowych

$$f(\mathbf{x}) = \begin{cases} sign\left(\sum_{i=1}^N \alpha_i y_i \Phi(\mathbf{x}_i) \Phi(\mathbf{x}) + b\right) \\ sign\left(\sum_{i=1}^N \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right) \end{cases}$$

- Model klasyfikacji binarnej rozszerza się na zagadnienie wieloklasowe  $K > 2$ 
  - Specyficzne konstrukcje złożone (jak w tzw. zespołach):
    - one-versus-all
    - Pairwise classification (one against one)

# Inne definicje funkcji jądrowych

- Tworzenie dziedzinowych specjalizowanych funkcji jądrowych
  - $K(x,y)$  przyjmuje wyższe wartości, jeśli  $x$  oraz  $y$  są podobne do siebie.
- Rozważmy, np. klasyfikacje tekstów: dwa dokumenty  $D_1$  oraz  $D_2$  –  $K$  skonstruowane z wykorzystaniem liczby wspólnych słów
- Inne: tzw. Tree kernels, graphs kernels. ... (Scholkopf i inni)

# Liczne zastosowania

Można się zapoznać z listą:

<http://www.clopinet.com/isabelle/Projects/SVM/applist.html>

Od końca poprzedniego wieku niezwykle popularny:

- Rozpoznawanie ręcznego pisma – historyczne prace Vapnik i współpracownicy
- Rozpoznawanie obiektów na zdjęciach
- Intrusion Detection Systems (IDSs)
- Klasyfikacja obrazów
- Zastosowania medyczne (diagnostyka, ...)
- Fizyka wysokich cząstek
- Bioinformatyka: analiza mikromacierzy, własności białek
- Wyszukiwanie informacji
- Przetwarzanie dokumentów tekstowych

# Trochę historii



- Wczesne lata sześćdziesiąte – została opracowana metoda “support vectors” w celu konstruowania hiperpłaszczyzn do rozpoznawania obrazu (Vapnik i Lerner 1963, Vapnik i Czervonenkis 1964 / Inst. Akademii Nauk ZSRR) – liniowa SVM.
- Początek lat 1990-siątych: uogólnienie metody pozwalające na konstruowanie nieliniowych funkcji separujących (Boser 1992, Cortes i Vapnik 1995).
- 1995: dalsze rozszerzenie pozwalające otrzymać estymację funkcji ciągłej na wyjściu – regresja (Vapnik 1995).
- Później grupowanie z SVM (Support Vector Clustering)
- Ciekawy wykład V.Vapnika (Complete Statistical Theory of Learning):

<https://www.youtube.com/watch?v=Ow25mjFjSmg>

# Kilka zagadn. efektywnego stosowania SVM

- Normalizuj sygnały wejściowe
- Dobra wybierz wartość C
- Wybór funkcji jądrowej – ważna decyzja + parametryzacja wybranej funkcji
- Uogólnienia dla problemów wieloklasowych
- ... co jeszcze?
  - Na ile są skuteczne SVM w analizie danych niebalansowanych?
  - Tzw. częściowo-etykietowane SVM
- Spójrz na mój inny przedmiot Projekt Eksploracji Danych (sem. 10)

<http://www.cs.put.poznan.pl/jstefanowski/PSE.html>

# Parę uwag podsumowujących

- **Dane odwzorowane (przy pomocy funkcji jądrowych) w nową przestrzeń cech – silna przewaga nad innymi metodami**
- **Wykorzystanie modelu optymalizującego – jedno rozwiązanie**
- W nowej przestrzeni dane powinny być liniowo separowalne
- W porównaniu do innych podejść wielowymiarowość przekształcenia jest „rozwiązana” przez trick kernelowy
- Pośrednio ogranicza się bezpieczeństwo przeuczenia – ale nadal może wystąpić
- Teoretycznie poszukują minimum globalnego a nie lokalnego (jak podejścia heurystyczne – np. MLP, i inne ANN)
- Ograniczenia
  - Dobór parametrów – trudne, także wybór funkcji kernela
  - Odpowiednik podejścia „black box”

# Mocne strony SVM

Stopień skomplikowania/pojemność jest niezależna od liczby wymiarów.

Bardzo dobra podbudowa statystyczno-teoretyczna

Znajdowanie minimum globalnego. Minimalizujemy funkcję kwadratową co gwarantuje zawsze znalezienie minimum.

SVM generuje “prawie optymalny” klasyfikator

Dobre uogólnianie dzięki wielowymiarowej “feature space”.

**Najważniejsze: poprzez użycie odpowiedniej funkcji jądra SVM bardzo duża skuteczność w praktyce**

# Słabe strony SVM

Powolny trening – minimalizacja funkcji, szczególnie dokuczliwy przy dużej ilości danych użytych do treningu.

Rozwiązańa też są skomplikowane (czasami >60% wektorów użytych do nauki staje się wektorami wspierającymi), szczególnie dla dużych ilości danych.

*Przykład (Haykin): poprawa o 1.5% ponad wynik osiągnięty przez MLP. Ale MLP używa 2 ukrytych węzłów, a SVM 285 wektorów.*

Trudno dodać własną wiedzę (prior / background knowledge) !

# SVM oprogramowanie

- SVM Website
  - <http://www.kernel-machines.org/>
- Popularne
  - **LIBSVM**: efektywna implementacja, także dla problemów wieloklasowych, one-class SVM, dostępna dla java, python, etc.
  - SVM-light: prostsza niż LIBSVM, na ogół dla binarnych problemów
  - SVM-torch: inna napisana w C.
- scikit learn – implementacja oparta na libsvm
- Ponadto – liczne inne implementacje, Weka, R

# Scikit-learn

- Proszę sprawdzić dokumentacje klasy `sklearn.svm.SVC`
- Główne parametry
  - Współczynnik C
  - Wybór funkcji jądrowej ('linear', 'poly', 'rbf', 'sigmoid', 'precomputed') – domyślnie rbf
  - Parametry funkcji (np. gamma w rbf, degree do funkcji wielomianowej)

# Odnośniki literaturowe

**Warto sprawdzić moje stronę:**

Tradycyjnie strona materiały dodatkowe:

<http://www.cs.put.poznan.pl/jstefanowski/ml/dodatki.html>

Oprócz linków na stronie www dodatki, warto czytać książki:

- T.Hastie, R.Tibshirani, J.Friedman: *The Elements of Statistical Learning*. Springer → poszukaj wersji elektronicznej pdf
  - J.Koronacki, J.Ćwik: *Statystyczne systemy uczące się* (rozdz. 6)
  - M.Krzyśko, W.Wołyński, T.Górecki, M.Skorzybut: *Systemy uczące się*.
  - S.Ossowski: *Sieci neuronowe w przetwarzaniu informacji*
- 
- Poszukujcie dalej w Internecie – sprawdźcie sami i referujcie

# Inne rozszerzenia SVM

- Specjalne wieloklasowe SVM - poza O-a-A lub Pairwise coupling – także inne sformułowanie problemu optymalizacyjnego
- Sformułowanie regresyjnego SVM
- One-class SVM (estymacja obszaru o wysokiej gęstości przykładów i dyskryminacja od obszarów o niskiej gęstości)
  - Przydatna dla wykrywania obserwacji nietypowych (ang. outlier detection) oraz uczenia się z rzadkich, danych niezbalansowanych.
- Specjalne wersje tzw. transductive support-vector machines – dla uczenia częściowo nadzorowanego

# One-Class Kernel Machines

Naucz się sfery z centrum  $\alpha$  oraz promieniem  $R$

$$\min R^2 + C \sum_t \xi^t$$

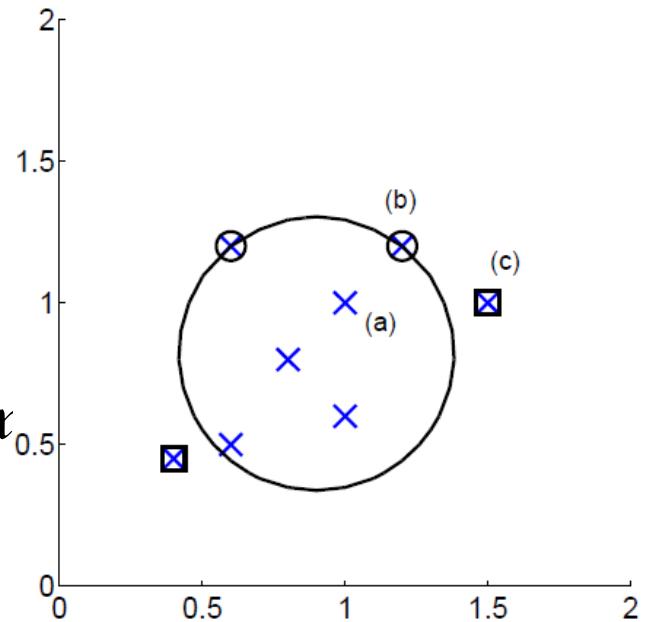
organiczenia

$$\|\mathbf{x}^t - \alpha\| \leq R^2 + \xi^t, \xi^t \geq 0$$

$$L_d = \sum_t \alpha^t \left( \mathbf{x}^t \right)^T \mathbf{x}^s - \sum_{t=1}^N \sum_s \alpha^t \alpha^s r^t r^s \left( \mathbf{x}^t \right)^T \mathbf{x}^s$$

organiczenia

$$0 \leq \alpha^t \leq C, \sum_t \alpha^t = 1$$



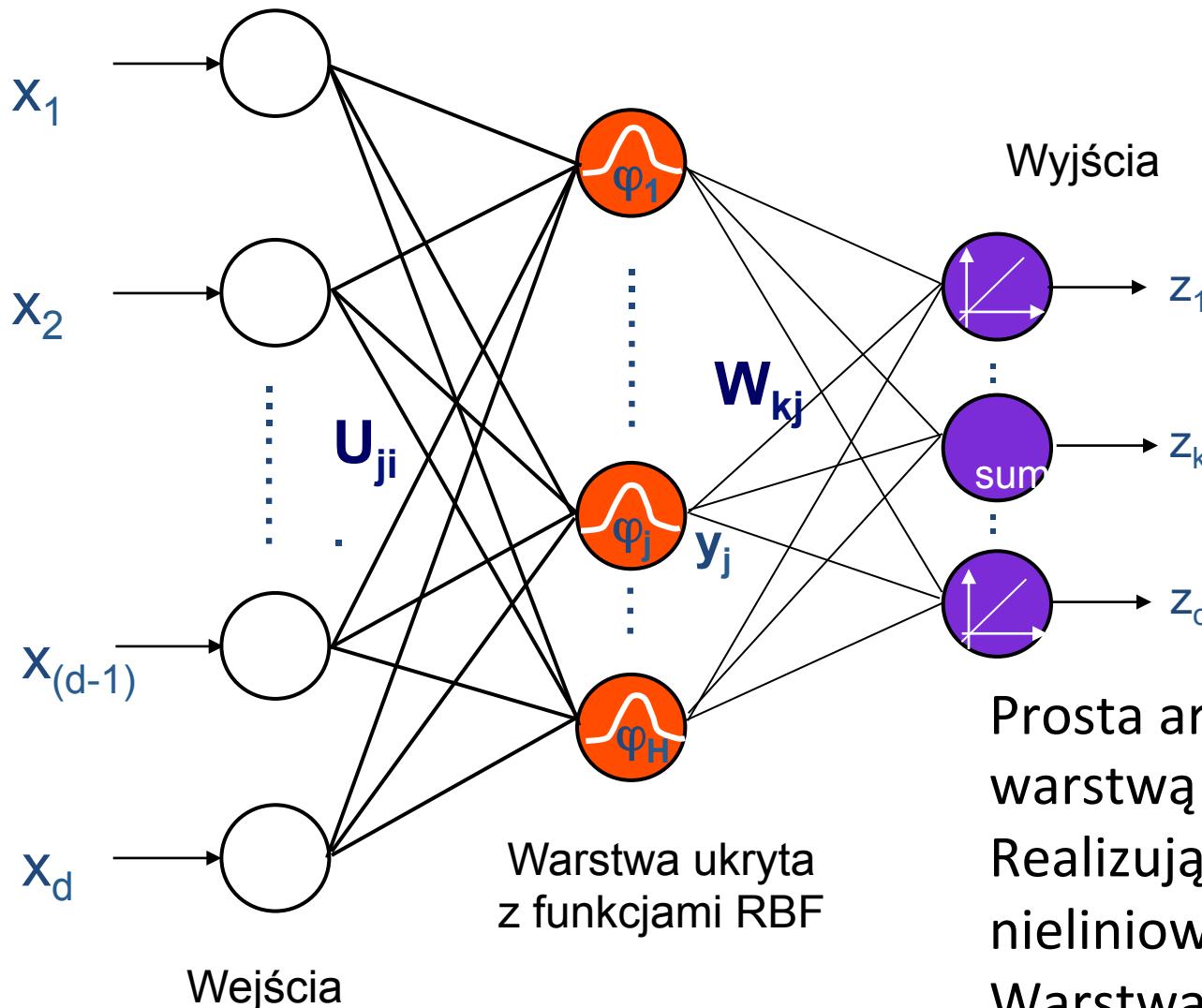
# Inne materiały - internet

- A.Bartkowiak: Kernele, siecie SVM i sieci GDA.
- W.Duch: wykłady nt. Computational Intelligence
- Angielska wersja Wikipedii
- Thorsten Joachims: Support Vector and Kernel Methods – SIGIR Tutorial

# Metody kernelowe

- Szersza klasa
  - Wykorzystują przekształcenie oryginalnej przestrzeni cech funkcjami jądrowymi
    - np. w zaawansowanej wizualizacji danych
  - Niektóre powrócą na laboratoriach
- **Przykład sieci neuronowych RBF**
  - Sieci z funkcjami o symetrii kołowej
  - Neurony ukryte realizują przekształcenie nieliniowe funkcjami jądrowymi

# Typowa topologia sieci RBF



Warstwa ukryta  
z funkcjami RBF

Wejścia

Prosta architektura z warstwą ukrytą – realizującą przekształcenie nieliniowe (funkcje RBF)  
Warstwa wyjściowa – proste (liniowe) ważone sumowanie

# Aproksymacja RBF funkcji ciągłej

- Zadanie aproksymacji złożonej funkcji  $f(\mathbf{x})=z$
- Przyjmijmy funkcje liniową względem parametrów  $w_i$ , wykorzystującą funkcje o symetrii kołowej RBF

$$f(\mathbf{x}) = \sum_{i=1}^m w_i \cdot \varphi(\|\mathbf{x} - \mathbf{c}_i\|)$$

- Radialna funkcja bazowa  $\rightarrow$  funkcja  $\varphi$  o postaci  $\varphi(\mathbf{x}, \mathbf{c}) = \varphi(r(\mathbf{x}, \mathbf{c}))$ , gdzie  $r$  jest odległością między punktami  $\mathbf{x}$  i  $\mathbf{c}$ . Punkt  $c$  nazywamy „centrum”
- Związek funkcji radialnym z funkcjami jądrowymi (ang. kernels), z parametrem  $\sigma$  szerokością jądra
- Najczęściej funkcja Gaussowska

# **Pytanie i komentarze?**

Dalszy kontakt:

[jerzy.stefanowski@cs.put.poznan.pl](mailto:jerzy.stefanowski@cs.put.poznan.pl)

<http://www.cs.put.poznan.pl/jstefanowski/>



**Fundusze  
Europejskie**  
Polska Cyfrowa



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz  
Rozwoju Regionalnego



# Systemy uczące się Ocena zdolności predykcyjnej klasyfikatorów

## wykład 5

Jerzy Stefanowski  
Instytut Informatyki PP  
2021

Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI-TECH)  
projekt finansowany z środków Programu Operacyjnego Polska Cyfrowa  
POPC.03.02.00-00-0001/20



Fundusze  
Europejskie  
Polska Cyfrowa



Rzeczpospolita  
Polska

Unia Europejska  
Europejski Fundusz  
Rozwoju Regionalnego



# Ocena wiedzy klasyfikacyjnej oraz klasyfikatorów

1. Perspektywy oceny klasyfikacji / regresji
2. Miary oceny zdolności predykcyjnych
  - Miary punktowe
  - Miary ROC
  - Uczenie się z kosztami pomyłek
3. Eksperymentalna ocena klasyfikatorów
4. Porównanie wielu klasyfikatorów w studiach przypadków – wykorzystanie testów statystycznych

# Różne perspektywy wiedzy klasyfikacyjnej

- Wiedza / klasyfikatory odkryte z danych
  - Predykcja (klasyfikacji) – przewidywanie przydziału nowych obiektów do klas / wykorzystanie jako tzw. klasyfikator (ocena zdolności klasyfikacyjnej – na ogólnie jedno wybrane kryterium).
  - Opis klasyfikacji obiektów – wyszukiwanie wzorców charakteryzujących właściwości danych i prezentacja ich użytkownikowi w zrozumiałej formie (ocena bardziej trudniejsza i bardziej subiektywna) – typowe dla tzw. data mining.

Spójrz też do książki : J.Stefanowski Algorytmy indukcji reguł decyzyjnych w odkrywaniu wiedzy 2001. pdf dostępny na mojej stronie WWW

# Dlaczego oceniać klasyfikatory?

- Wyzwania praktyczne – potrzeba predykcji
  - Patrz przykłady laboratorium i pierwszy wykład
- Prowadzą do skupienia działania wokół precyzyjnego celu i wspierają decyzje, co do zastosowania
- Pozwalają na porównanie (obecne działanie vs. tzw. baseline; aktualne działania vs. oczekiwane – optymalizacja; porównywanie wielu alternatywnych rozwiązań, ...)
- Wspierają tzw. monitoring lub badanie skuteczności systemu
- oraz ....

# Tworzenie i ocena klasyfikatorów

Jest procesem trzyetapowym:

1. Konstrukcja modelu w oparciu o zbiór danych wejściowych (przykłady uczące - etykietowane).

Przykładowe modele :

- drzewa decyzyjne, reguły (IF .. THEN ..),
- Naive Bayes, regresja logistyczna,
- sieci neuronowe, SVM, zespoły.

2. Ocena modelu (przykłady testujące – ukryte etykiety)
3. Użycie/ wdrożenie modelu (klasyfikowanie nowych obiektów – bez etykiet)

# Popularne kryteria

- **Trafność predykcji** (klasyfikacja / regresja)
- Zdolności interpretacji modelu: np. drzewa decyzyjne vs. sieci neuronowe => patrz dalsze wykłady
- Złożoność struktury, np.
  - rozmiar drzew decyzyjnego,
  - miary oceny reguły
- Odporność na różne charakterystyki danych
  - Szum (noise),
  - Inne trudności rozkładu danych,  
----- oraz wymagania obliczeniowe
- Szybkość i skalowalność:
  - czas uczenia się,
  - szybkość samego klasyfikowania

# Trafność klasyfikowania

- Użyj przykładów testowych nie wykorzystanych w fazie uczenia klasyfikatora:
  - $N_t$  – liczba przykładów testowych
  - $N_c$  – liczba poprawnie sklasyfikowanych przykładów testowych
- Trafność klasyfikowania (ang. classification accuracy) – najczęściej wyrażania w procentach:

$$\eta = \frac{N_c}{N_t}$$

- Alternatywnie błąd klasyfikowania.  $\varepsilon = \frac{N_t - N_c}{N_t}$

Pomyśl – czy oba błędy się zawsze uzupełniają (np. do 1,0 lub 100%)?

# Predykcja zmiennej y (liczbowej)

- Zmienna wyjściowa liczbowa : ocena jak odbiega predykcja  $\hat{y}$  od właściwej wyjściowej y
- Odpowiednik funkcji straty : ocena różnicy  $y_i$  oraz predykcji  $\hat{y}_i$ 
  - Absolute error:  $|y_i - \hat{y}_i|$
  - Squared error:  $(y_i - \hat{y}_i)^2$
- Popularne uśrednione wartości błędów
  - Mean absolute error:  $\frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{n}$       **Mean squared error:**  $\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}$
  - Relative absolute error:  $\frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{\sum_{i=1}^n |y_i - \bar{y}|}$       **Relative squared error:**  $\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$
- Na ogół stosowane (square) root mean-square error, oraz root relative squared error

# Wiele innych miar oceny predykcji

<code>roc_curve (y_true, y_score[, pos_label, ...])</code>	Compute Receiver operating characteristic (ROC)
<code>balanced_accuracy_score (y_true, y_pred[, ...])</code>	Compute the balanced accuracy

Others also work in the multiclass case:

<code>cohen_kappa_score (y1, y2[, labels, weights, ...])</code>	Cohen's kappa: a statistic that measures inter-annotator agreement.
<code>confusion_matrix (y_true, y_pred[, labels, ...])</code>	Compute confusion matrix to evaluate the accuracy of a classification
<code>hinge_loss (y_true, pred_decision[, labels, ...])</code>	Average hinge loss (non-regularized)
<code>matthews_corrcoef (y_true, y_pred[, ...])</code>	Compute the Matthews correlation coefficient (MCC)

Some also work in the multilabel case:

<code>accuracy_score (y_true, y_pred[, normalize, ...])</code>	Accuracy classification score.
<code>classification_report (y_true, y_pred[, ...])</code>	Build a text report showing the main classification metrics
<code>f1_score (y_true, y_pred[, labels, ...])</code>	Compute the F1 score, also known as balanced F-score or F-measure
<code>fbeta_score (y_true, y_pred, beta[, labels, ...])</code>	Compute the F-beta score
<code>hamming_loss (y_true, y_pred[, labels, ...])</code>	Compute the average Hamming loss.
<code>jaccard_similarity_score (y_true, y_pred[, ...])</code>	Jaccard similarity coefficient score
<code>log_loss (y_true, y_pred[, eps, normalize, ...])</code>	Log loss, aka logistic loss or cross-entropy loss.
<code>precision_recall_fscore_support (y_true, y_pred)</code>	Compute precision, recall, F-measure and support for each class
<code>precision_score (y_true, y_pred[, labels, ...])</code>	Compute the precision
<code>recall_score (y_true, y_pred[, labels, ...])</code>	Compute the recall
<code>zero_one_loss (y_true, y_pred[, normalize, ...])</code>	Zero-one classification loss.

And some work with binary and multilabel (but not multiclass) problems:

# WEKA evaluation

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier Choose J48 -C 0.25 -M 2

Test options

Use training set  
 Supplied test set Set...  
 Cross-validation Folds 10  
 Percentage split % 66  
More options...

(Nom) CLASSE\_INF

Start Stop

Result list (right-click for options)

- 04:33:49 - trees.J48
- 04:34:01 - trees.J48
- 04:37:47 - trees.J48
- 04:38:00 - trees.J48
- 04:40:19 - trees.J48
- 04:40:34 - trees.J48
- 05:23:51 - trees.J48
- 05:25:57 - trees.J48
- 05:29:19 - trees.J48
- 05:29:43 - trees.J48
- 05:34:15 - trees.J48

Classifier output

```
|  |  v4_neadiness > 79
|  |  | 03_Percentage_Bypass <= 75: Good (4.0)
|  |  | 03_Percentage_Bypass > 75: VeryGood (2.0)
|  | 01_Number_of_Patients > 287: VeryGood (3.0)
05_Notoriety > 87: VeryGood (4.0)
```

Number of Leaves : 5

Size of the tree : 9

Time taken to build model: 0 seconds

== Stratified cross-validation ==

== Summary ==

Correctly Classified Instances	11	55	%
Incorrectly Classified Instances	9	45	%
Kappa statistic	0.0625		
Mean absolute error	0.4536		
Root mean squared error	0.6082		
Relative absolute error	90.7222 %		
Root relative squared error	121.0868 %		
Total Number of Instances	20		

== Detailed Accuracy By Class ==

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.333	0.273	0.5	0.333	0.4	0.571	VeryGood	
0.727	0.667	0.571	0.727	0.64	0.571	Good	
Weighted Avg.	0.55	0.489	0.539	0.55	0.532	0.571	

== Confusion Matrix ==

```
a b    <- classified as
3 6 | a = VeryGood
3 8 | b = Good
```

Źródło – własne uruchomienie oprogramowania

# Miary – zależność od zadania

## Klasyfikacja binarna

Wskazanie etykiety vs. scoring predictions

Miary punktowe np. Accuracy,

Ocena prawdopodobieństwa – Kappa statistics

Zainteresowanie wybraną klasą

Precision, Recall / Sensitivity, Specificity, F-score, G-mean

Miary graficzne: ROC, PRcurves, Lift curves

## Wieloklasowość / Wielo-etykietowość

Nie wszystkie miary binarne można uogólnić

## Specyfika danych

Tzw. Imbalanced data oraz cost sensitive learning

## Predykcja ciągła

Błędy RSME, oceny różnic rozkładów (dywergencje KL)

# Macierz pomyłek

- Analiza pomyłek w przydiale do różnych klas przy pomocy tzw. macierz pomyłek (ang. *confusion matrix*)
- Macierz  $r \times r$ , gdzie wiersze odpowiadają poprawnym klasom decyzyjnym, a kolumny decyzjom przewidywanym przez klasyfikator; na przecięciu wiersza  $i$  oraz kolumny  $j$  - liczba przykładów  $n_{ij}$  należących oryginalnie do klasy  $i$ -tej, a zaliczonej do klasy  $j$ -tej

Przykład:

		Przewidywane klasy decyzyjne		
Oryginalne klasy	$K_1$	$K_2$	$K_3$	
$K_1$	50	0	0	
$K_2$	0	48	2	
$K_3$	0	4	46	

# Klasyfikacja binarna

- Niektóre zastosowania → jedna z klas posiada szczególne znaczenie, np. diagnozowanie poważnej choroby. Zadanie → klasyfikacja binarna.

Oryginalne klasy	Przewidywane klasy decyzyjne	
	Pozytywna	Negatywna
Pozytywna	$TP$	$FN$
Negatywna	$FP$	$TN$

- Nazewnictwo (inspirowane medycznie):

- $TP$  (ang. *true positive*) – liczba poprawnie sklasyfikowanych przykładów z wybranej klasy (ang. *hit*),
- $FN$  (ang. *false negative*) – liczba błędnie sklasyfikowanych przykładów z tej klasy, tj. decyzja negatywna podczas gdy przykład w rzeczywistości jest pozytywny (błąd pominięcia - z ang. *miss*),
- $TN$  (ang. *true negative*) – liczba przykładów poprawnie nie przydzielonych do wybranej klasy (poprawnie odrzuconych z ang. *correct rejection*),
- $FP$  (ang. *false positive*) – liczba przykładów błędnie przydzielonych do wybranej klasy, podczas gdy w rzeczywistości do niej nie należą (ang. *false alarm*).

# Trudności oceny trafności

Oryginalna →	Pos	Neg
Yes	200	100
No	300	400
	P=500	N=500

Oryginalna →	Pos	Neg
Yes	400	300
No	100	200
	P=500	N=500

Oba klasyfikatory = 60% trafność (accuracy)

Lecz różnie w predykcji poszczególnych klas:

Lewa tabela: niski TPR /wysoka rozpoznawalność Neg

Prawa tabela: dobre rozpoznawanie klasy Pos, słabe Neg

# Zainteresowanie pojedyncza klasą

- **Dane niebalansowane** (na ogólnie dwie klasy)
  - (ang. imbalanced data) klasy nie są w przybliżeniu równo liczne; Klasa mniejszościowa (ang. minority class) zawiera wyraźnie mniej przykładów niż inne klasy
  - Przykłady z klasy mniejszościowej są często najważniejsze i ich poprawne rozpoznawanie jest głównym celem.
    - Rozpoznawanie rzadkiej, niebezpiecznej choroby
- Powoduje trudności w fazie uczenia i obniża zdolność predykcyjną
  - Niektóre klasyfikatory pomimo wysokiej globalnej trafności nie rozpoznają kl. mniejszościowej
  - Przykład klasyfikacji tekstów (Catlett) trafność 99% , lecz brak rozpoznania specjalnych dokumentów (TPR 0%)

# Miary punktowe dla niezbalansowania klas

Rozpoznawanie klasy mniejszościowej z ...

Wiele miar definiowanych na podstawie macierzy pomyłek

$$Sensitivity = Recall = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP} \quad Precision = \frac{TP}{TP + FP}$$

Oryginalne	Przewidywane	
	+	-
+	TP	FN
-	FP	TN

Inne miary:

*False-positive rate* =  $FP / (FP+TN)$ , czyli 1 – specyficzność

Agregacje:  $G\text{-mean} = \sqrt{Sensitivity * Specificity}$

$$F\text{-measure} = \frac{(1 + \beta)^2 * Precision * Recall}{\beta^2 * Recall + Precision}$$

# Analiza macierzy... spróbuj rozwiązać...

$$Sensitivity = \frac{TP}{TP+FN} = ?$$

$$Specificity = \frac{TN}{TN+FP} = ?$$

*Co przewidywano*

*Rzeczywista  
Klasa*

		1	0
1	1	60	30
	0	80	20

$60+30 = 90$  przykładów w danych należało do Klasy 1

$80+20 = 100$  przykładów było w Klasy 0

$90+100 = 190$  łączna liczba przykładów

Który klasyfikator jest najlepszy – miary mogą oceniać inne aspekt, np. eksperymenty UCI Breast Cancer

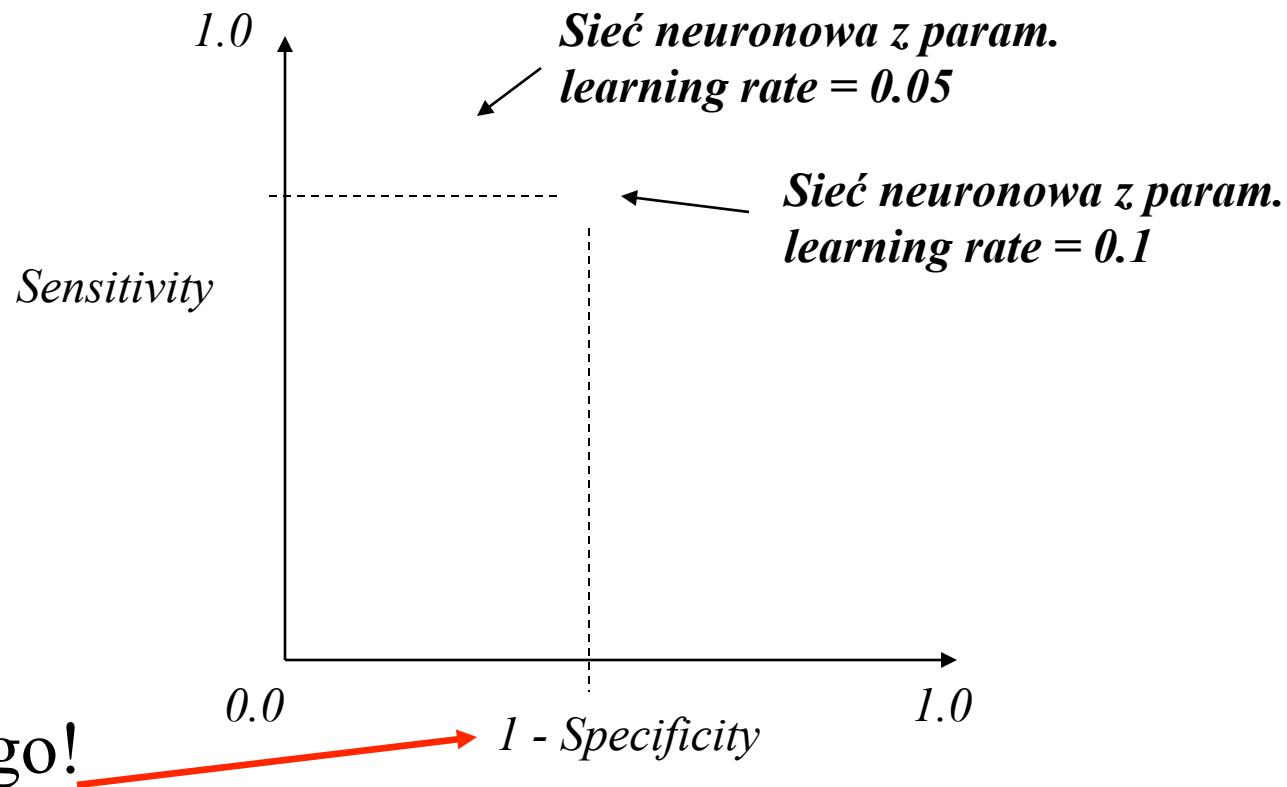
Algo	Acc	RMSE	TPR	FPR	Prec	Rec	F	AUC	Info S
NB	71.7	.4534	.44	.16	.53	.44	.48	.7	48.11
C4.5	75.5	.4324	.27	.04	.74	.27	.4	.59	34.28
3NN	72.4	.5101	.32	.1	.56	.32	.41	.63	43.37
Ripper	71	.4494	.37	.14	.52	.37	.43	.6	22.34
SVM	69.6	.5515	.33	.15	.48	.33	.39	.59	54.89
Bagg	67.8	.4518	.17	.1	.4	.17	.23	.63	11.30
Boost	70.3	.4329	.42	.18	.5	.42	.46	.7	34.48
RanFR	69.23	.47	.33	.15	.48	.33	.39	.63	20.78

# Scoring classifier – odpowiedź także liczbowa (np. NB, ANN, regresja logistyczna)

- Klasyfikator oprócz wskazania klasy pokazuje także wartość ilościową z nią związaną
  - Pomyśl o Naiwnym klasyfikatorze Bayesowskim
- Ponadto tzw. klasyfikator ciągły- możliwość progowania wyjścia modelu – zwłaszcza dla dwóch klas

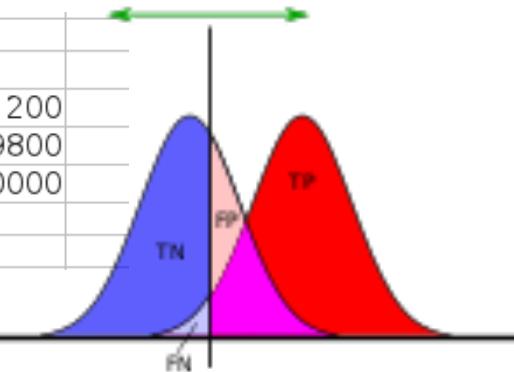
# Analiza krzywej ROC

Każda technika budowy klasyfikatora może być scharakteryzowana poprzez pewne wartości miar ‘sensitivity’ i ‘specificity’. Graficznie można je przedstawić na wykresie ‘sensitivity’ vs.  $1 - \text{specificity}$ .



# Interpretacja progu klasyfikatora

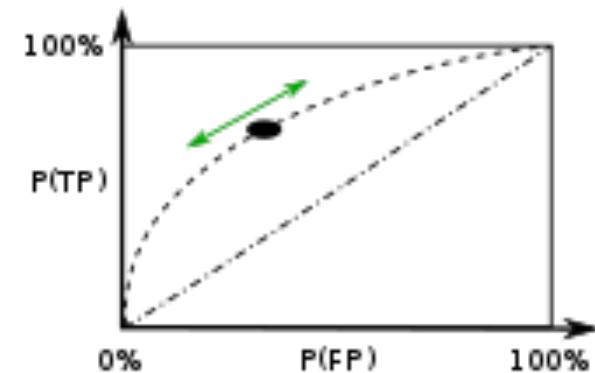
	Test Positive	Test Negative	Total
Patient Diseased	160	40	200
Patient Healthy	29940	69860	99800
Total	30100	69900	100000



Maximum, np. 1

Próg - T

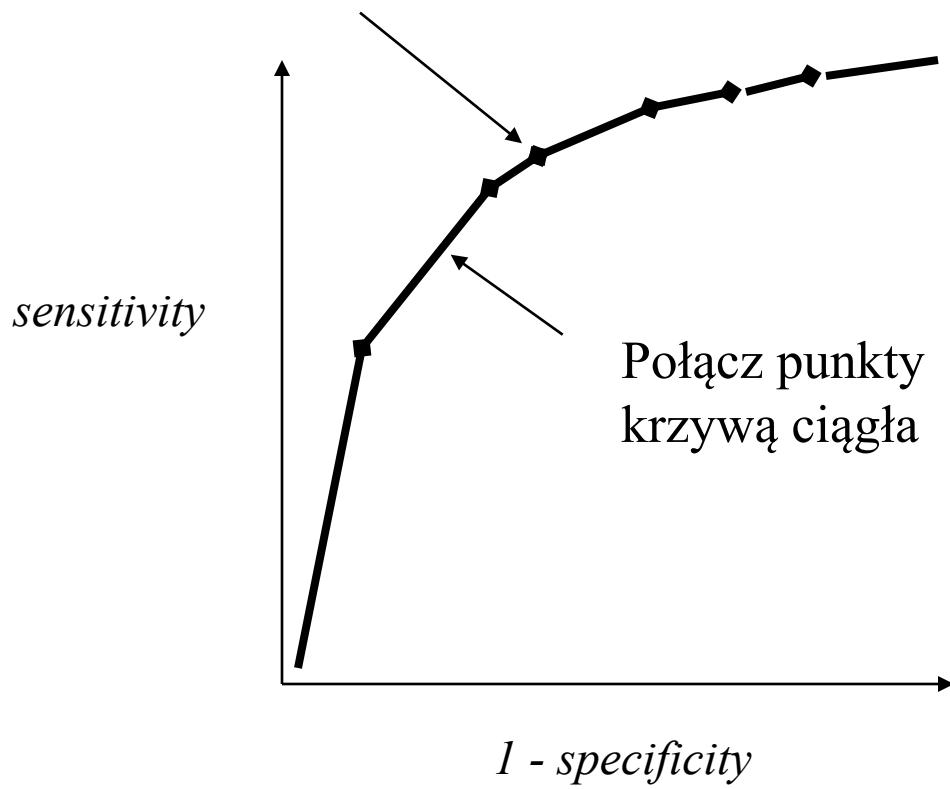
Minimum, np. 0



Źródło - Wikipedia

# ROC - analiza

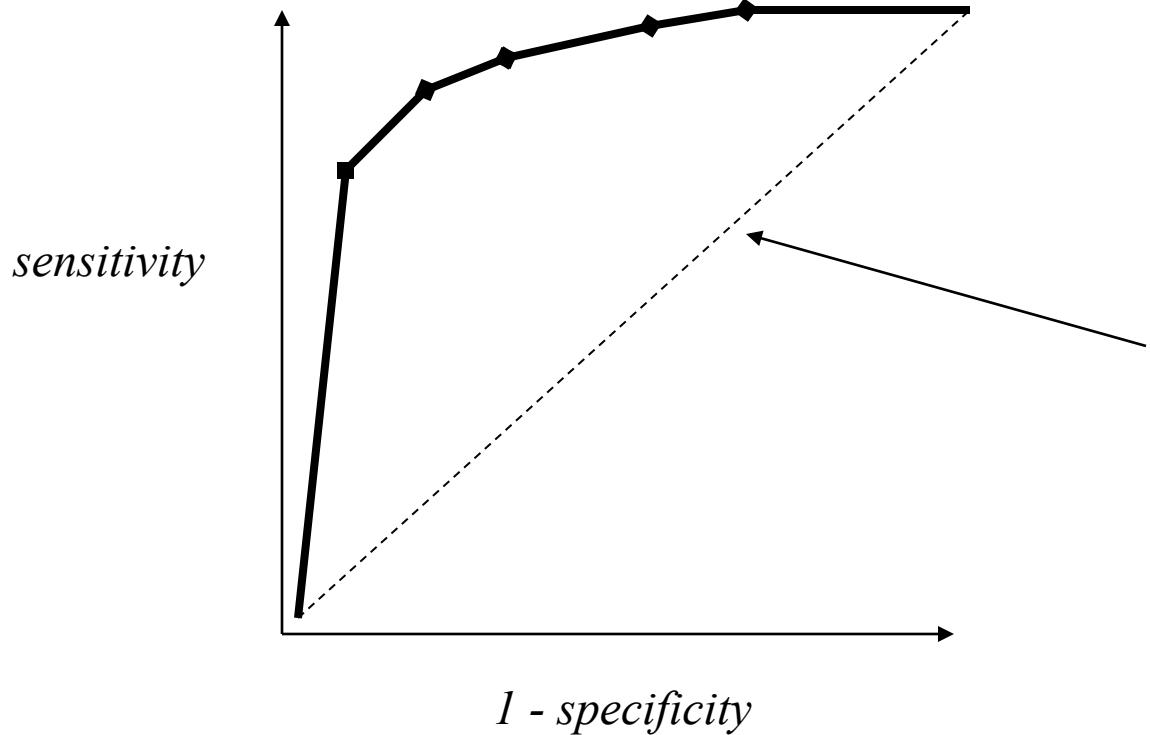
Algorytm może być parametryzowany, i w rezultacie otrzymuje się serie punktów odpowiadających doborowi parametrów



Połącz punkty  
krzywą ciągłą

Wykres nazywany  
'krzywą' ROC.

# Krzywa ROC



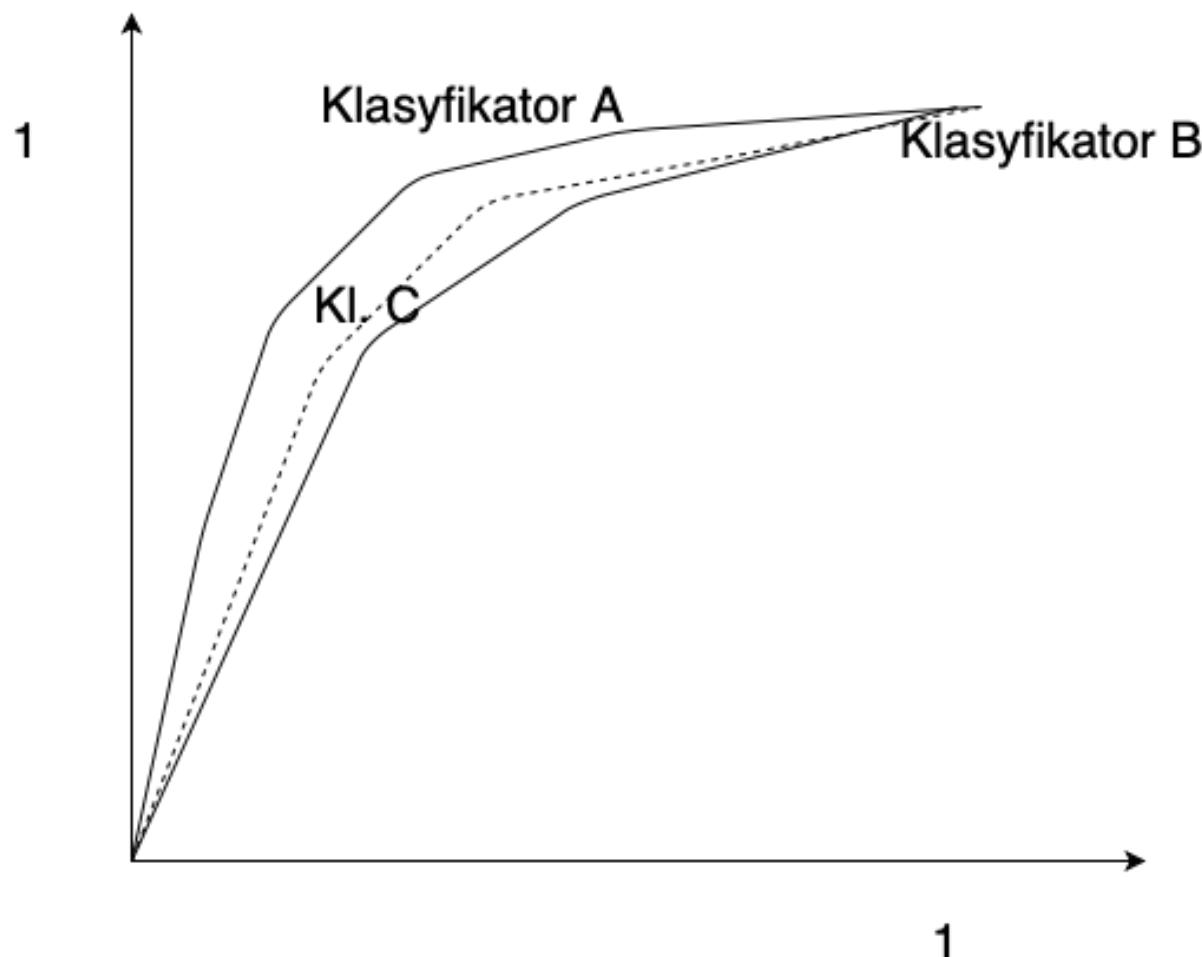
Im krzywa bardziej wygięta ku górnemu lewemu narożnikowi, tym lepszy klasyfikator .

Przekątna odpowiada losowemu „zgadywaniu”. Im bliżej niej, tym gorszy klasyfikator

Można porównywać działanie kilku klasyfikatorów.

Miary oceny np. **AUC** – pole pod krzywą. Wartość z zakresu 0 do 1

# Porównywanie działania klasyfikatorów na ROC



Krzywe dla 3 różnych klasyfikatorów – A najlepszy

Krzywe mogą się przecinać

# Macierze kosztów

Oryginalne klasy	Przewidywane klasy decyzyjne	
	Pozytywna	Negatywna
Pozytywna	$C(TP)$	$C(FP)$
Negatywna	$C(FN)$	$C(TN)$

Koszty  $C(TP)$  i  $C(TN) \rightarrow 0$ ; a  $C(FP)$  na ogół większe niż  $C(FN)$

Oryginalne klasy	Przewidywane klasy decyzyjne	
	Pozytywna	Negatywna
Pozytywna	0	15
Negatywna	5	0

Pomyłki mają różną interpretacje i praktyczne znaczenie

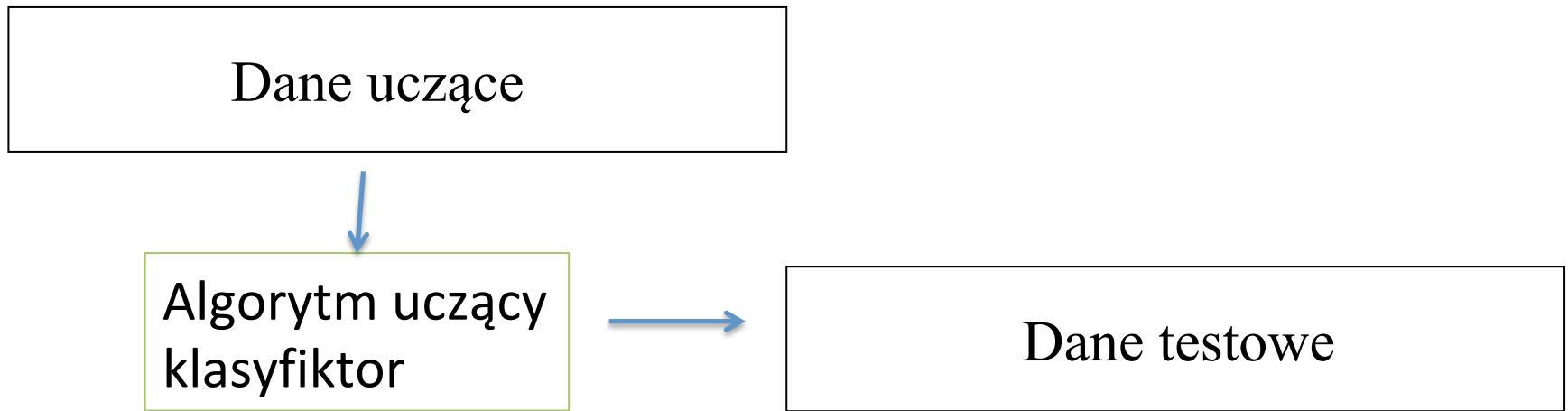
Implementacje: np. WEKA costsensitiveclassifiers

# Jak szacować wiarygodnie ?

- Zależy od perspektywy użycia wiedzy:
  - Predykcja klasyfikacji albo opisowa
- Ocena na zbiorze uczącym nie jest wiarygodna jeśli rozważamy predykcję nowych faktów!
  - Nowe obserwacje najprawdopodobniej nie będą takie same jak dane uczące!
  - Choć zasada reprezentatywności próbki uczącej ...
- Problem przeuczenia (ang. overfitting)
  - Nadmierne dopasowanie do specyfiki danych uczących powiązane jest najczęściej z utratą zdolności uogólniania (ang. generalization) i predykcji nowych faktów!

# Zasada eksperymentalnej oceny

Niezależny zbiór przykładów testowych - nie wykorzystuj w fazie uczenia klasyfikatora!



Nie dopuszczaj do tzw. przecieku informacji (ang. information leak)

Błąd treningowy – niebezpieczeństwo przeuczenia.

# Podejście empiryczne

- Zasada „Train and test” (ucz i testuj)
- Gdy nie ma podziału zadanego przez nauczyciela, to co wykorzystasz - losowe podziały.
  - **Podziały – próba losowa LECZ ile i jakie przykłady!**
- Nadal pytanie jak szacować wiarygodnie?

Dane pełne - etykietowane przykłady

Podział losowy

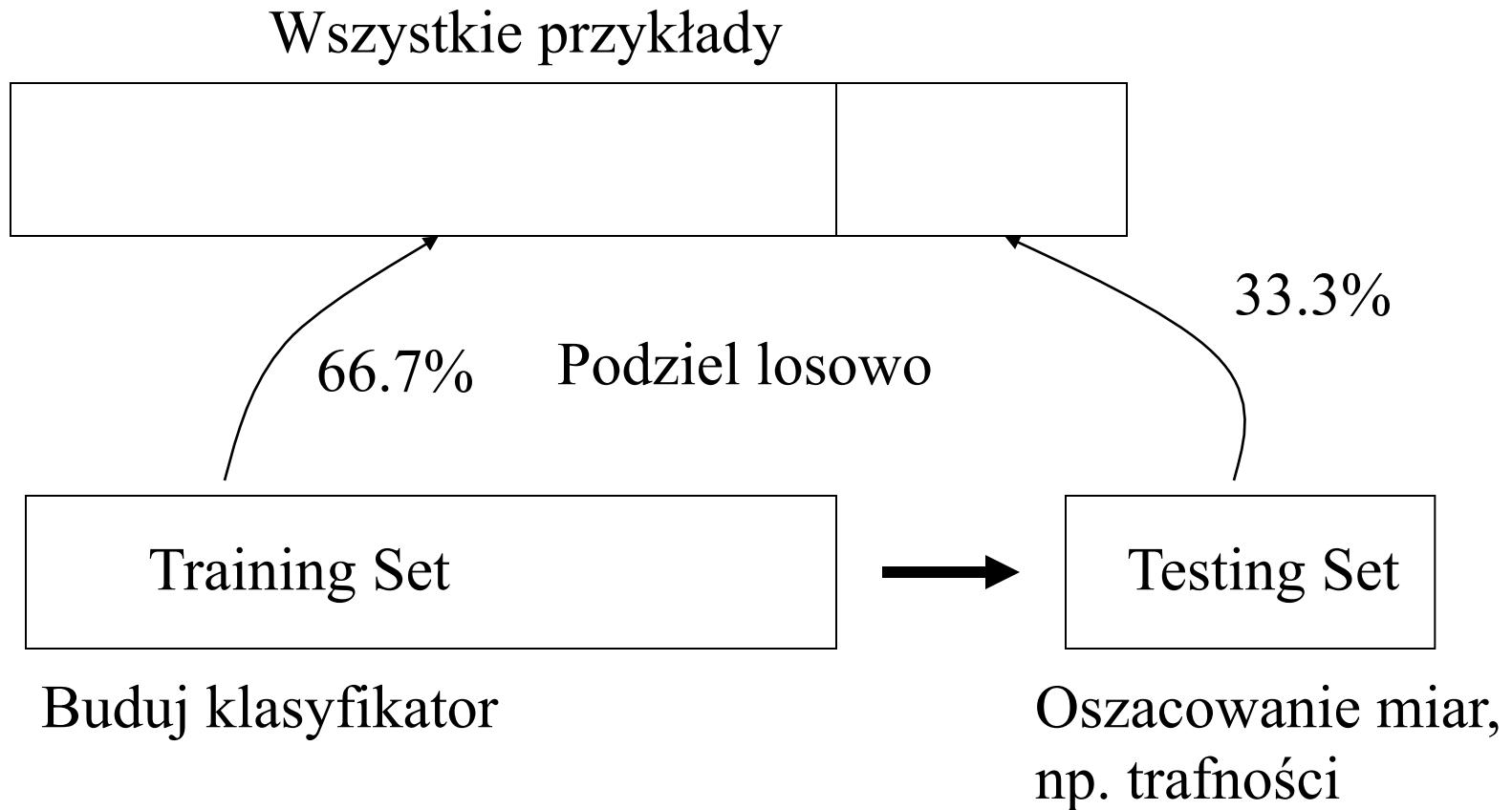


- Typowo – każdy przykład ma równe prawdopodobieństwa wylosowania podziału
- Wersje spec. losowania – zmienne prawdopodobieństwa

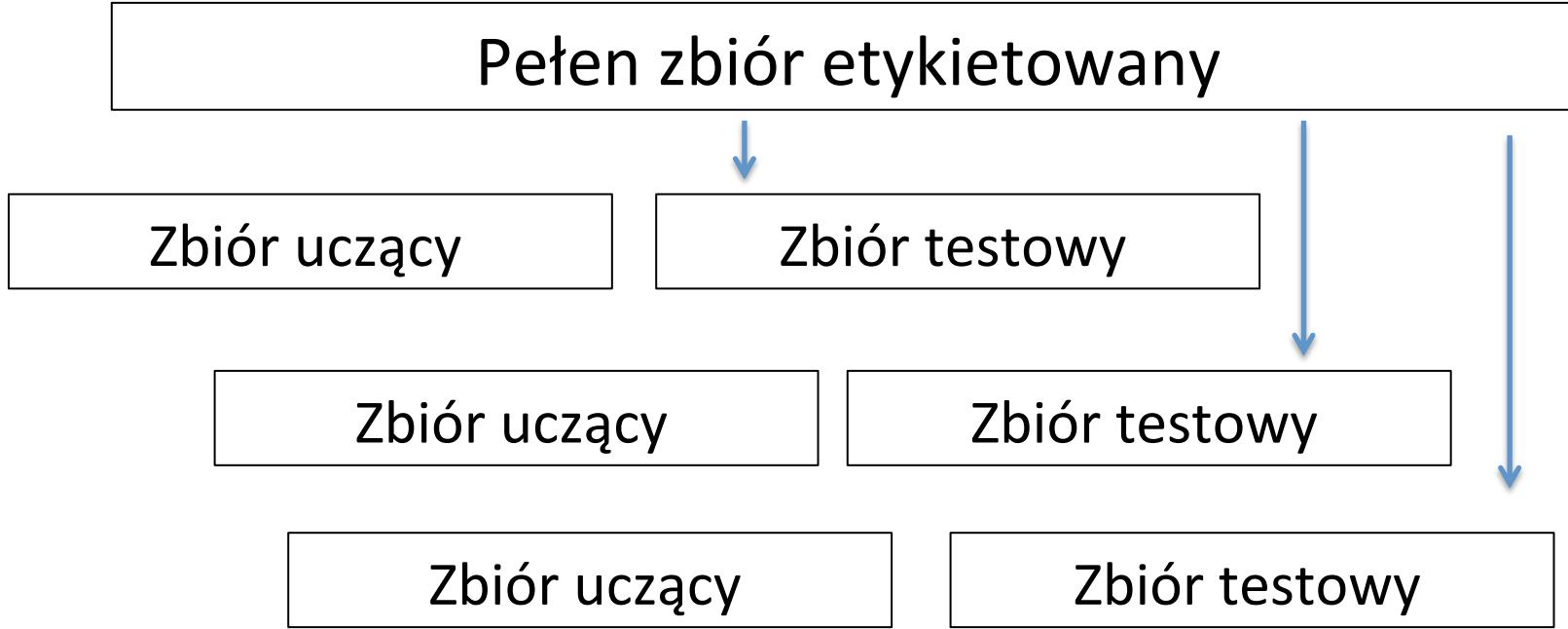
# Empiryczne metody estymacji

- Techniki podziału: „**hold-out**” (bardzo duża l. przykładów)
  - Użyj dwóch niezależnych zbiorów: uczącego (2/3), testowego (1/3)
  - Jednokrotny podział losowy stosuje się dla dużych zbiorów (**hold-out**)
- „**Cross-validation**” - Ocena krzyżowa
  - Podziel losowo dane w  $k$  podzbiorów (równomierne lub warstwowe)
  - Użyj  $k-1$  podzbiorów jako części uczącej i pozostałej jako testującej ( $k$ -fold cross-validation).
  - Oblicz wynik średni.
  - Stosowane dla danych o średnich rozmiarach (najczęściej  $k = 10$ )
- **leaving-one-out** = Dla małych rozmiarów danych  $< 100$  przykładów.
  - „Leaving-one-out” jest szczególnym przypadkiem, dla którego liczba iteracji jest równa liczbie przykładów
- Specjalne techniki statystyczne dla mniejszej l. przykładów

## Jednokrotny podział (hold-out) – duża liczba przykładów (> tysięcy)



# Wielokrotne podziały losowe

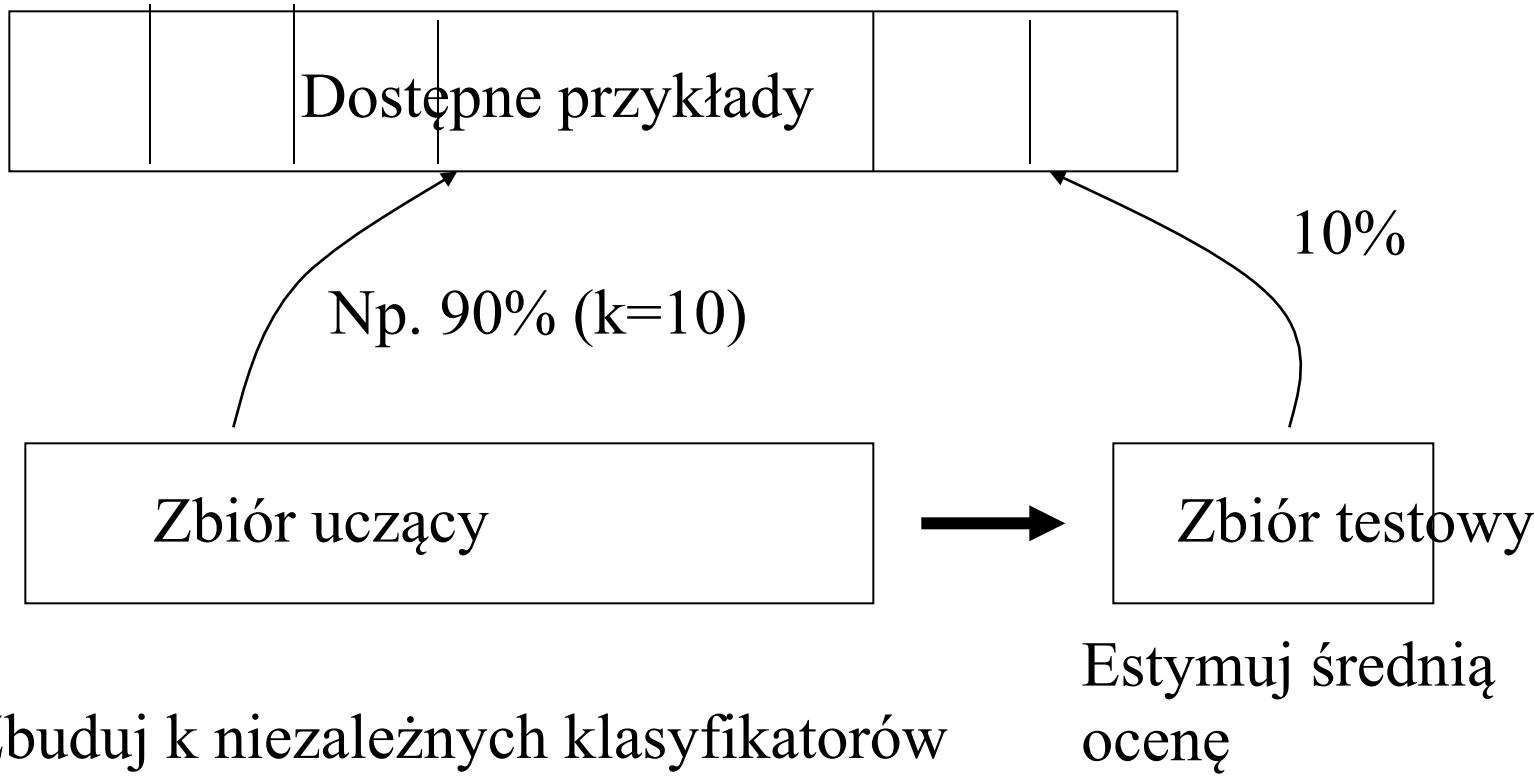


Po wielokrotnych podziałach losowych – oblicz wynik średni wybranej miary oceny każdego z klasyfikatorów

# Mniejsza liczba przykładów (od 100 do kilku tysięcy)

ang. k fold cross-validation

**Powtórz k razy**

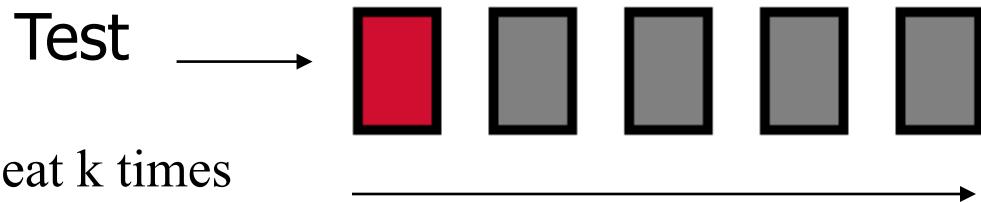


# K –fold cross-validation

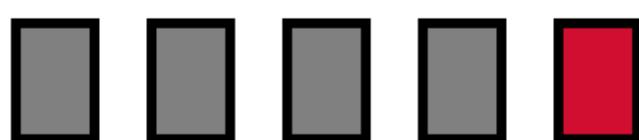
- Podziel losowo w k części (folds) w przybliżeniu tej samej wielkości



- Użyj jednego podziału do testowania a reszty do budowy klasyfikatora



- Repeat k times



# Uwagi o 10 fold cross-validation

- Stosuj wersję: **stratified** ten-fold cross-validation
- Dlaczego 10? Doświadczenie badaczy głównie eksperymentalne (zwłaszcza związane CART)
- Stratification – warstwowość ogranicza wariancje estymaty błędy!
- Lepsza wersja: repeated stratified cross-validation”
  - np. 10-fold cross-validation jest powtarzane kilka razy (z innym ziarnem rozkładu prawdopodobieństwa) i wynik średni z wielu powtórzeń.
  - Minimalizuje wariancje oszacowania

# Losowanie warstwowe (stratified)

Pełen zbiór uczący etykietowany

np. 70% klasa 1 i 30% klasa 2

Podział losowy

Zbiór uczący  
przybliż. 70% klasa 1 i 30% klasa 2

Zbiór testowy  
przyb. 70% kl 1 i 30% kl 2

Podobne proporcje losowania klas w ew. zbiorze walidacyjnym

Zachowujemy proporcje klas w losowaniu

# Przykład – C4.5 cross validation

The screenshot shows the C4.5 CRX software interface. The main window title is "C4.5 CRX (15 attributes, 490 training cases, 200 test cases)". The menu bar includes Data, Tree, Rules, Cross-validation, Special, and Help. Below the menu is a toolbar with icons for file operations and help.

**Before pruning**

Tree	Before pruning			After pruning				Estimate
	Size	Errors	Errors (test)	Size	Errors	Errors (test)		
1	101	18 ( 4.1%)	5 ( 10.2%)	50	28 ( 6.3%)	4 ( 8.2%)	15.0	
2	91	16 ( 3.6%)	9 ( 18.4%)	44	26 ( 5.9%)	9 ( 18.4%)	13.0	
3	95	16 ( 3.6%)	8 ( 16.3%)	48	23 ( 5.2%)	8 ( 16.3%)	13.0	
4	94	20 ( 4.5%)	8 ( 16.3%)	46	27 ( 6.1%)	7 ( 14.3%)	14.0	
5	102	17 ( 3.9%)	6 ( 12.2%)	51	26 ( 5.9%)	6 ( 12.2%)	14.0	
6	98	23 ( 5.2%)	11 ( 22.4%)	9	54 ( 12.2%)	5 ( 10.2%)	15.0	
7	112	21 ( 4.8%)	4 ( 8.2%)	41	30 ( 6.8%)	5 ( 10.2%)	14.0	
8	107	19 ( 4.3%)	13 ( 26.5%)	3	58 ( 13.2%)	8 ( 16.3%)	15.0	
9	88	25 ( 5.7%)	7 ( 14.3%)	40	29 ( 6.6%)	7 ( 14.3%)	14.0	
10	121	24 ( 5.4%)	7 ( 14.3%)	46	30 ( 6.8%)	7 ( 14.3%)	14.0	
Avg.	100.9	19.9 ( 4.5%)	7.8 ( 15.9%)	37.8	33.1 ( 7.5%)	6.6 ( 13.5%)	14.0	

**Cross-validation (rules)**

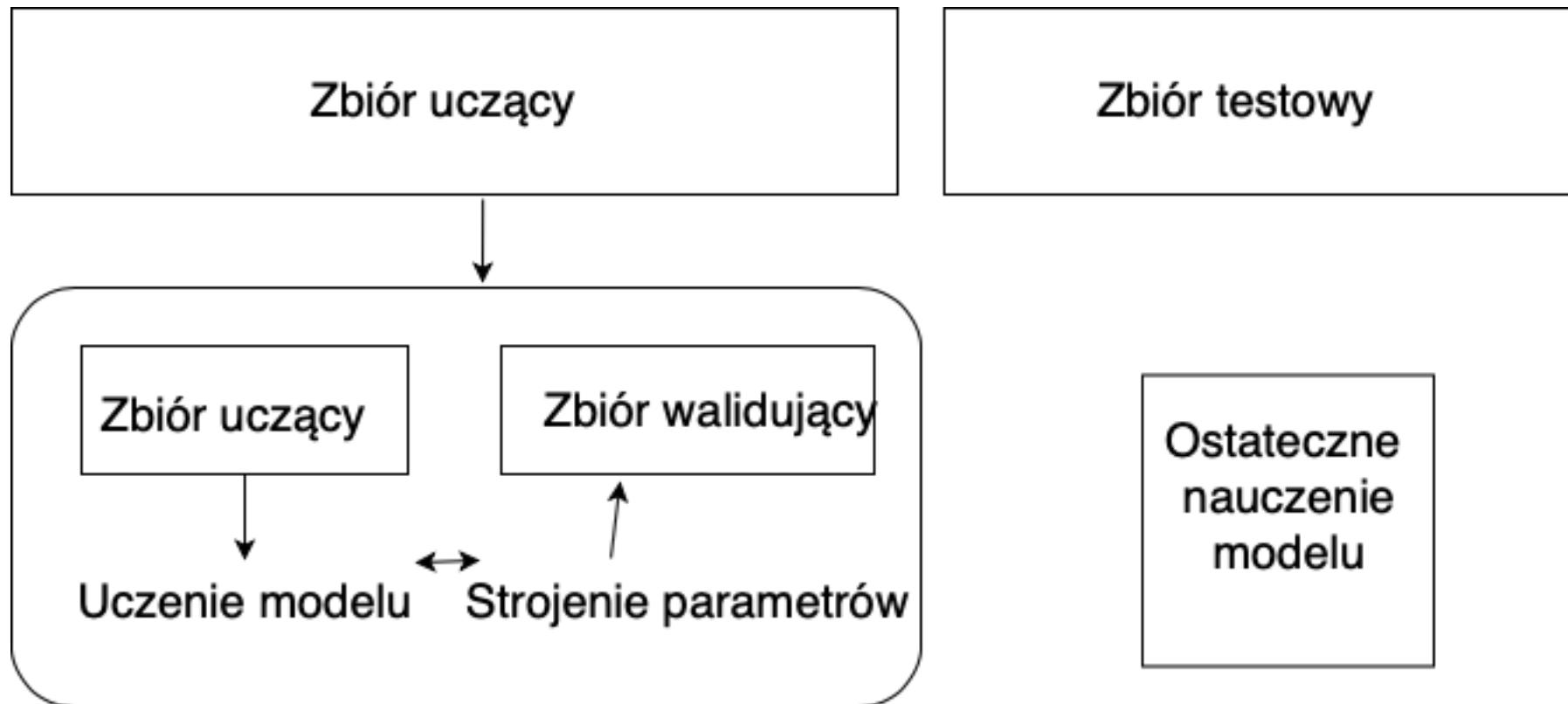
Ruleset	Size	Errors	Errors (test)
1	5	60 ( 13.6%)	1 ( 2.0%)
2	15	32 ( 7.3%)	10 ( 20.4%)
3	10	38 ( 8.6%)	9 ( 18.4%)
4	7	42 ( 9.5%)	7 ( 14.3%)
5	6	47 ( 10.7%)	5 ( 10.2%)
6	4	51 ( 11.6%)	6 ( 12.2%)
7	8	43 ( 9.8%)	6 ( 12.2%)
8	2	58 ( 13.2%)	8 ( 16.3%)
9	10	40 ( 9.1%)	6 ( 12.2%)
10	5	49 ( 11.1%)	7 ( 14.3%)
Avg.	7.2	46.0 ( 10.4%)	6.5 ( 13.2%)

Źródło – aplikacja wenw. PP

# Strojenie parametrów klasyfikatora i późniejsza ocena

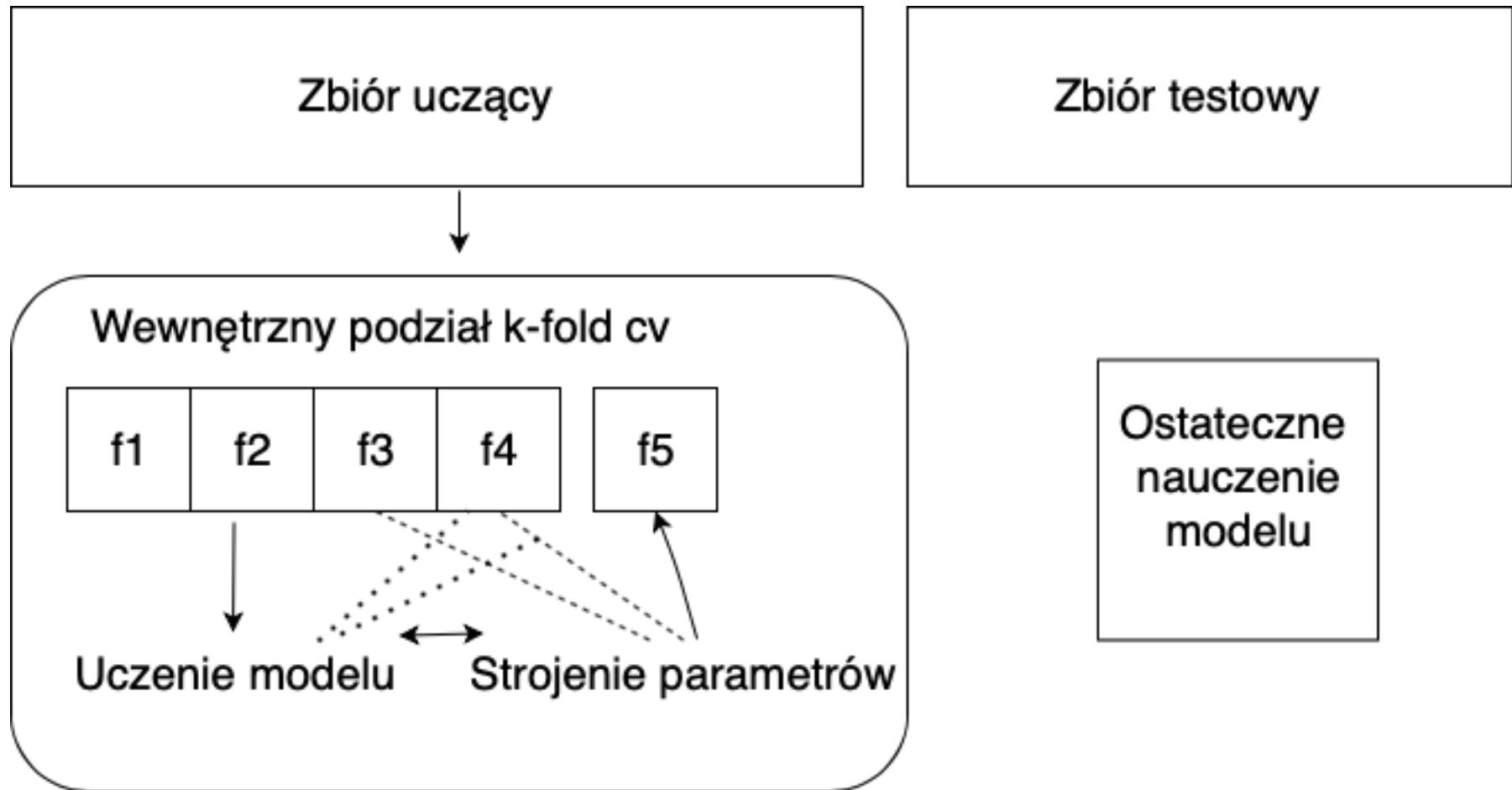
- Potrzeba specjalnego zbioru walidacyjnego, na którym prowadzi się eksperymentalne sprawdzanie wartości parametru (czasami wspomagany oprogramowaniem np. grid search)
  - Patrz np. redukcja (pruning) drzew, dobór k w algorytmie K-NN, strojenie parametrów ANN
- Wydzielony ze zbioru uczącego:
  - Właściwy zbiór uczący i walidujący = Niezależne od przykładów testowych
  - Może być tzw. wewnętrzna (w zbiorze uczącym) ocena krzyżowa = wtedy podwójna pętla oceny (cross validations)

# Strojenie klasyfikatora – potrzeba zbioru walidującego



Wydzielenie zbioru walidującego z części uczącej do doboru parametrów; Dla nich nauczenie klasyfikatora na pełnym zbiorze uczącym

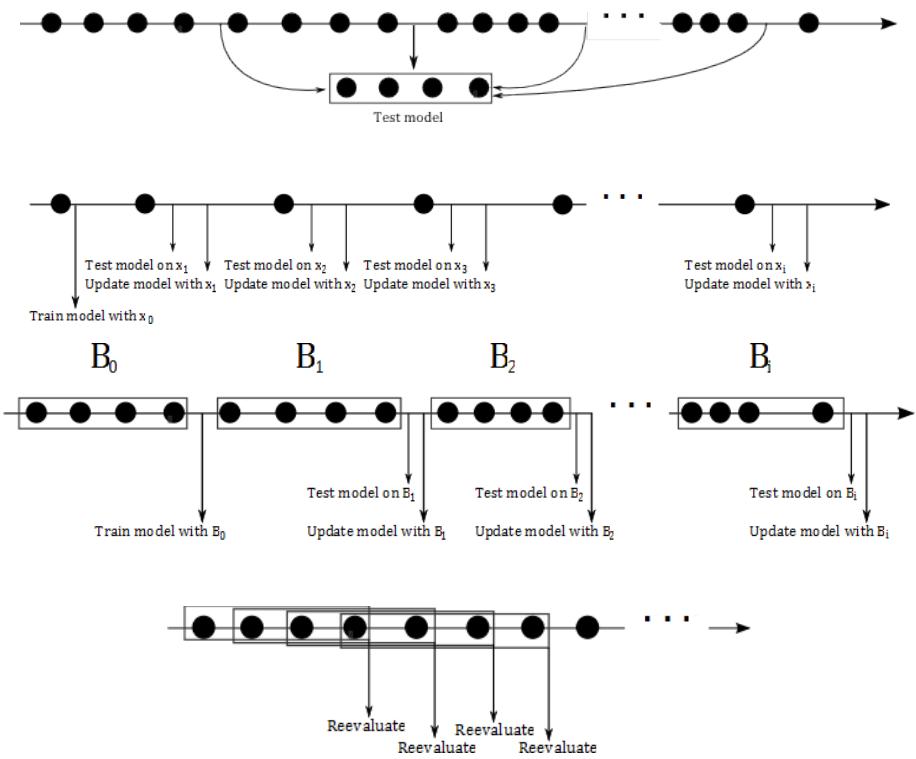
# Strojenie klasyfikatora – wewnętrzne wielokrotne podziały



Wykorzystaj wewnętrzną kocenę krzyżową – wymiana f bloków

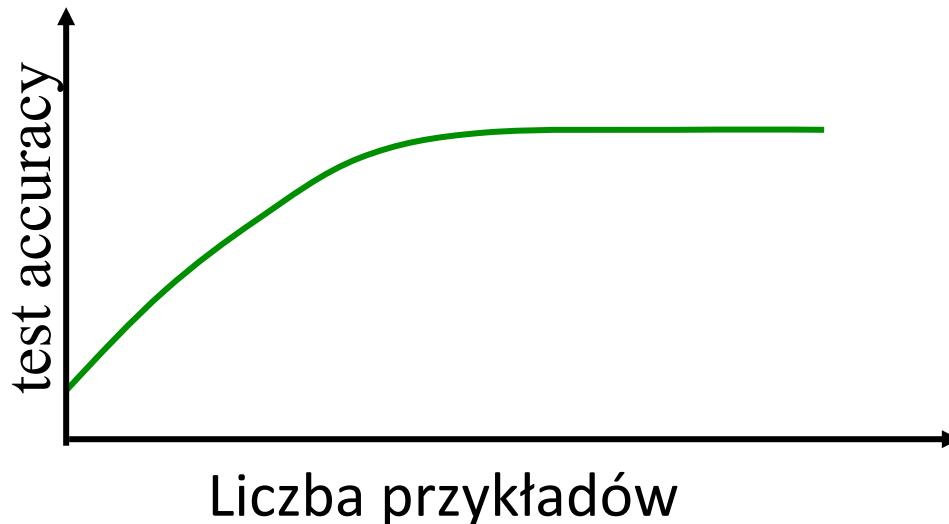
# Ocena klasyfikatorów przyrostowych strumieniowych

- Holdout
  - [np., Kirkby 2007]
- Test-then-train
  - [np., Kirkby 2007]
- Block-based evaluation
  - [np., Brzezinski & Stefanowski 2010]
- Prequential accuracy
  - [Gama et al. 2013]
- Inne miary
  - [Bifet & Frank 2010, Zliobaite et al. 2014]



Rysunek artykułu Brzeziński, Stefanowski Reacting to different types of concept drift: The accuracy updated ensemble algorithm

# Krzywe przyrostowego uczenia się



Klasyfikatory przyrostowe – wizualizacja graficzna uczenia się w odniesieniu do kolejno dostępnych przykładów; Jeśli przyrostowo dostępne dane uczące są stacjonarne, dobre algorytmy powinny prowadzić do stopniowego przyrostu zdolności predykcyjnej

Inna sytuacja z tzw. zmiennych strumieniach danych – dryft definicji pojęcia (ang. concept drift)

# Porównywanie wielu klasyfikatorów

- Często należy porównać dwa klasyfikatory
- Uwaga: porównanie z niezależnością od danych?
  - Generatory losowe
  - Rzeczywiste dane (problem dependent)
- Oszacuj 10-fold CV estimates.
- Trudność: wariancja oszacowania.
- Możesz oczywiście zastosować „repeated CV”.
- Lecz, jak wiarygodnie ustalić konkluzję – który jest lepszy?

# Porównywanie klasyfikatorów

- Jak oceniać skuteczność klasyfikacyjną dwóch różnych klasyfikatorów na tych samych danych?
- Ograniczamy zainteresowanie wyłącznie do trafności klasyfikacyjnej – oszacowanie techniką 10-krotnej oceny krzyżowej (ang. *k-fold cross validation*).
- Zastosowano dwa różne algorytmy uczące *AL1* i *AL2* do tego samego zbioru przykładów, otrzymując dwa różne klasyfikatory *KL1* i *KL2*. Oszacowanie ich trafności klasyfikacyjnej (10-fcv):
  - klasyfikator *KL1* → 86,98%
  - klasyfikator *KL2* → 87,43%.
- Czy uzasadnione jest stwierdzenie, że klasyfikator *KL2* jest skuteczniejszy niż klasyfikator *KL1*?

# Analiza wyniku oszacowania trafności klasyfikowania

<b>Podział</b>	<b>KI_1</b>	<b>KI_2</b>
1	87,45	88,4
2	86,5	88,1
3	86,4	87,2
4	86,8	86
5	87,8	87,6
6	86,6	86,4
7	87,3	87
8	87,2	87,4
9	88	89
10	85,8	87,2
<b>Srednia</b>	<b>86,98</b>	<b>87,43</b>
<b>Odchylenie</b>	<b>0,65</b>	<b>0,85</b>

- Test statystyczny (t-Studenta dla par zmiennych/zależnych)
  - $H_0$  : średnie oceny kl1 i kl2 się nie różnią znacząco
  - $H_1$ : średnia ocena jednego z klasyfikatorów jest wyższa niż drugiego
  - $temp = 1,733 \quad (p = 0,117) \quad ???$
  - ALE !!! W art. naukowych zastosuj odpowiednie poprawki przy wykonaniu testu (kwestia naruszenia założeń co do rozkładu  $t$ ).

**Porównanie działania dwóch klasyfikatorów DT oraz n<sup>2</sup> na wielu zbiorach danych (wyniki średnie z 10-oceny krzyżowej wraz z przedziałem ufności  $\alpha=0,95$ )**

<b>Data set</b>	<b>Classification accuracy DT (%)</b>	<b>Classification accuracy n<sup>2</sup> (%)</b>	<b>Improvement n<sup>2</sup> vs. DT (%)</b>
<b>Automobile</b>	<b>85.5 ± 1.9</b>	<b>87.0 ± 1.9</b>	1.5*
<b>Cooc</b>	<b>54.0 ± 2.0</b>	<b>59.0 ± 1.7</b>	5.0
<b>Ecoli</b>	<b>79.7 ± 0.8</b>	<b>81.0 ± 1.7</b>	1.3
<b>Glass</b>	<b>70.7 ± 2.1</b>	<b>74.0 ± 1.1</b>	3.3
<b>Hist</b>	<b>71.3 ± 2.3</b>	<b>73.0 ± 1.8</b>	1.7
<b>Meta-data</b>	<b>47.2 ± 1.4</b>	<b>49.8 ± 1.4</b>	2.6
<b>Primary Tumor</b>	<b>40.2 ± 1.5</b>	<b>45.1 ± 1.2</b>	4.9
<b>Soybean-large</b>	<b>91.9 ± 0.7</b>	<b>92.4 ± 0.5</b>	0.5*
<b>Vowel</b>	<b>81.1 ± 1.1</b>	<b>83.7 ± 0.5</b>	2.6
<b>Yeast</b>	<b>49.1 ± 2.1</b>	<b>52.8 ± 1.8</b>	3.7

# Dalsze porównania klasyfikatorów

- Dwa modele na wielu zbiorach danych – test rangowy Wilcooxona
  - Detale za chwile
- Wiele modeli/klasyfikatorów na wielu zbiorach danych
  - Test Friedmana (odpowiada na  $H_0$ : że nie ma znaczących różnic w ocenie klasyfikatorów;  $H_1$  negacja);
  - Jeśli odrzucimy  $H_0$ , przedstaw średnie rangi przypisane każdemu klasyfikatorowi;
  - Wykonanie posthoc analizy (np. Nemenyi) – policzenie CD krytycznej różnicy rang

# Globalna ocena (2 alg. wiele zb. danych)

Wilcoxon test (sparowany test rangowy)

H0: nie ma różnicy oceny klasyfikatorów

1. Różnice oceny klasyfikatorów uporządkuj wg. wartości bezwzględnych i przypisz im rangi.
2. R+ suma rang dla sytuacji gdy klasyfikator 1 jest lepszy niż klasyfikator 2 // R- sytuacja odwrotna
3. Oblicz statystykę  $T = \min\{R+; R-\}$   
Rozkład T jest stabelaryzowany / prosta reguła decyzyjna
4. Dla odpowiednio dużej liczby  $m$  zbiorów danych można stosować przybliżenie z

$$z = \frac{\min\{R+; R-\} - \frac{1}{4}m(m-1)}{\sqrt{\frac{1}{24}m(m+1)(2m+1)}}$$

# Porównanie dwóch klasyfikatorów

Dane	Klasyf B	Klas M	Różnica	ranga
D1	0,763	0,768	+0,005	3,5
D2	0,599	0,591	-0,008	7
D3	0,954	0,971	+0,017	9
...	...	...	...	...
D12	0,619	0,666	+0,047	13
D13	0,972	0,981	+0,009	8
D14	0,957	0,978	+0,021	10

- Obliczenie średnich rang  $R+=3,5+9+12+5+6+14+11+13+8+10+1,5 = 83$
- $R- = 7 + 3,5 + 1,5 = 12$
- $Z = -2.51 < - 1,96 / H_0$  odrzucamy, klasyfikator M średnio lepszy niż Klasyfikator B

# Test Friedmana

- H0: oceny wszystkich klasyfikatorów nie różnią się
- H1: Oceny niektórych klasyfikatorów są lepsze niż pozostałych

Dla każdego zbioru danych ( $i=1,..,N$ ) ustawiamy rangi  $m$  klasyfikatorów wg. ich rezultatów

Następnie oblicz średnie rangi klasyfikatorów  $r_j$  ( $j=1,..m$ )

Statystyka Friedmana ma rozkład  $\chi^2$  z  $N-1$  stopniami swobody

$$\chi_F^2 = \frac{12m}{N(N+1)} \left( \sum_{j=1}^m r_j^2 - \frac{N(N+1)^2}{4} \right)$$

Jeśli odrzucamy H0, to liczymy post-hoc analize (np. Nemeyi test)

$$CD = q_\alpha \sqrt{\frac{N(N+1)}{6m}}$$

Algorytmy z różnicą średnich rang większą niż CD są statystycznie lepsze

# Test Friedmana

Dane	Klasyfikator1	Klasyfikator2	Klasyfikator3
Zb danych 1	1	3	2
Zb danych 2	1,5	1,5	3
Zb danych 3	1	2	3
Zb danych 4	2	3	1
Zb danych 5	2,5	2,5	1
Średnie rangi	1,6	2,4	2,0

$F_{obl} = 37,1$  a krytyczna statystka  $k = 9,488$  = odrzucamy  $H_0$

CD wartość krytyczna - klasyfikatory są nierozróżnialne

# Podejścia teoretyczne

- Obliczeniowa teoria uczenia się (COLT)
  - PAC model (Valiant)
  - Wymiar Vapnik Chervonenkis → VC Dimension
- Pytania o ogólne prawa dotyczące procesu uczenia się klas pewnych funkcji z przykładów - rozkładów prawdopodobieństwa.
- Silne założenia i ograniczone odniesienia do problemów praktycznych.

# Perspektywa opisowa

- Trudniejsza niż ocena zdolności klasyfikacyjnych.
- Rozważmy przykład reguł:
  - Klasyfikacyjne  
Jeżeli (atr1=wartość) and (atr3=wartość) to (klasa=A)
  - Asocjacyjne.  
Jeżeli ACD to B
- Pojedyncza reguła oceniana jako potencjalny reprezentant „interesującego” wzorca z danych
  - W literaturze propozycje tzw. ilościowych miar oceny reguł oraz sposoby definiowania „interesujących” reguł, także na podstawie wymagań podawanych przez użytkownika.

# Przykład reguł klasyfikacyjnych

Minimalny zbiór pewnych reguł

- *if ( $a_2 = s$ )  $\wedge$  ( $a_3 \leq 2$ ) then ( $d = C_1$ )*  
 $\{x_1, x_7\}$
- *if ( $a_2 = n$ )  $\wedge$  ( $a_4 = c$ ) then ( $d = C_1$ )*  
 $\{x_3, x_4\}$
- *if ( $a_2 = w$ ) then ( $d = C_2$ )*    $\{x_2, x_6\}$
- *if ( $a_1 = f$ )  $\wedge$  ( $a_4 = a$ ) then ( $d = C_2$ )*  
 $\{x_5, x_8\}$

Reguła z  $conf < 1$

- *if ( $a_1 = m$ ) then ( $d = C_1$ )*  
 $\{x_1, x_3, x_7 \mid x_6\}$    3/4

id.	$a_1$	$a_2$	$a_3$	$a_4$	$d$
$x_1$	m	s	1	a	C1
$x_2$	f	w	1	b	C2
$x_3$	m	n	3	c	C1
$x_4$	f	n	2	c	C1
$x_5$	f	n	2	a	C2
$x_6$	m	w	2	c	C2
$x_7$	m	s	2	b	C1
$x_8$	f	s	3	a	C2

# Opisowe miary oceny reguł

- Miary dla reguły  $r$  (*jeżeli P to Q*) definiowane na podstawie zbioru przykładów  $U$ , z którego została wygenerowana.
- Tablica kontyngencji dla reguły *jeżeli P to Q* :

	$Q$	$\neg Q$	
$P$	$n_{PQ}$	$n_{P \neg Q}$	$n_P$
$\neg P$	$n_{\neg P Q}$	$n_{\neg P \neg Q}$	$n_{\neg P}$
	$n_Q$	$n_{\neg Q}$	$n$

- Przegląd różnych miar, np.: Ya Y.Y, Zhong N.: An analysis of quantitative measures associated with rules, w: Proc. of the Third Pacific-Asia Conference on Knowledge Discovery and Data Mining, LNAI 1574, Springer, 1999, s. 479-488.
- Także rozprawa habilitacyjna J.Stefanowski: Algorytmy indukcji reguł w odkrywaniu wiedzy (dostępna przez WWW) oraz rozprawa doktorska p. Izabeli Szczęch.

# Popularne miary oceny reguł

- Wsparcie reguły jeśli P to Q (ang. *support*) zdefiniowane jako:

$$G(P \wedge Q) = \frac{n_{PQ}}{n}$$

- Dokładność (ang. *rule accuracy*) / wiarygodność (ang. *confidence*) reguły (bezwzględne wsparcie konkluzji Q przez przesłankę P):

$$AS(Q | P) = \frac{n_{PQ}}{n_P}$$

- Względne pokrycie (ang. *coverage*) reguły zdefiniowane jako:

$$AS(P | Q) = \frac{n_{PQ}}{n_Q}$$

# Zaawansowane miary oceny reguł

**Change of support** – rodzaj konfirmacji wsparcia hipotezy Q przez wystąpienie przesłanki P (propozycja Piatetsky-Shapiro)

$$CS(Q | P) = AS(Q | P) - G(Q)$$

gdzie

$$G(Q) = \frac{n_Q}{n}$$

Zakres wartości od -1 do +1 ; Interpretacja: różnica między prawdopodobieństwami a prior i a posterior; dodatnie wartości wystąpienie przesłanki P powoduje konkluzję Q; ujemna wartość wskazuje że nie ma wpływu.

**Degree of independence:**

$$IND(Q, P) = \frac{G(P \wedge Q)}{G(P) \cdot G(Q)}$$

# Złożone miary oceny reguł

Połączenie miar podstawowych

Significance of a rule (propozycja Yao i Liu)

$$S(Q|P) = AS(Q|P) \cdot IND(Q, P)$$

Klosgen's measure of interest

$$K(Q|P) = G(P)^\alpha \cdot (AS(Q|P) - G(Q))$$

Michalski's weighted sum

$$WSC(Q|P) = w_1 \cdot AS(Q|P) + w_2 \cdot AS(P|Q)$$

The relative risk (Ali, Srikant):

$$r(Q|P) = \frac{AS(Q|P)}{AS(Q|\neg P)}$$

# Przykład diagnostyki technicznej

- Bada się stan techniczny 76 autobusów tego samego typu (dokładnie ich silników) na podstawie symptomów stanu technicznego - parametrów pochodzących z okresowych badań diagnostycznych [dane prof. J.Zak, analiza J.Stefanowski]
  - Autobusy są podzielone na dwie klasy: dobry i zły stan techniczny pojazdu
- Cel analizy
  - Ocenia się jakość diagnostyczną symptomów stanu technicznego
  - Poszukuje się zależności pomiędzy wartościami najistotniejszych w tych symptomów a przydziałem do klas = konieczność interpretacji wzorców w postaci reguł
  - Konstruuje się klasyfikator stanu technicznego

# Rozważane symptomy

$s_1$  – prędkość maksymalna [km/h],

$s_2$  – ciśnienie sprężania [Mpa],

$s_3$  – zawartość elementów smołowatych w spalinach wylotowych [%],

$s_4$  – moment obrotowy silnika [Nm],

$s_5$  – letnie zużycie paliwa [l/100lm],

$s_6$  – zimowe zużycie paliwa [l/100km],

$s_7$  – zużycie oleju [l/1000km],

$s_8$  – aktualna moc silnika [KM].

Dwie klasy decyzyjne:

1. Autobusy z silnikami w dobrym stanie – dalsza eksploatacja (46),
2. Autobusy z silnikami w złym stanie – konieczność napraw (30).

# Minimalny zbiór reguł klasyfikujących

1. if ( $s_2 \geq 2.4 \text{ MPa}$ ) & ( $s_7 < 2.1 \text{ l}/1000\text{km}$ ) then  
(technical state=good) [46]
2. if ( $s_2 < 2.4 \text{ MPa}$ ) then (technical state=bad) [29]
3. if ( $s_7 \geq 2.1 \text{ l}/1000\text{km}$ ) then (technical state=bad) [24]

Oszacowana trafność klasyfikowania  
**(‘leaving one out’ test)** 98.7%.

Lecz trudność ich interpretacji

# Poszukiwanie innych reguł z danych

## Próg satysfakcji dla miary support (51%):

1. if ( $s1 > 85 \text{ km/h}$ ) then (technical state=good) [34]
2. if ( $s8 > 134 \text{ kM}$ ) then (technical state=good) [26]
3. if ( $s2 \geq 2.4 \text{ MPa}$ ) & ( $s3 < 61 \%$ ) then (technical state=good) [44]
4. if ( $s2 \geq 2.4 \text{ MPa}$ ) & ( $s4 > 444 \text{ Nm}$ ) then (technical state=good) [44]
5. if ( $s2 \geq 2.4 \text{ MPa}$ ) & ( $s7 < 2.1 // 1000 \text{ km}$ ) then (technical state=good) [46]
6. if ( $s3 < 61 \%$ ) & ( $s4 > 444 \text{ Nm}$ ) then (technical state=good) [42]
7. if ( $s1 \leq 77 \text{ km/h}$ ) then (technical state=bad) [25]
8. if ( $s2 < 2.4 \text{ MPa}$ ) then (technical state=bad) [29]
9. if ( $s7 \geq 2.1 // 1000 \text{ km}$ ) then (technical state=bad) [24]
10. if ( $s3 \geq 61 \%$ ) & ( $s4 \leq 444 \text{ Nm}$ ) then (technical state=bad) [28]
11. if ( $s3 \geq 61 \%$ ) & ( $s8 < 120 \text{ kM}$ ) then (technical state=bad) [27]

# Uwagi do źródeł

Wykorzystano książki:

- S.Weiss, C.Kulikowski: Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning and Expert Systems, Morgan Kaufmann 1991.
- **N.Japkowicz, M. Shah: Evaluating Learning Algorithms: A Classification Perspective, Cambridge Presss 2011.**
- I.Konennko, M.Kukar: Machine Learning and Data Mining, 2007.
- J.Han, M.Kember: Data mining. Morgan Kaufmann 2001.

oraz inspiracje ze slajdów wykładów:

- J.Han; G.Piatetsky-Shapiro; D.Page, A.Avati + materiały związane z WEKA i prezentacji W.Kotłowski nt. Statistical Analysis of Computational Experiments in Machine Learning

Wybrane artykuły

- Patrz następny slajd

# Wybrane artykuły

1. Kohavi, R. (1995): A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proc. of the 14th Int. Joint Conference on Artificial Intelligence*, 1137–1143.
2. Salzberg, S. L. (1997): On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery*, 1, 317–328.
3. Dietterich, T. (1998): Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:7, 1895–1924.
4. Bouckaert, R. R. (2003): Choosing between two learning algorithms based on calibrated tests. *ICML 2003*.
5. Bengio, Y., Grandvalet, Y. (2004): No unbiased estimator of the variance of k-fold cross-validation. *Journal of Machine Learning Research*, 5, 1089–1105.
6. Demsar, J. (2006): Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7, 1–30.
7. S. Raschka (2018) Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning, arXiv 2018
8. Sesja spacjalna nt. oceny systemów uczących (N.Japkowicz) na ICML 2007 + tutorial  
Oraz nowsze artykuły, np. o testach statystycznych Salvador Garcia Univ. Granada

# **Pytanie i komentarze?**

Dalszy kontakt:

[jerzy.stefanowski@cs.put.poznan.pl](mailto:jerzy.stefanowski@cs.put.poznan.pl)

<http://www.cs.put.poznan.pl/jstefanowski/>



**Fundusze  
Europejskie**  
Polska Cyfrowa



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz  
Rozwoju Regionalnego



# Systemy uczące się Drzewa regresji wykład 6

Jerzy Stefanowski  
Instytut Informatyki PP  
2021

Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI-TECH)  
projekt finansowany z środków Programu Operacyjnego Polska Cyfrowa  
POPC.03.02.00-00-0001/20



**Fundusze  
Europejskie**  
Polska Cyfrowa



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz  
Rozwoju Regionalnego



# Plan wykładu

1. Zadanie regresji w uczeniu maszynowym
2. Ograniczenia klasycznych modeli liniowych, podejścia nieparametryczne i wykorzystanie podziałów dziedziny zmiennych niezależnych
3. Drzewa - rekurencyjny podział przestrzeni cech oraz estymacja predykcji zmiennej wyjściowej – ilustracja oraz przykład
4. Drzewa regresji vs. drzewa klasyfikacyjne
5. Kryterium podziału w węźle
6. Zatrzymanie budowy drzewa vs. tzw. post-pruning
7. Inne rodzaje drzew, tzw. model trees

# Przypomnienie regresji

## Zadanie regresji (predykcja zmiennej liczbowej)

- Metoda oszacowania wartości liczbowej zmiennej zależnej (objaśnianej)  $y$  na podstawie wartości zmiennych niezależnych  $x$  [klasyczne w statystyce]
- Poszukujemy modelu  $\hat{y} = f(x, \beta)$  – wybór postaci funkcji  $f$  oraz estymacja parametrów

Popularne modele liniowe – regresja wieloraka / wielowymiarowa

$$y = x_1 w_1 + x_2 w_2 + \dots + x_m w_m + w_0$$

Na ogół minimalizacja funkcji straty w postaci RSME (dot reszty  $y - \hat{y}$ )

$$S = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Różne metody szacowania (MNK, Est.Najw. Wiaryg., Spadek Gradientu)

Dostępna w wielu programach, np. SAS, SPSS, R lub Statistica,...

Liczne zastosowania praktyczne

# Liczne zastosowania

Predykcja:

- Wyceny produktów finansowych, akcji giełdowych, portfolio analiza
- Cen sprzedaży, wynajmu mieszkań
- Sektor sprzedaży różnych produktów
- Poziomu satysfakcji klientów oraz czasu współpracy w CRM, rynku ubezpieczeniowym, itd.

Ocena pracochłonności projektów (COCOMO)

Model oceny efektywności systemów (np. komputerowych)

Analiza ryzyka przedsięwzięć

I wiele innych, ....

# Przykład predykcji cen mieszkań

- Harrison i Rubinfeld – badanie związku między różnymi wskaźnikami jakości życia a cenami nieruchomości w okolicach Bostonu tzw. **Boston Housing** patrz [lib.stat.cmu.edu/datasets/boston](http://lib.stat.cmu.edu/datasets/boston)
- 506 domów opisanych przez 14 cech
- Zadanie – predykcja ceny nieruchomości, pośrednio poziom zanieczyszczenia (koncentracja tlenku azotu)
  1. CRIM - per capita crime rate by town
  2. ZN - proportion of residential land zoned for lots over 25,000 sq.ft.
  3. INDUS - proportion of non-retail business acres per town.
  4. CHAS - Charles River dummy variable (1 if tract bounds river; 0 otherwise)
  5. NOX - nitric oxides concentration (parts per 10 million)
  6. RM - average number of rooms per dwelling
  7. AGE - proportion of owner-occupied units built prior to 1940
  8. DIS - weighted distances to five Boston employment centres
  9. RAD - index of accessibility to radial highways
  10. TAX - full-value property-tax rate per \$10,000
  11. PTRATIO - pupil-teacher ratio by town
  12. B -  $1000(Bk - 0.63)^2$  where Bk is the proportion of blacks by town
  13. LSTAT - % lower status of the population
  14. MEDV - Median value of owner-occupied homes in \$1000's

# Inne repozytoria

- **Kaggle** – kilka konkursów predykcji cen nieruchomości (np. Ames data lub new york)
- **UCI ML repository** – specjalna kolekcja benchmarkowych danych dla regresji (**134 zbiory**)  
<https://archive.ics.uci.edu/ml/index.php>
- Spojrzeć do artykułu: PMLB: a large benchmark suite for machine learning evaluation and comparison (2017) – repozytorium <https://github.com/EpistasisLab/pmlb>

# Poprzednie wykłady i laboratorium

**Modele liniowe** – też regresja

Różne formy funkcji straty (nie tylko błąd resztowy  $y - \hat{y}$ )

Zasady estymacji metodą największej wiarygodności

Przeuczenie – na przykładzie różnych rodzajów regresji

W systemach uczących – modele predykcji zmiennych liczbowych

Nie tylko o podłożu liniowych modeli statystycznych

Także nauczone sieci neuronowe – predykcja wielu zmiennych liczbowych  $y_1, y_2, \dots, y_k$  – neurony wyjściowe

# Regularizacja w regresji

- Ridge (regresja grzbietowa) - Czynnik dodatkowy

$$\sum_{j=1}^m (w_j)^2 \leq t$$

gdzie  $t$  jest ograniczeniem, a całość sformułowania

$$w = \operatorname{argmin} \left( \sum_{i=1}^n (y_i - \sum_{j=1}^m x_{ij} w_j - w_0)^2 + \lambda \sum_{j=1}^m w_j^2 \right)$$

$\lambda$  – mnożniki Lagrange'a

Lasso – inny czynnik dodatkowy

$$\sum_{j=1}^m |w_j| \leq t$$

# Dyskusja klasycznej regresji

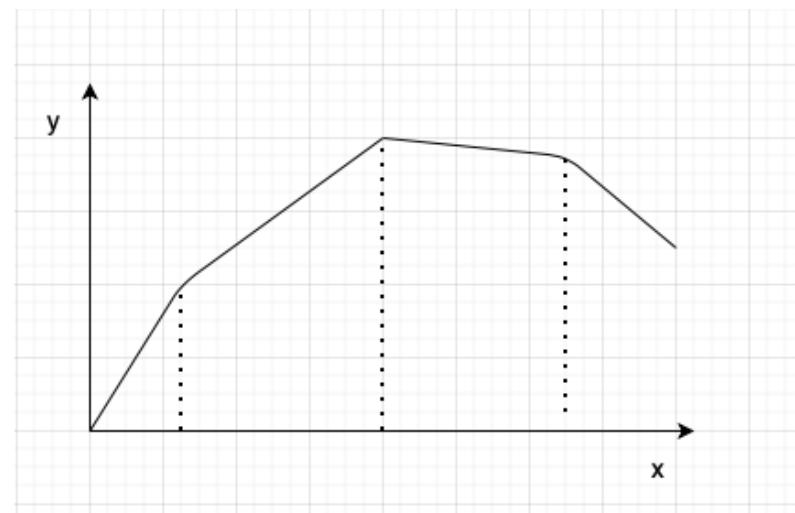
- **Liniowa regresja**
  - Model globalny – zmienne x obejmują całą przestrzeń cech,
  - Założenie liniowości i nieskorelowanie zmiennych – lecz rzeczywiste dane / przykłady uczące mają rozkłady, gdzie cechy mogą być wzajemnie nieliniowe („świat dla sztucznych sieci neuronowych”)
  - W problemach uczenia maszynowego na ogół „mieszanki” różnych typów cech / atrybutów
- **Nieliniowa regresja**
  - Metody nieparametryczne estymacji – patrz literatura, np. książka J.Koronackiego Statystyczne systemy uczące
  - Także podział funkcji na segmenty / części

# Aproksymacje lokalne – regresja nieparametryczna

- Więcej w książce J.Koronacki, J.Ćwik: Statystyczne systemy uczące się.
- Estymując funkcję regresji staramy się uwzględnić w modelu własności lokalne
- Składanie kilku „funkcji podstawowych” zdolnych lokalnie przybliżyć własności pewnych podobszarów dziedziny
- Regresyjne funkcje sklejane z „węzłami”
- Tzw. regresja lokalnie ważona

Locally weighted regression

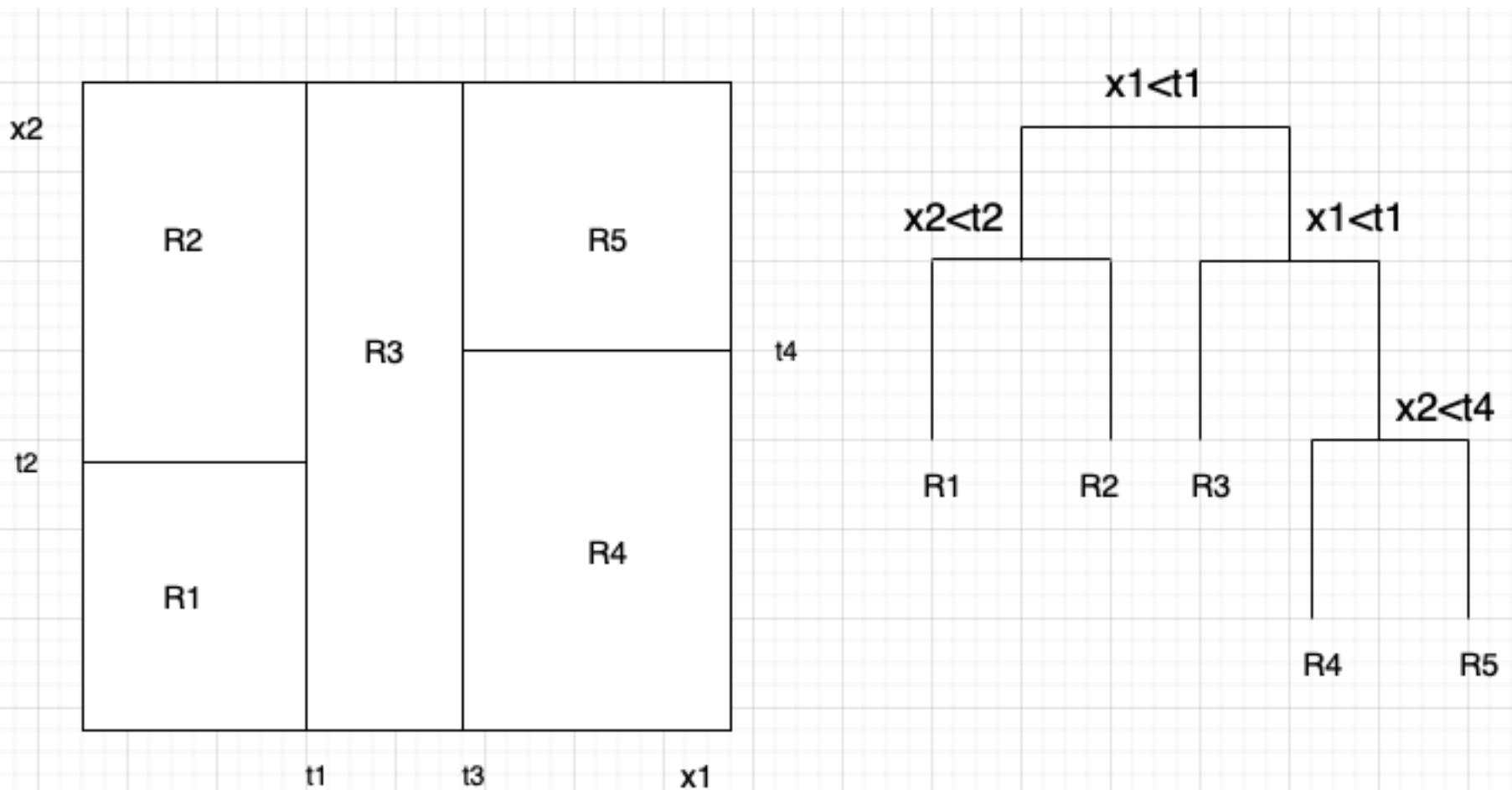
$$y = \alpha + \sum_{j=1}^p f_j(\mathbf{x}, \beta)$$



# W stronę drzew regresji

- **Inne podejście do podziałów wielowymiarowej przestrzeni cech /atrybutów**
  - Stopniowo podziel przestrzeń na obszary (CART – hiperkostki - prostopadłościany),
  - Procedura rekurencyjna podziału (top-down) jak w drzewach klasyfikacyjnych)
  - Uwzględnianie różnych typów cech / atrybutów
- **Preidykcja**
  - W końcowym obszarze można zbudować prostszy model predykcji  $y$
  - Drzewa regresji – estymacja pojedynczej wartości  $y$  na podstawie rozkładu przykładów należących do obszaru
  - ang. Model trees – zbuduj model regresji liniowej, jeśli jest wystarczająca liczba przykładów w obszarze

# Ilustracja drzewa regresji i zasad podziału przestrzeni cech



# Drzewa regresji

- **Węzły drzewa** – sekwencja pytań o testy na wartościach atrybutów (np. is horsepower > 50 and is gradutestudent)
- **Predykcja w obszarze** (hiper-kostce)
  - CART – przykłady należące do hiperkostki  $R_j$  – w miarę jednorodne (ze względu na charakterystykę  $x$  + możliwe wyjście  $y$  – czyli posiadają dość podobne wartości  $y$  dla  $x_i$  z  $R_j$ )
  - Oszacowujemy wartość przeciętną wśród dla  $x_i$  z  $R_j$  -> średnia arytmetyczna  $\hat{y}$  w  $(R_j)$

$$\hat{y} = \frac{1}{|R_j|} \sum_{x_i \in R_j} y_i$$

- Podziały w liściach muszą być dość homogeniczne i reprezentowane są przez stała wartość (średnią)! - lepszy estymator niż mediana z uwagi na kryterium oceny drzewa

# Przykład danych cars

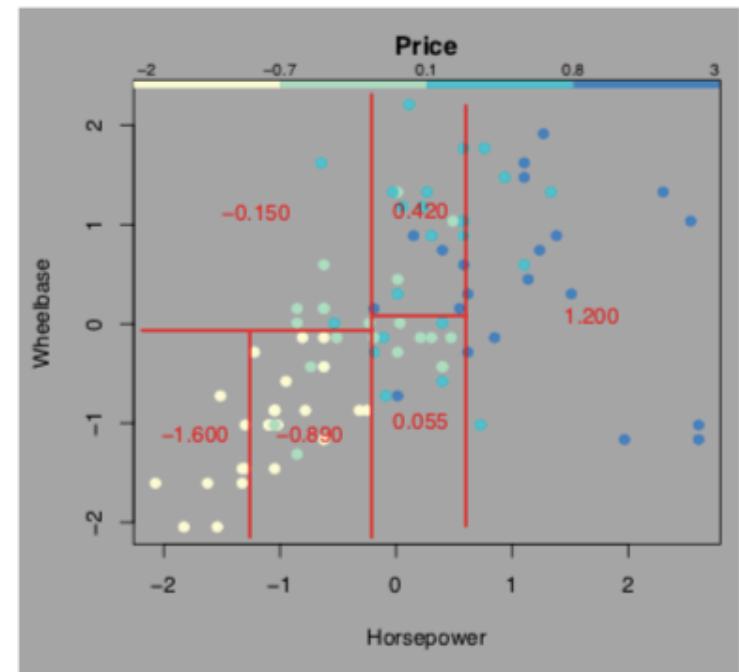
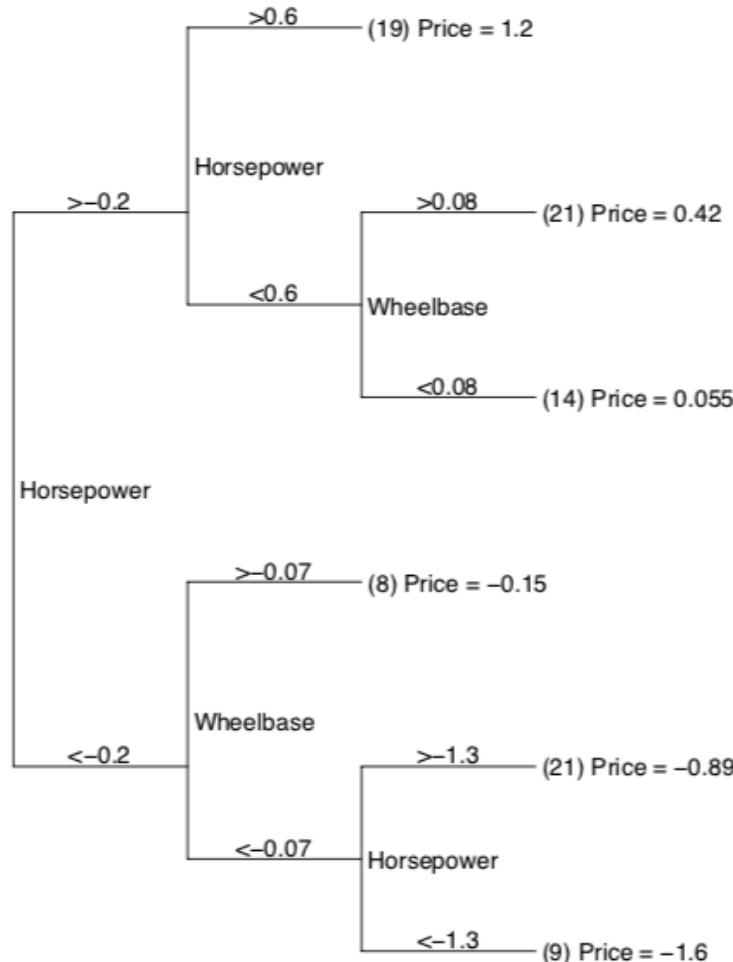


Figure 2: The partition of the data implied by the regression tree from Figure 1

Figure 1: Regression tree for predicting price of 1993-model cars. All features

# Drzewa regresyjna a klasyfikacyjne

- Podobne zasady rekurencyjnego podziału zbioru przykładów uczących
- Podobna struktura drzewa
- Inne kryteria (podziału, stopu, ...)
- Inne miara oceny predykcji – funkcja straty ciągła, błąd średniokwadratowy  $(y - \hat{y})^2$
- Ponadto możliwość upraszczania, redukcji wielkości drzewa
- Spójrz do pracy przeglądowej Wei-Yin Loh: Fifty Years of Classification and Regression Trees, International Statistical Review (2014)

# Problemy w budowie drzew regresji

- Kryteria oszacowania jakości drzewa
- Zasady wykonania podziału w węźle
- Określenie kryterium stopu (kiedy węzeł drzewa stanie się liściem)
- Alternatywne upraszczanie drzewa (tzw. ang. pruning)

# Kryterium oceny

- Ogólne kryterium oceny predykcji drzewa  $T$ , wybierz drzewo min. błąd predykcji

$$\min_T \frac{1}{n} \sum_{i=1}^n (\gamma_i - T(x_i))^2$$

- W przypadku drzewa regresji dzielącego przestrzeń na  $J$  obszarów  $R_j$  – można minimalizować

$$\sum_{j=1}^J \sum_{i \in R_j} (\gamma_i - \hat{\gamma}_{R_j})$$

- Dalsze wersje drzew regresji – można dodać czynnik regularizacji, zwłaszcza dla upraszczania (patrz książka E. Gatnara)

# Kryterium podziału w węźle

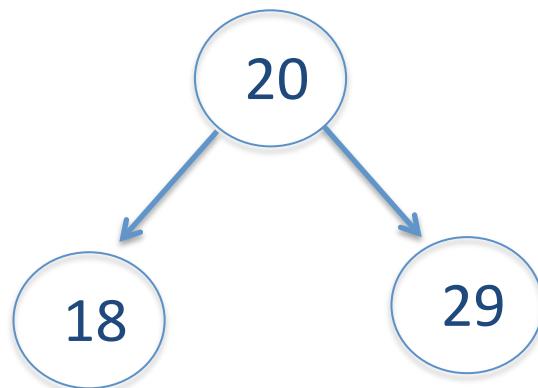
- Zbiór przykładów S zostanie podzielony na dwie części S<sub>1</sub> i S<sub>2</sub> (np., wg. progu  $\tau$  na atrybutie A)
- Ocena podziału wg. kryterium błędu

$$\sum_{i \in S_1} (\gamma_i - \hat{\gamma}_{S_1})^2 + \sum_{i \in S_2} (\gamma_i - \hat{\gamma}_{S_2})^2$$

- Dla atrybutu A sprawdza się możliwe progi  $\tau$  i wybiera ten, który minimalizuje błąd po podziale
- Dla atrybutów nominalnych – możliwe więcej podziałów niż dwa S = S<sub>1</sub>, S<sub>2</sub>, ..., S<sub>v</sub> – wtedy suma v elementów
- Niektóre źródła – ważenie liczbą przykładów w S<sub>j</sub> lub ich prawdopodobieństwem  $p_j$

# Intuicja podziałów

- Niezależnie od minimalizacji kryterium błędu
- Dąży się do tego, aby w obszarach (kostkach) przykłady miały zblżone do siebie wartości
- Ponadto stara się rozdzielić wartości y niższe od wyższych i przydzielać je wydzielonych węzłów



Średnie wartości y  
w kostce / obszarze

# Ogólny schemat

- Rozpocznij od pojedynczego węzła z S przykładami – oblicz  $\hat{Y}$  na ich podstawie
- Jeśli wszystkie przykłady w S mają tą samą wartość – stop;

W przeciwnym razie - po wszystkim atrybutach poszukaj najlepszego podziału S, który minimalizuje błąd; jeśli spadek błędu jest zbyt mały lub zbyt mało przykładów (najczęściej wymaga się się aby  $n_j \geq 5$ ) to także zatrzymaj

- Gdy stop, to utwórz liść odpowiadający  $R_j$  (obl.  $\hat{Y}_{Rj}$ ), w przeciwnym razie powróć do pierwszego punktu

Na ogół w liściu/węźle wymaga się minimalnej liczby przykładów dla obliczeń (CART do 5 przykładów)

# Upraszczanie drzew

Podobnie jak w drzewach klasyfikacyjnych – zbuduj pełne drzewo i i zastosuj tzw. post-pruning

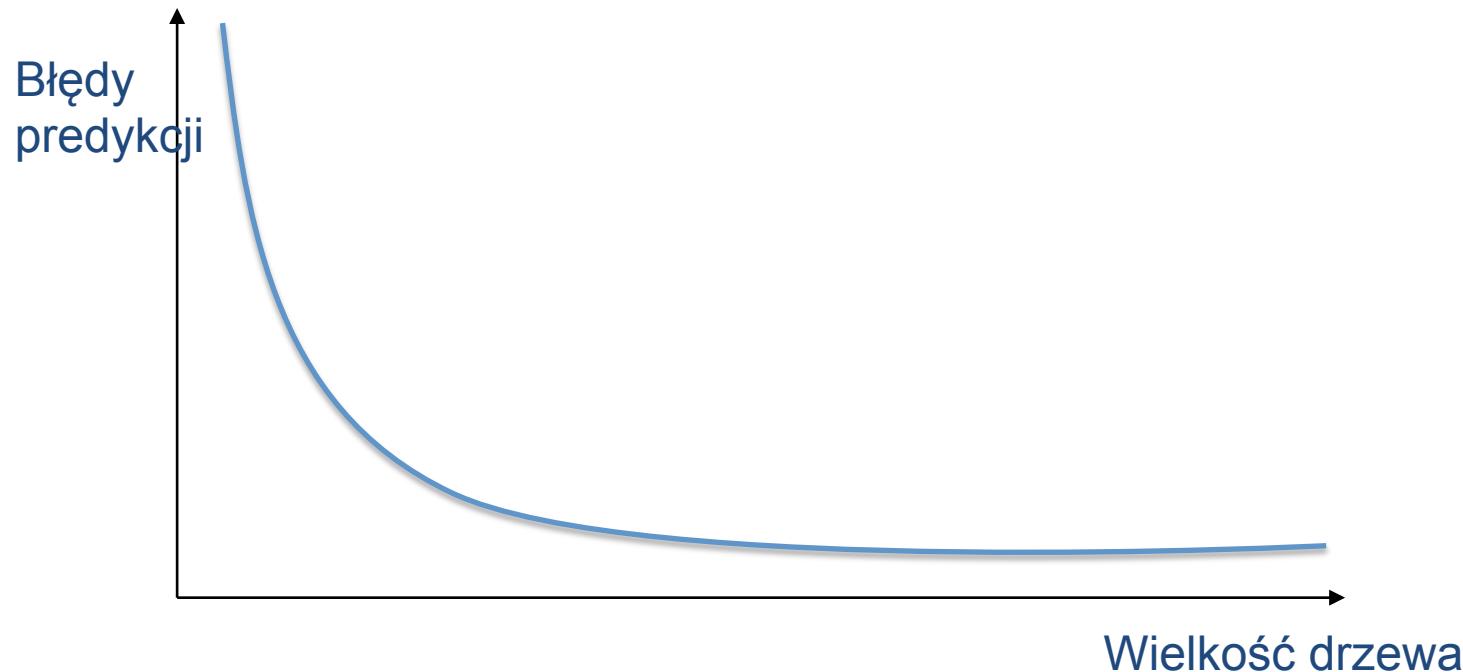
- CART – rozwiążanie cost-complexity
- T – (pod)drzewo do ew. uproszczenia z  $|T|$  węzłami ( $R_m$ )
- $N_m = \#\{x_i \text{ in } R_m\}$  oraz  $\hat{y}_{Rm} = \frac{1}{N_m} \sum_{x_i \in R_m} y_i$

$$Q_m(T) = \frac{1}{N_m} \sum_{x_i \in R_m} (y_i - \hat{y}_{Rm})^2$$

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha |T|$$

- $\alpha$  - współczynnik przetargu pomiędzy dobrym dopasowaniem a preferencją dla mniejszych drzew (odpowiednik regularizacji)
- Ocena na zbiorze walidacyjnym (CART wewn.ocena krzyżowa) tworzy się sekwencje coraz mniejszych drzew i poszukuje najm,

# Upraszczanie drzew regresyjnych

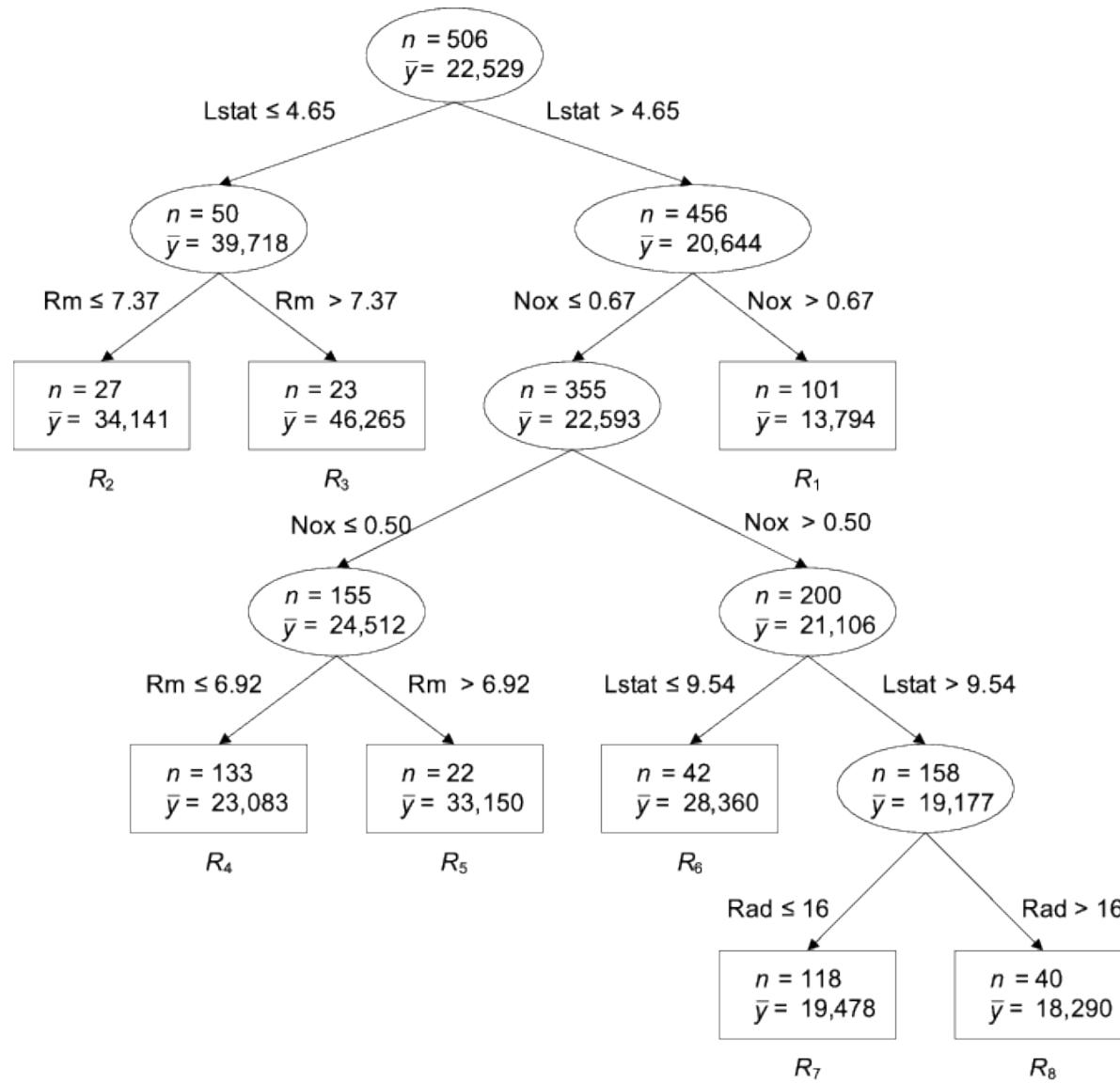


Wykres zmian błędu, w odróżnieniu od drzew klasyfikacyjnych bardziej płaski wykres;

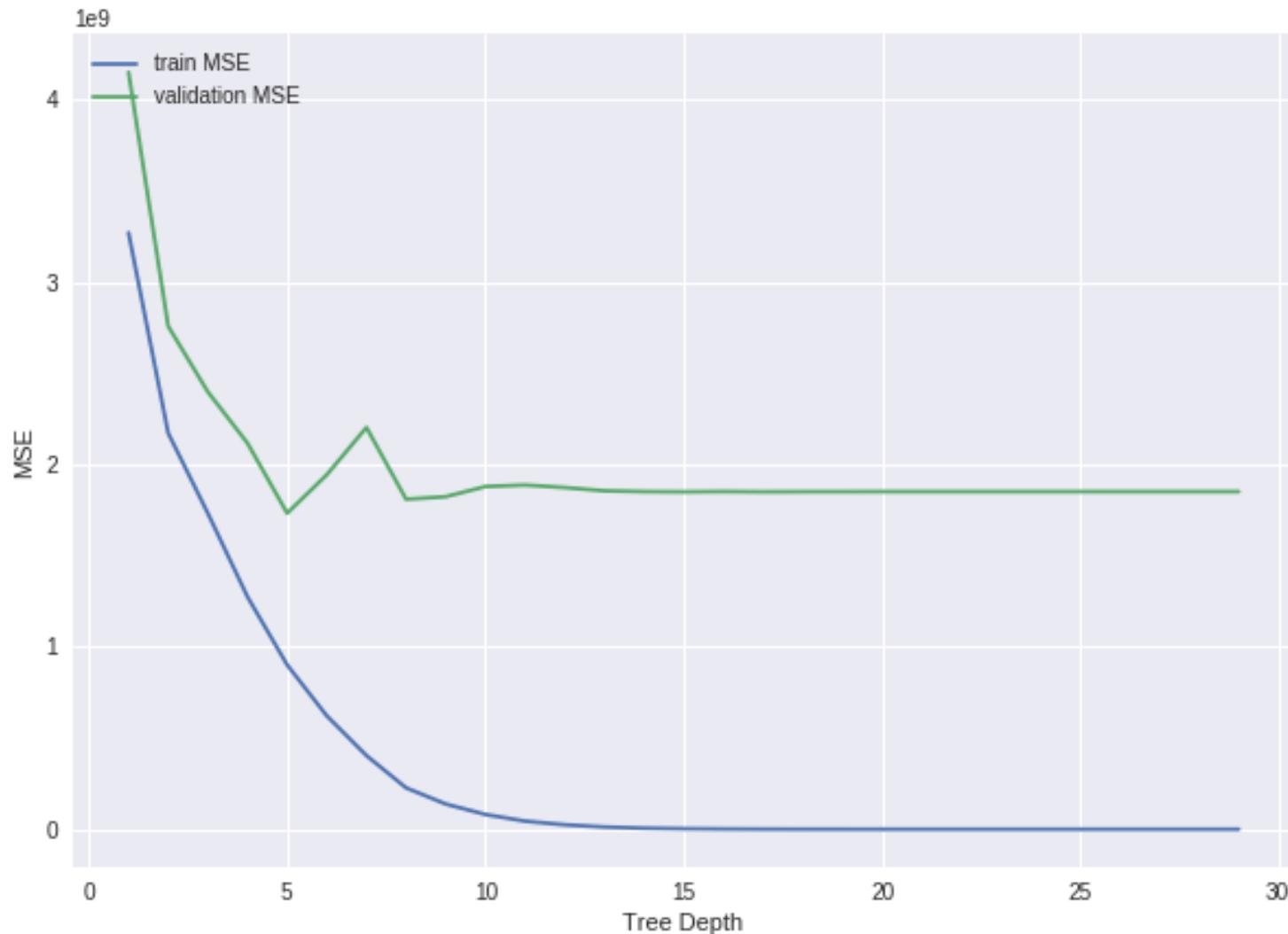
Przycinając drzewo – redukujemy liczbę liści przy jak najmniejszym przyroście błędu – błąd  $Q_m$  mniejszy od najmniejszego, powiększonego o jedno odchylenie standardowe w sekwencji drzew

Liczne inne propozycje, także z wykorzystaniem metody LASSO

# Przykład drzewa CART nauczonego z danych Boston housing (pruned)



# Boston housing – upraszczanie drzew



# Cechy drzew regresji

Różnice wobec klasycznych metod regresji:

- Możliwość bezpośredniego użycia różnorodnych zmiennych, w tym wielowartościowych jakościowych (bez specjalnego kodowania zerojedynkowego)
- Nie ma potrzeby standaryzacji, normalizacji zmiennych
- Rozkłady zmiennych nie muszą być normalne (typowa regresja liniowe)
- Modelowanie złożonych nieliniowych współzależności zmiennych (kiedy powierzchnia wielowymiarowej regresji jest bardzo złożona i nieregularna0
- Szybka predykcja
- Wspierania interpretacji struktury danych oraz oceny ważności zmiennych

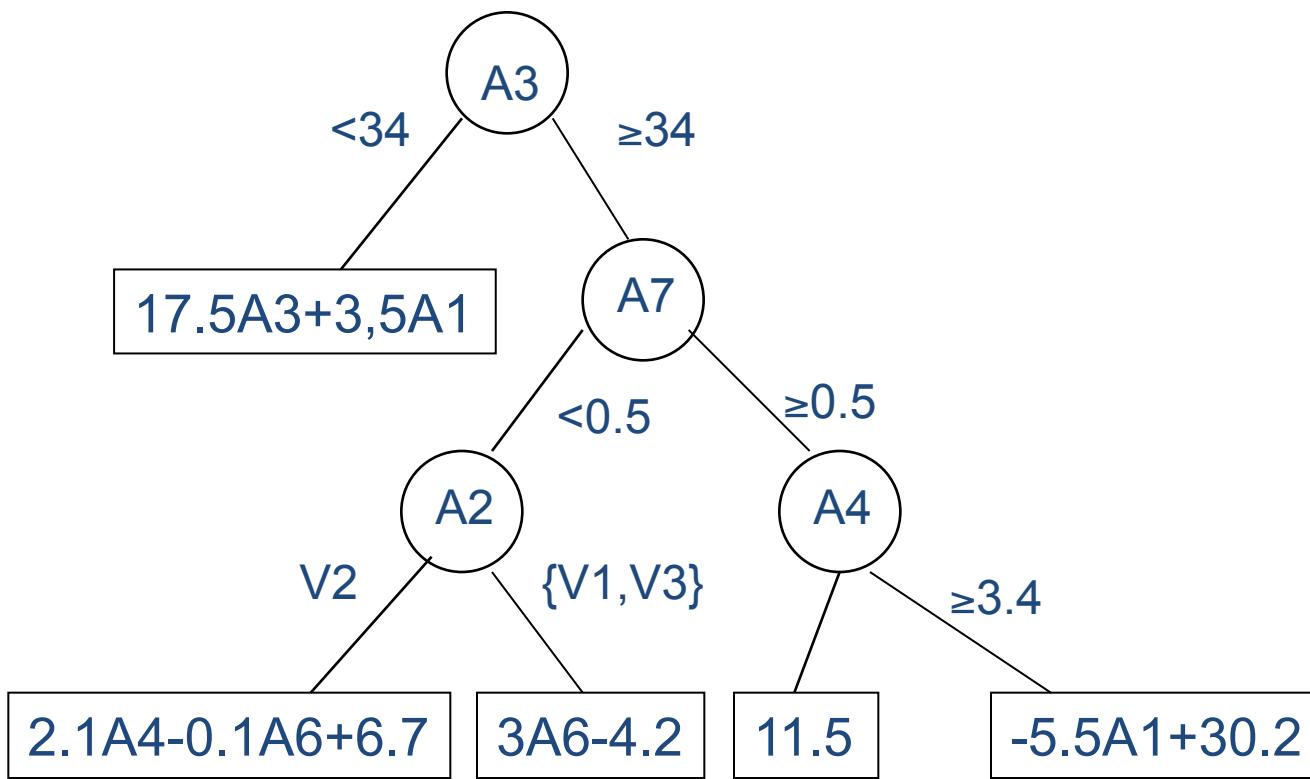
# Rozwój drzew

- Drzewa regresji wprowadzono w CART - (Breiman et al., 1984)
- Później model trees, od M5 (Quinlan, 1992)
- Wykorzystanie w zespołach klasyfikatorów (bagging, Random Forest) Breiman 1994, 2001
- Gradient boosted trees (Friedman 1999)
- Option trees (Buntine)
- Wersje przyrostowe dla strumieni danych (Ekomorovska)
- Multi-target trees – wiele wyjść  $y$  (S.Dzeroski et al.)
- Oraz wiele innych – patrz artykuł przeglądowy Wei-Yin Loh: Fifty Years of Classification and Regression Trees, International Statistical Review (2014)

# Rozwinięcie do model trees

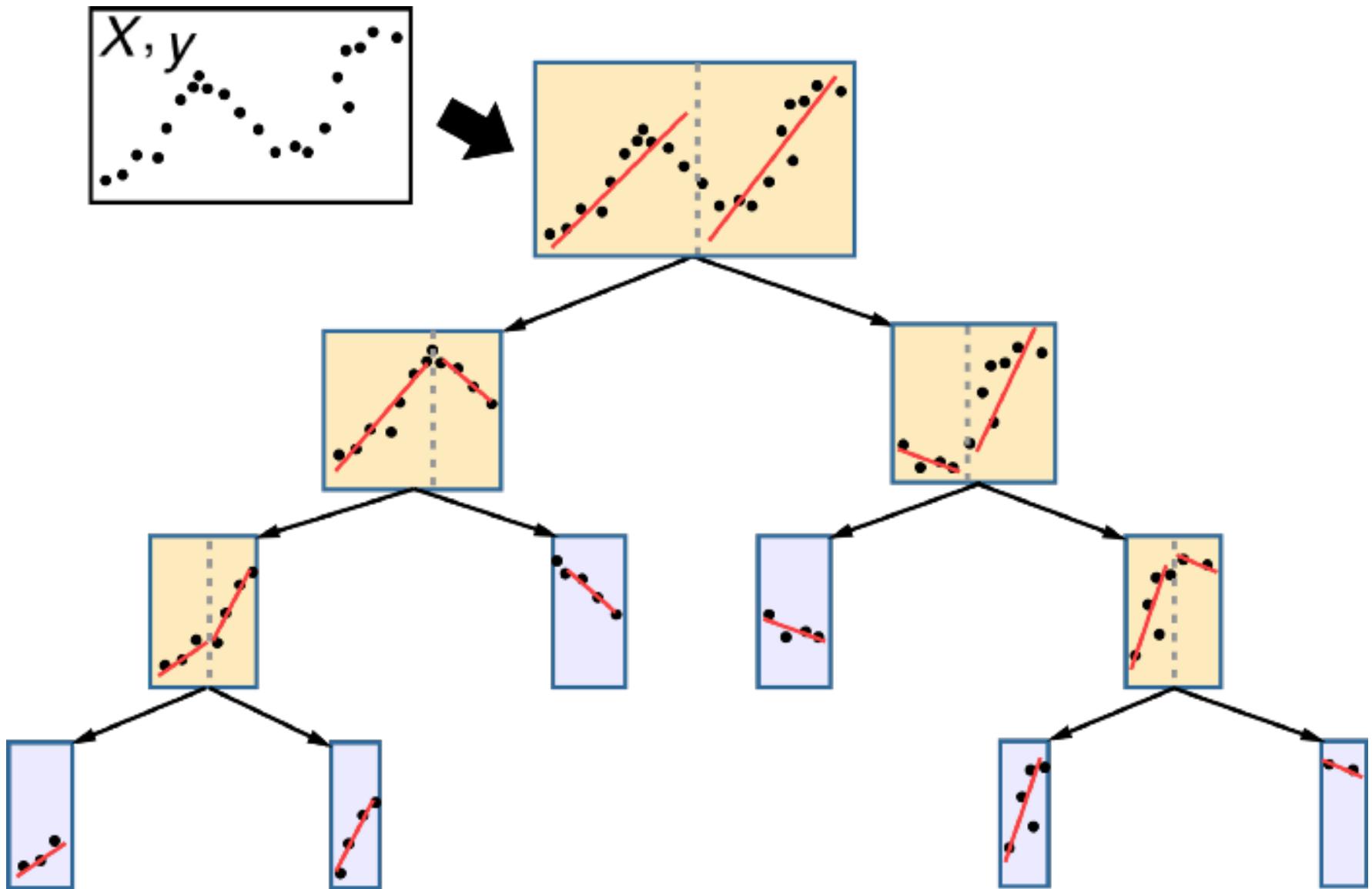
- W liściach wprowadzono odcinkami linowe funkcje regresji
- Efektywnie rozwiązane przez Quinlana w M5, rozwijane później, np. a stepwise linear regression model w węzłach (Kardic, Malerba,...)
- Torgo 1997 – zaproponował użycie regresji z funkcjami jądrowymi
- Oraz wiele innych – spójrz do literatury oraz wypróbuj oprogramowanie

# Model trees



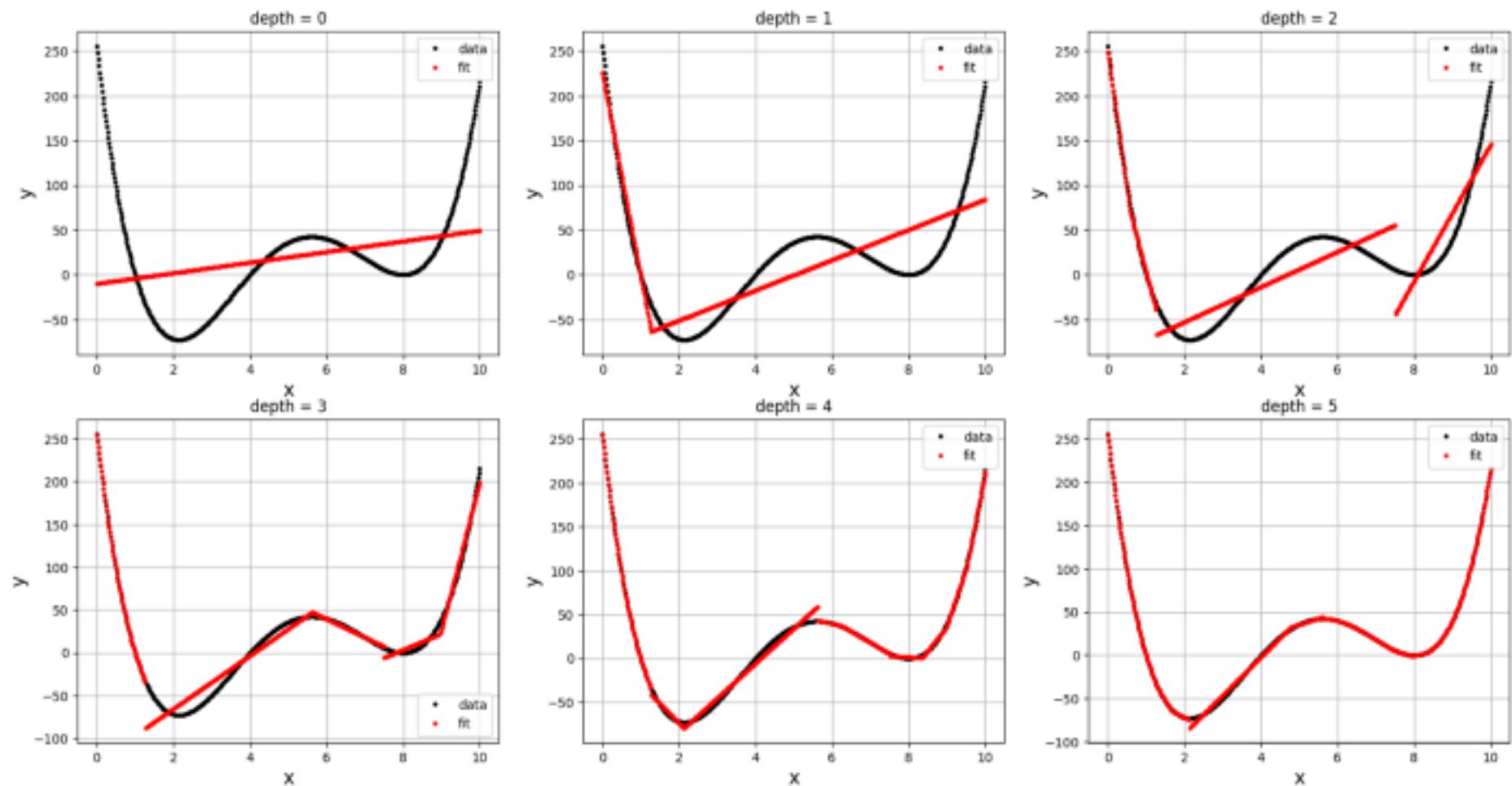
Jeśli w kostce obszaru związanej z liściem jest wystarczająca liczba przykładów uczących, to buduj lokalny model regresji liniowej

# Ilustracja działania model tree



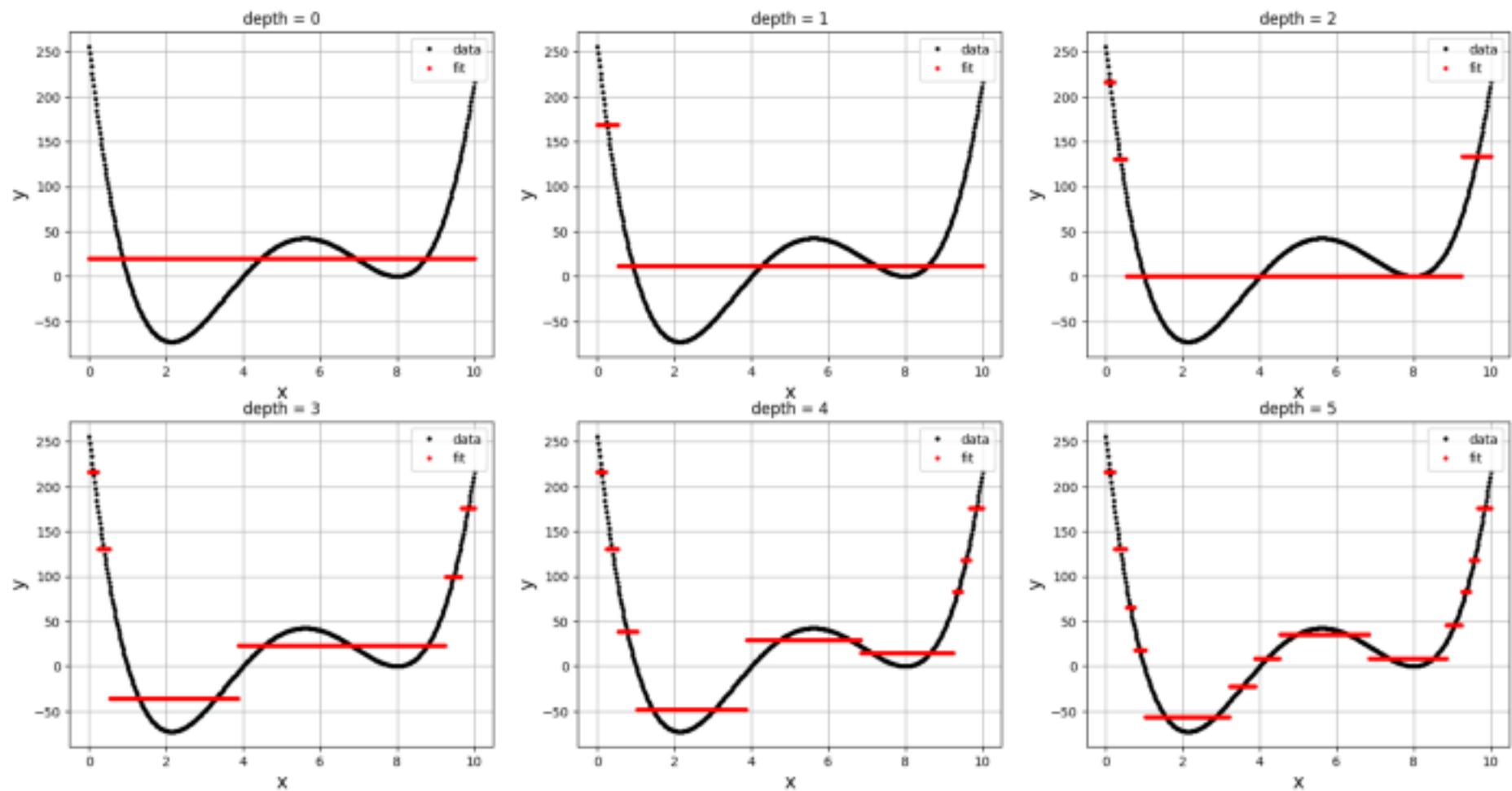
# Przybliżenie modelem linowym funkcji nieliniowej na różnych poziomach drzewa (model tree)

Model tree (model = linear\_regr) fits for different depths



# Przybliżenie modelem linowym funkcji nieliniowej na różnych poziomach drzewa regresji (średnia w kostce)

Model tree (model = mean\_regr) fits for different depths

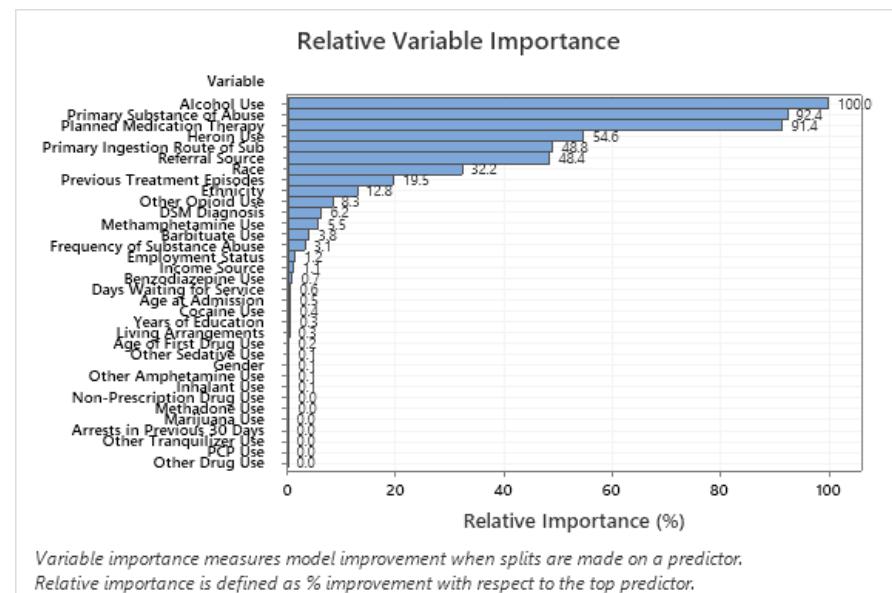


# Ilustracja przybliżenia funkcji

- Rysunki oraz szerszy opis intuicji budowy model trees z odcinkami liniowymi funkcjami regresji pochodzą z blogu pt. introduction to Model Trees from scratch | Anson Wong | na Towards Data Science – znajdź samodzielnie w internecie

# Interpretowalność drzewa i predykcji

- Możliwość interpretacji symbolicznej struktury drzewa – jeśli nie jest bardzo złożone
  - Podobnie jak dla drzew klasyfikacyjnych
- Ocena znaczenia najważniejszych cech dla predykcji
  - wykorzystanie propozycji tzw. feature importance zaproponowanej przez Breimana



# Dalsze pytania

- Złożone pytania w węzłach drzewa – lepsze przybliżenie skomplikowanych funkcji
- Lecz koszty obliczeniowe
- Otwarte pytania na dalszy wykład:
  - Jak wykorzystać drzewa regresji i model trees w zespołach klasyfikatorów
  - Inne funkcje straty  $L$  do optymalizacji i lepsze dopasowanie drzewa do danych, zwłaszcza w zespołach drzew regresyjnych (patrz np. gradient boosted trees)

# Odnośniki do literatury

- Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. Classification and Regression Tree. (1984).
- Rozdziały w książce Hastie, Tibshirani, Friedman: Elements of statistical learning (dostępna online pdf)
- Wei-Yin Loh: Fifty Years of Classification and Regression Trees, International Statistical Review (2014) – dobra lista cytowań do podstawowych prac oraz wielu rozszerzeń drzew
- Po polsku: M.Krzyśko, T.Górecki i inni, książka pt. Systemy uczące się
- E.Gatnar: Nieparametryczna metoda dyskryminacji i regresji
- J.Koronacki: Statystyczne systemu uczące się

Przykład analizy Boston house w Python

[https://quantdev.ssri.psu.edu/sites/qdev/files/07\\_Trees\\_2017\\_1125.html](https://quantdev.ssri.psu.edu/sites/qdev/files/07_Trees_2017_1125.html)

# **Pytanie i komentarze?**

Dalszy kontakt:

[jerzy.stefanowski@cs.put.poznan.pl](mailto:jerzy.stefanowski@cs.put.poznan.pl)

<http://www.cs.put.poznan.pl/jstefanowski/>



**Fundusze  
Europejskie**  
Polska Cyfrowa



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz  
Rozwoju Regionalnego



# Uczenie klasyfikatorów z niebalansowanych danych

## Wykład 7

Jerzy Stefanowski  
Instytut Informatyki PP  
2021

Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI-TECH) projekt finansowany z środków Programu Operacyjnego Polska Cyfrowa POPC.03.02.00-00-0001/20



Fundusze  
Europejskie  
Polska Cyfrowa



Rzeczpospolita  
Polska

Unia Europejska  
Europejski Fundusz  
Rozwoju Regionalnego



# Plan wykładu

---

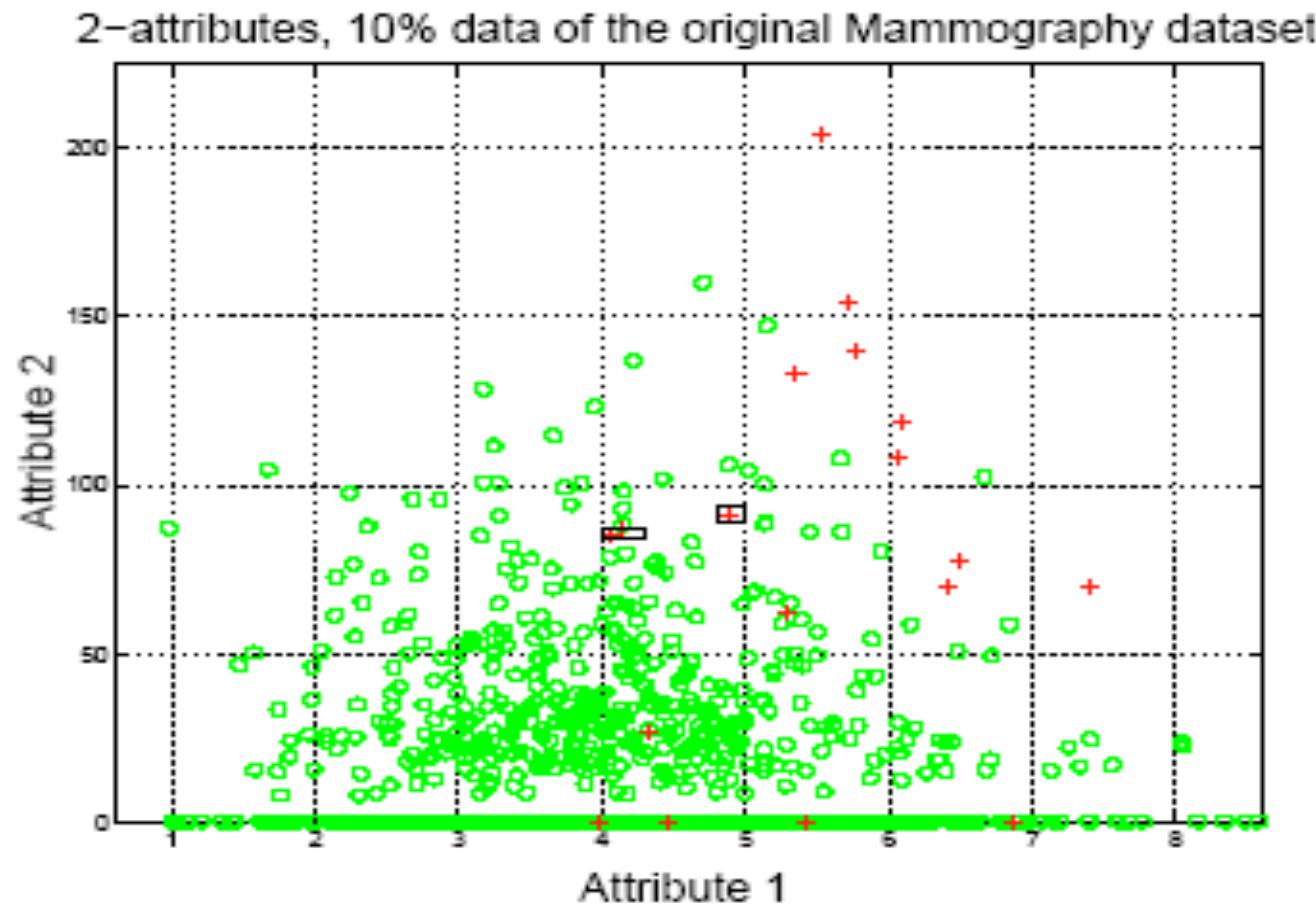
1. Niebalansowanie liczności klas (przykłady; miary oceny - wstęp)
2. Czynniki trudności i charakterystyka danych
3. Taksonomia podstawowych metod
4. Przetwarzanie wstępne
  - Under-, over- sampling, SMOTE
  - Metody hybrydowe
5. Wybrane modyfikacje algorytmów
  1. Cost sensitive learning
  2. Zespoły klasyfikatorów (RBB i inne generalizacje) = częściowo przesunięte na późniejszy wykład
6. Ocena klasyfikatorów
7. Inne zagadnienia i wyzwania

# Uczenie się klasyfikatorów z niebalansowanych danych

---

- Zadajmy pytanie o rozkład przykładów w klasach w zbiorze uczącym
- Standardowe założenie:
  - Dane są zrównoważone /zbalansowane - rozkłady liczności przykładów w klasach względnie podobne
  - Czy takie założenie jest realistyczne?
    - Lecz: „Czy medyczne dane o diagnozie rzadkich chorób są zbalansowane”
    - Rozważ, np., dane N.Chawla nt. badań mammograficznych - 11183 przykładów, 6 atrybutów, klasa mniejszościowa 2.3%
- Inne przykłady praktyczne

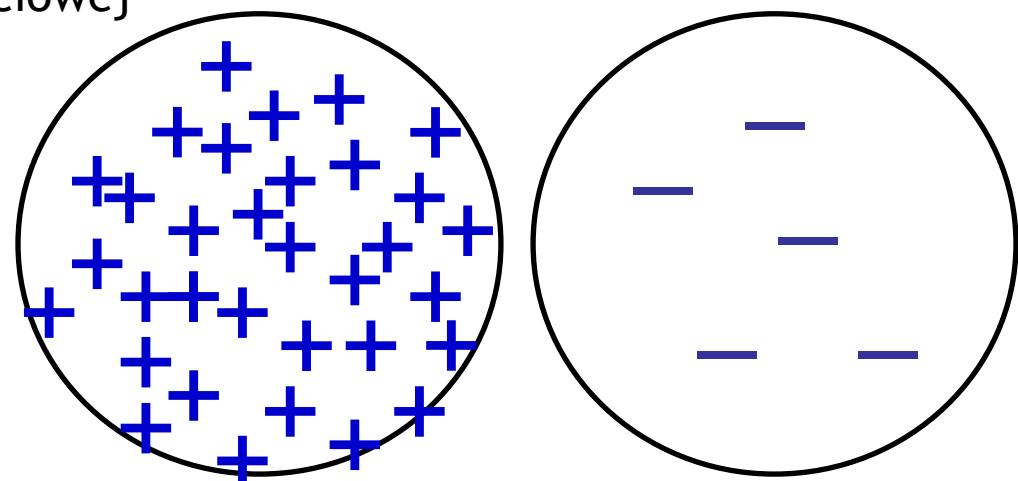
# Przykład danych medycznych Chawla et al. SMOTE 2002



Dane – 11183 przykładów, 6 atrybutów, klasa mniejszościowa 2.3%

# Niebalansowanie rozkładu przykładów w klasach

- Dane są niebalansowane (imbalanced) jeśli klasy nie są w przybliżeniu równo liczne
  - Klasa mniejszościowa (minority class) zawiera wyraźnie mniej przykładów niż inne klasy
- Przykłady z klasy mniejszościowej są często najważniejsze i ich poprawne rozpoznawanie jest głównym celem, np.:
  - Rozpoznawanie rzadkiej, niebezpiecznej choroby
- CLASS IMBALANCE → powoduje trudności w fazie uczenia i obniża zdolność predykcyjną
  - Niektóre klasyfikatory pomimo wysokiej globalnej trafności nie rozpoznają kl. mniejszościowej



„Niebalansowanie to nie to samo co uczenie z kosztami”

# Przykłady rzeczywistych problemów

---

- Niebalansowanie klas naturalne w :
  - Analiza danych medycznych - leczenie i diagnostyka
  - Monitorowanie uszkodzeń urządzeń technicznych
  - Odróżnianie trzęsień ziemi od prób nuklearnych
  - Filtrowanie wiadomości
  - Marketing bezpośredni
  - Tzw. problem ucieczki klientów (kompanie telekomunikacyjne)
  - .....
- Przegląd innych problemów i zastosowań
  - Japkowicz N., Learning from imbalanced data. AAAI Conf., 2000.
  - Weiss G.M., Mining with rarity: a unifying framework. ACM Newsletter, 2004.
  - Chawla N., Data mining for imbalanced datasets: an overview. In The Data mining and knowledge discovery handbook, Springer 2005.
  - He H, Garcia, Mining imbalanced data. IEEE Trans. Data and Knowledge 2009.
  - Książki = dwie monografie anglojęzyczne

# Globalne niezbalansowanie (Imbalance Ratio)

---

- Naturalnie rozważany problem binarny - klasa mniejszościowa  
-> specjalne znaczenie w zastosowaniu
  - Przykład: diagnoza rzadkiej, lecz niebezpiecznej choroby; błędne nierozpoznanie chorego pacjenta ważniejsze niż sytuacja odwrotna
  - Problemy wieloklasowe – rzadziej badane
- Prosta charakterystyka – stopień niezbalansowania
  - $N_W$  – liczba przykładów z klasy większościowej
  - $N_M$  – liczba przykładów z klasy mniejszościowej
- Różne definicje w literaturze
  - $IR = N_W / N_M$  (ile razy większa klasa W)
  - $IR [\%]$  – jaki procent  $N_M$  w całości  $N_M + N_W$
- Brak wyraźnej granicy IR, kiedy zbiór jest mniejszościowy
  - Może być 15%, 10%, 5%, 1%, itd

# Przykład charakterystyk benchmark data

Dataset	No of examples	Imbalance ratio [%]	No of attributes (numeric)	Minority class name
breast-w	699	34.47	9(9)	malignant
abdominal-pain	723	27.94	13 (0)	positive
acl	140	28.57	6 (4)	1
new-thyroid	215	16.28	5 (5)	hyper
vehicle	846	23.52	18 (18)	van
nursery	12960	2.53	8(0)	very-recom
satimage	4435	9.35	36(36)	4
car	1728	3.99	6 (0)	good
scrotal-pain	201	29.35	13 (0)	positive
credit-g	1000	30	20 (7)	bad
ecoli	336	10.42	7 (7)	imU
hepatitis	155	20.65	19 (6)	die
ionosphere	351	35.89	34 (34)	bad
haberman	306	26.47	3 (3)	died
cmc	1473	22.61	9 (2)	l-term
breast-cancer	286	29.72	9 (0)	rec-events
cleveland	303	11.55	13 (6)	positive
glass	214	7.94	9 (9)	v-float
hsv	122	11.48	11 (9)	4.0
abalone	4177	8.02	8 (7)	0-4 16-29
postoperative	90	26.66	8 (0)	S
seismic-bumps	2584	6.57	18(14)	1
solar-flare	1066	4.03	12 (0)	F
transfusion	748	23.8	4 (4)	yes
yeast	1484	3.44	8 (8)	ME2
balance-scale	625	7.84	4(4)	B

Za artykuł: K.Napierała, J.Stefanowski:

Types of minority class examples and their influence on learning classifiers. JIIS (2016)

# Jak oceniać klasyfikatory dla niezbalansowanych danych?

- Standardowa trafność bezużyteczna
  - Wyszukiwanie informacji (klasa mniejszościowa ~ 1%)  
→ ogólna trafność klasyfikowania ~100%, lecz źle rozpoznawana wybrana klasa
- Miary powiązane z klasą mniejszościową
  - Analiza binarnej macierzy pomyłek confusion matrix
  - Sensitivity i specificity → G-mean
  - ROC curve analysis (AUC)
  - Cost-Precision curves

		Predicted class	
		Yes	No
Actual class	Yes	TP: True positive	FN: False negative
	No	FP: False positive	TN: True negative

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

$$\text{Specificity} = \frac{TN}{TN + FP}$$

$$G\text{-mean} = \sqrt{\text{Sensitivity} * \text{Specificity}}$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad \text{Recall} = \frac{TP}{TP + FN}$$

$$F\text{-measure} = \frac{(1 + \beta)^2 * \text{Precision} * \text{Recall}}{\beta^2 * \text{Recall} + \text{Precision}}$$

Więcej informacji później

# Złożone miary

---

Najpopularniejsze:

$$G-mean = \sqrt{sensitivity \cdot specificity}$$

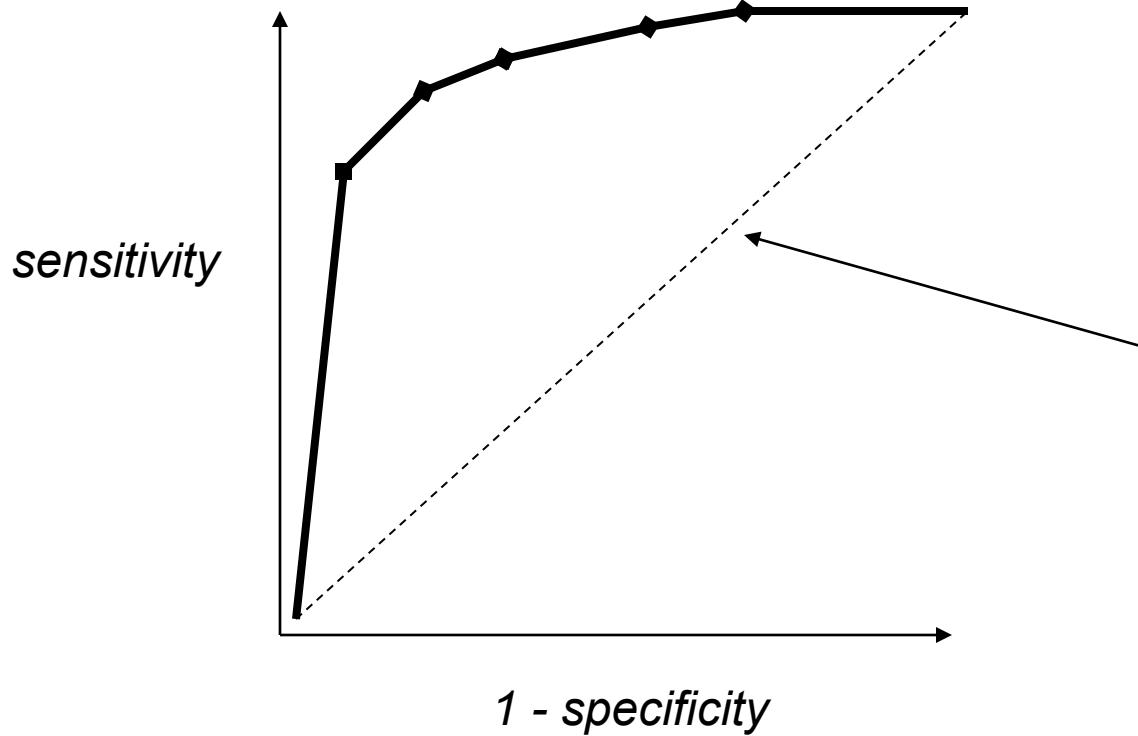
$$F_{\beta} = \frac{(1 + \beta^2) \cdot Recall \cdot Precision}{\beta^2 \cdot Recall \cdot Precision}$$

Matthews correlation coefficient, MCC

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

(MCC) expresses a correlation between the actual and predicted classification and returns a value between -1 (total disagreement) and +1 (perfect agreement); 0 classifiers performs randomly

# Krzywa ROC oraz AUC



Im krzywa bardziej wygięta ku górnemu lewemu narożnikowi, tym lepszy klasyfikator .

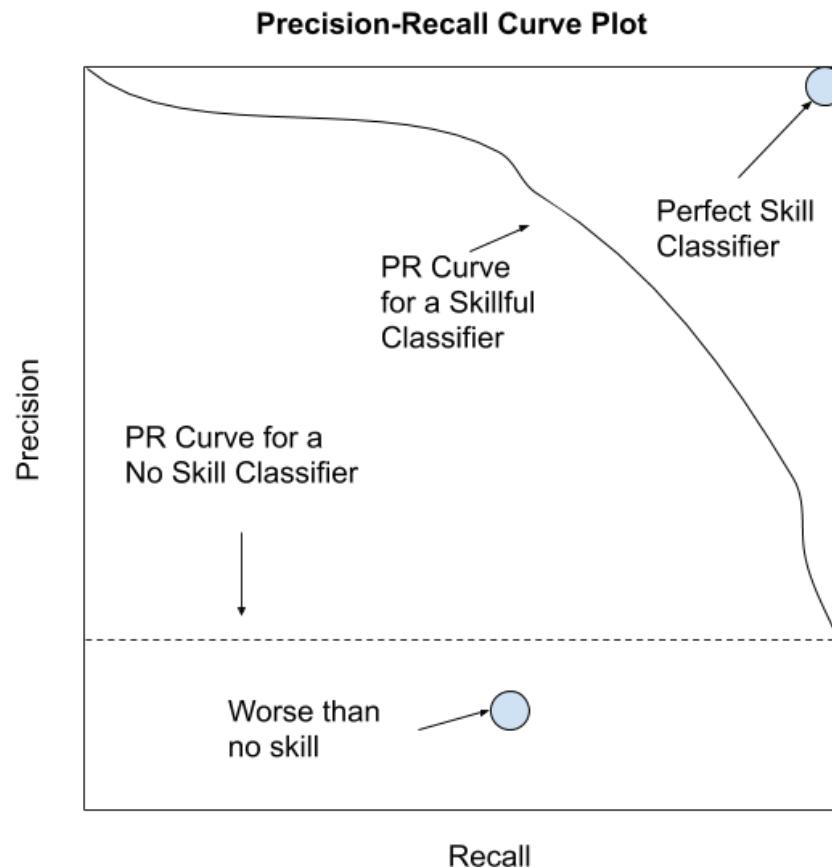
Przekątna odpowiada losowemu „zgadywaniu”. Im bliżej niej, tym gorszy klasyfikator

Można porównywać działanie kilku klasyfikatorów.  
Miary oceny np. AUC – pole pod krzywą,,. Powinno być więcej niż 0.5

# Precision Recall Curve

Pomimo dobrego zachowania AUC, może być zbyt optymistyczna dla silnego niezbalansowania /b. mała liczba przykładów mniejszościowych

Alternatywa - analiza krzywej precision recall - mocniej skupia się na predykcji klasyfikatora dla klasy mniejszościowej



# Standardowe klasyfikatory?

---

- Standardowe algorytmy uczące
  - zakłada się w przybliżeniu zrównoważenie klas
- Typowe strategie przeszukiwania optymalizują globalne kryteria (błąd, miary entropii, itp.)
  - Przykłady uczące są liczniej reprezentowane przy wyborze hipotez
- Metody redukcji (ang. pruning) faworyzują przykłady większościowe
- Strategie klasyfikacyjne ukierunkowane na klasy większościowe

Konkluzja – nie są wystarczająco dobrze przystosowane do radzenia sobie z niezbalansowaniem

# Słaba skuteczność klasyfikatorów

Table 1. Characteristics of evaluated data sets ( $N$  – the number of examples,  $N_A$  – the number of attributes,  $C$  – the minority class,  $N_C$  – the number of examples in the minority class,  $N_O$  – the number of examples in the majority class,  $R_C = N_C/N$  – the ratio of examples in the minority class)

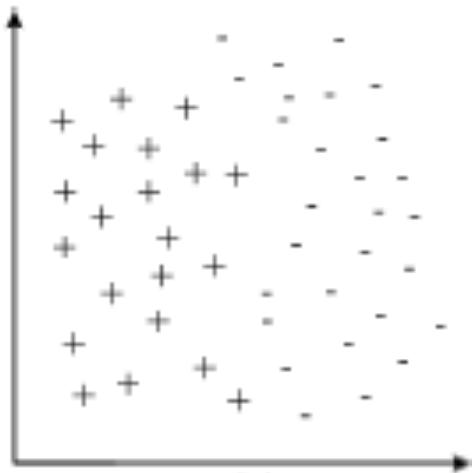
Data set	$N$	$N_A$		$C$	$N_C$	$N_O$	$R_C$
Acl	140	6	with knee injury	40	100	0.29	
Breast cancer	286	9	recurrence-events	85	201	0.30	
Bupa	345	6		145	200	0.42	
Cleveland	303	13		positive	35	268	0.12
Ecoli	336	7		imU	35	301	0.10
Haberman	306	3		died	81	225	0.26
Hepatitis	155	19		die	32	123	0.21
New-thyroid	215	5		hyper	35	180	0.16
Pima	768	8		positive	268	500	0.35

## Sensitivity klasy mniejszościowej

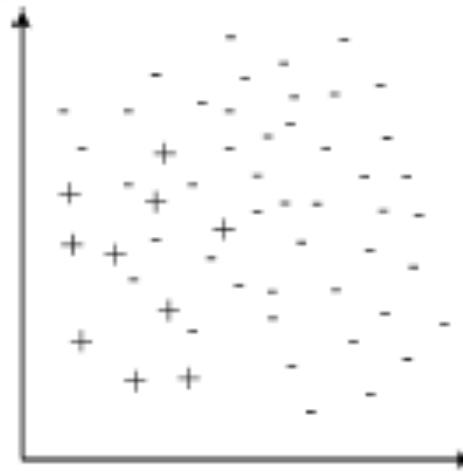
Data	Modlem rules	C4.5 trees
Acl	0.805	0.855
Breast	0.319	0.387
Bupa	0.520	0.491
Cleveland	0.085	0.237
Ecoli	0.400	0.580
Haberman	0.240	0.410
Hepatitis	0.383	0.432
New-thyr.	0.812	0.922
Pima	0.485	0.601

Lepsze rozpoznawania  
tylko Acl i New Thyroid

# Na czym polega trudność?



Łatwiejszy problem



Trudniejszy

## Źródła trudności:

- Zbyt mało przykładów z klasy mniejszościowej (IR),
- „Zaburzenia” brzegu klas,
- Segmentacja klasy
- ...

Przeglądowe prace:

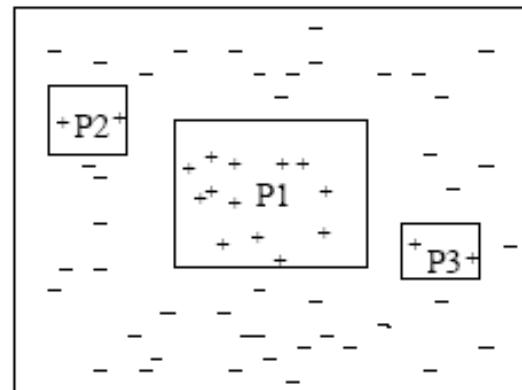
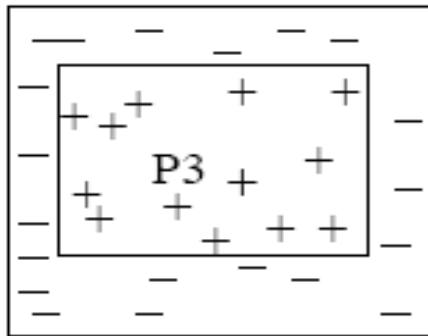
- Japkowicz N., Learning from imbalanced data. AAAI Conf., 2000.
- Weiss G.M., Mining with rarity: a unifying framework. ACM Newsletter, 2004.

Klasa większa „nakłada” się na mniejszościowe:

- Niejednoznaczne przykłady brzegowe
- Outliers and rare cases
- Wpływ „szumu” (noisy examples)

# Czy zawsze „niezbalansowanie” jest trudnością?

- Przeanalizuj studia eksperymentalne N.Japkowicz lub przeglądy G.Weiss – nie wszystkie niezbalansowane dane są trudne dla standardowych algorytmów.
- Japkowicz „The minority class contains small sub-clusters of interesting examples surrounded by other examples” (pełnią rolę tzw, small „disjuncts”, które częściej prowadzą do błędnych decyzji - Holte)

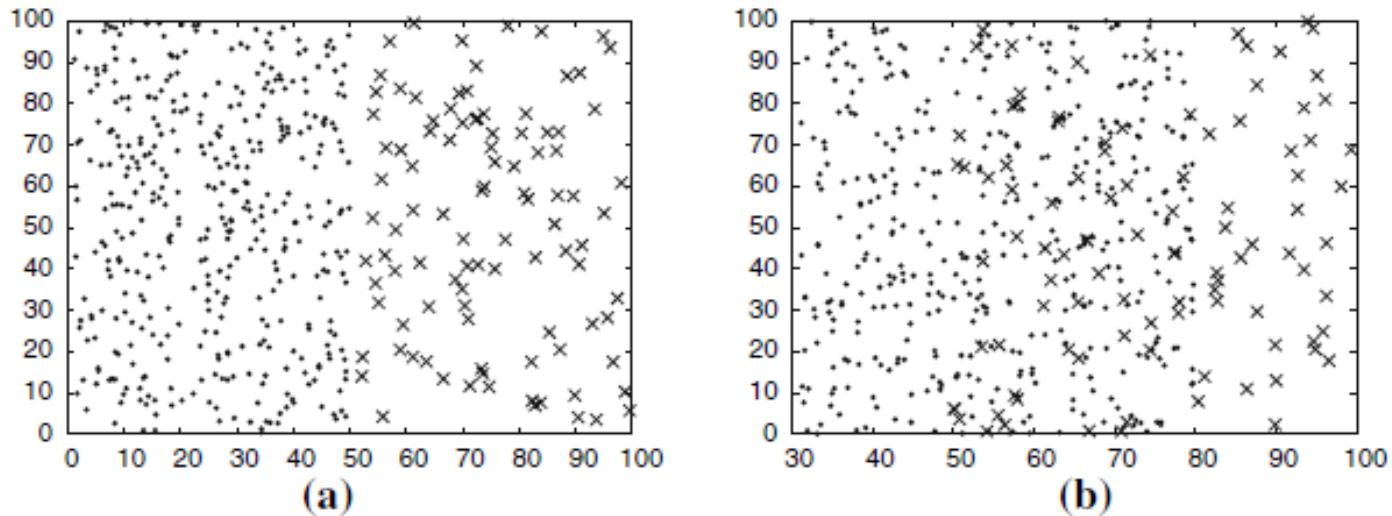


Niektóre prace eksperymentalne z dyskusją źródeł trudności, e.g:

- T. Jo, N. Japkowicz. Class imbalances versus small disjuncts. SIGKDD Explorations 6:1 (2004) 40-49
- V. García, R.A. Mollineda, J.S. Sánchez. On the k-NN performance in a challenging scenario of imbalance and overlapping. Pattern Anal Applic (2008) 11: 269-280
- Stefanowski J et al. Learning from imbalanced data in presence of noisy and borderline examples. RSCTC 2010.

# Ekspertymenty Garcia et al. ze strefami brzegowymi

## Nakładanie się rozkładów klas (ang.overlapping)

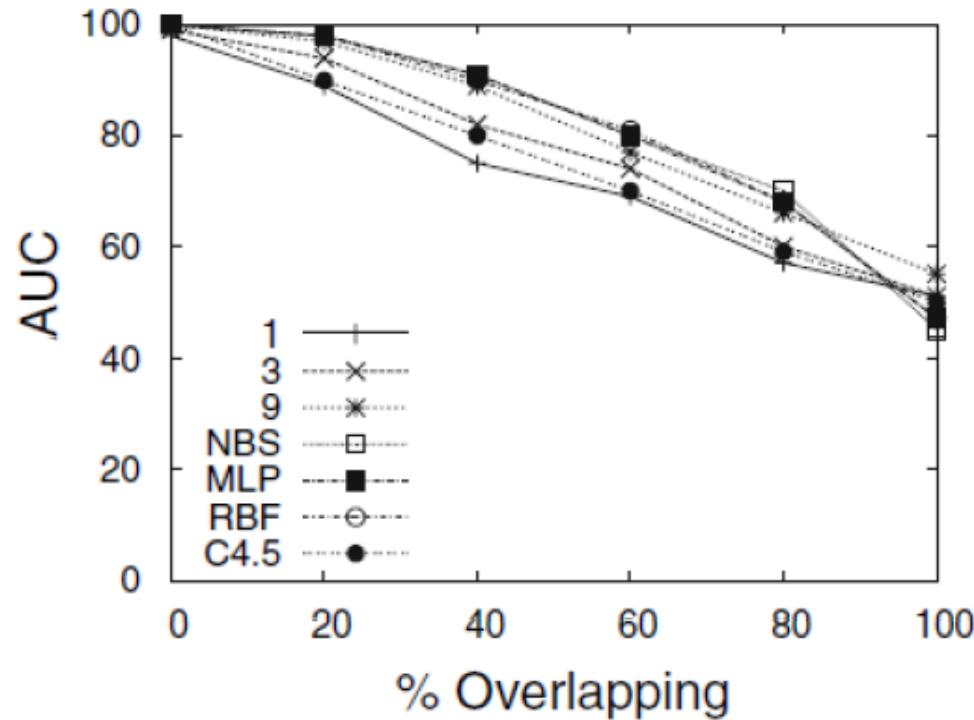


Dwa różne poziomy nakładania się 0% i 60%

**Źródło:** V García, J Sánchez, R Mollineda: An empirical study of the behavior of classifiers on imbalanced and overlapped data sets. 2007.

# Ekspertymenty Garcia et al. ze strefami brzegowymi

## Niektóre z wyników

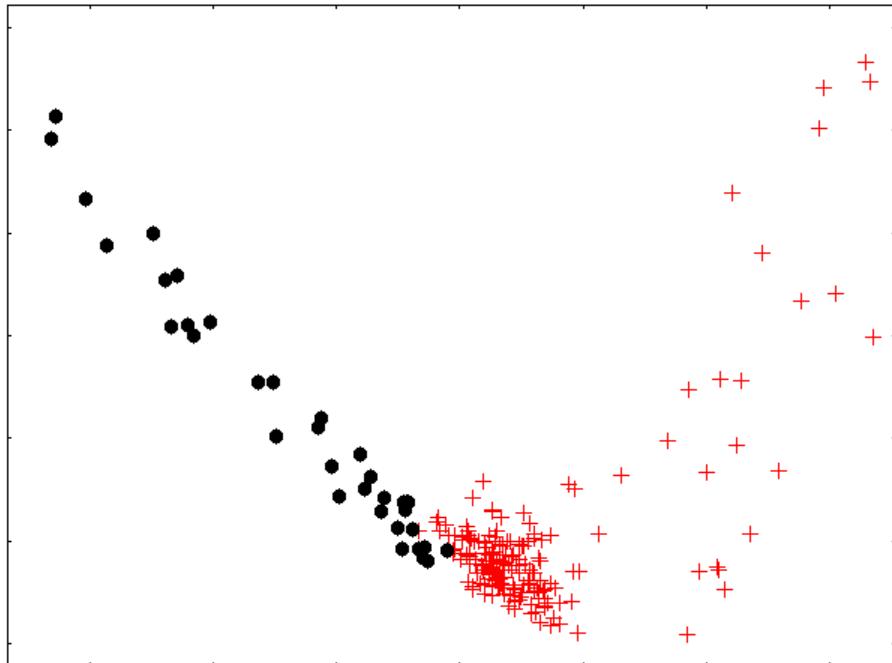


Skuteczność różnych klasyfikatorów – wzrost niejednoznaczności strefy brzegowej silnie obniża AUC niż wzrost niezrównoważenia

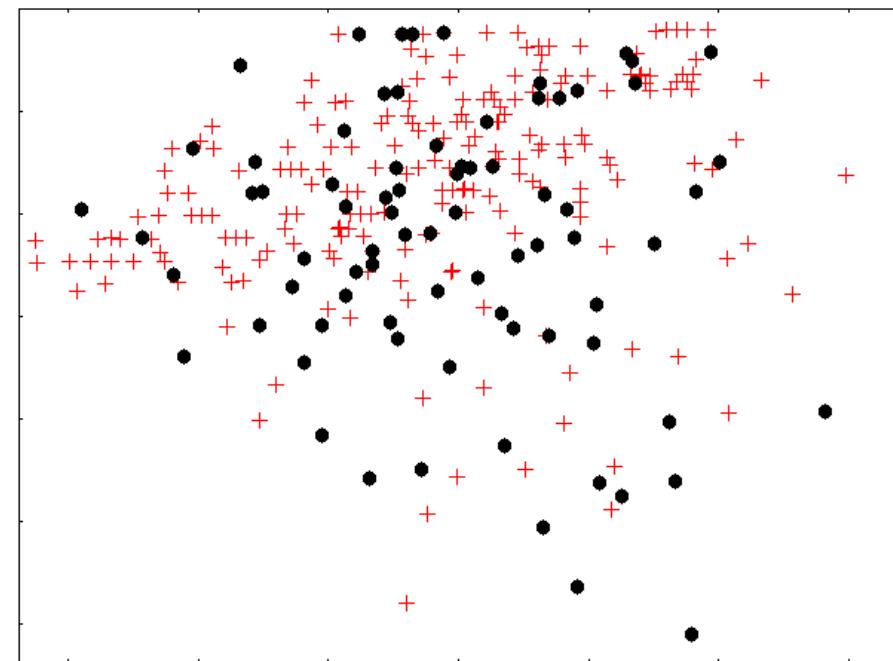
W dalszych eksperymentach zauważony wpływ lokalnej gęstości przykładów!

# Co z rzeczywistymi danymi?

Wizualizacja 2 pierwszych składowych w metodzie MDS (PCA)  
eksperyment własny autora i K.Napierały



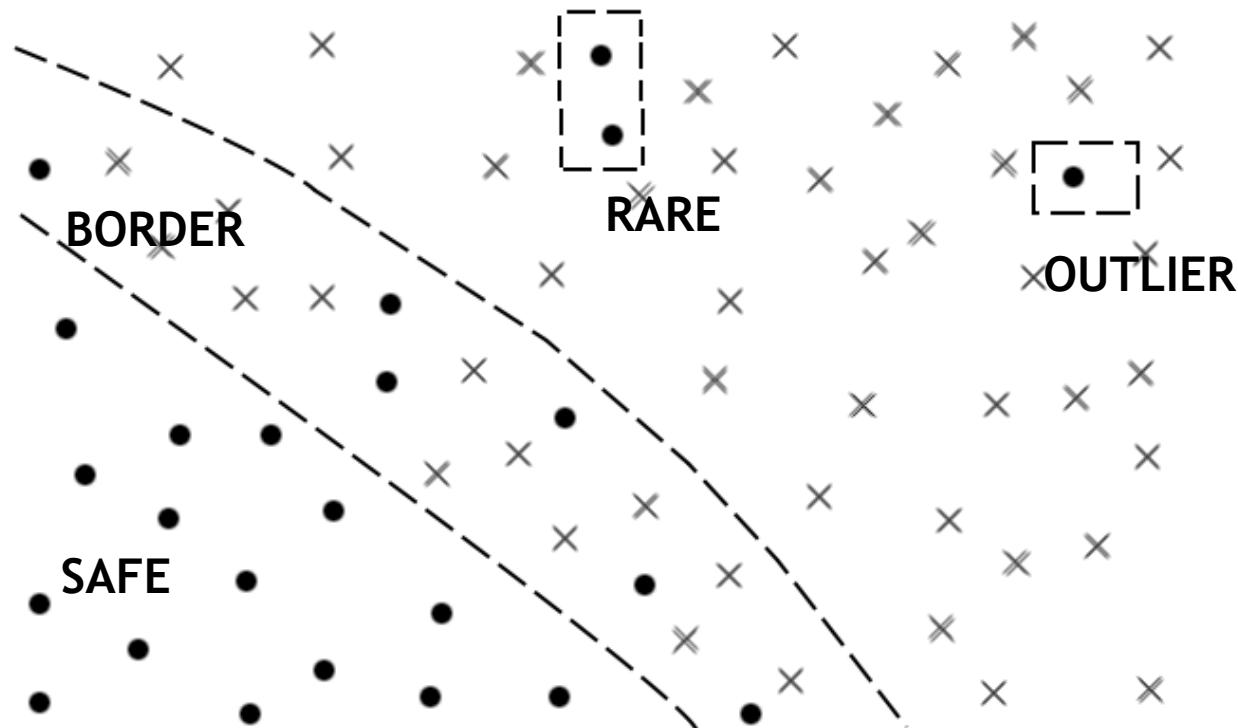
Thyroid  
215 ob./ 35 mniejsz.  
Prosty dla klasyfikacji



Haberman  
306 ob. / 81 mniejsz.  
trudny

# Data Difficulty Factors → Różna lokalna charakterystyka rozkładu (typów) przykładów

Rozróżniamy 4-y typy przykładów:



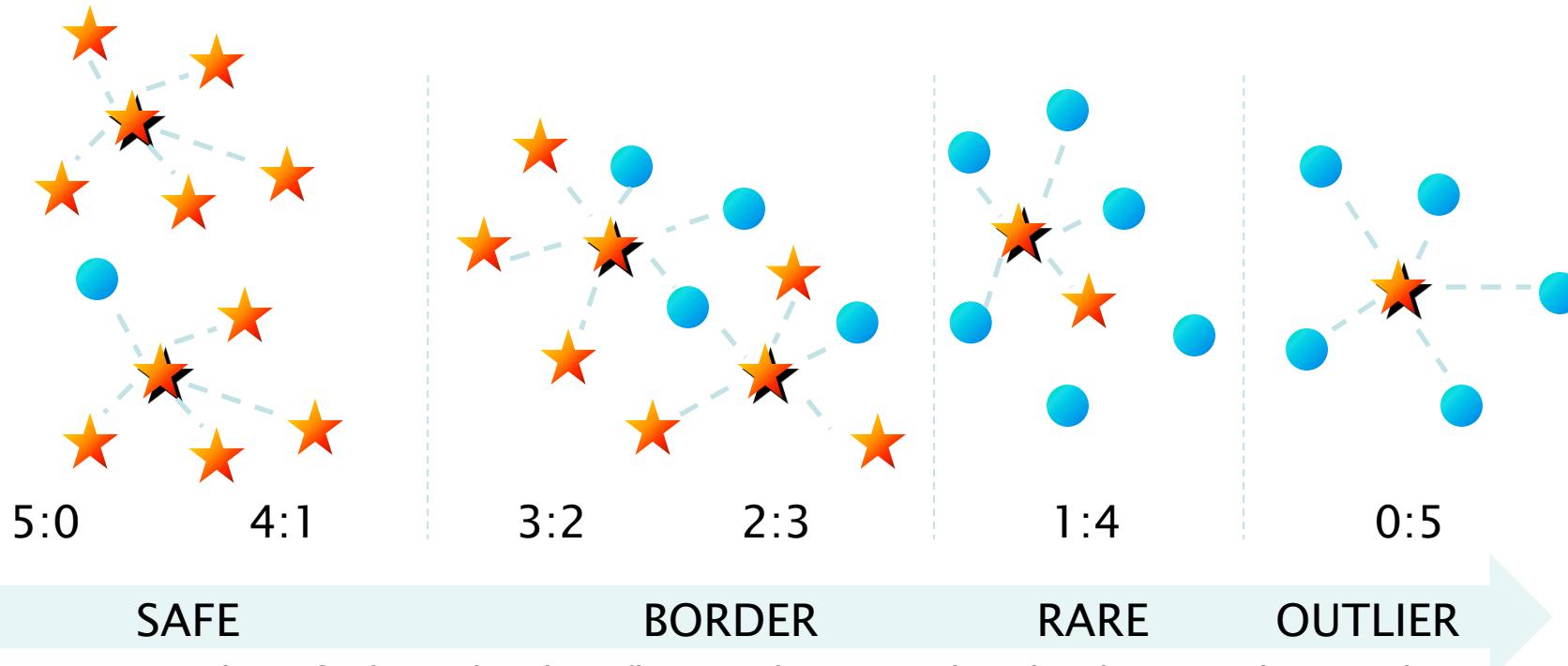
Więcej → K.Napierała, J. Stefanowski: The influence of minority class distribution on learning from imbalance data. HAIS, 2012.

# Podejście do identyfikacji typów przykładów

Analizuj rozkład etykiet wśród najbliższych sąsiadów x

- K-NN ( $k=5, 7, \dots$ ) - HVDM distance
- Kernel functions (parametr rozproszenia)

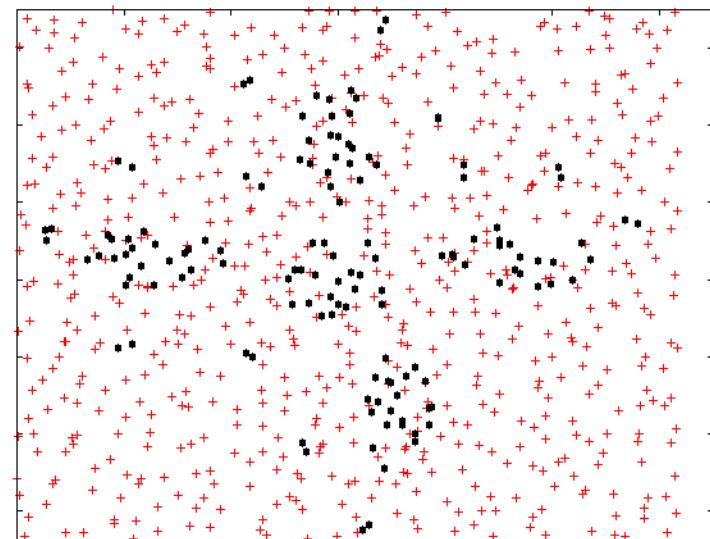
Określ typ przykładu x



Details → K.Napierała, J. Stefanowski: The influence of minority class distribution on learning from imbalance data. HAIS, 2012.

# Sprawdzenie na sztucznych rozkładach danych

Imbalance Ratio	Sub- concepts	Dataset Description			Identified Labels			
		Border [%]	Rare [%]	Outlier [%]	Safe [%]	Border [%]	Rare [%]	Outlier [%]
1:5	1	60	20	0	17.04	60.74	21.48	0.74
1:5	3	60	20	0	18.52	57.78	23.70	0.00
1:5	5	60	20	0	17.78	64.44	17.78	0.00
1:5	5	0	0	10	64.44	25.93	0.00	9.63
1:7	5	0	0	10	54.00	36.00	0.00	10.00
1:9	5	0	0	10	52.00	36.00	2.00	10.00



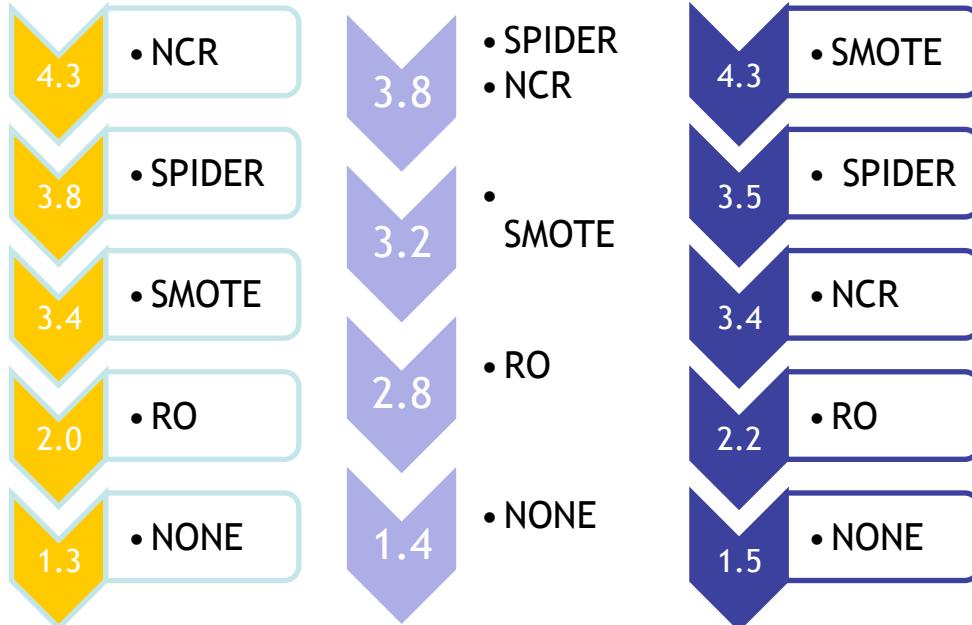
# Labeling Minority Examples → UCI Data sets

Dataset	S [%]	B [%]	R [%]	O [%]
abdominal-pain	59.90	22.28	8.90	7.92
acl	67.50	30.00	0.00	2.50
new-thyroid	68.57	31.43	0.00	0.00
vehicle	74.37	24.62	0.00	1.01
car	47.83	39.13	8.70	4.35
scrotal-pain	38.98	45.76	10.17	5.08
ionosphere	44.44	30.95	11.90	12.70
credit-g	9.33	63.67	10.33	16.67
ecoli	28.57	54.29	2.86	14.29
hepatitis	15.63	62.50	6.25	15.63
haberman	4.94	61.73	18.52	14.81
breast-cancer	24.71	25.88	32.94	16.47
cmc	17.72	44.44	18.32	19.52
cleveland	0.00	31.43	17.14	51.43
glass	0.00	35.29	35.29	29.41
hsv	0.00	0.00	28.57	71.43
abalone	8.36	20.60	20.60	50.45
postoperative	0.00	41.67	29.17	29.17
solar-flare	0.00	48.84	11.63	39.53
transfusion	18.54	47.19	11.24	23.03
yeast	5.88	47.06	7.84	39.22

- Very unsafe distribution of the minority examples
  - cleveland → 51% outliers, no safe ones
  - solar flare, balance scale
- Majority class → quite safe
  - yeast → 98,5% S
  - ecoli → 91,7% S
- Experiences with k (7,9,...) or kernels → similar categorizations of data
- Unsafe data → deteriorate classifier performance and difficult for improvement

# Zróżnicowane działanie metod w zależności od kategorii przykładów mniejszościowych w danych

## Friedman Tests



Resultaty sensitivity PART rules

1NN, J48: podobne rankingi

RBF: RO wyżej w rankingu

B

R

O

# Python – co robić

Imbalanced-learn Toolbox (Lemaitre et al 2017)  
Under- (11), over-sampling (7), some ensembles (4)

## imbalanced-learn 0.6.2

`pip install imbalanced-learn`



Latest version

Released: Feb 16, 2020

Toolbox for imbalanced dataset in machine learning.

### Navigation

[Project description](#)

[Release history](#)

[Download files](#)

### Project links

[Homepage](#)

### Project description

Azure Pipelines succeeded build failing build failing codecov 98% circleci passing python 3.6 | 3.7 | 3.8  
 pypi package 0.6.2 chat on gitter

### imbalanced-learn

imbalanced-learn is a python package offering a number of re-sampling techniques commonly used in datasets showing strong between-class imbalance. It is compatible with [scikit-learn](#) and is part of [scikit-learn-contrib](#) projects.

### Documentation

# WEKA i inne

---

Podstawowa WEKA = resampling (SMOTE oraz random),  
cost-sensitive classifiers, MetaCost

KEEL - więcej algorytmów (45) over i under sampling (20)  
oraz rozbudowane zespoły klasyfikatorów (21)

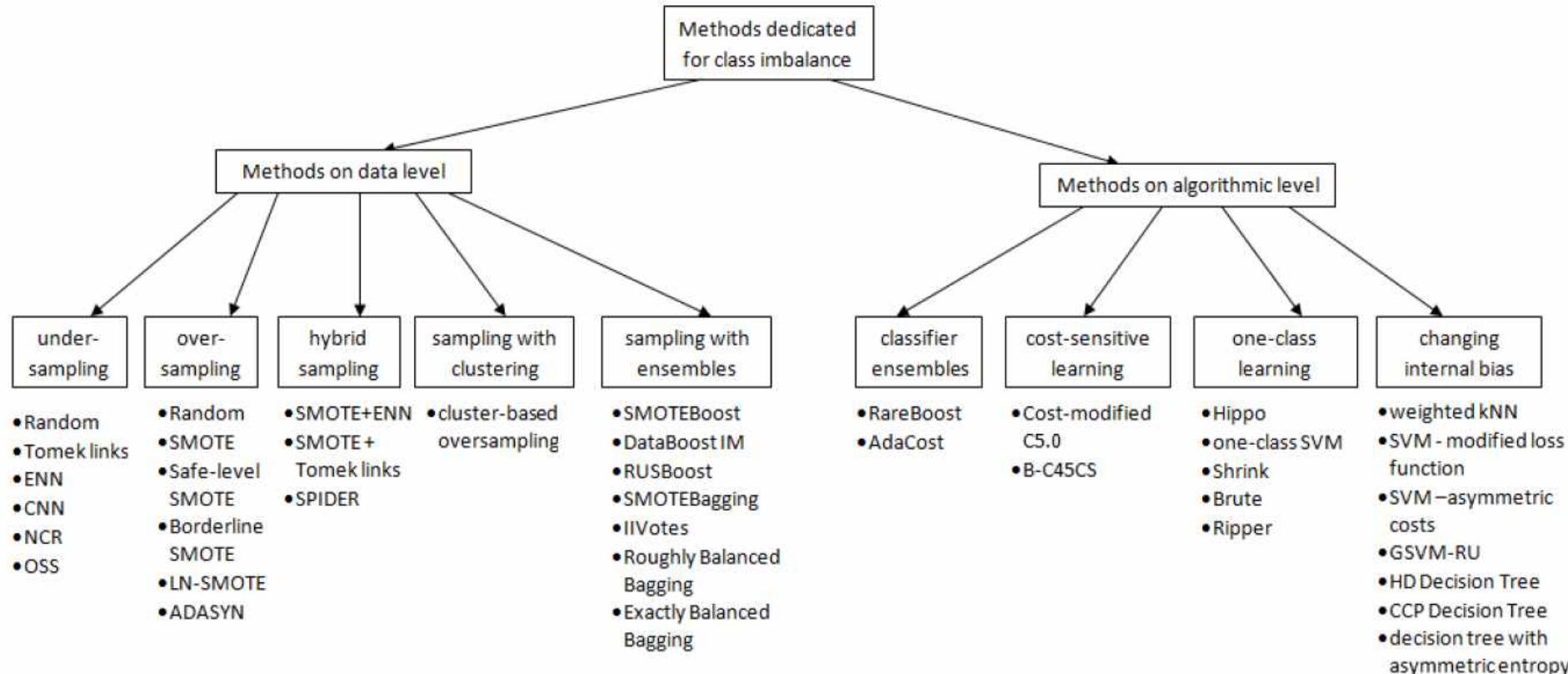
R package ‘imbalance’ oraz IRIC: An R library for binary  
imbalanced classification: 29 metod re-sampling, 4 zespoły  
klasyfikatorów, 1 cost sensitive

# Literaturowa kategoryzacja metod

---

- Dwa podstawowe kierunki działanie
  - Modyfikacje danych (preprocessing)
  - Modyfikacje algorytmów
- Najbardziej popularne grupy metod
  - **Re-sampling** or re-weighting,
  - Zmiany w strategiach uczenia się, użycie nowych miar oceny (np. AUC)
  - Nowe strategie eksploatacji klasyfikatora (classification strategies)
  - Ensemble approaches (najczęściej adaptacyjne klasyfikatory złożone typu bagging)
  - Specjalizowane systemy hybrydowe
  - One-class-learning
  - Transformacje do zadania „cost-sensitive learning”
  - ...

# Quick view at methods for class imbalance



and even more, ...

Review →

He H., Yungian, Ma (eds): Imbalanced Learning. Foundations, Algorithms and Applications. IEEE - Wiley, 2013  
A.Fernandez et al.: Learning from imbalanced data sets. Springer 2018.

# Inne podejścia do modyfikacji algorytmów uczących

---

## □ Zmiany w indukcji drzew decyzyjnych

- Weiss, G.M. Provost, F. (2003) "Learning When Training Data are Costly: The Effect of Class Distribution on Tree Induction" JAIR.
- Hellinger distances i asymetryczne entropie (Chawla et al.)

## □ Modyfikacje w klasyfikatorach bayesowskich

- Jason Rennie: Tackling the Poor Assumptions of Naive Bayes Text Classifiers ICML 2003.

## □ Wykorzystanie „cost-learning” w algorytmach uczących

- Domingos 1999; Elkan, 2001; Ting 2002; Zadrozny et al. 2003; Zhou and Liu, 2006

## □ Modyfikacje zadania w SVM

- K.Morik et al., 1999.; Amari and Wu (1999)
- Wu and Chang (2003),
- B.Wang, N.Japkowicz: Boosting Support Vector Machines for Imbalanced Data Sets, KAIS, 2009.

# Metody modyfikujące zbiór uczący

---

Zmiana rozkładu przykładów w klasach przed indukcją klasyfikatora (ang. pre-processing):

- Proste techniki losowe
  - „Over-sampling” - klasa mniejszościowe
  - „Under-sampling” - klasa mniejszościowa
- Specjalizowane nadlosowanie
  - Cluster-oversampling (Japkowicz)
- Ukierunkowane transformacje**
  - Klasa większościowe
    - One-side-sampling (Kubat, Matwin) z Tomek Links
    - Laurikkala’s edited nearest neighbor rule
  - Klasa mniejszościowe
    - SMOTE → Chawla et al.
    - Borderline SMOTE, Safe Level, Surrounding SMOTE, ...
  - Podejścia łączone (hybrydowe)
    - SPIDER
    - SMOTE i undersampling
  - Powiązanie z budową klasyfikatorów złożonych



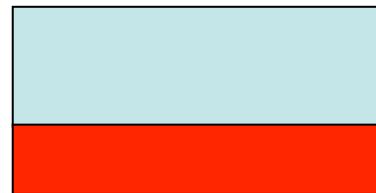
# Resampling – modyfikacja zbioru uczącego przed budową klasyfikatora

---

„Resampling” → pre-processing; celowa zmiana rozkładu przykładów; „balansowanie” liczności klas po to aby w kolejnej fazie móc lepiej nauczyć klasyfikator

Raczej heurystyka ukierunkowana na uzyskanie lepszych rozkładów klas niż uzasadnione teoretycznie podejście [F.Herrera 2010].

Brak teoretycznej gwarancji znalezienia optymalnej postaci rozkładu!



Oryginalny zbiór



**Wybór lub  
nadlosowanie**



Zmodyfikowany  
zbalansowany  
rozkład

# Losowe nadlosowanie lub usuwanie przykładów

---

ang. undersampling vs oversampling



**under-sampling**



**over-sampling**



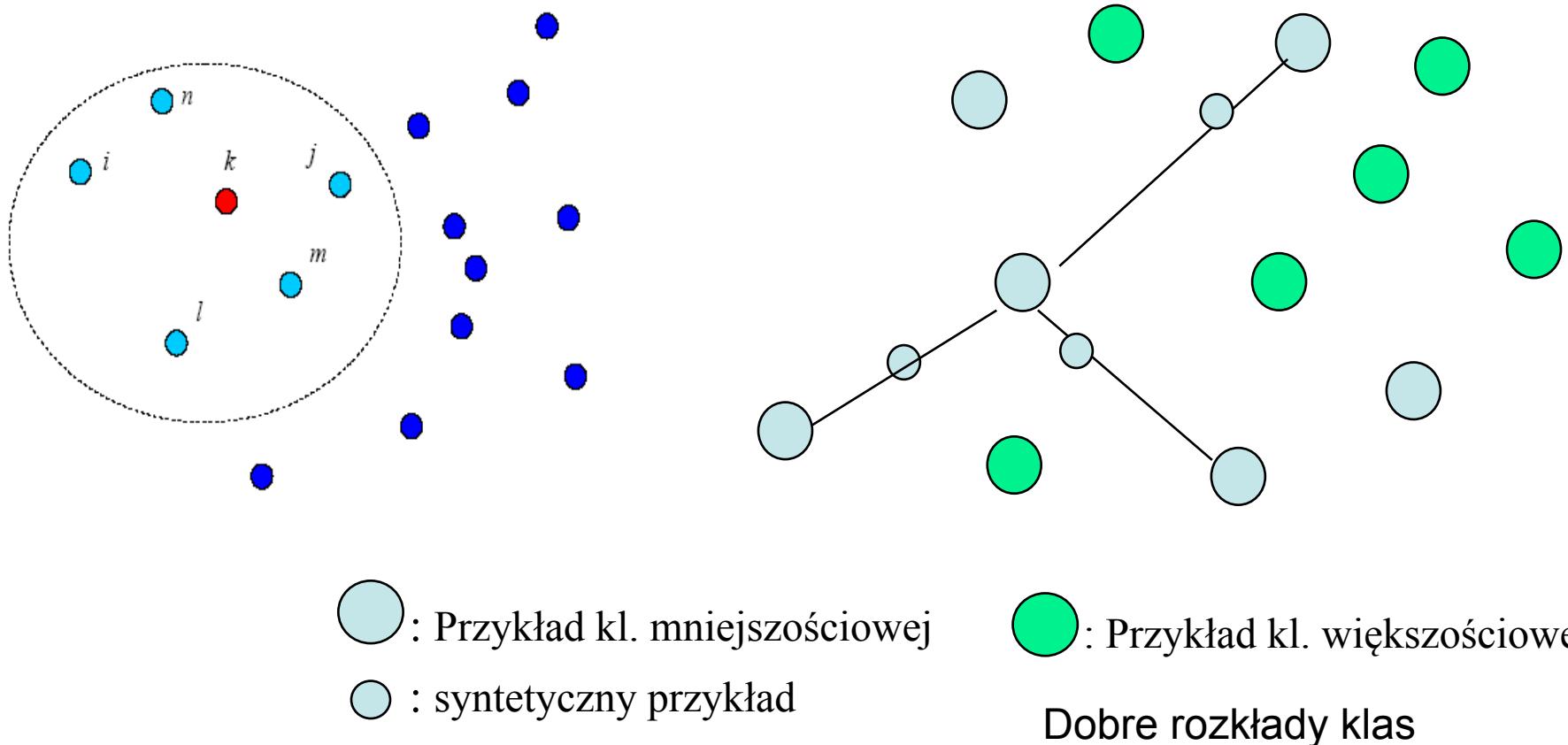
# SMOTE - Synthetic Minority Oversampling Technique

---

- Wprowadzona przez Chawla, Hall, Kegelmeyer 2002
- Dla każdego przykładu  $p$  z klasy mniejszościowej
  - Znajdź jego  $k$ -najbliższych sąsiadów (UWAGA wyłącznie z klasy mniejszościowej!)
  - Losowo wybierz  $j$  z powyższych sąsiadów
  - Losowo stwórz sztuczny przykład wzdłuż lini łączącej  $p$  z wybranym losowo jego sąsiadem  
**(parametr j - the amount of oversampling desired)**
- Porównując z simple random oversampling - SMOTE rozszerza regiony klasy mniejszościowej starając się robić je mniej specyficzne, „paying attention to minority class samples without causing overfitting”.
- SMOTE - uznawana za bardzo skuteczną zwłaszcza w połączeniu z odpowiednim undersampling (wyniki Chawla, 2003).

# Oversampling klasy mniejszościowej w SMOTE

SMOTE – analiza WYŁĄCZNIE klasy mniejszościowej!



SMOTE – może wstawić sztuczne przykłady w regionach klasy większościowej / wprowadza zakłócenia, szum

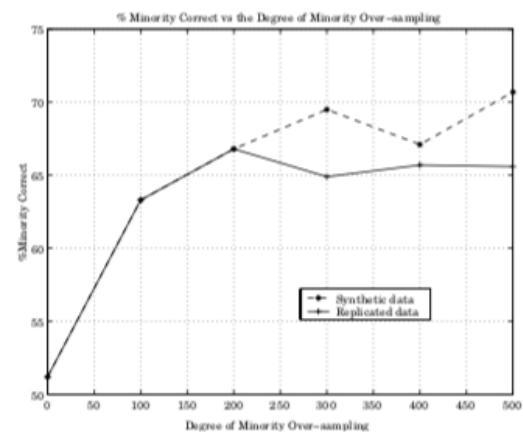
# SMOTE zbiorcza ocena

$k = 5$  sąsiadów, różny stopień nadlosowania (np. 100% to dwukrotne zwiększenie liczności klasy mniejszościowej)

Dataset	Under	50 SMOTE	100 SMOTE	200 SMOTE	300 SMOTE	400 SMOTE	500 SMOTE
Pima	7242		<b>7307</b>				
Phoneme	8622		8644	<b>8661</b>			
Satimage	8900		8957	<b>8979</b>	8963	8975	8960
Forest Cover	9807		9832	9834	<b>9849</b>	9841	9842
Oil	8524		8523	8368	8161	8339	<b>8537</b>
Mammography	9260		9250	9265	9311	<b>9330</b>	9304
E-state	6811		6792	<b>6828</b>	6784	6788	6779
Can	9535	<b>9560</b>	9505	9505	9494	9472	9470

Table 3: AUC's [C4.5 as the base classifier] with the best highlighted in bold.

SMOTE



: Comparison of % Minority correct for replicated over-sampling and SMOTE for the Mammography dataset

# SMOTE - uwagi krytyczne

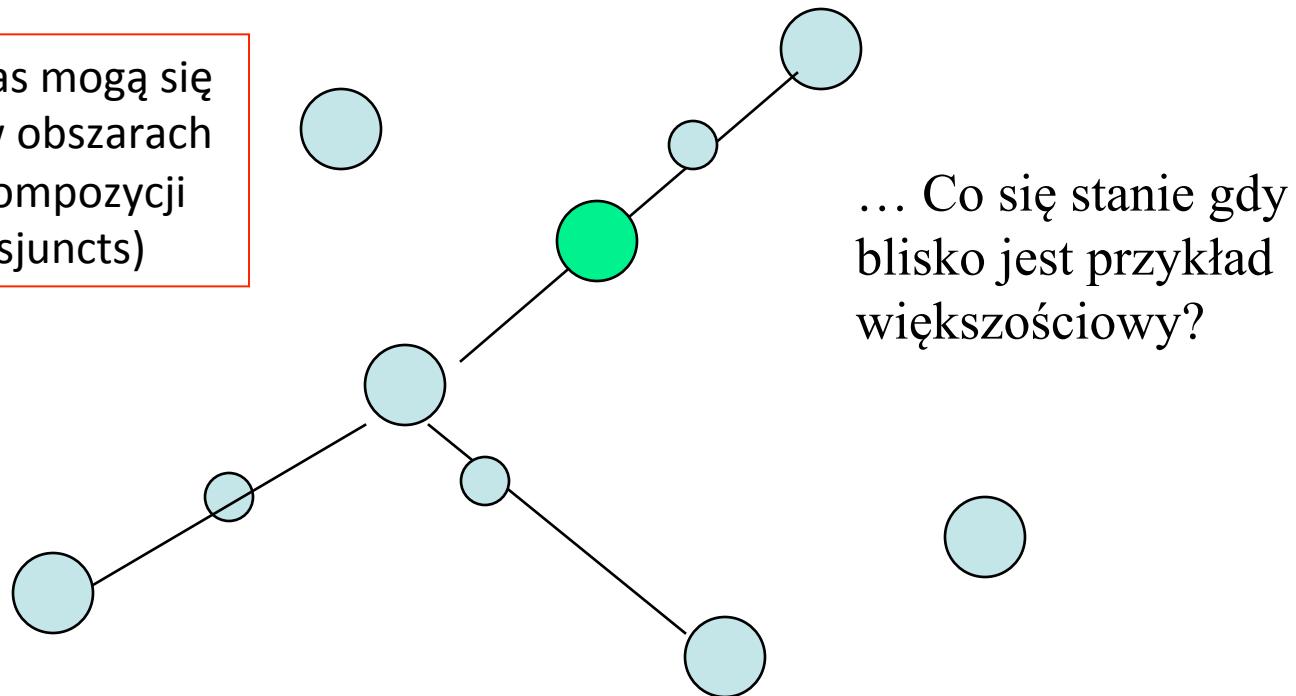
---

- Ślepe nadlosownie
  - SMOTE jest „naturalnie” niebezpieczna, gdyż ślepo uogólnia mniejszościowe przykłady bez rozważania rozkładów klasy większościowej
  - Szczególnie problematyczne dla mocno rozproszonych klas z tzw. small disjuncts → zwiększa szanse na nakładanie się rozkładów klas
- Trudność strojenia
  - Liczba przykładów do nadlosowania klasy mniejszościowej musi być znana przed uruchomieniem procedury.
  - Właściwe dostrojenie parametrów silnie zależne od zadania

# Oversampling klasy mniejszościowej w SMOTE

Oversampling – nie rozważa rozkładów klasy większościowej

Pamiętaj, że rozkłady klas mogą się „przenikać” zwłaszcza w obszarach brzegowych i przy dekompozycji klas (sparse small disjuncts)



: Minority sample



: Synthetic sample



: Majority sample

# Najnowsze rozszerzenia SMOTE

---

**Borderline\_SMOTE:** H. Han, W.Y. Wang, B.H. Mao. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. International Conference on Intelligent Computing (ICIC'05). Lecture Notes in Computer Science 3644, Springer-Verlag 2005, Hefei (China, 2005) 878-887

**Safe\_Level\_SMOTE:** C. Bunkhumpornpat, K. Sinapiromsaran, C. Lursinsap. Safe-level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-09). LNAI 5476, Springer-Verlag 2005, Bangkok (Thailand, 2009) 475-482

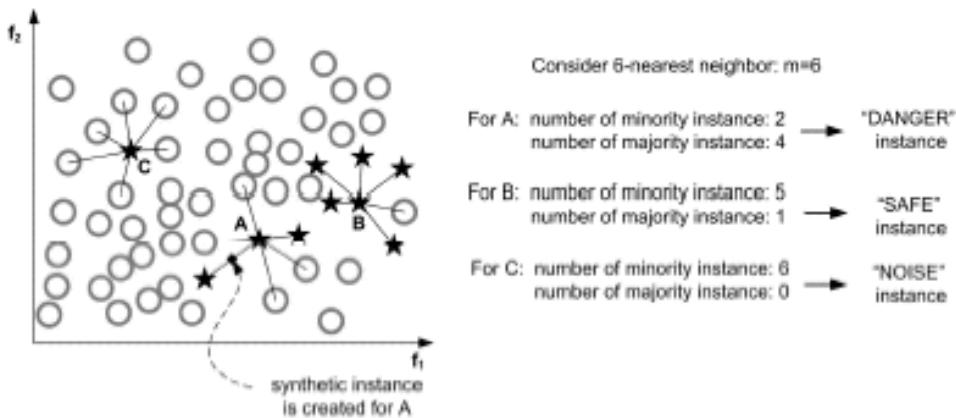
**SMOTE\_LLE:** J. Wang, M. Xu, H. Wang, J. Zhang. Classification of imbalanced data by using the SMOTE algorithm and locally linear embedding. IEEE 8th International Conference on Signal Processing.

**LN-SMOTE:** J. Stefanowski, T. Maciejewski: Local Neighbourhood in SMOTE for Mining Imbalanced Data. IEEE CIDM, 2010

# SMOTE Borderline

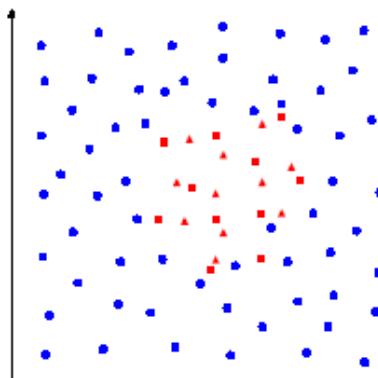
## Przykład ilustracyjny Borderline

Trzy typy przykładów mniejszościowych DANGER, SAFE, NOISE

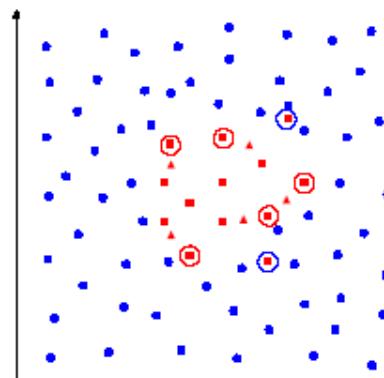


Nadlosowyj tylko  
DANGER wg  
zasady SMOTE

Fig. 4. Data creation based on Borderline instance.



RYSUNEK 6.1: SMOTE



RYSUNEK 6.2: Borderline-SMOTE1

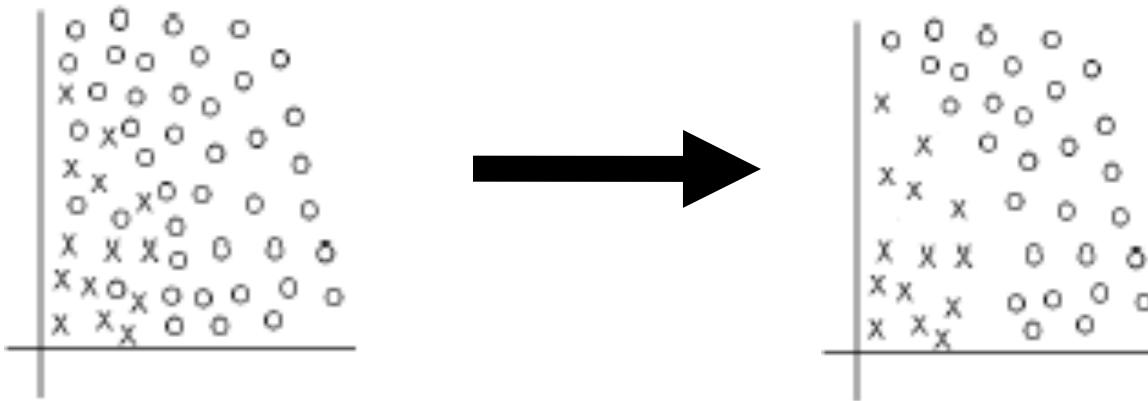
# Różne rozszerzania SMOTE i C4.5 trees → F-measure

	<b>None</b>	<b>SMO</b>	<b>BS1</b>	<b>BS2</b>	<b>SLS</b>	<b>LN1</b>	<b>LN2</b>
<b>Balance scale</b>	0.00	9.29	8.40	11.33	8.58	16.54	16.08
<b>Breast cancer</b>	39.83	43.83	43.02	44.37	45.15	43.83	45.64
<b>Cleveland</b>	19.29	26.71	25.27	28.33	26.03	29.27	29.70
<b>CMC</b>	40.81	41.64	42.05	44.16	41.64	44.95	45.94
<b>Ecoli</b>	58.86	64.31	62.38	64.02	63.98	62.01	66.96
<b>Flags</b>	30.89	44.51	41.35	42.68	43.15	39.46	42.03
<b>Germ. credit</b>	45.51	50.30	49.98	51.01	50.02	50.91	50.46
<b>Haberman</b>	30.36	43.70	41.84	43.58	40.08	44.56	42.59
<b>Hepatitis</b>	49.20	52.10	53.94	53.00	57.10	58.57	57.86
<b>Pima</b>	62.05	65.51	65.68	65.61	65.02	65.13	65.06
<b>Post-operative</b>	5.84	22.03	22.86	19.06	20.56	20.42	19.44
<b>Solar flare</b>	28.79	27.84	28.85	29.93	28.68	31.60	33.08
<b>Transfusion</b>	47.27	48.80	50.05	51.12	48.94	49.19	50.30
<b>Yeast</b>	35.02	39.64	42.23	42.02	40.07	41.39	42.58

- LN SMOTE - największa poprawa (balance 7.25., solar flare 5.24)
- Najlepszy dla 11 z 14 danych; LN SMOTE ver 2 > LN SMOTE ver. 1
- Podobne trendy dla G-means + PART rules, k-NN

# Ukierunkowane modyfikacje danych

Focused resampling (Informed approaches): przetwarzaj tylko trudne obszary



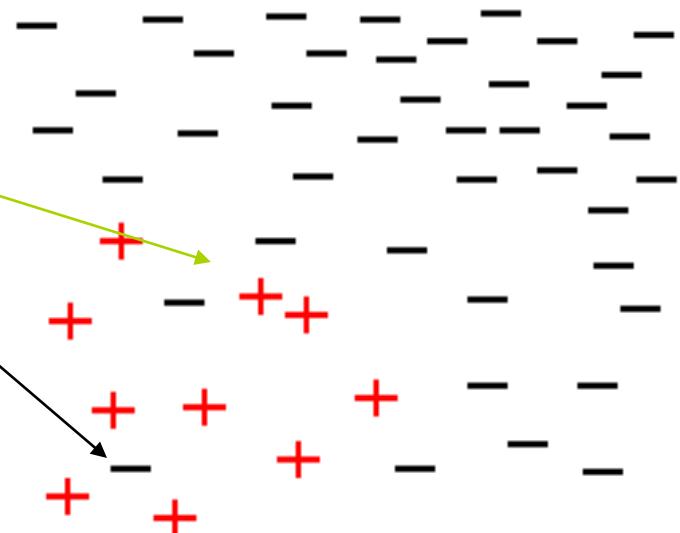
- Czyszczenie borderline, redundant examples: Tomek links i one-side sampling
- Czyszczenie szumu i borderline: NCR
- Metoda SPIDER (J.Stefanowski, Sz.Wilk)
- SMOTE i jej rozszerzenia
- Czy są to typowe tricki „losowania” oraz edytowanie danych (np. rozszerzania k-NN)?

# Powróćmy do charakterystyki przykładów

**Typy przykładów →** techniki „resampling” powinny skupić swoje działanie na niektórych z nich

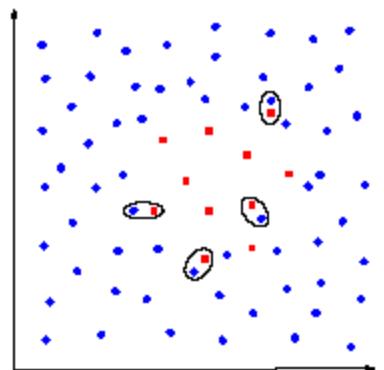
Różne typy przykłady

- Noise przykłady zaszumione
- Borderline examples  
Trudne przykłady w strefie  
brzegowej oraz tuż przy  
granicy.
- Rzadkie przykłady
- Safe bezpieczne przykłady



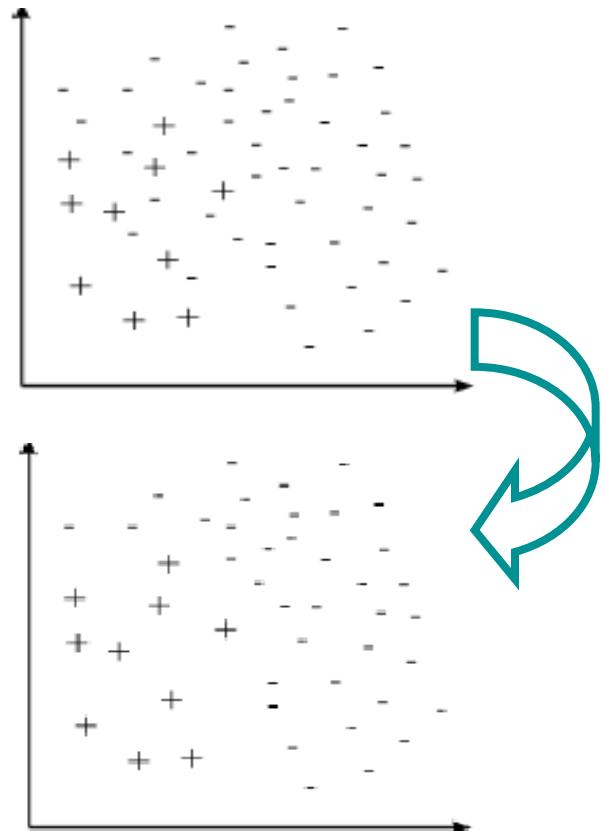
**Modyfikacje undersampling:** Znajdź i usuń 2 lub 3 pierwsze typy przykładów

# Under-sampling z wykorzystaniem Tomek links



Przykład Tomek Links

- Usuwaj przykłady graniczne i szum z klasy większościowej
- „Tomek link”
  - $E_i, E_j$  należą do różnych klas,  
 $d(E_i, E_j)$  odległość między nimi.
  - para  $(E_i, E_j)$  jest tzw. Tomek link jeśli nie istnieje inny przykład  $E_l$ , spełniający  
 $d(E_i, E_l) < d(E_i, E_j)$  or  $d(E_j, E_l) < d(E_i, E_j)$ .



# Nearest Cleaning Rule

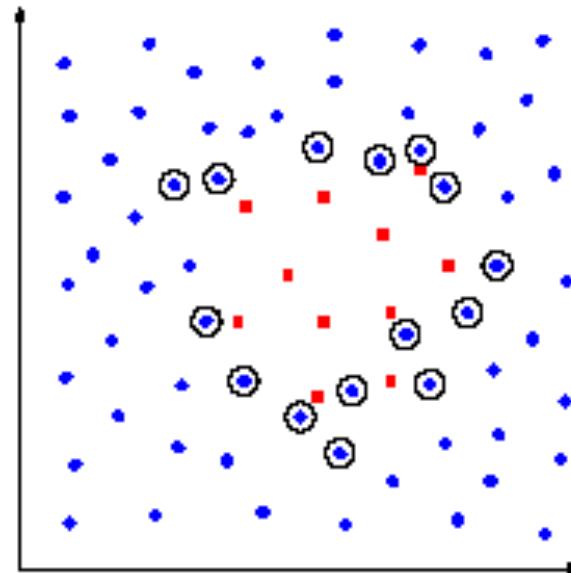
- **NCL** Nearest Cleaning Rule - Jorma Laurikkala 2001,

Inne od OSS, bardziej „czyści” obszary brzegowe klas niż redukuje przykłady

Algorytm:

- Find three nearest neighbors for each example  $E_i$  in the training set
- If  $E_i$  belongs to majority class, & the three nearest neighbors classify it to be minority class, then remove  $E_i$
- If  $E_i$  belongs to minority class, and the three nearest neighbors classify it to be majority class, then remove the three nearest neighbors

Ilustracja – które przykłady większościowe usuwamy



Rysunek 5.3: Neighbourhood Cleaning Rule

# Selective Preprocessing of Imbalanced Data → SPIDER

---

- Ukierunkowane na wzrost **czułości** (ang. **sensitivity**) dla **klasy mniejszościowej** przy możliwie jak najmniejszym spadku specyficzności
- Rozróżnienie rodzaju przykładów: bezpieczne safe (certain lub possible); unsafe (brzegowe, noise, outliers)
- Metoda hybrydowa → ograniczony undersampling i lokalizowany over-sampling

Dwie fazy

- W przypadku klasy większościowej **selektywne usunięcie** noise certain i części z noise possible
  - **Możliwość przetykietowania** przykładów noise certain
- W przypadku klasy mniejszościowej - modyfikacje przykładów brzegowych i noise (**nadlosowania**)
  - weak or strong amplification / SPIDER 1 kopiowanie wybranych przykładów lub **relabel** (zmień etykietę większościowego)
  - Stopień wzmocnienia zależny od analizy sąsiedztwa (ENN)

# Miara czułości klasy mniejszościowej

---

Dane	Pojed. Klasyfik.	Under- sampling	Over- sampling	SPIDER
<i>breast ca</i>	0.3056	0.5971	0.4043	<b>0.6264</b>
<i>bupa</i>	0.7290	0.6707	0.5935	<b>0.8767</b>
<i>ecoli</i>	0.4167	<b>0.8208</b>	0.5150	0.7750
<i>pima</i>	0.4962	0.7093	0.5519	<b>0.8098</b>
<i>Acl</i>	0.7250	0.8485	0.7840	<b>0.8750</b>
...	...	...	...	...
<i>Wisconsin</i>	0.9083	0.9521	0.8326	<b>0.9625</b>
<i>hepatitis</i>	0.4833	<b>0.7372</b>	0.5447	0.6500

Nowe podejście zwiększa znaczaco wartość miary Sensitivity

# Cost learning

Potrzeba zdefiniowania macierzy kosztów pomyłek

		Actual = negative	Actual = positive	
		$TN$	$FN$	Positive – Minority class
		$FP$	$TP$	Imbalanced FN is more dangerous than FP !
		True = 0	True = 1	
Predict = 0		$C(0,0)$	$C(0,1)$	Zwykle $C(0,1)$ większe niż $C(1,0)$
Predict = 1		$C(1,0)$	$C(1,1)$	

# Cost learning

---

The cost of labeling an example incorrectly should always be greater than the cost of labeling it correctly.[C.Elklan]

$C(0,1) >> C(1,0)$  i ....

	True = 0	True = 1
Predict = 0	0	80
Predict = 1	5	0

Jak zdefiniować precyzyjne wartości kosztów?

Jak je wykorzystać w klasyfikacji niezbalansowanych danych?

“In cost-sensitive learning instead of each instance being either correctly or incorrectly classified, each class (or instance) is given a misclassification cost. Thus, instead of trying to optimize the accuracy, the problem is then to **minimize the total misclassification cost.**”

# Definiowanie kosztów (globalne dla klasy)

Wiedząc, że koszt nieroznalezienia klasy mniejszościowej jest większy  
 $C(0,1) >> C(1,0)$

Prosto - ustal koszty proporcjonalnie do stopnia niezbalansowania, np.

	True = 0	True = 1
Predict = 0	$\theta$	$1 * IR$
Predict = 1	$1$	$\theta$

Nguyen, Gantner, Schmidt-Thieme: Cost-sensitive learning methods for imbalanced data

Potraktuj to jako hiper-parametr o lokalnej optymalizacji  
(wewnętrzna ocena krzyżowa)

Koszty pomyłek mogą być zdefiniowane dla poszczególnych przykładów z klasy = bardziej skomplikowane podejście

# Cost sensitive learning

---

Cost-Sensitive Learning is a type of learning that takes the misclassification costs (and possibly other types of cost) into consideration. The goal of this type of learning is to minimize the total cost [Ling,Sheng]

Dla danej macierzy kosztów, przykład klasyfikuje się do klasy z minimalnym oczekiwany kosztem

$$R(i|x) = \sum_j P(j|x) \cdot C(i,j)$$

gdzie  $P(j|x)$  jest estymatą prawdopodobieństwa przydziału  $x$  do  $j$ -tej klasy.

# Cost-sensitive learning

---

Przydziel x do klasy pozytywnej / mniejszościowej, gdy  
 $P(0|x)C(1,0)+P(1|x)C(1,1) \leq P(0|x)C(0,0)+P(1|x)C(0,1)$   
można przekształcić do

$$P(0|x)(C(1,0)-C(0,0)) \leq P(1|x)(C(0,1)-C(1,1))$$

wiedząc, że  $C(0,0)=C(1,1)=0$  otrzymujemy

$$P(0|x)C(1,0) \leq P(1|x)C(0,1) \text{ oraz } P(0|x)=1-P(1|x)$$

Otrzymujemy próg  $p^*$  pozwalający na klasyfikację przykładu x do klasy pozytywnej, gdy

$$p^* = \frac{C(1,0)}{C(1,0) + C(0,1)}$$

Kalibracja – dane zbalansowane  $p^*=0.5$

Niezbalsowanie mniejszościowa  $p^* < 0.5$

# Reguły i niezrównoważenie klas

---

- zbiór uczący Ecoli: 336 ob. i 35 ob. w klasie M ; 7 atr. liczbowych
- MODLEM (noprunе) 18 reguł, w tym 7 dla Minority class

r1.(a7<0.62)&(a5>=0.11) => (Dec=O); [230, 76.41%, 100%]

r2.(a1<0.75)&(a6>=0.78)&(a5<0.57) => (Dec=O); [27, 8.97%, 100%]

r3.(a1<0.46) => (Dec=O); [148, 148, 49.17%, 100%]

r4.(a1<0.75)&(a5<0.63)&(a2∈[0.49,0.6]) => (Dec=O); [65, 21.59%, 100%]

r5.(a1<0.75)&(a7<0.74)&(a2>=0.46) => (Dec=O); [135, 44.85%, 100%]

r6.(a2>=0.45)&(a6>=0.75)&(a1<0.69) => (Dec=O); [34, 11.3%, 100%]

...

r12.(a7>=0.62)&(a6<0.78)&(a2<0.49)&(a1 ∈[0.57,0.68]) => (Dec=M) [6, 17.14%, 100%]

r13.(a7>=0.62)&(a6<0.76)&(a5<0.65)&(a1 ∈[0.73,0.82]) => (Dec=M)[7, 20%, 100%]

r14.(a7>=0.74)&(a1>=0.47)&(a2>=0.45)&(a6<0.75)&(a5>=0.59) => (Dec=M); [3, 8.57%, 100%]

r15.(a5>=0.56)&(a1>=0.49)&(a2 ∈[0.42,0.44]) => (Dec=M); [3, 8.57%, 100%]

r16.(a7>=0.74)&(a2 ∈[0.53,0.54]) => (Dec=M); [2, 5.71%, 100%]

...

- A strategia klasyfikacyjna:

- Niejednoznaczne wielokrotne dopasowanie? Głosowanie większościowe
- Brak dopasowania? - reguły najbliższe

# BRACID

Bottom-up induction of Rules And Cases from Imbalanced Data

---

## Assumptions:

- Hybrid knowledge representation: rule and instances
- Induction rules by bottom-up strategy
- Resigning from greedy sequential covering
- Some inspirations from RISE [P.Domingos 1996]
- Considering info about types of difficult examples
- Local neighbors with HVDM
- Internal evaluation criterion (F-miara)
- Local nearest rules classification strategy

More →

K.Napierała, J. Stefanowski: BRACID A comprehensive approach to rule induction from imbalanced data. Int. Journal of Intelligent Information Systems. 2012

# Comparing classifiers - G-mean

---

Zbiór	BRACID	RISE	kNN	C45.rules	CN2	PART	RIPPER	Modlem	Modlem-C
abalone	<b>0,65</b>	0,34	0,36	0,57	0,40	0,42	0,42	0,48	0,51
b-cancer	<b>0,56</b>	0,54	0,47	0,49	0,46	0,53	0,48	0,49	0,53
car	0,87	0,75	0,08	0,86	0,71	<b>0,94</b>	0,71	0,88	0,88
cleveland	<b>0,57</b>	0,23	0,08	0,26	0,00	0,38	0,26	0,15	0,23
cmc	<b>0,64</b>	0,51	0,52	0,59	0,26	0,54	0,25	0,47	0,54
credit-g	0,61	0,54	0,57	0,55	0,47	0,60	0,44	0,56	<b>0,65</b>
ecoli	<b>0,83</b>	0,64	0,70	0,72	0,28	0,55	0,59	0,57	0,63
haberman	<b>0,58</b>	0,38	0,33	0,43	0,35	0,47	0,36	0,40	0,53
hepatitis	<b>0,75</b>	0,60	0,62	0,51	0,05	0,55	0,50	0,50	0,64
new-thyroid	<b>0,98</b>	0,95	0,92	0,90	0,92	0,95	0,91	0,88	0,90
solar-flareF	<b>0,64</b>	0,14	0,00	0,27	0,00	0,32	0,02	0,13	0,32
transfusion	<b>0,64</b>	0,51	0,53	0,58	0,34	0,60	0,27	0,53	0,58
vehicle	<b>0,94</b>	0,90	0,91	0,91	0,51	0,92	0,92	0,92	<b>0,94</b>
yeast-ME2	<b>0,71</b>	0,44	0,34	0,51	0,00	0,42	0,45	0,34	0,37

# Adaptacje zespołów klasyfikatorów

---

## □ Data preprocessing + ensemble

### ■ Boosting-based

- SMOTEBoost, DataBoost

### ■ Bagging-based

- Exactly Balanced Bagging
- Roughly Balanced Bagging
- OverBagging
- UnderOverBagging
- SMOTEBagging
- NBBag

## □ Inne or Hybrid (EasyEnsemble)

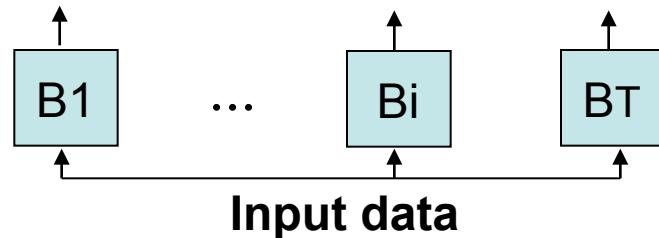
## □ Cost Sensitive Boosting

### ■ AdaCost (C1-C3)

### ■ RareBoost

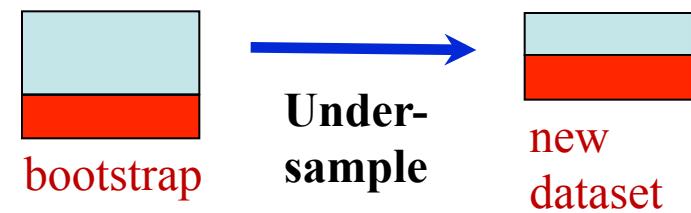
# Under- Bagging – popularne rozszerzania

- Standardowy Bagging → wykorzystuje bootstraps
  - sampling N examples (with replacements) equal probability



## Propozycje z Undersampling

- Exactly Balanced Bagging [Ch03]
  - bootstrap samples = copy of the minority class + randomly drawn subset of the majority class ( $N_{maj} = N_{min}$ )
- Rough Balanced Bagging [Hido 09]
  - Inaczej - wyrównuje prawdopodobieństwa klas w losowaniu



# Roughly Balanced Bagging

Hido S., Kashima H.: Roughly balanced bagging for imbalance data (2008)

## Data preprocessing + ensemble

- Under-sampling modification of Exactly Balanced Bagging
- Instead of fixing the constant sample size, it equalizes the sampling probability of each class
- For each of  $T$  iterations the size of the majority class in the bootstrap  $BS_{\text{maj}}$  is determined probabilistically according to the negative binomial distribution

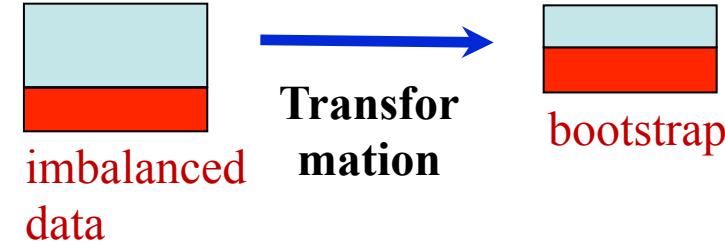
For each bootstrap

- Random size  $BS_{\text{maj}}$
- Sample with replacement  $N_{\text{min}}$  and  $BS_{\text{maj}}$

Prediction with majority voting

## Przykładowe rozszerzenia:

- Attribute Selection with RBBag for highly dimensional data
- Multi-class generalization (changing sampling idea)



Lango M., Stefanowski J.: The Usefulness of Roughly Balanced Bagging for Complex and High-dimensional Imbalanced Data (2016)

# Wybrane otwarte problemy

---

- Lepsze zrozumienie problemu
  - Analiza sztucznych i rzeczywistych danych
  - Lepsze wykrywanie dekompozycji na pod-pojęcia
  - Teoretyczna analiza wybranych metod
- Multi-class imbalanced data
- Nowe miary oceny
- Rozważanie danych wielowymiarowych
- Uczenie przyrostowe
- Niebalansowane strumienie danych i zmiany podjęć
- Niebalansowanie regresji, alg. skupień, its.
- Large scale imbalanced learning i Big Data



Spójrz do B.Krawczyk Learning from imbalanced data: open challenges and future directions (2016)

# Literatura przeglądowa

---

1. G. M. Weiss. Mining with Rarity: A Unifying Framework. SIGKDD Explorations, 6(1):7-19, June 2004
2. Chawla N., Data mining for imbalanced datasets: an overview. In The Data mining and knowledge discovery handbook, Springer 2005.
3. Garcia V., Sánchez J.S., Mollineda R.A., Alejo R., Sotoca J.M. The class imbalance problem in pattern classification and learning. pp. 283-291, 2007
4. Visa, S. and Ralescu, A. Issues in mining imbalanced data sets - a review paper. Proceedings of the Midwest Artificial Intelligence and Cognitive Science Conference, Dayton, pp.67-73, 2005
5. Y. Sun, A. K. C. Wong and M. S. Kamel. Classification of imbalanced data: A review. International Journal of Pattern Recognition 23:4 (2009) 687-719.
6. He, H. and Garcia, E. A. Learning from Imbalanced Data. IEEE Trans. on Knowl. and Data Eng. 21, 9 (Sep. 2009), pp. 1263-1284, 2009

*IEEE ICDM noted “Dealing with Non-static, Unbalanced and Cost-sensitive Data” among the 10 Challenging Problems in Data Mining Research*

# Inne odnośniki literaturowe

---

- J.Błaszczyński, M.Deckert, J.Stefanowski, Sz.Wilk: Integrating Selective Pre-processing of Imbalanced Data with Ivotes Ensemble. RSCTC 2010, LNAI vol. 6086, Springer Verlag 2010, 148-157
- J.W. Grzymala-Busse, J.Stefanowski, S. Wilk: A Comparison of Two Approaches to Data Mining from Imbalanced Data, Proc. of the 8th Int. Conference KES 2004, Lecture Notes in Computer Science, vol. 3213, Springer-Verlag, 757-763
- K.Napierała, J.Stefanowski: Identification of Different Types of Minority Class Examples in Imbalanced Data. Proc. HAIS 2012, Part II, LNAI vol. 7209, Springer Verlag 2012, 139-150.
- K.Napierała, J.Stefanowski, Sz.Wilk: Learning from Imbalanced Data in Presence of Noisy and Borderline Examples. RSCTC 2010, LNAI vol. 6086, 2010, 158-167
- K. Napierała, J. Stefanowski: BRACID Journal of Intelligent Information Systems 2013
- T. Maciejewski, J. Stefanowski: Local Neighbourhood Extension of SMOTE for Mining Imbalanced Data. Proc. of IEEE Symposium on Computational Intelligence and Data Mining, SSCI IEEE, April 11-15, 2011, Paris, IEEE Press, 104–111
- J.Stefanowski, S. Wilk: Rough sets for handling imbalanced data: combining filtering and rule-based classifiers. Fundamenta Informaticae, vol. 72, no. (1-3) July/August 2006, 379-391.
- J.Stefanowski, Sz.Wilk: Improving Rule Based Classifiers Induced by MODLEM by Selective Pre-processing of Imbalanced Data. Proceedings of the RSKT Workshop ECML/PKDD, 2007, 54-65.
- J.Stefanowski, Sz.Wilk: Selective pre-processing of imbalanced data for improving classification performance. Proc. of 10th Int. Conf. DaWaK 2008, LNCS vol. 5182, Springer Verlag, 2008, 283-292.
- I wiele inne

---

# Pytanie i komentarze?

Dalszy kontakt:

[jerzy.stefanowski@cs.put.poznan.pl](mailto:jerzy.stefanowski@cs.put.poznan.pl)

<http://www.cs.put.poznan.pl/jstefanowski/>



**Fundusze  
Europejskie**  
Polska Cyfrowa



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz  
Rozwoju Regionalnego



# Zespoły modeli predykcyjnych

## Systemy uczące się wykład 8

Jerzy Stefanowski

Instytut Informatyki PP

2021

Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI-TECH)  
projekt finansowany z środków Programu Operacyjnego Polska Cyfrowa  
POPC.03.02.00-00-0001/20



Fundusze  
Europejskie  
Polska Cyfrowa



Rzeczpospolita  
Polska

Unia Europejska  
Europejski Fundusz  
Rozwoju Regionalnego



# Plan wykładów 8 i 9

- Motywacje do tworzenia zespołów modeli
- Kiedy łączenie klasyfikatorów jest skuteczne
- Różne podejścia do tworzenia zróżnicowanych klasyfikatorów
- Zasady agregacji odpowiedzi klasyfikatorów składowych
- Metoda bagging
- Feature esmebles i Random forest

# Plan wykładów 8 i 9

- Metody Boosting
  - AdaBoost
- Porównania Bagging vs. Boosting
- Zróżnicowanie klasyfikatorów składowych
- Generalizacja stosowa (stacking) i tzw. mixture of experts
- Podejścia zespołowe do danych silnie wieloklasowych
- Boosted trees ensembles
- Podsumowanie

# Ogólne założenia

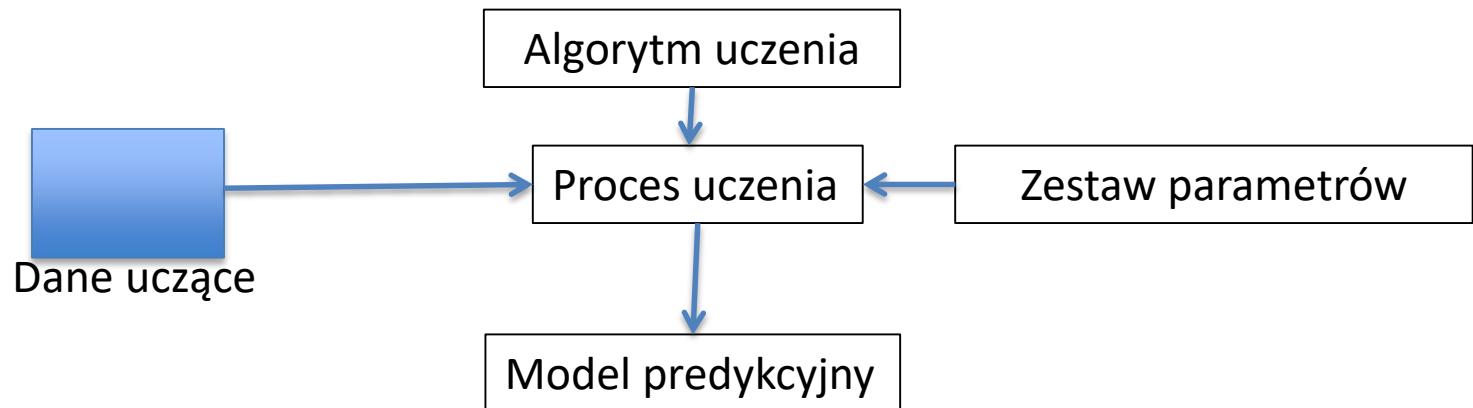
- Zespoły modeli (szerszy termin) obejmują integrację wielu składowych modeli nauczonych dla
  - Klasyfikatorów
  - Modeli regresji
  - Algorytmów tworzenia skupień (nie zajmujemy się w tym wykładzie)

# Typowe podejście do uczenia nadzorowanego

Dla danego problemu – danych (przykładów uczących)

Poszukuje się jednego najlepszego modelu (klasyfikatora)

Wybór algorytmu uczenia klasyfikatora, dobór parametrów, porównanie z innych możliwymi klasyfikatorami (algorytmami), intensywne oceny eksperymentalne (np.z wykorzystaniem k-fold cross validation)



# Czy zawsze szukać jednego modelu?

## Lekcje z doświadczeń

- Nie ma jednego algorytmu najlepszego dla wielu możliwych problemów!
- Można skonstruować model wystarczająco skuteczny dla wybranego zestawu problemów.
- Skomplikowane, złożone problemy często mogą być zdekomponowane / rozłożone na prostsze podproblemy (rozwiążane niezależnie i zintegrowane)

Motywacja – integracja wielu niezależnych modeli!

# W stronę rozwiązań zespołowych

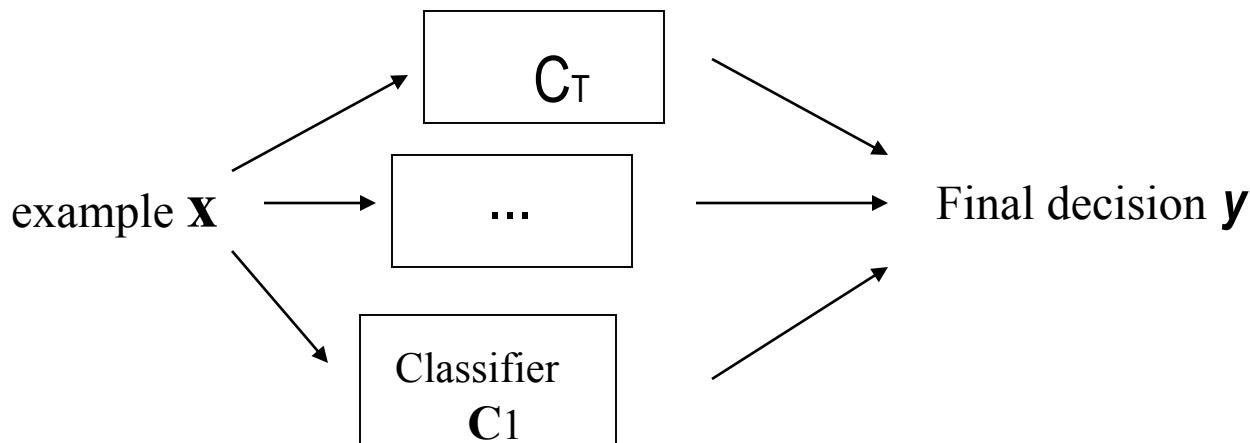
- Integracja wielu modeli w jeden system predykcyjny może polepszyć trafność predykcji oraz pozwolić rozwiązać bardziej złożone problemy
  - Cytat:

*„Multiple learning systems try to exploit the local different behavior of the base learners to enhance the accuracy of the overall learning system”*

- G. Valentini, F. Masulli
- Terminy angielskie - ensembles lub multiple learning classifiers

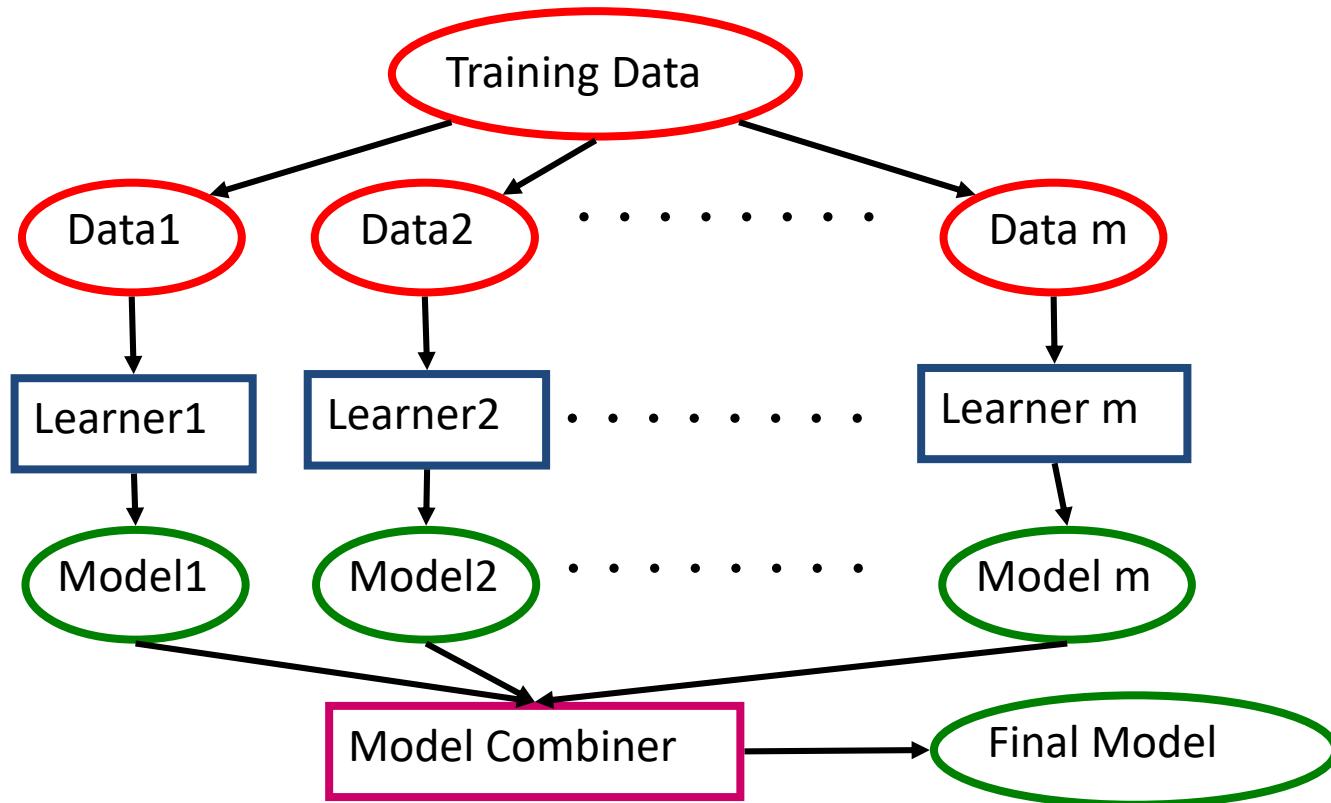
# Definicja

- Zbiór wielu modeli (klasyfikatorów, predyktorów regresji) których decyzje są integrowane tak aby klasyfikować nowe przykłady
- Różne nazwy po angielsku: **ensemble** methods, multiple classifiers, committee, classifier fusion, combination,...
- Przetarg pomiędzy złożonością systemu (ang. complexity) a zdolnością poprawy predykcji



# Uczenie zespołów

- Naucz się alternatywnych definicji problemów – modeli poprzez ich zróżnicowanie (np. wiele różnych wersji danych uczących albo różnych algorytmów)
- Integracja predykcji – np. poprzez różne formy głosowania



# Kiedy integracja jest skuteczna?

- Kiedy zespół może być skuteczniejszy niż pojedynczy model?
- Łączenie tak samo działających modeli jest nieskuteczne!
- Niezbędny jest pewien poziom niezgodności klasyfikatorów składowych, rozumiany w ten sposób, że jeśli popełniają błędne decyzje, to są one niezależne pomiędzy nimi (czyli nie popełniają równocześnie takich samych błędów)
- Pierwsze systematyczne prace (np. Hansen&Salamon90, Ali&Pazzani96), cyt:  
Member classifiers should **make uncorrelated errors** with respect to one another; each classifier should perform better than a random guess.

# Intuicja nieskorelowania błędnych predykcji

Rozważmy 3 klasyfikatory i głosowanie większościowe

Poprawna klasa	Model 1	Model 2	Model 3	Zespół
A	A	B	A	A
A	A	A	B	A
B	A	B	B	B
A	B	B	B	B
B	B	B	A	B
B	A	B	B	B
B	B	A	B	B
A	B	A	A	A
B	B	B	A	B
A	A	A	A	A
Accuracy	60%	70%	60%	90%

# Oczekiwana trafność predykcji zespołu

Niech:  $L$  – nieparzysta liczba składowych klasyfikatorów;  $p$  – prawdopodobieństwo poprawnej klasyfikacji składnika; wszystkie predykcje składników są prob. niezależne

Oczekiwana trafność predykcji zespołu  $L$  klasyfikatorów w głosowaniu większościowym:

$$p_{maj} = \sum_{m=\lfloor L/2 \rfloor + 1}^L \binom{L}{m} p^m (1-p)^{L-m}$$

$P > 0.5$  to  $p_{maj} \rightarrow 1$  wraz ze wzrostem  $L$

$P < 0.5$  to  $p_{maj} \rightarrow 0$  wraz ze wzrostem  $L$

$P = 0.5$  to  $p_{maj} = 0.5$

Dokładniejsza analiza w książce L.Kuncheva

# Głosowanie większościowe $p_{\text{maj}}$

$L$  niezależnych klasyfikatorów, każdy z oczekiwana trafnością  $p$

	L=5	L=7	L=9
$p=0.6$	0.6827	0.7102	0.7334
$p=0.7$	0.8369	0.8740	0.9012
$p=0.8$	0.9421	0.9667	0.9804
$p=0.9$	0.9914	0.9973	0.9991

# Poprawa predykcji wobec pojedynczego klasyfikatora

- Binarny zbalansowany problem; składowe klasyfikatory (o tym samym spodziewanym błędzie) i braku korelacji predykcji; głosowanie większościowe w zespole klasyfikatorów
- Oczekiwany błąd predykcji zespołu maleje wraz ze liczbą klasyfikatorów

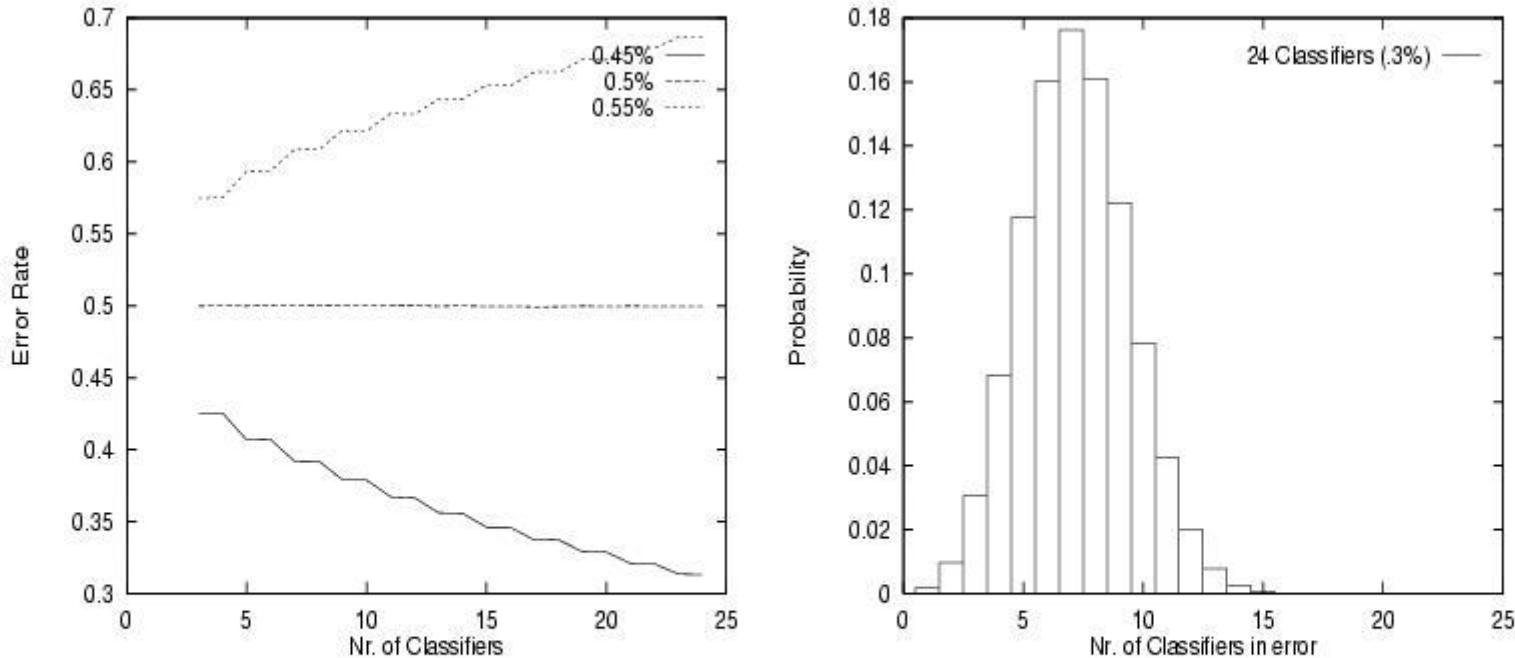
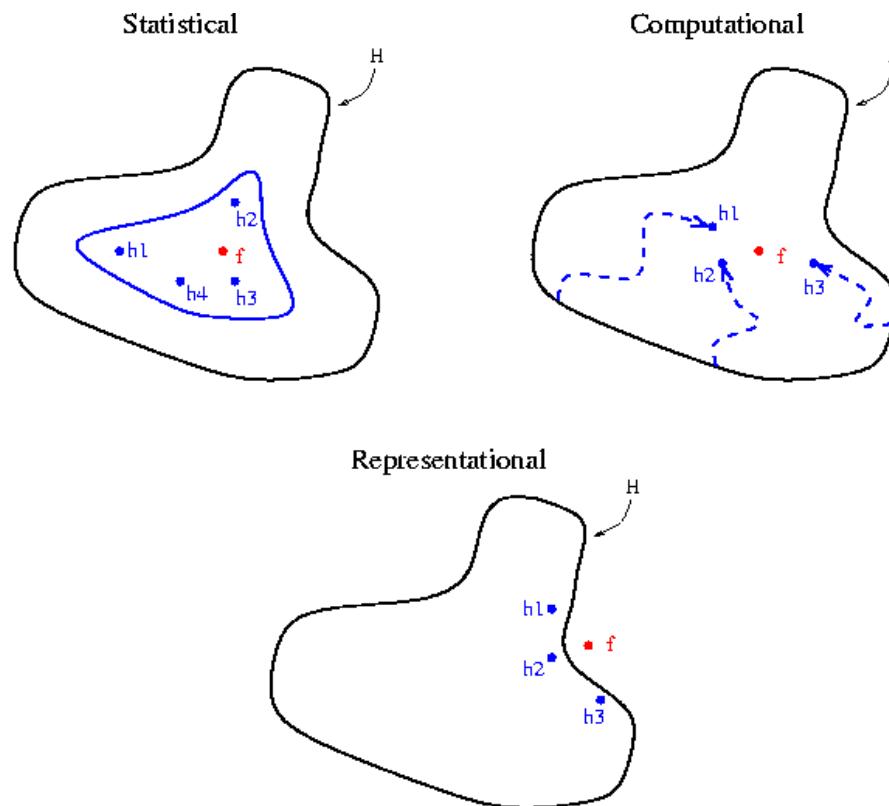


Figure 5.1: (a) Error rate versus nr. of classifiers in an ensemble. (b) Probability that exactly  $n$  of 24 classifiers will make an error.

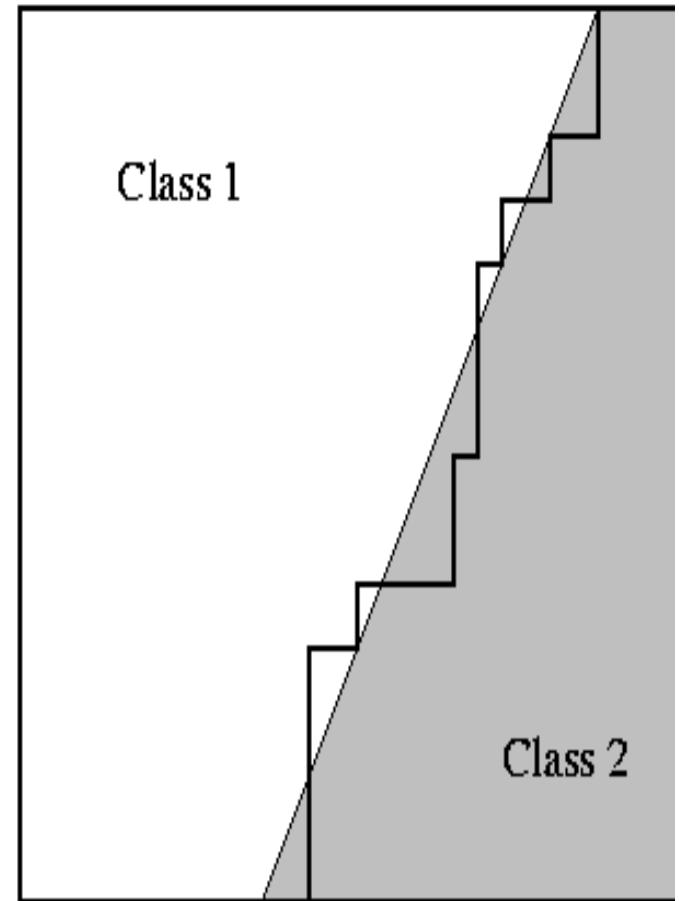
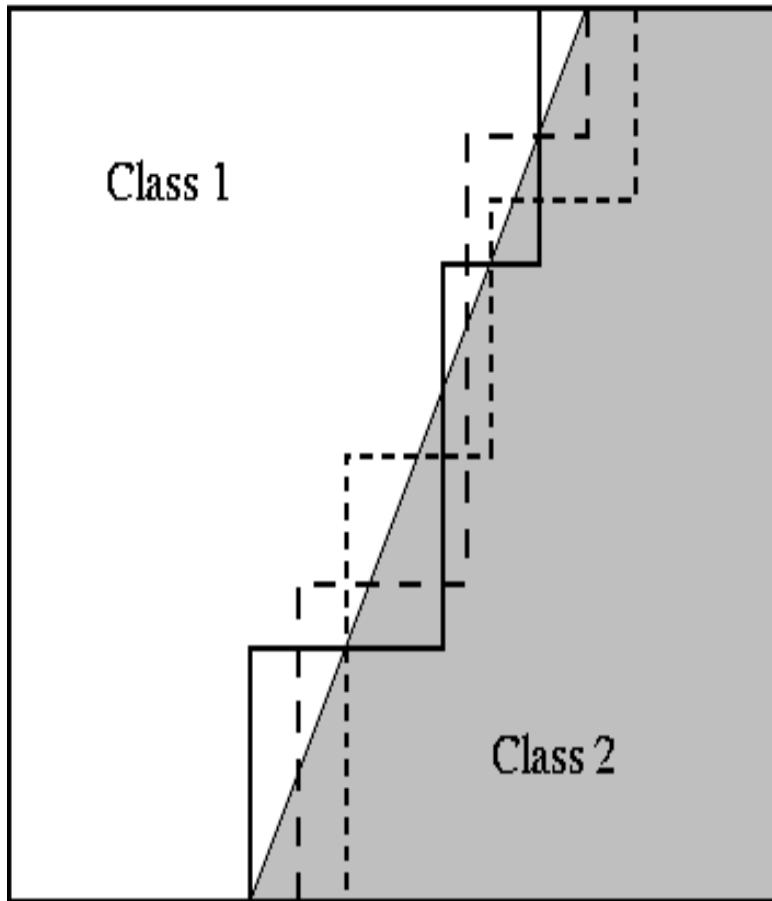
# Inna interpretacja poprawy zespół vs. pojedynczy model

T.Dietterich: statystyczna (dobór próby uczącej); wybór języka reprezentacji (rodzaju klasyfikatora); perturbacja parametrami uczenia.



**Fig. 2.** Three fundamental reasons why an ensemble may work better than a single classifier

# Zespół drzew klasyfikacyjnych vs. pojedyncze drzewo



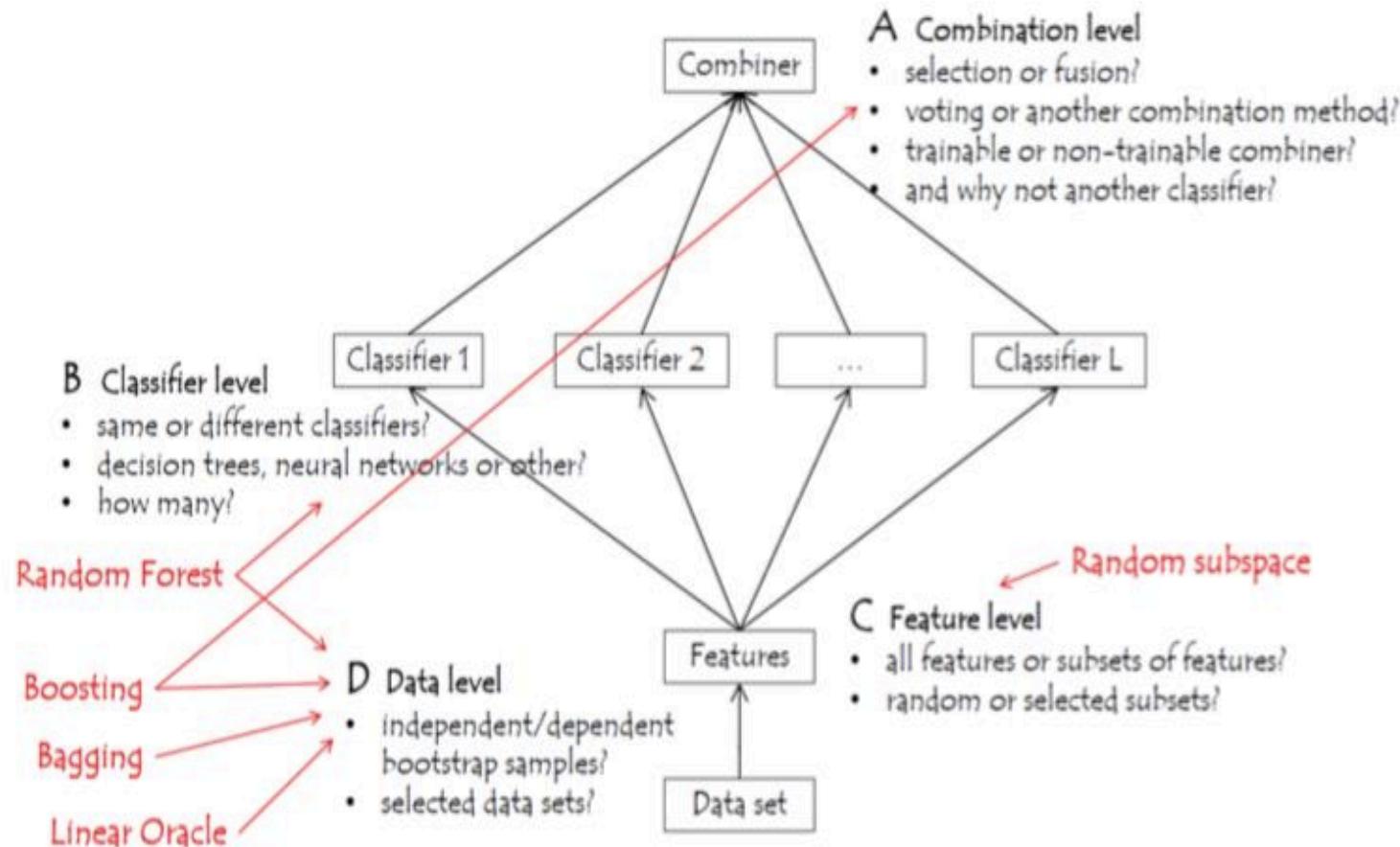
# Tworzenie zróżnicowanych klasyfikatorów składowych w zespołach

- Różne zbiory uczące (losowanie, wagowanie przykładów, inne podziały oryginalnego zbioru uczącego)
- Różne algorytmy uczące (zastosowane do tego samego zbioru uczącego)
- Wybór różnych podzbiorów atrybutów (częste dla tekstów, mowy, obrazów)
- Zmiany parametrów algorytmu uczącego (np. w ANN, stopień uproszczenia drzewa)
- Inne inicjalizacje algorytmów (stochastycznych)

# Typowa kategoryzacja algorytmów

- **Jednorodne modele (Homogeneous classifiers)** – użycie tego samego algorytmu na wielu zróżnicowanych zbiorach danych
  - Bagging (Breiman)
  - Boosting (Freund, Schapire)
  - Random Forest (Breiman)
  - Inne podziały zbioru (np. Ho, Lattine)
  - Specjalizowane dla wieloklasowości, (np.. ECOC pairwise classification)
- **Niejednorodne (Heterogeneous classifiers)** – jawnie inne algorytmy zastosowane do tego samego zbioru danych
  - Stacked generalization lub meta-learning
  - Rozwiązania dla innych złożonych danych

# Ogólne spojrzenie na tworzenie zespołów



# Jak agregować predykcje?

- Klasyfikacja
  - Głos (zerojedynkowy) lub tzw. współczynnik score / prawdopodobieństwa)
- Regresja
  - Uśrednianie predykcji liczbowych  $y_{maj}^- = \sum_{i=1}^L y_i$
  - Inne formy
- Czy wszystkie modele składowe biorą udział w wypracowaniu decyzji zespołu?

# Agregacje predykcji klasyfikatorów

## Głosowanie vs. inne podejścia

- Odmiany głosowania
  - Każdy model ma taki głos o tej samej wadze – decyzja końcowa największa liczba głosów na klasę
  - Głos każdego klasyfikatora ma wagę (suma wag albo interpretacja klasyfikacji Bayesowskiej)
- Non-voting - najczęściej dotyczy agregacji wskazań liczbowych (współczynniki score, prawdopodobieństwa klas)
  - Specjalne formuły (product, sum, min, max, median,...)
- Uczenie się b. złożonych agregacji – tzw. meta uczenie (extra meta-learner / combiner)

# Przykłady formuł dla wskazań liczbowych

Rule	Fusion function $f(\cdot)$			
Sum	$y_i = \frac{1}{L} \sum_{j=1}^L d_{ji}$			
Weighted sum	$y_i = \sum_j w_j d_{ji}, w_j \geq 0, \sum_j w_j = 1$			
Median	$y_i = \text{median}_j d_{ji}$			
Minimum	$y_i = \min_j d_{ji}$			
Maximum	$y_i = \max_j d_{ji}$			
Product	$y_i = \prod_j d_{ji}$			

	$C_1$	$C_2$	$C_3$
$d_1$	0.2	0.5	0.3
$d_2$	0.0	0.6	0.4
$d_3$	0.4	0.4	0.2
Sum	0.2	<b>0.5</b>	0.3
Median	0.2	<b>0.5</b>	0.4
Minimum	0.0	<b>0.4</b>	0.2
Maximum	0.4	<b>0.6</b>	0.4
Product	0.0	<b>0.12</b>	0.032

# Grupowe lub selektywne podejmowanie decyzji

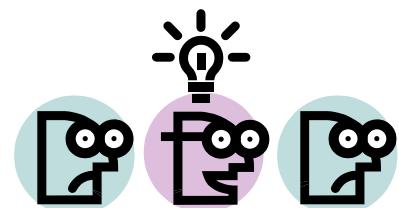
- **Grupowe** (statyczne) – wszystkie klasyfikatory składowe biorą udział w wypracowaniu decyzji końcowej zespołu.
- **Specjalizowany wybór / dynamiczna integracja** – wybór podzbioru “kompetentnych” klasyfikatorów
  - Poszukiwanie tych klasyfikatorów, które są ekspertami dla opisu klasyfikowanego przykładu

# Dynamiczne składanie głosów

Zamiast wyboru najbardziej kompetentnych klasyfikatorów

**Dynamic voting:** [propozycja A.Tsymbal]

- For każdy nowy obiekt do klasyfikacji:
  - Znajdź jego ***h-nearest neighbors*** w oryginalnym zbiorze uczącym
  - Reklasyfikuj je przez klasyfikatory składowe
  - Użyj informacji o skuteczności reklasyfikacji do oszacowania wag klasyfikatorów.

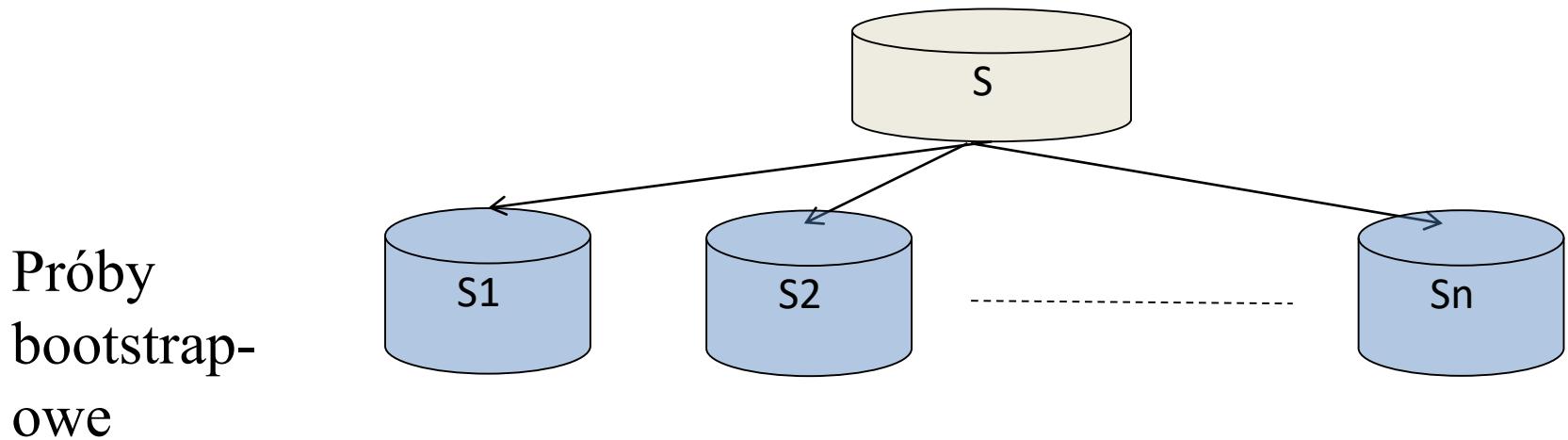


# Jednorodne klasyfikatory

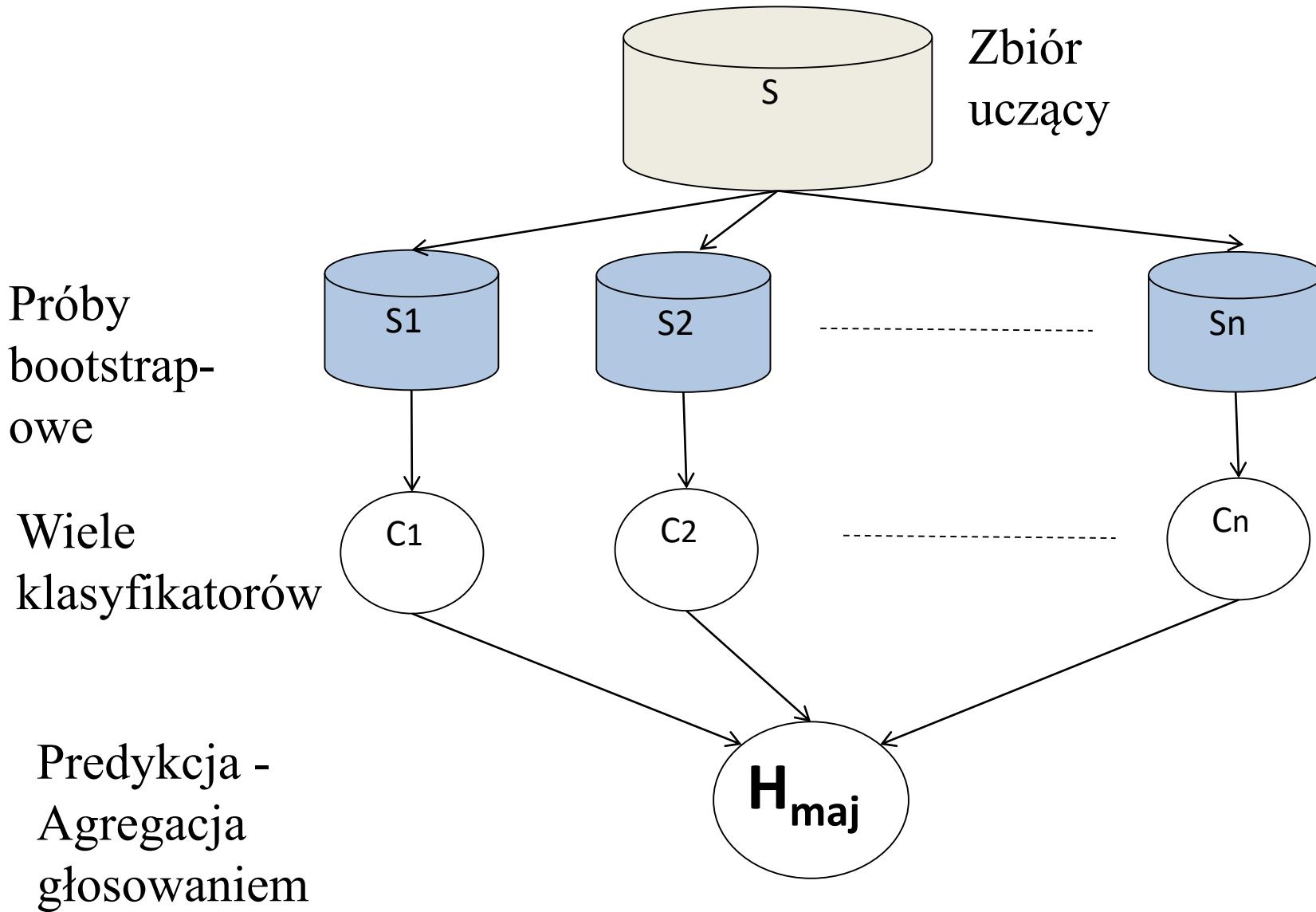
- Zróżnicowanie poprzez modyfikacje zbiorów uczących
  - **Data1 ≠ Data2 ≠ ... ≠ DataT**
- Użycie tego samego algorytmu uczącego
  - Różne dane -> różne klasyfikatory
- Najpopularniejsze propozycje
  - Bagging: losowanie bootstrapowe
  - Boosting: modyfikacja wag przykładów
  - Random Subspace (losowanie cech)
  - Random Forest

# Bagging [L.Breiman, 1996]

- Bagging = **Bootstrap aggregation**
- Wielokrotne losowanie różnych podzbiorów przykładów z początkowego zbioru
- Zastosowanie tego samego algorytmu uczącego
- Agregacja predykcji klasyfikatorów składowych
  - Klasyfikacja – różne formy głosowania
  - Regresja – uśrednianie odpowiedzi



# Schemat zespołu klasyfikatorów bagging



# Bootstrap aggregation

- Losowanie bootstrapowe – losowanie ze zwracaniem
  - Do danej próbki niektóre przykłady zostaną wylosowane kilka razy, a niektóre nie zostaną wylosowane
  - Przy wielkości próbki zbliżone do wielkości oryginalnego zbioru danych, średnio do próbki trafia 63.2% przykładów z tego zbioru
- Oszacowanie:
  - Dla danych o wielkości  $N$ , każdy przykład ma prawdopodobieństwo wylosowania przynajmniej raz równe  $1-(1-1/N)^N$
  - Dla wysokich  $N$  – dąży to do  $(1-1/e)$  or 0.632 [Bauer and Kohavi, 1999]
- Próbka bootstrapowa – wielkość zbliżona do oryginalnego zbioru albo może być mniejsza (np. pasting small votes lub inne uogólnienia bagging)

# Losowanie bootstrapowe

Początkowy zbiór przykładów

$$S = \{x_1, x_2, x_3, x_4\}$$

Wylosowane próbki do bagging

$$S_1 = \{x_1, x_2, x_3, x_3\}$$

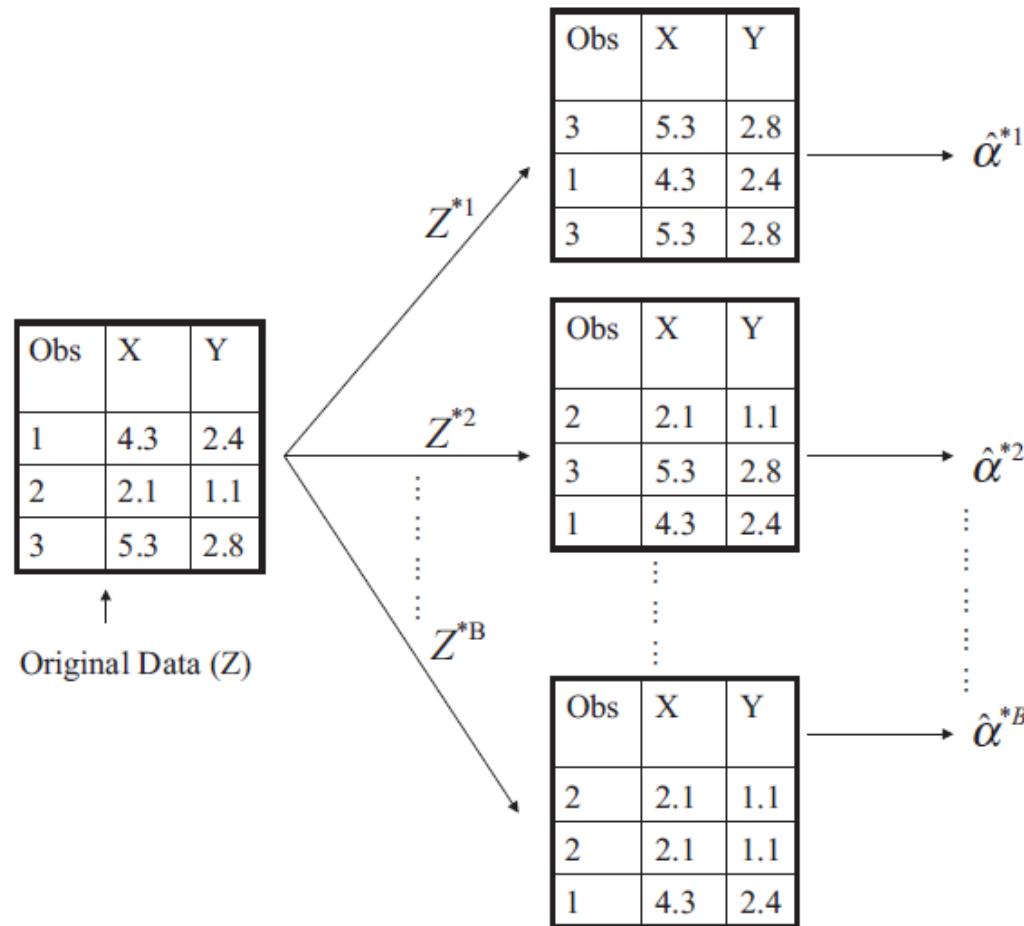
$$S_2 = \{x_1, x_4, x_4, x_4\}$$

$$S_3 = \{x_1, x_1, x_2, x_2\}$$

$$S_4 = \dots$$

# Losowanie bootstrap-owe

## Ilustracja



# Ogólny zapis metody bagging

## Uczenie

**input**  $S$  – zbiór uczący,  $T$  – # składowych model,  $LA$  – algorytm uczący

**output**  $E$ - zespół złożony z  $H_i$  składników

**for**  $i=1$  **to**  $T$  **do**

**begin**

$S_i :=$ bootstrap sample from  $S$ ;

$H_i := LA(S_i)$ ;

    add  $H_i$  to ensemble  $E$

**end;**

Predykcja – przykład  $x$

Klasyfikuj  $x$  przez każdy klasyfikator  $H_i$  – wskaznie etykiety klasy

Agreguj do  $D_j$  wskazania (oryginalnie sumuj głosy za każdą z klas)

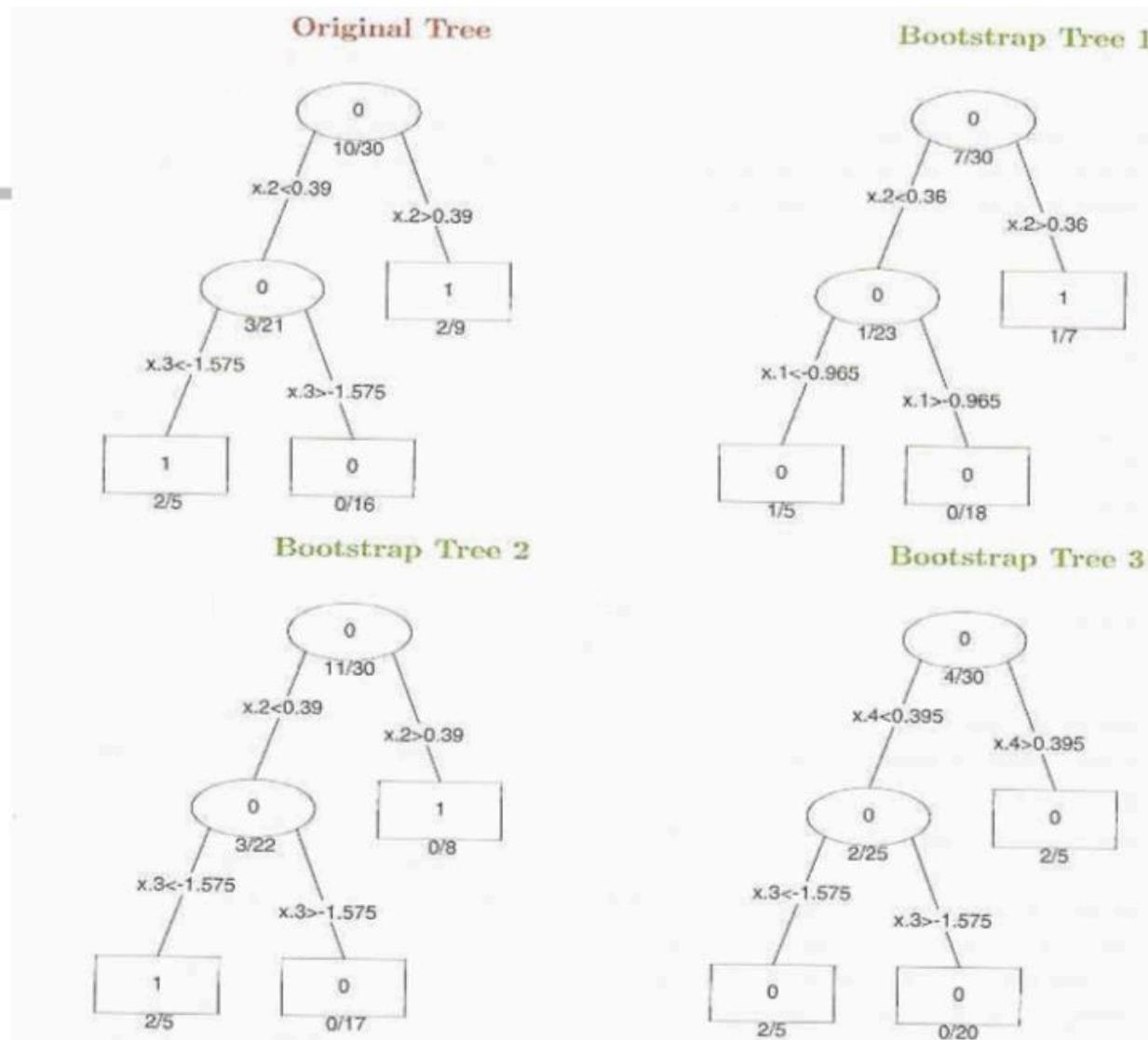
Wybierz klase maksymalizującą  $D_j$

# Przykład oceny eksperymentalnej

Misclassification error rates [Percent]

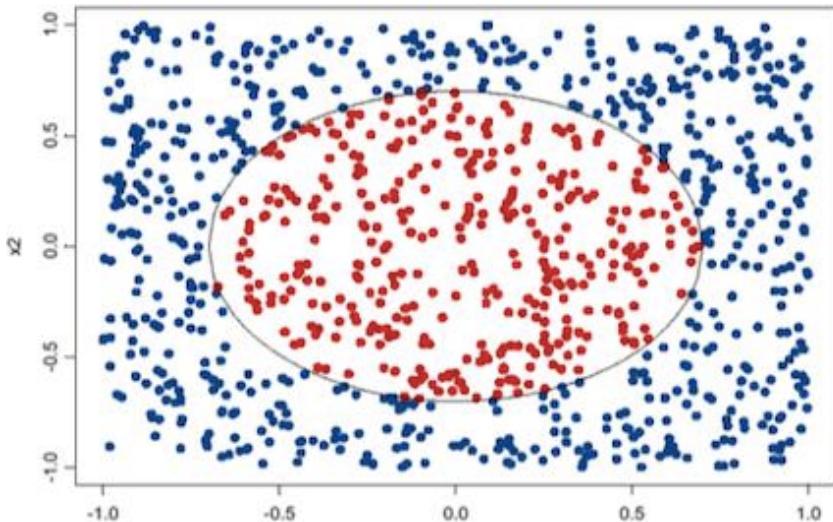
Data	Single	Bagging	Decrease
waveform	29.0	19.4	33%
heart	10.0	5.3	47%
breast cancer	6.0	4.2	30%
ionosphere	11.2	8.6	23%
diabetes	23.4	18.8	20%
glass	32.0	24.9	22%
soybean	14.5	10.6	27%

# Ilustracja perturbacji i różnych drzew



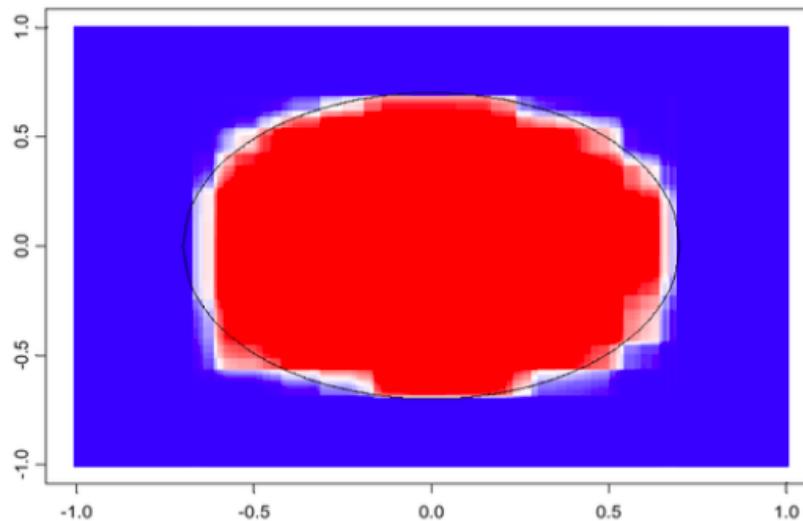
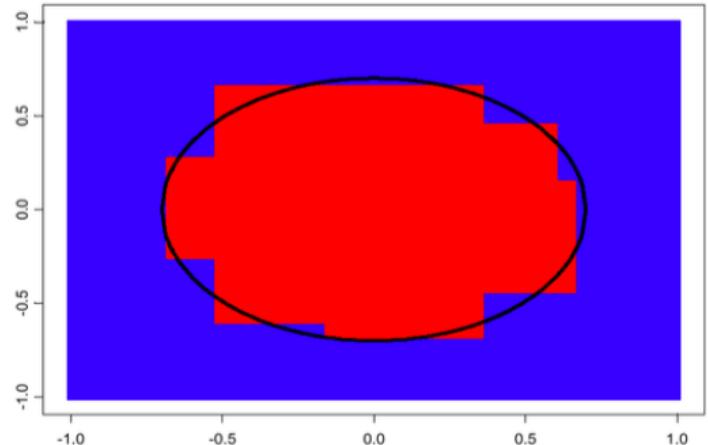
from Hastie, Tibshirani, Friedman: The Elements of Statistical Learning, Springer Verlag 2001

# Inne ilustracja graficzna agregacji bagging



100 bagged trees

CART decision boundary



# Bagging – dlaczego może działać?

- Liczne studia eksperymentalne (zwłaszcza dla drzew) – poprawa trafności
- Tzw. małe perturbacje w zbiorze uczącym + hipoteza o tzw. niestabilnym algorytmie generującym klasyfikatory
- Cytat z prac Breiman
  - This approach works well for **unstable algorithms**:
  - Whose major output classifier undergoes major changes in response to small changes in learning data.
- Typowe niestabilne algorytmy – drzew, decision stumps, reguly, liniowa regresja
- Inne spojrzenie na błąd – bias – variance decomposition
- Bagging naturalnie redukuje składnik wariancji

# Bias-variance decomposition

- Oczekiwany błąd predykcji klasyfikatora
  - Dwa składniki: bias + variance
  - “The *bias* of a classifier” oczekiwany element błędu wynikający z założeń algorytmu uczącego, które nie pasują do problemu
  - “The *variance* of a classifier” wynika z rozważania konkretnego zbioru przykładów, który może wpływać na działanie klasyfikatora
- Najczęściej przetarg pomiędzy nimi:
  - niski bias => wyższa variance
  - niska variance => wyższy bias

Bagging głównie redukuje wariancje proporcjonalnie do liczby składników T oraz stopnie nieskorelowanie ich predykcji

# Analiza teoretyczna zmian wariancji

## Probability detour - Variance reduction by averaging

Let  $z_b$ ,  $b = 1, \dots, B$  be identically distributed random variables with mean  $\mathbb{E}[z_b] = \mu$  and variance  $\text{Var}[\sigma^2]$ . Let  $\rho$  be the correlation between distinct variables.

Then,

$$\mathbb{E}\left[\frac{1}{B} \sum_{b=1}^B z_b\right] = \mu,$$

$$\text{Var}\left[\frac{1}{B} \sum_{b=1}^B z_b\right] = \underbrace{\frac{1-\rho}{B} \sigma^2}_{\text{small for large } B} + \rho \sigma^2.$$

The variance is reduced by averaging (if  $\rho < 1$ ) !

# Analiza w próbach bootstrapowych

## Bagging (I/II)

---

For now, assume that we have access to  $B$  **independent** datasets  $\mathcal{T}^1, \dots, \mathcal{T}^B$ . We can then train a separate deep tree  $\hat{y}^b(\mathbf{x})$  for each dataset,  $1, \dots, B$ .

- Each  $\hat{y}^b(\mathbf{x})$  has a **low bias** but **high variance**
- By averaging

$$\hat{y}_{\text{bag}}(\mathbf{x}) = \frac{1}{B} \sum_{b=1}^B \hat{y}^b(\mathbf{x})$$

the bias is kept small, but variance is reduced by a factor  $B$ !

# Analiza redukcji wariancji

Rozpatrzmy bagging regresyjny  $f_{bag} = \frac{1}{B} \sum_{i=1}^B f_i^*$  oparty na próbie  $D$  (predyktory składowe nieskorelowane)

Wtedy

$$E(f(x) - f^\wedge(x))^2 = \text{Var}(f^\wedge(x)) + (Ef^\wedge(x) - f(x))^2$$

$$E(f(x) - f_{bag}^\wedge(x))^2 = \text{Var}(f_{bag}^\wedge(x)) + (Ef_{bag}^\wedge(x) - f(x))^2$$

$$\text{Var}(f_{bag}^\wedge(x)) = \frac{1}{B} \text{Var}(f^\wedge(x))$$

Wariancja zespołu (bagging) będzie  $B$  razy mniejsza niż wariancja pojedynczego predyktora (klasyfikatora)

# Analiza wariancji baggingu

W rzeczywistości próby bootstrapowe są zależne więc zyskujemy mniej na uśrednianiu

$$\begin{aligned}\text{Var}(\hat{f}_{bag}(x)) &= \frac{1}{B} \text{Var}(\hat{f}(x)) + \frac{B(B-1)}{B^2} \text{Cov}(f_i f_j) = \\ &= \frac{1-\rho}{B} \sigma^2 + \rho \sigma^2 < \sigma^2\end{aligned}$$

# Drzewa regresji i redukcja wariancji

- Dla  $T$  składników naucz drzewa regresji
- Uśrednij wyniki predykcji
- Jeśli tworzymy niezredukowane (unpruned) drzewa, to będą się charakteryzować większą wariancją i mniejszym obciążeniem (bias)
- Połączenie drzew w zespół bagging – zredukuje wariancje i częściowo bias

# Bagging – decyzja zespołu

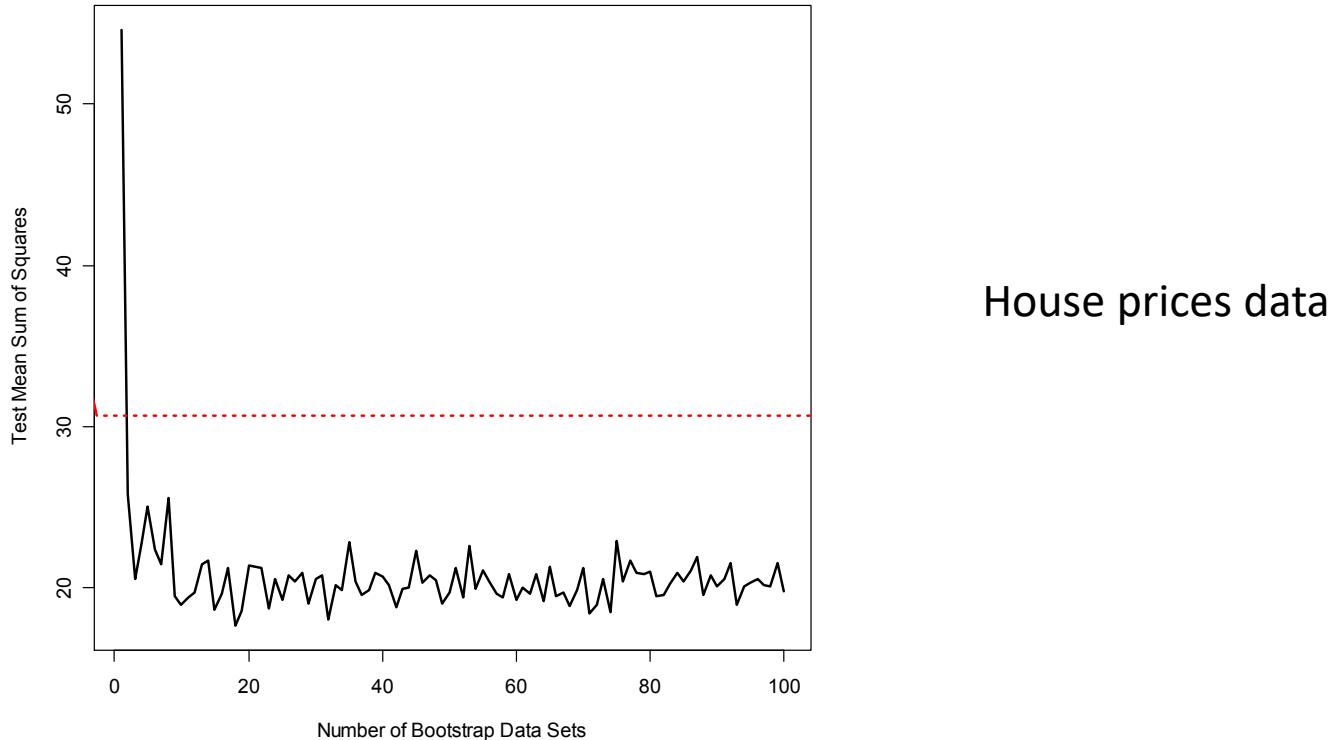
Wskazanie deterministyczne vs. prawdopodobieństwa

- Inne operatory agregacji niż max suma predykcji
- Standard – głosowanie z równymi wagami
- Głosowanie większościowe => każdy z klasyfikatorów maną wagę podczas agregacji “głosów” / predykcji
- Lecz jak ocenić wagę klasyfikatora składowego / jego kompetencje?
  - Globalnie, statycznie – oceń jego zdolności predykcyjne
  - Lecz, oszacowanie wymaga zbioru walidującego; czy jest inna alternatywa?
  - **Out-of-bag (OOB) estimate** (2/3 przykładów wylosowano do próbki bootstrapowej, lecz 1/3 pozostaje na zewnątrz.

# Liczba składowych modeli

Czy redukcja oczekiwanej błędu zmienia się wraz ze wzrostem składników w bagging?

- Im więcej, tym lepiej ? Nie aż tak bardzo – Breiman wskazywał, że dla większości jego zbiorów danych 20-50 drzew wystarczało
- Trochę związane z wielkością i charakterystyką danych oraz miarami oceny



# Interpretowalność zespołu

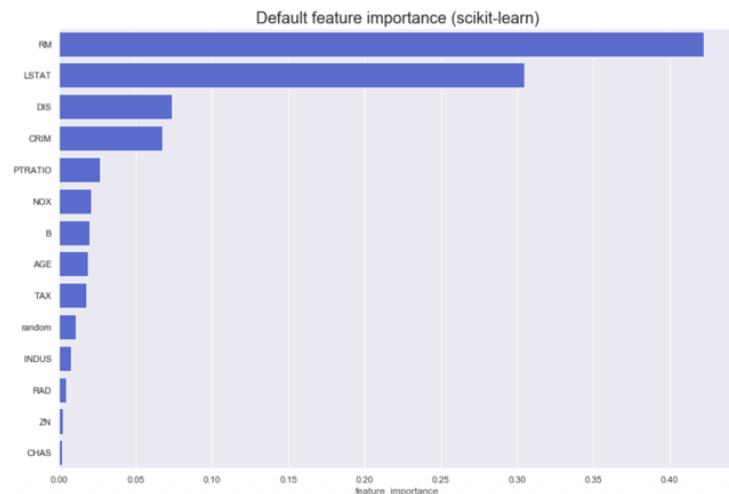
Bagging obejmuje wiele różnych drzew

- Pytanie – czy są dostatecznie zróżnicowane
- Lecz równocześnie tracimy interpretowalność która była osiągalna dla pojedynczego drzewa

Jak to rozwiązać? = pomysł L.Breimana / analiza feature importance

analizy struktury drzewa -> Oceń dla warunku redukcję miary (impurity) oraz wagę – liczbę przykładów w węźle

Permutations – “noised-up” method



# Feature-Selection Ensembles

*Niektóre dane/zadania* – zbyt wiele cech

**Pomysł:** Użyj innego zbioru cech dla każdego z algorytmów (częste w przetwarzaniu tekstów, obrazów, sygnałów,...)

**Przykład:** Venus&Cherkauer (1996) zespół 32 sieci ANN, każda na innym podzbiorze cech – co doprowadziło do poprawy trafności

**Propozycja:** Random Subspace Methods autorstwa Ho.

Dla każdego klasyfikatora składowego – losowo wybierz podzbiór atrybutów nie zmieniając wyboru przykładów

Ho proponowała :  $m = 50\%$  losowo wybranych atrybutów

Lattine (pozniej Stefanowski) – połączenie bootstrap sampling z losowym wyborem cech

# Random forests [Breiman 2001]

**Motywacja:** Oprócz radzenia sobie z wysoką wymiarowością cech, dodatkowo zdekorelować / zróżnicować klasyfikatory składowe

Resampling przykładów jest niewystarczający

**Pomysł:** Dodatkowa perturbacja w tworzeniu drzewa

- Wykorzystaj losowanie bootstrapowe przykładów
- W każdym z węzłów drzewa losowo wybierz podzbior  $m$  cech z oryginalnie  $q$  cech i znajdź warunek podziału z wykorzystaniem kryterium Gini index lub entropii
  - Breiman proponuje  $m = \text{sqrt}(q)$  dla drzew klasyfikacyjnych i  $m = q/3$  drzew regresji
- Predykcje drzew agregowane tak jak w bagging

# Intuicja losowego wyboru cech

- Założmy, że wśród  $q$  cech jest wyjątkowo silny predyktor wyjścia  $y$  oraz ew. Inne silne cechy
- Wtedy każde drzewo będzie używało tego predyktora w korzeniu drzева, a inne na wysokich poziomach
- Drzewa będą zbyt podobne i skorelowane
- Agregacja, uśrednianie zbyt podobnych drzew nie zredukuje wariancji i nie poprawi predykcji
- Losowania podzbiorów cech – zapobiega powyższym ograniczeniom

# Random forest pseudo-code

**input**  $S$  – learning set ( $n$ ),  $T$  – no. of bootstrap samples,  $LA$  – learning algorithm

**output**  $C^*$  - multiple classifier

**for**  $i=1$  to  $T$  **do**

**begin**

$S_i :=$ bootstrap sample from  $S$  –  $n$  examples ;

$C_i :=$  learn tree from  $S_i$  with extra conditions

    for each node

        In each node select  $m$  out of the  $q$  input attributes uniformly at random

        Choose the best split test among  $m$  attributes and split tree

    until a stopping condition (may be max depth)

**end;**

$$C^*(x) = \operatorname{argmax}_y \sum_{i=1}^T (C_i(x) = y)$$

Lub średnia predykcji dla wersji drzew regresji

# Oceny eksperymentalne

- Praca Breiman, L., Random forests, Machine Learning, 2001, vol 45, 5-32
- Random forest często trafniejszy niż podstawowy bootstrap bagging i konkurencyjny do Adaboost
- Jest mniej podatny na przeuczenie wobec trudnych danych

# Random forests

Podatność na przeuczenie

- Czy wzrost liczby drzew składowych nie wpływa na nadmierne dostosowanie się do specyfiki zbiorów uczących?
- NIE!

Łatwa implementacja, równoległa oraz przydatna dla analizy “Big data”

Można także wylosowywać mniej przykładów do próby bootstrapowej

Cytat, We could bootstrap fewer than  $q$  features, say  $\text{sqrt}(q)$  useful for “big data” problems.

Interpretowalność – tak jak bagging

Bardzo dogodne dla analizy wysoko-wymiarowych danych

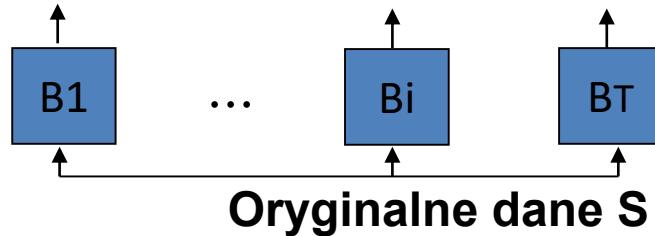
... Liczne zastosowania (być może najpopularniejszy zespół klasyfikatorów)

# Uwagi nt. rozszerzeń

- Zarówno podstawowy bagging jak i random forest są często rozszerzane dla innych problemów (schemat jest elastyczny), np.
  - Online bagging dla uczenia przyrostowego i dalsze modyfikacja dla klasyfikacji zmiennych strumieni danych
  - Podstawy typ zespołu modyfikowany dla niebalansowanych danych
  - Równoległe implementacje dla Big Data
  - Metody specjalnych perturbacji do odkrywania nieskorelowanych, znaczących cech dla danych wysoce wielowymiarowych (np. bioinformatyczne eksperymenty)

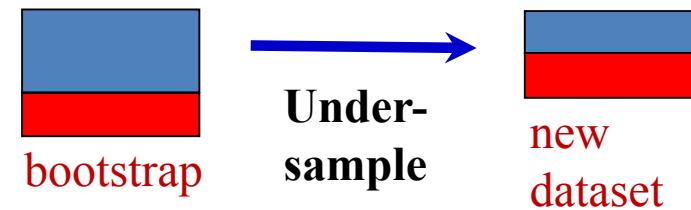
# Under-Bagging – dane niezbalansowane

- Standardowy Bagging → wykorzystuje bootstraps
  - Losowanie przykładów ze zwracaniem
  - Nie rozwiązuje obciążenia w stronę klasy większościowej



## Propozycje z Undersampling

- Exactly Balanced Bagging [Ch03]
  - Bootstraps = przekopij wszystkie przykłady mniejszościowe + wylosuj podobną liczbę przykładów większościowych ( $N_{maj} = N_{min}$ )
- Rough Balanced Bagging [Hido 09]
  - Inaczej wyrównuje prawdopodobieństwa klas w losowaniu do próbek bootstrapowych



# Roughly Balanced Bagging

Hido S., Kashima H.: Roughly balanced bagging for imbalance data (2008)

## Modyfikacja losowania

- Under-sampling - zmniejszanie liczności klas większościowej
- Zamiast ustawienia sztywnych liczności klas jak w EBB, wyrównać prawdopodobieństwa losowania w klasach - czyli na poziomie rozkładu prawdopodobieństwa
- Dla każdej  $T$  iteracji licznosć klasy większościowej w próbie bootstrapowej  $Bs_{maj}$  jest zmienną losową określona wg. negatywnego rozkładu dwumianowego

For each bootstrap

- Random size  $BS_{maj}$
- Wylosuj ze zwracaniem  $N_{min}$  oraz  $BS_{maj}$

Predykcja - odmiany losowania większościowego

- **Przykładowe rozszerzenia:**
  - Attribute Selection with RBBag dla wysoko wymiarowych danych
  - Multi-class generalization (zmiana rozkładów prawdopodobieństwa)

Lango M., Stefanowski J.: The Usefulness of Roughly Balanced Bagging for Complex and High-dimensional Imbalanced Data (2016)

# Porównanie wielu zespołów klasyfikatorów

Studia eksperymentalne

Galar, Herrera et al [2011]

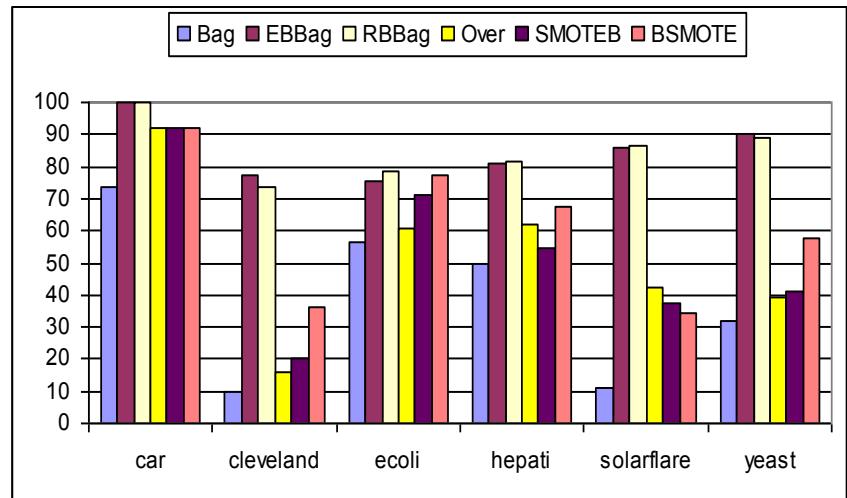
- Bagging działa lepiej niż inne zespoły w tym tzw. cost based

Khoshgoftaar et al. [2011]

- EBBag, RBBag lepsze niż SMOTEBoost and RUBoost

Własne studium [2013]

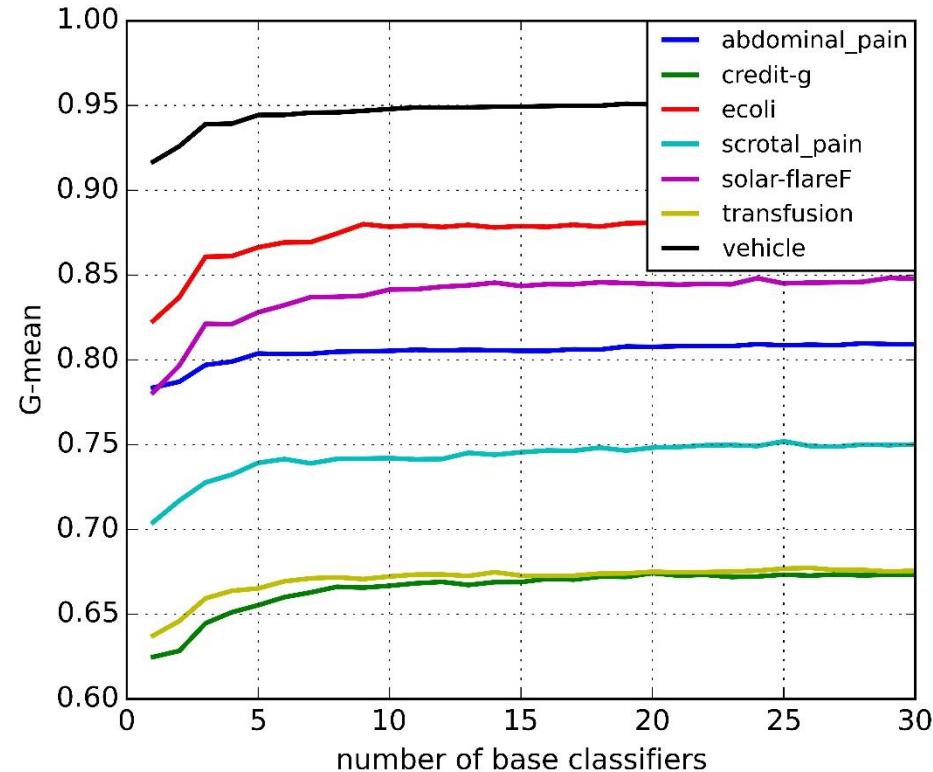
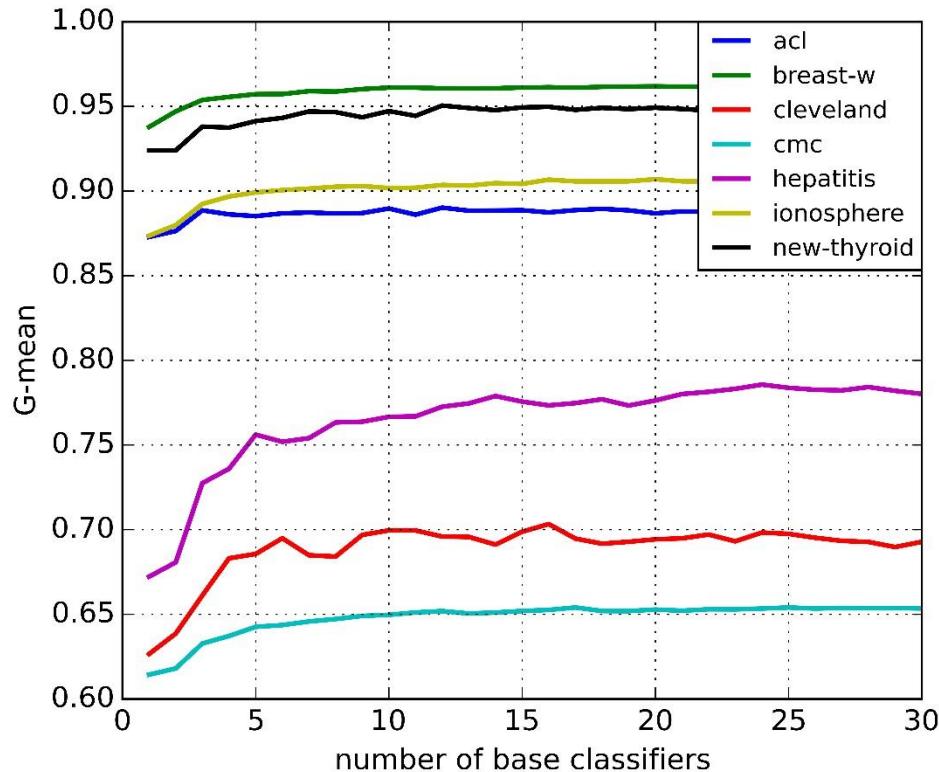
- **RBBag**  $\approx$  **EBBag** > **OverBag** > **SMOTEBag** > **Bagging**



Dataset	Bag	EBBag	RBBag	OvBag	SmBag	BagSm
breast-w	95.88	96.03	96.37	96.23	95.88	96.77
abdominal-pain	78.95	80.65	80.35	79.44	80.85	79.86
acl	88.18	90.71	89.35	88.35	88.64	87.81
new-thyroid	92.41	96.91	96.58	95.36	95.18	92.89
vehicle	93.91	94.58	95.44	94.61	94.34	94.20
car	84.53	96.73	96.58	95.29	95.26	95.18
scrotal-pain	70.75	73.18	75.65	72.01	70.42	70.68
ionosphere	88.96	90.44	90.67	90.47	90.30	90.26
pima	71.54	74.22	75.64	73.54	72.33	71.38
credit-g	63.98	65.82	67.82	71.75	80.68	66.11
ecoli	68.67	72.24	88.85	51.42	58.38	80.11
hepatitis	62.81	78.93	78.66	72.16	68.47	74.29
haberman	43.11	65.41	63.43	58.11	60.02	62.82
breast-cancer	54.30	58.82	59.37	56.17	52.57	57.25
cmc	52.76	64.61	65.27	59.95	57.74	62.77
cleveland	12.61	72.32	71.02	22.77	25.03	50.96
hsv	0.00	36.27	35.74	2.84	5.37	16.61
abalone	49.58	78.93	79.32	61.95	63.67	69.65
postoperative	1.99	24.97	34.03	15.01	1.57	11.55
solar-flare	13.70	85.39	83.21	58.07	55.04	54.40
transfusion	55.72	66.75	67.32	64.83	63.96	65.76
yeast	51.48	84.55	84.68	59.70	59.41	57.94
balance-scale	0.00	59.07	54.23	1.40	0.00	0.67
average rank	5.61	1.96	1.61	3.65	4.26	3.91

J. Blaszczyński, J., Stefanowski: Extending bagging for imbalanced data. Proc. CORES 2013.

# RBBag (liczba drzew decyzyjnych)



Relatywnie mała:

- Dla większości danych wystarczy kilkanaście

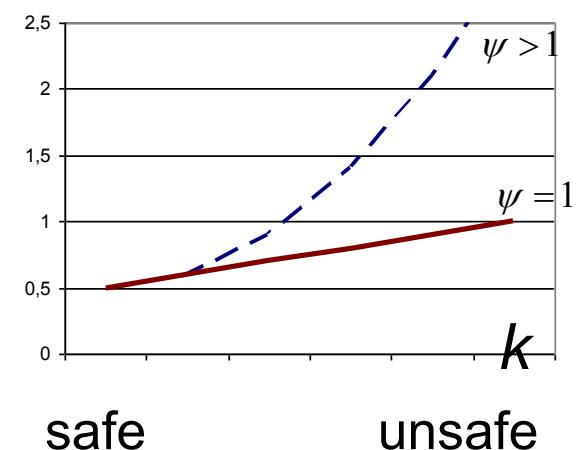


# Neighbourhood Balanced Bagging

- Propozycja wykorzystujące inne zasady:
  - Zmodyfikuj prawdopodobieństwo losowania do próbki bootstrapowej z wykorzystaniem “safe level” przykładu
  - Zwiększ szanse wyboru przykładów mniejszościowych kosztem większościowych (global prob.)
- Poziom globalny
  - $p_{\min}^1 = 1$  (mniejszościowa)
  - $p_{maj}^1 = N_{\min} / N_{maj}$  (decrease → inverse global imbalance)
- Lokalny
  - **Minority local neighb.**  $\psi \geq 1$
- Eksperymenty - porównywalny do RBBag, lepszy dla b. trudnych danych

$$L = \frac{(N'_{maj})^\psi}{k}$$

$$P_{global} \cdot P_{local}$$



# Odnośniki do literatury

- Intensywny rozwój od lat 90 poprzedniego wieku
- Wiele różnych propozycji
- Przykładowe pozycje:
  - L.Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, 2004 (large review + list of bibliography).
  - T.Dietterich, Ensemble methods in machine learning, 2000.
  - J.Gama, Combining classification algorithms, 1999.
  - G.Valentini, F.Masulli, Ensemble of learning machines, 2001 [obszerna lista referencyjna]
  - R.Polikar, Ensemble based systems in decision making, IEEE Circuits and Systems Magazine, vol. 6, no. 3, pp. 21–45, 2006.
  - See also many papers by L.Breiman, J.Friedman, Y.Freund, R.Schapire, T.Hastie, R.Tibshirani,
  - W Polsce – przykładowo prace M.Woźniak i współpracownicy

# Pytanie i komentarze?

Dalszy kontakt:

[jerzy.stefanowski@cs.put.poznan.pl](mailto:jerzy.stefanowski@cs.put.poznan.pl)

<http://www.cs.put.poznan.pl/jstefanowski/>



Fundusze  
Europejskie  
Polska Cyfrowa



Rzeczpospolita  
Polska

Unia Europejska  
Europejski Fundusz  
Rozwoju Regionalnego



# Zespoły modeli predykcyjnych boosting i inne wykład 9

Jerzy Stefanowski

Instytut Informatyki PP

2021

Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI-TECH)  
projekt finansowany z środków Programu Operacyjnego Polska Cyfrowa  
POPC.03.02.00-00-0001/20



Fundusze  
Europejskie  
Polska Cyfrowa



Rzeczpospolita  
Polska

Unia Europejska  
Europejski Fundusz  
Rozwoju Regionalnego



# Plan wykładu 9 i częściowo 10tego

- Metody Boosting
  - AdaBoost
- Porównania Bagging vs. Boosting
- Boosted trees ensembles (funkcyjny boosting)  
----- kolejny wykład
- Zróżnicowanie klasyfikatorów składowych
- Generalizacja stosowa (stacking) i tzw. mixture of experts
- Podejścia zespołowe do danych silnie wieloklasowych
- Podsumowanie

# Boosting

- Schapire [1990] rozważanie teoretyczne, jak można tzw. słabe modele (ang. weak learner) rozbudować do predyktorów o lepszej trafności (ang. strong learner)
  - **Weak learner** – algorytm tworzenia klasyfikatora o trafności trochę lepszej niż głosowanie większościowe ( $>0.5$  dla klasifikacji binarnej)
- Wprowadził ideę tzw. **wzmacniania klasyfikatora** (ang. boosting) poprzez odpowiednio ukierunkowane losowanie przykładów do zbiorów uczących dodatkowych klasyfikatorów
- Później [Freund & Shapire, 1996] rozwinięte do praktycznego i efektywnego algorytmu AdaBoost
  - Wagowanie przykładów – może być realizowane przez zmianę prawdopodobieństwa wylosowania
- Liczne rozszerzenia i zastosowania

# Boosting - inspiracje

- Schapire [1990] weak learners – proponuje użyć/uczyć trzy klasyfikatory C1,C2 lub C3 na podzbiorach z oryginalnych danych uczących  $S$ 
  - Pierwszy C1 uczyony na wylosowanym podzbiorze  $S_1$  z  $S$
  - Stwórz kolejny podzbiór  $S_2$  z  $S$  złożony w połowie z przykładów poprawnie sklasyfikowanych przez C1, a w połowie źle sklasyfikowanych (te przykłady odpowiednio losowane z  $S$ )
  - Naucz nowy klasyfikator C2 na  $S_2$
  - Trzeci klasyfikator C3 uczyony na tych przykładach z  $S$ , na których predykcje C1 i C2 są niezgodne
- Faza klasyfikacji nowego  $x$ 
  - Użyj klasyfikatorów C1 i C2, jeśli predykcje zgodne, to wynik
  - Jeśli ich predykcje są niezgodne, to użyj C3

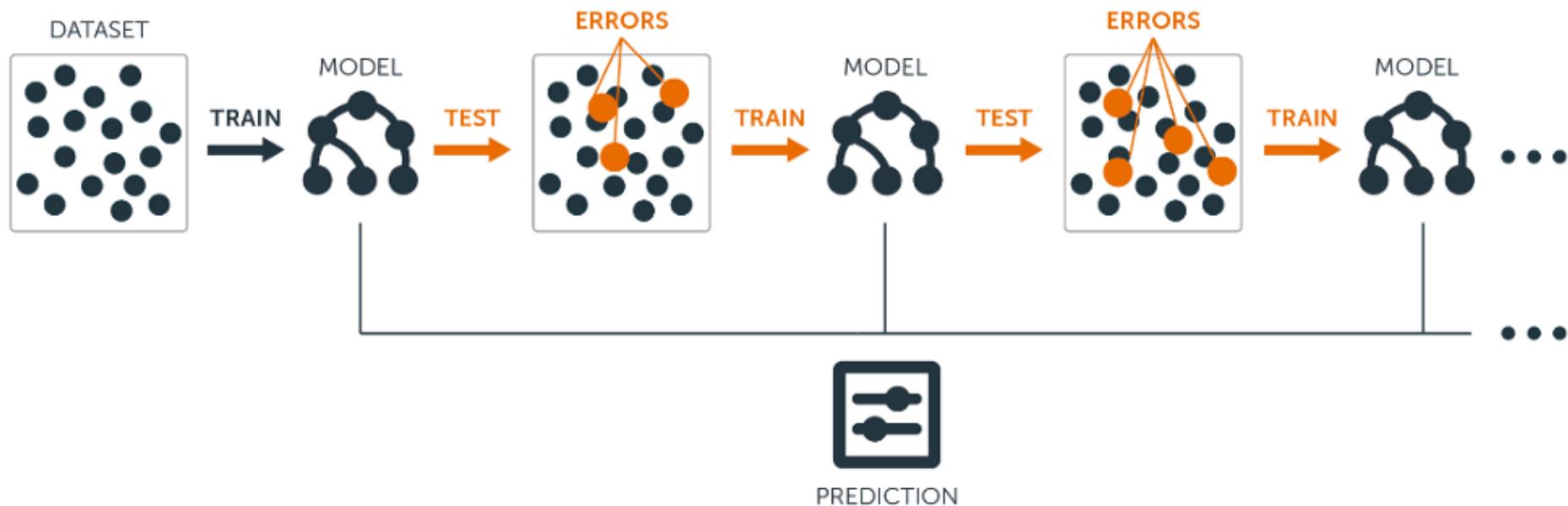
# Pierwszy boosting

- Schapire – podał dowód, że dla binarnej klasyfikacji, błąd takiego złożonego klasyfikatora jest z góry ograniczony przez połowę błędu najlepszego z klasyfikatorów składowych C1, C2, C3
- Należy oczekiwąć poprawy trafności całego klasyfikatora złożonego
- Możliwe tworzenie innych sekwencyjnych / iteracyjnych rozwiązań -> ...

# Boosting - AdaBoost

- (Freund & Shapire, 1996) wprowadzili algorytm AdaBoost jako uogólnienie wzmacnia poprzez **iteracyjne wagowanie przykładów**
- Podejście iteracyjne – **sekwencyjne** dodawanie kolejnego klasyfikatora do zespołu.
- W każdej iteracji zmiana rozkładu wag przykładów w S
- Kolejne klasyfikatory w sekwencji,  $C_t, C_{t+1}$  uczone ze zmodyfikowanego zbioru tak, aby skupiać “zainteresowanie” na przykładach poprzednio źle klasyfikowanych (podniesienie wag źle sklasyfikowanym przykładom, obniżenie wag dobrze sklasyfikowanym)
  - Kolejne składowe klasyfikatory to eksperci od trudnych przykładów
- Predykcja całego systemu – głosowanie ważone / predykcja składowego klasyfikatora z nieliniową funkcją zależną od błędu w fazie uczenia

# Ilustracja graficzna



# Schemat uczenia AdaBoost

Input: dane oryginalne  $S$  ( $n$  przykładów), Algorytm uczący  $L$ , liczba iteracji  $T$  (klasyfikatorów składowych  $C_i$ )

Inicjalizacja zbioru  $D_i = \text{przypisz wagi przykładom } w_p = 1/n$

For  $i = 1, \dots, T$  do

1. Naucz klasyfikator  $C_i$  na zbiorze  $D_i$
2. Oblicz błąd  $e_i$  na zbiorze  $D_i$
3. Jeśli  $e_i > \text{próg}$  (0.5 dla binarnej klas), to przerwij
4. Oblicz  $\beta_i = e_i / (1 - e_i)$
5. Zmodyfikuj wagi każdego przykładu dobrze sklasyfikowanemu  $w_p = w_p * \beta_i$ , a źle sklasyfikowanemu przemnóż przez 1
6. Znormalizuj nowe wagi aby ich suma była 1 ( $w_p / \sum w$ ) -> utwórz nowy zbiór  $D_{i+1}$
7. Przejdź do punktu 1

Proces powtarzany do wyczerpania liczby iteracji lub warunku stop  $e_i$

# AdaBoost klasyfikowanie

Z każdym klasyfikatorem składowym  $C_i$  jest związany współczynnik  $\beta_i = e_i / (1 - e_i)$

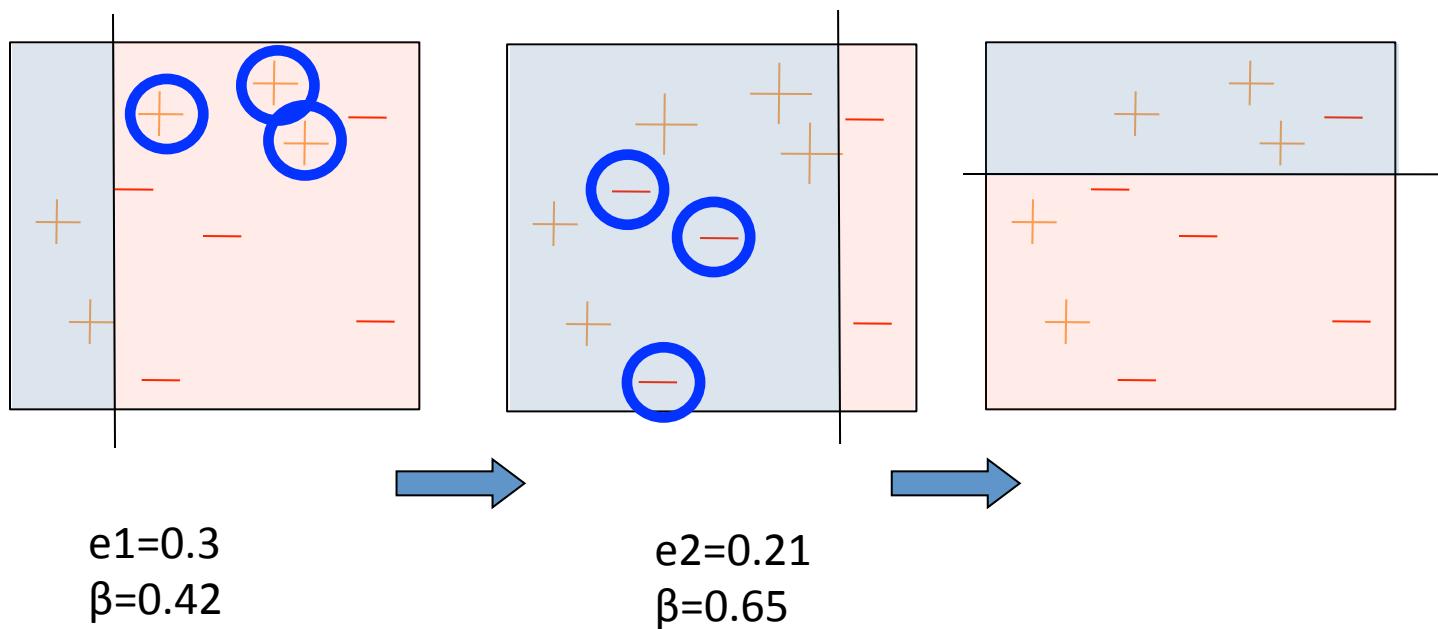
Klasyfikowanie nowego przykładu  $x$

1. Oblicz predykcje klasy  $x$  wg każdego klasyfikatora  $C_i$
2. Oblicz sumę wskazań dla każdej z klas  $K_j$

$$V_j = \sum_{\{i: C_i = k_j\}} \log(1/\beta_i)$$

3. Wybierz klasę z maksymalną wartością  $V_j$

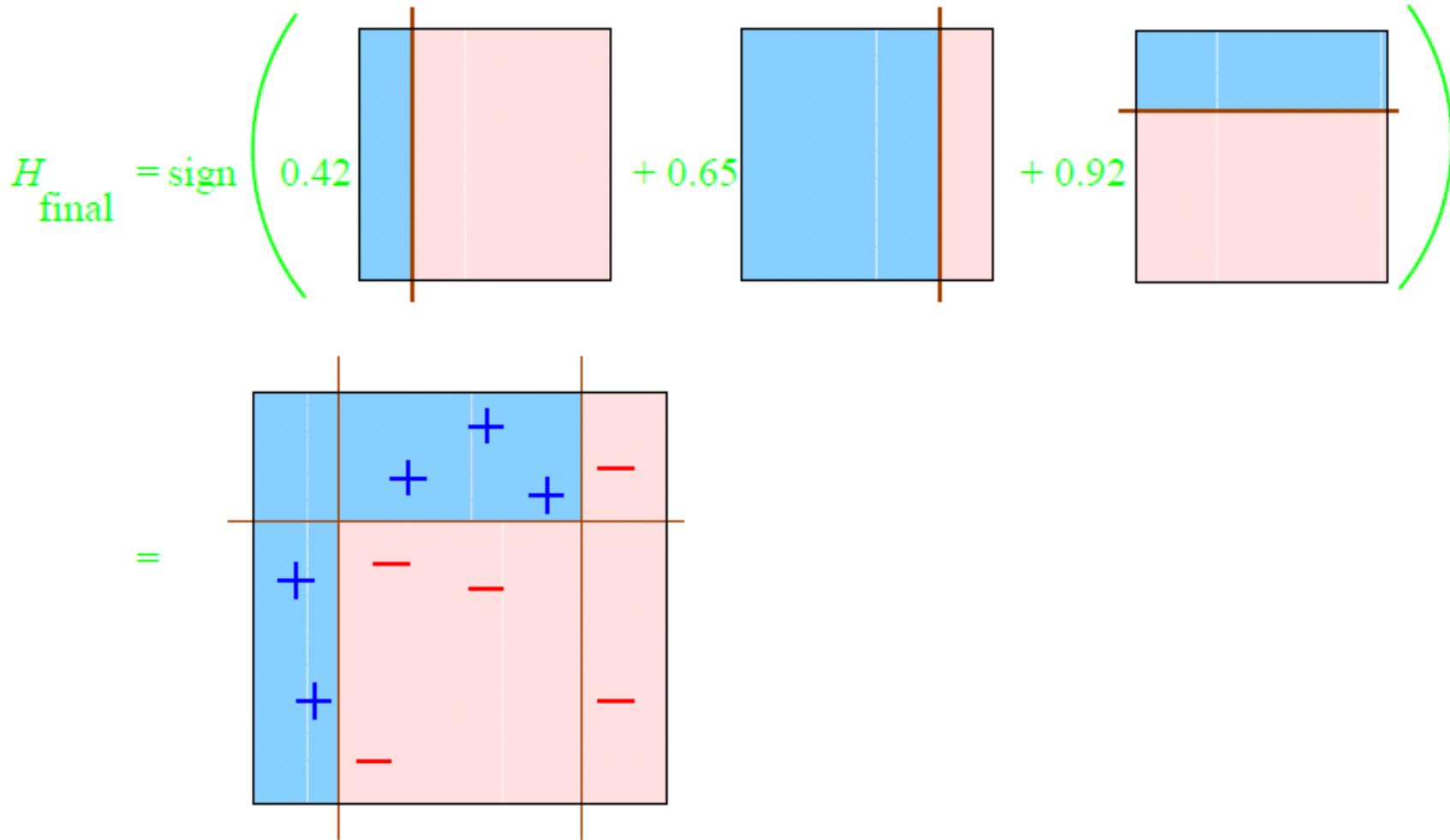
# Przykład ilustracyjny wzmacniania



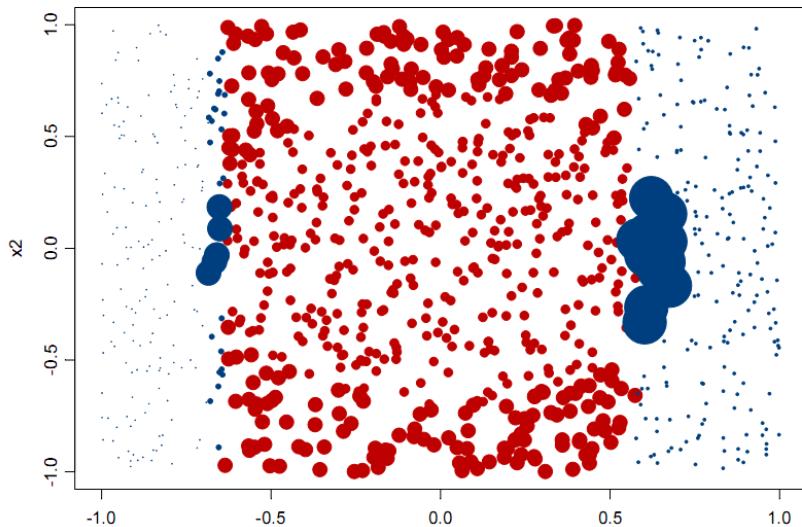
Kolejne iteracje uczenia klasyfikatorów składowych w AdaBoost

= za tutorialem “A Tutorial on Boosting” by Yoav Freund and Rob Schapire

# Końcowy zespół klasyfikatorów

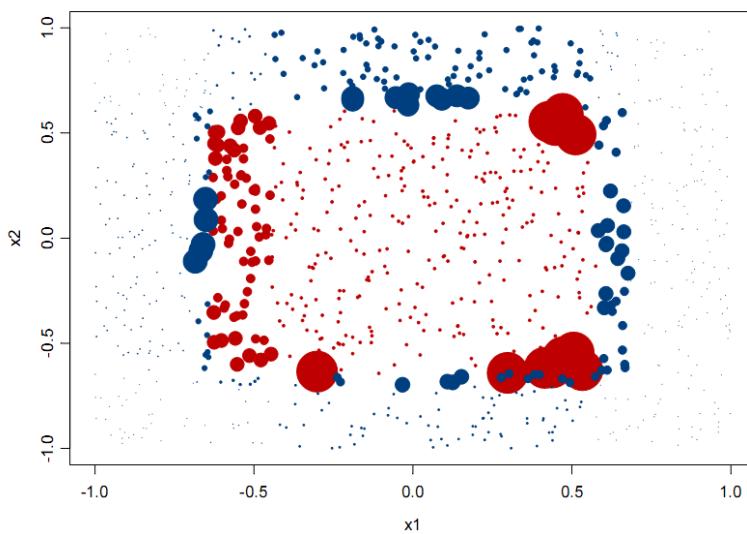


za tutorialem “A Tutorial on Boosting” by Yoav Freund and Rob Schapire

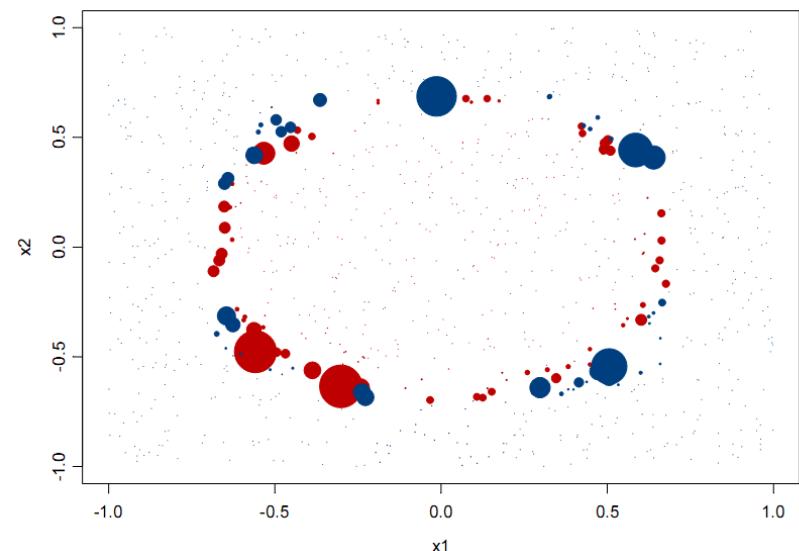


Klasy (kolor) i waga (wielkość) przykładów po 1 iteracji AdaBoost

**3 iteracje**



**20 iteracje**



Z wykładu Elder, John. From Trees to  
Forests and Rule Sets - A Unified  
Overview of Ensemble Methods. 2007.

# Analiza teoretyczna

Freund i Schapire – oszacowanie błędu AdaBoost

$$E < 2^T \prod_{t=1}^T \sqrt{e_t(1-e_t)}$$

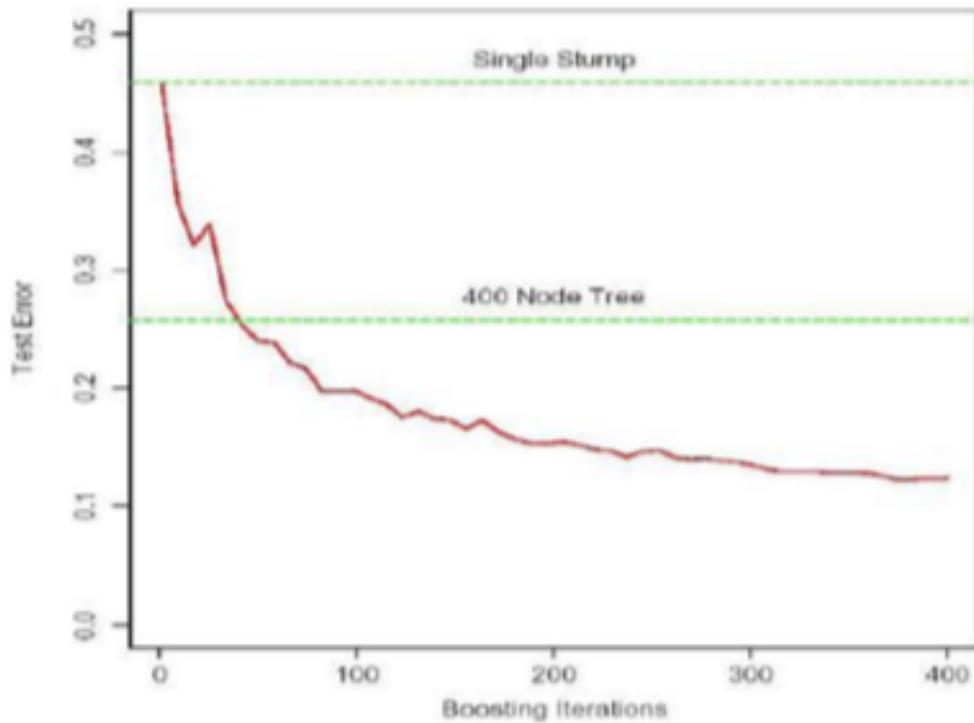
gdzie oczekuję się, że błąd klasyfikatora  $e_t < 0.5$

Przy pewnych założeniach, błąd powinien maleć wraz ze zwiększeniem liczby składowych klasyfikatorów  $T$ , lecz nie zawsze!  
– istnieje w praktyce możliwość przeuczenia

# Przykład budowy dużego zespołu

---

boosting of decision stumps on simulated data



from Hastie, Tibshirani, Friedman: The Elements of Statistical Learning, Springer Verlag 2001

---

# Interpretacje działania boosting

- Spojrzenie z punktu widzenia statystycznej teorii uczenia się – sprawdź Tibishirani, Hastie, Friedman: The Elements of Statistical Learning (pdf wersji autorskiej dostępny, np. <https://web.stanford.edu/~hastie/Papers/ESLII.pdf>)
- Prace samych autorów = patrz np. R.Schapire Theoretical Views of Boosting.
- Ciekawa własność wzmacniania i sposobów wypracowania decyzji – tzw. **minimalizacja marginesu**, patrz np. <https://www.cc.gatech.edu/~isbell/tutorials/boostingmargins.pdf>

# Uwagi

- Adaptacja wag odbywa się w ten sposób, aby uczynić problem możliwie trudnym dla następnego klasyfikatora. Jednocześnie dla takich wag poprzedni klasyfikator stanie się słabym klasyfikatorem / działa inaczej niż kolejne
- Klasyfikator z mniejszym błędem  $e_i$ , a co za tym idzie, z większym  $\log(1/\beta_i)$  ma większy wpływ na ostateczną decyzję zespołu
- Boosting może często dać poprawę trafności, lecz nie zawsze zwiększanie iteracji jest właściwe / przeuczenie
- Algorytm uczący  $L$  musi być zdolny do uwzględnienia wag przykładów
- Jeśli nie, alternatywne podejścia – zmiana prawdopodobieństwa wylosowania przykładu proporcjonalna do wag

# Charakterystyka Adaboost

## Pro

- Relatywnie prosty, łatwy w implementacji i szybki
- Poza liczbą modeli, ew. warunkiem stopu nie wymaga wielu parametrów do strojenia (w odróżnieniu od XGBoost)
- Nie potrzeba dodatkowej wiedzy nt. słabszego uczenia klasyfikatora, może być użyty z wieloma algorytmami
- Może być uogólniany – później

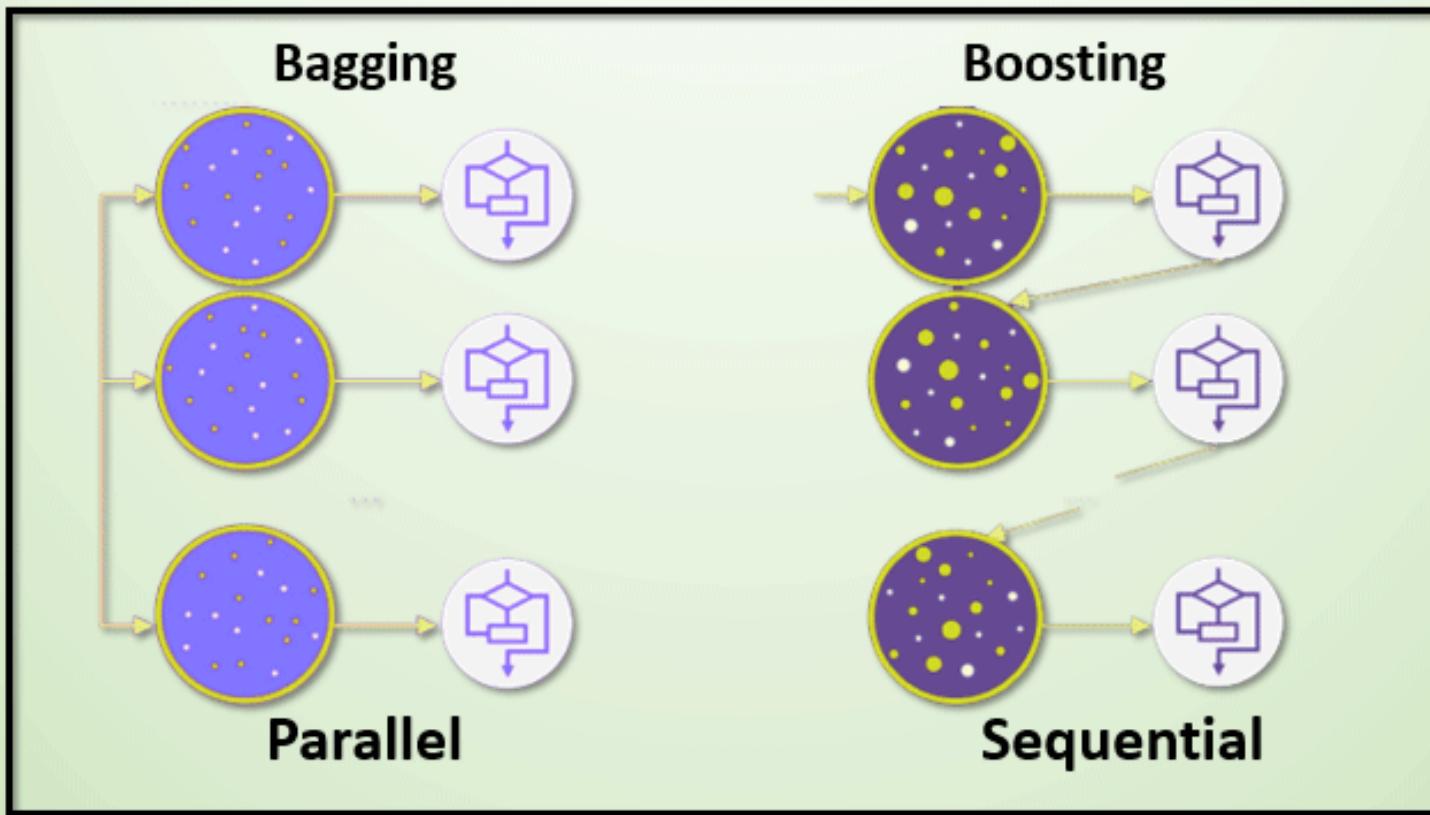
## Cons

- Faktyczna poprawa predykcji zależy od danych oraz właściwości algorytmu uczącego podstawowe klasyfikatory
- Może się przeuczyć
- Większa trudność skalowalnej implementacji dla większych danych (Big data) niż bagging lub RF

# Boosting vs. Bagging

- Bagging
  - modyfikacja danych poprzez losowanie przykładów,
  - klasyfikatory uczone niezależnie (możliwość zróżnoleglenie obliczeń)
  - Redukcja wariancji błędu
- Boosting
  - sekwencyjne rozbudowywanie zespołu – kolejny składnik zależny od działania wcześniejszych
  - modyfikacja poprzez zmianę rozkładu wag przypisywanych przykładom
  - Może redukować wariancje, lecz także bias

# Bagging and Boosting



# Boosting vs. Bagging with C4.5 [Quinlan 96]

	C4.5	Bagged C4.5 vs C4.5			Boosted C4.5 vs C4.5			Boosting vs Bagging	
	err (%)	err (%)	w-l	ratio	err (%)	w-l	ratio	w-l	ratio
anneal	7.67	6.25	10-0	.814	4.73	10-0	.617	10-0	.758
audiology	22.12	19.29	9-0	.872	15.71	10-0	.710	10-0	.814
auto	17.66	19.66	2-8	1.113	15.22	9-1	.862	9-1	.774
breast-w	5.28	4.23	9-0	.802	4.09	9-0	.775	7-2	.966
chess	8.55	8.33	6-2	.975	4.59	10-0	.537	10-0	.551
colic	14.92	15.19	0-6	1.018	18.83	0-10	1.262	0-10	1.240
credit-a	14.70	14.13	8-2	.962	15.64	1-9	1.064	0-10	1.107
credit-g	28.44	25.81	10-0	.908	29.14	2-8	1.025	0-10	1.129
diabetes	25.39	23.63	9-1	.931	28.18	0-10	1.110	0-10	1.192
glass	32.48	27.01	10-0	.832	23.55	10-0	.725	9-1	.872
heart-c	22.94	21.52	7-2	.938	21.39	8-0	.932	5-4	.994
heart-h	21.53	20.31	8-1	.943	21.05	5-4	.978	3-6	1.037
hepatitis	20.39	18.52	9-0	.908	17.68	10-0	.867	6-1	.955
hypo	.48	.45	7-2	.928	.36	9-1	.746	9-1	.804
iris	4.80	5.13	2-6	1.069	6.53	0-10	1.361	0-8	1.273
labor	19.12	14.39	10-0	.752	13.86	9-1	.725	5-3	.963
letter	11.99	7.51	10-0	.626	4.66	10-0	.389	10-0	.621
lymphography	21.69	20.41	8-2	.941	17.43	10-0	.804	10-0	.854
phoneme	19.44	18.73	10-0	.964	16.36	10-0	.842	10-0	.873
segment	3.21	2.74	9-1	.853	1.87	10-0	.583	10-0	.684
sick	1.34	1.22	7-1	.907	1.05	10-0	.781	9-1	.861
sonar	25.62	23.80	7-1	.929	19.62	10-0	.766	10-0	.824
soybean	7.73	7.58	6-3	.981	7.16	8-2	.926	8-1	.944
splice	5.91	5.58	9-1	.943	5.43	9-0	.919	6-4	.974
vehicle	27.09	25.54	10-0	.943	22.72	10-0	.839	10-0	.889
vote	5.06	4.37	9-0	.864	5.29	3-6	1.046	1-9	1.211
waveform	27.33	19.77	10-0	.723	18.53	10-0	.678	8-2	.938
average	15.66	14.11		.905	13.36		.847		.930

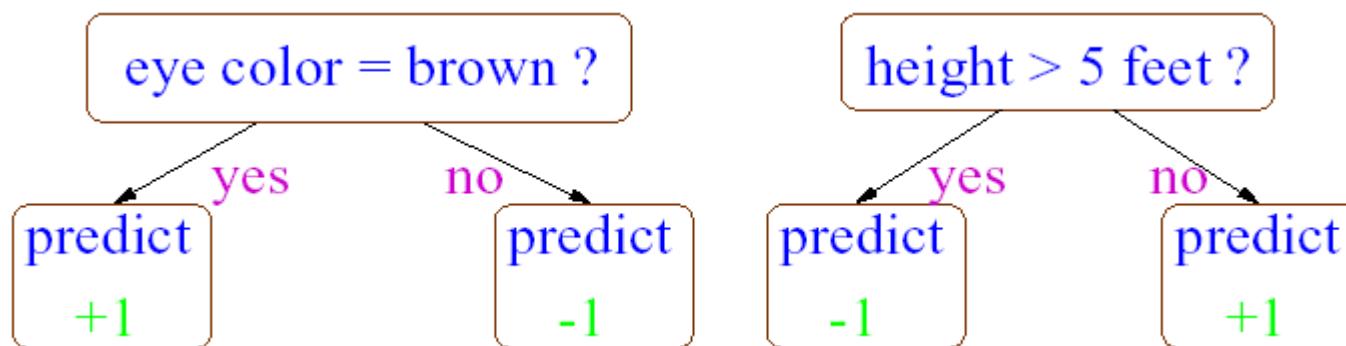
Table 1: Comparison of C4.5 and its bagged and boosted versions.

# Boosting vs. Bagging

- Bagging nie działa ze stabilnymi algorytmami, boosting może działać
- Boosting podatny na przeuczenie dla zaszumianych i trudnych danych. Bagging mniej, zwłaszcza odporny jest RF
- Boosting na ogół może prowadzić do wyższych przyrostów trafności, lecz może też prowadzić do pogorszenia na niektórych danych
- Bagging na ogół zawsze polepsza trafność chodź średnio mniej
- Bagging potencjalnie łatwiejszy do uogólnienia

# Boosting z drzewami

- Klasyfikatory o zbyt małym bias (np. k-NN) słabo wzmacniane [Rayens], często stosowany z drzewami
- W odróżnieniu od Random Forest (gdzie pozwala się budować głębsze, mniej zredukowane, drzewa) w koncepcji wzmacniania boosting na ogół wykorzystuje się płytsze drzewa (nawet tzw. decision stumps)
- Zbyt duża liczba drzew może prowadzić do przeuczenia dla niedoskonałych danych
- Dla mniejszych drzew – zwłaszcza dla function gradient boosting – głębokość i np. min. loss dla warunków podziału są globalnymi parametrami wymagającymi specjalnej optymalizacji



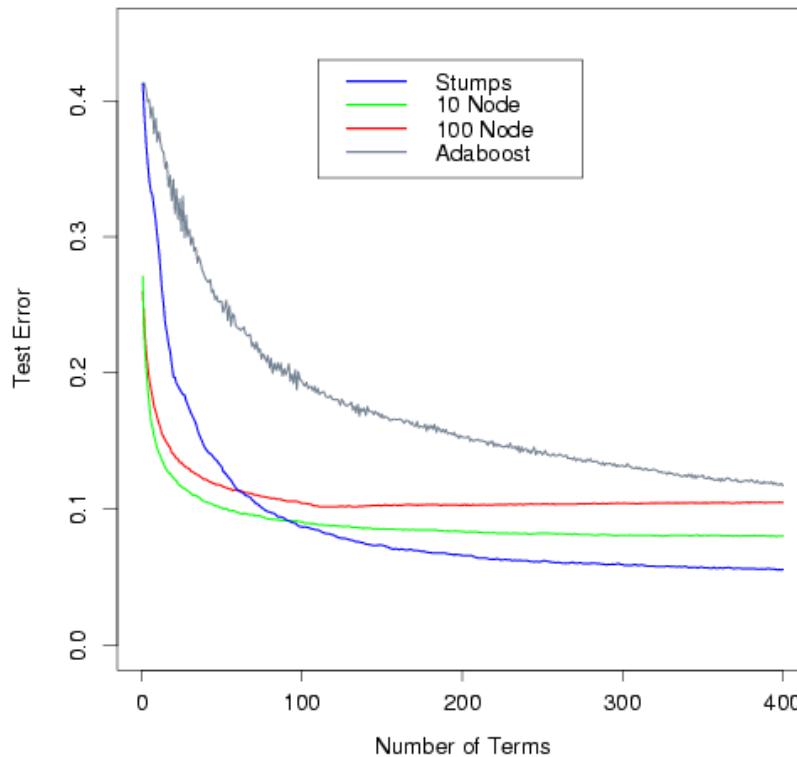


Figure 10.9: Boosting with different sized trees, applied to the example (10.2) used in Figure 10.2. Since the generative model is additive, stumps perform the best. The boosting algorithm used the binomial deviance loss in Algorithm 10.3; shown for comparison is the AdaBoost algorithm 10.1.

Rysunek za książka EST

## DECORATE (Melville & Mooney, 2003)

- Przykład uogólnienia boostingu poprzez wprowadzenie sztucznych przykładów do zbioru  $D_t$
- Zwiększa zróżnicowanie prób i klasyfikacji
- Skuteczne dla przetwarzanie mniejszych zbiorów danych, gdzie re-weighting i re-sampling ma mniejszy potencjał dywersyfikacji danych uczących

# Wzmacnienie – boosting - ogólniej

- Ogólna metoda służąca polepszeniu predykcji dowolnych klasyfikatorów uczonych dowolnymi algorytmami.
- Pierwsze inspiracje L.Valiant,M. Kearns; Rozwinięte przez Yoav Freund i Robert Schapire
- Pierwsze zastosowania (drzewa, ANN): OCR, rozpoznanie obrazów
- Liczne realizacji pomysłu wzmacniania słabych klasyfikatorów, nie tylko Adaboost

# Rozwój pomysłu wzmacniania klasyfikatorów

- Uogólnienia Adaboost, np.
  - Wprowadzenie uczenia z kosztami (MetaCost)
  - Specjalizowane przelosowanie obecności przykładów dla danych niezbalansowanych SmoteBoost
  - Cascade classifiers, np. dla rozpoznawania obrazów
  - W organiczonym stopniu w uczeniu przyrostowych (Learn++)
  - ...
- Wzmacnianie gradientowe – ang. gradient boosting (obecnie b. efektywne biblioteki, np. XGBoost, CatBoost i LightGBM)

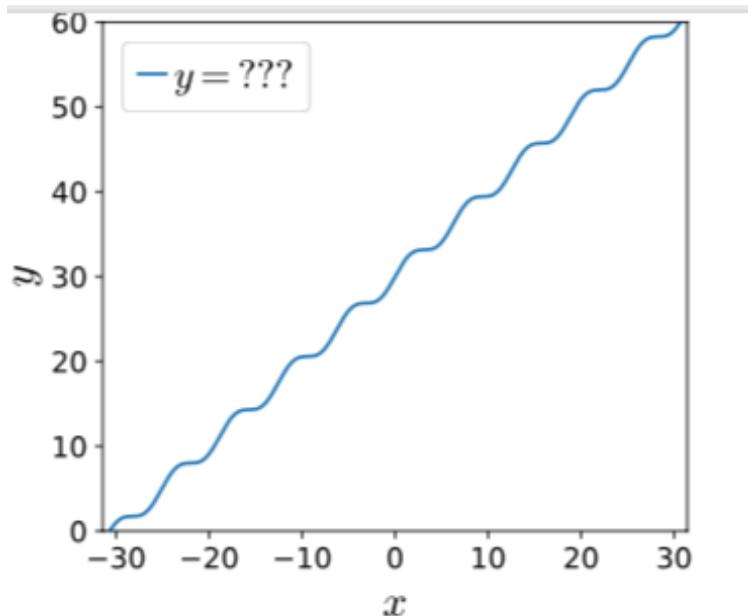
# Wzmacnianie gradientowe - krótko

## Gradient boosting

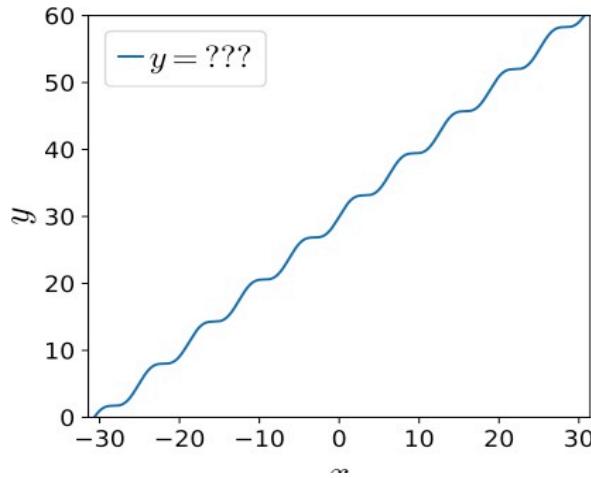
- Zasada dopasowywania **addytywnego modelu** do danych krok po kroku
- W każdym kroku wprowadza się kolejny “słaby” model w celu poradzenia sobie z błędami poprzedników
- W gradient boosting – są one identyfikowane poprzez **ujemne gradienty** wybranej funkcji straty
- W Adaboost – stosuje się wagowanie błędnie sklasyfikowanych przykładów
- W obu wersjach wskazują one kierunki zmiany

# Modele addytywne

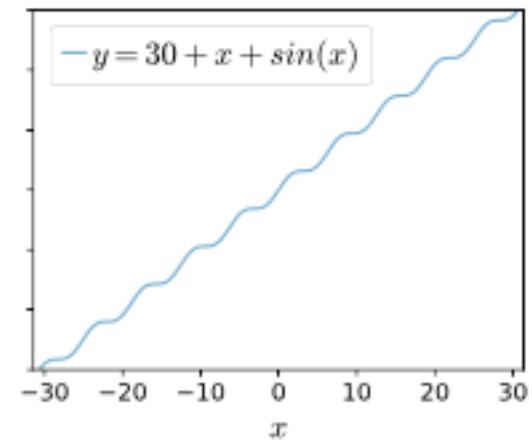
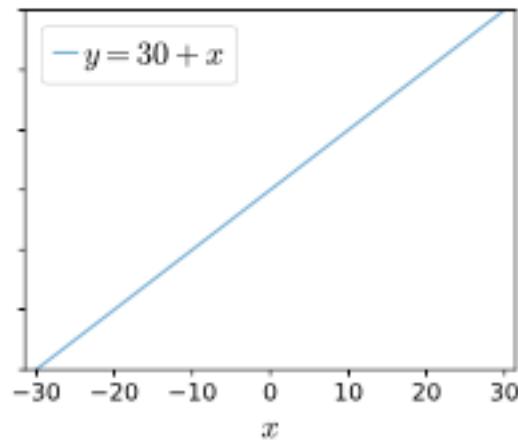
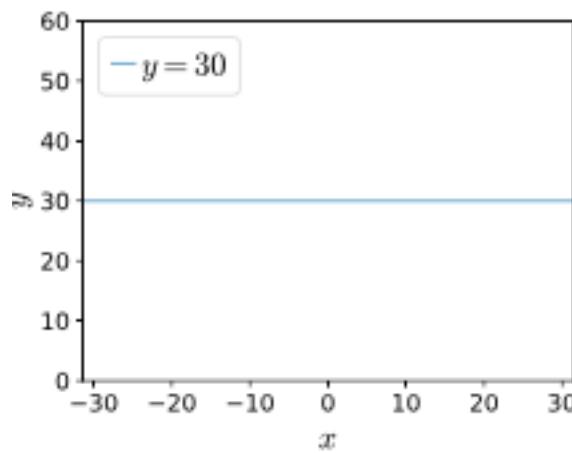
- Połączenie prostszych składników, po to aby zamodelować dopasowanie do złożonej sytuacji  $F(x)=f_1(x)+f_2(x)+f_3(x)+\dots$
- Przykład ilustracyjny – poszukiwanie funkcji regresji dopasowanej do złożonej krzywej  $y$  vs.  $x$



# Modele adytywne 2

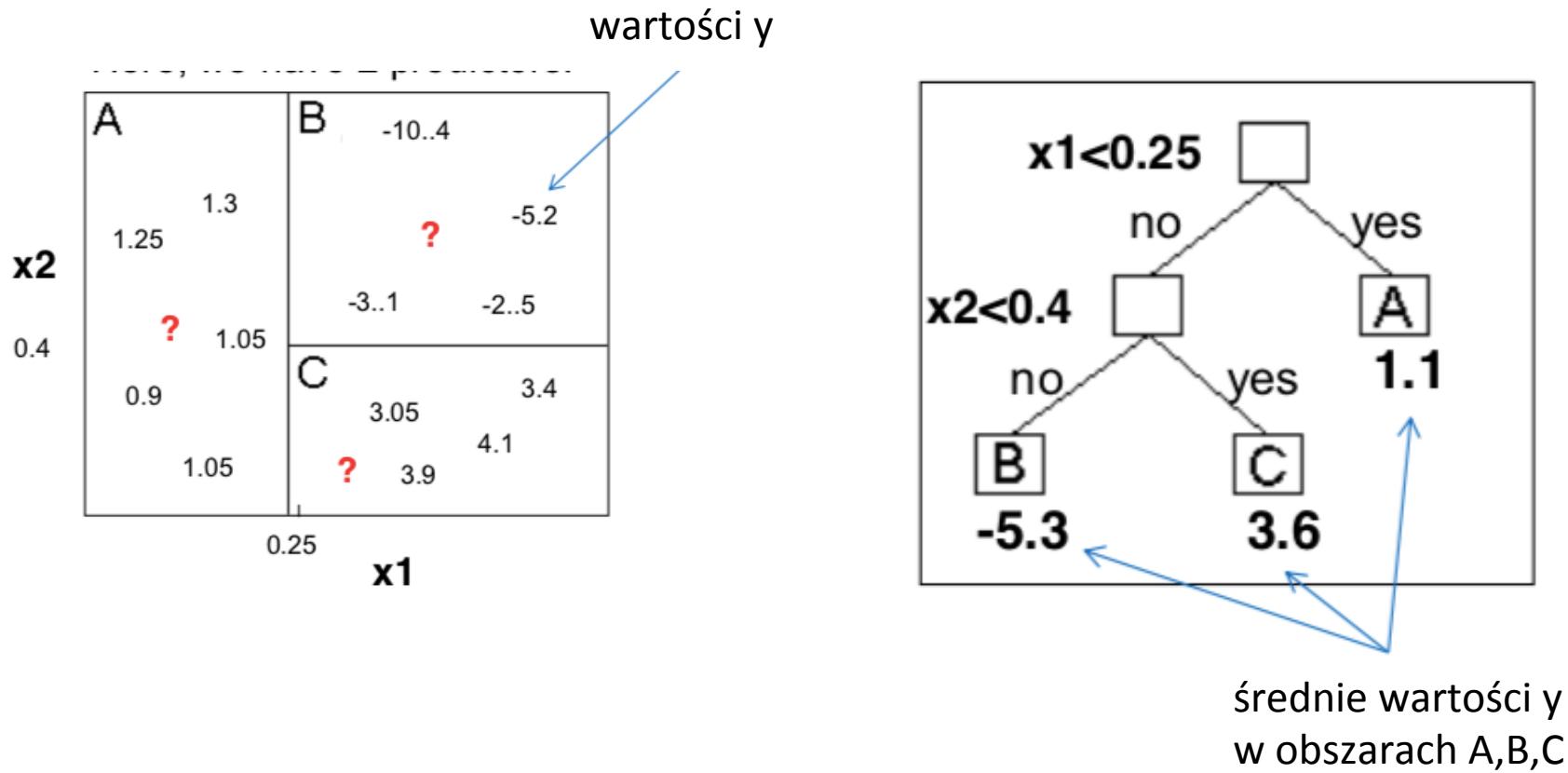


$$\hat{y} = \sum_{m=1}^M f_m(x)$$



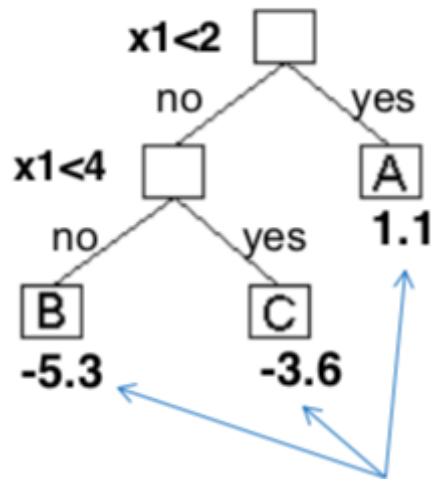
Zbudowano zespół  $F(x)=30+x+\sin(x)$

# Przykład z drzewami regresji

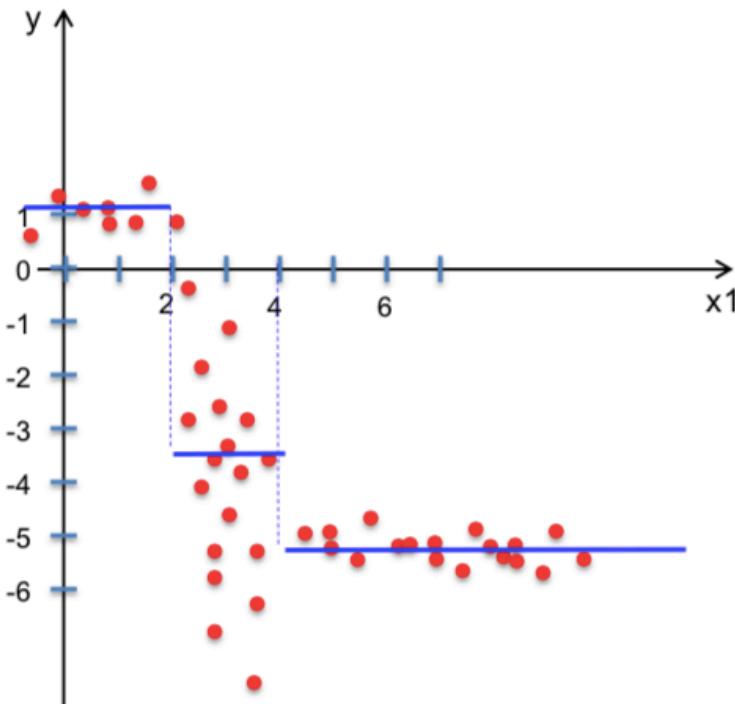


Cel: minimalizacja błędu średniokwadratowego MSE

# Drzewo regresji dla jednej zmiennej $x_1$

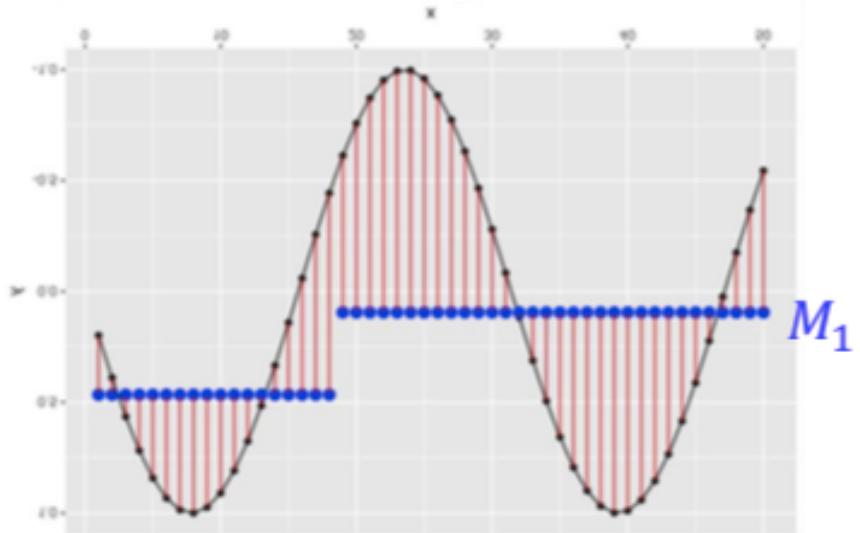


Kolejne podziały  $x_1$   
średnie wartości y  
w obszarach A,B,C

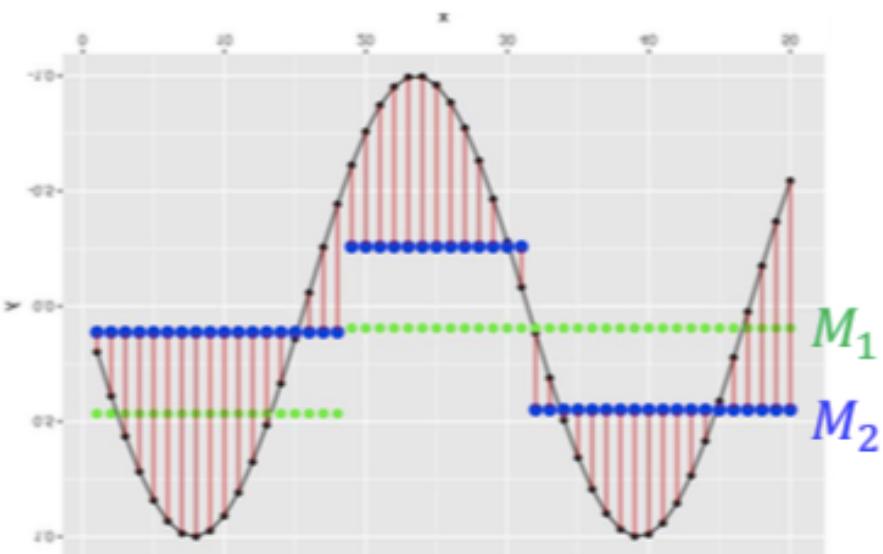


Rozważmy b. złożoną funkcję wymagającą modelu addytywnego  $F(x)$

# Funkcja $\sin(x)$

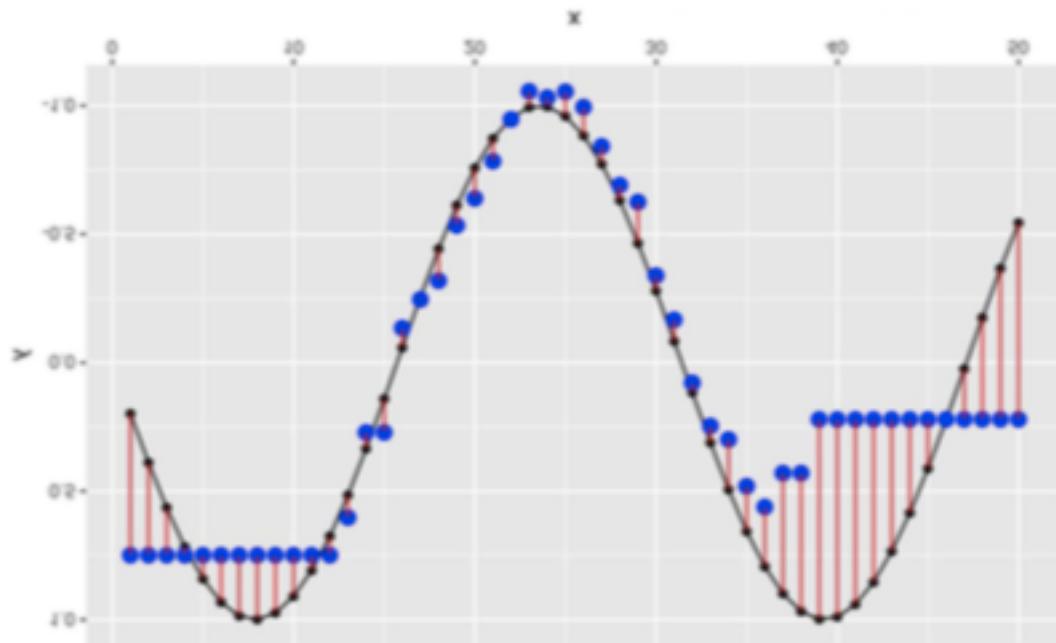


1. Zbuduj płytkie drzewo regresji  $T_1$  – pierwszy model dopasowany do danych  $M_1=T_1$ . Niedopasowanie  $T_1$  do danych opisują residua funkcji  $r_i=y_i-\hat{y}_i$
2. Zbuduj kolejne drzewo  $T_2$  na danych z wyjściami – residua  $r_i$ . Model zostaje rozszerzony  $M_2=M_1+\theta T_2$  gdzie  $\theta$  jest optymalizowany, w celu lepszego dopasowania do danych. Oblicz ponownie residua dla  $M_2$
3. Buduj kolejne drzewa dla dopasowania się do residiów z 2, i postępują tak do warunku stopu



# Boosted regression trees – model końcowy

Model końcowy jest ważonym uśrednieniem krokowo tworzonych modeli  $T$ ;  $M = T_1 + \eta \sum \theta_i T_i$  gdzie  $\eta$  jest prędkością uczenia (zapobieganie przeuczeniu)



# Dokładniej dla regresji

Ogólny schemat (dla regresji z MSE)

Dane uczące  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  => cel znaleźć model  $F(x)$  który minimalizuje błąd średniokwadratowy

Rozpoczynamy od pierwszego prostego modelu  $F_1(x)$ , który dla kolejnych  $x_i$  popełnia błędy niedopasowania (residua  $F$ ), np.  $F(x_1)=0.8$  gdy  $y_1=0.9$ ;  $F(x_2)=1.4$  gdy  $y_2=1.3$

Poszukujemy nowego modelu regresji  $h$ , który może być dodany do  $F$ , tak aby osiągnąć poprawę:

$$F(x_1) + h(x_1) = y_1$$

$$F(x_2) + h(x_2) = y_2$$

....

$$F(x_n) + h(x_n) = y_n$$

# Gradient boosting dla regresji

Alternatywnie poszukujemy nowego modelu regresji  $h$  np. drzewa regresji, który powinien

$$h(x_1) = y_1 - F(x_1)$$

$$h(x_2) = y_2 - F(x_2)$$

....

$$h(x_n) = y_n - F(x_n)$$

gdzie  $y - F(x)$  to residua  $r$  (wskazujące gdzie dotychczasowy model  $M$  źle działa). Utwórz nowy zbiór uczący  $x$  z wyjściem  $r$  ( $x, y - F(x)$ ) i naucz model  $h_1 = F_1$

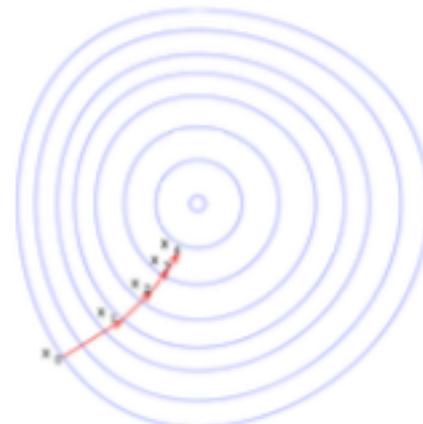
Model addytywny  $F(x) + h_1(x)$  może nadal nie dopasowywać się dość dobrze do danych, powtórz postępowanie w kolejnych iteracjach

Ogólny schemat boostingu - kolejny model poprawia ograniczenia wcześniejszych

# Spadek gradientu

Podejście spadku gradientu - uniwersalna metoda minimalizacji funkcji poprzez przesuwanie się w kierunku przeciwnym do gradientu:

$$\theta_{i+1} = \theta_i - \delta \frac{\partial J}{\partial \theta_i}$$



# Gradient boosting dla regresji

Jak podejście wzmacniania regresji ma się do spadku gradientu?

W regresji funkcja straty  $L(y, F(x)) = (y - F(x))^2 / 2$

Celem jest minimalizacja sumy kwadratów w danych

$$J = \sum_i L(y_i, F(x_i))$$

W przypadku regresji

$$\frac{\partial J}{\partial F(x_i)} = \frac{\partial \sum_i L(y_i, F(x_i))}{\partial F(x_i)} = F(x_i) - y_i$$

Czyli residua są ujemnymi gradientami

$$r_i = y_i - F(x_i) = -\frac{\partial J}{\partial F(x_i)}$$

# Gradient boosting dla regresji

Residua f. regresji  $\Leftrightarrow$  ujemny gradient f. straty

Dopasowanie  $h$  do residiów  $\Leftrightarrow$  dopasowanie  $h$  do  
ujemnych gradientów f. straty

Rozbudowana modeli  $F$  wg. residiów  $\Leftrightarrow$  krokowa  
rozbudowane wg. ujemnych gradientów

Prowadzi to do ogólnego sformułowanie podejścia  
funkcyjnego gradientowego wzmacnianie (gradient  
boosted ensemble)

# Przebieg wzmacniania gradientowego dla regresji

Zainicjuj pierwszy model  $F(x) = \sum_i y_i / n$  oraz  $j = 1$

Postępuj do warunku stopu

1. Oblicz ujemne gradienty funkcji straty  $L$  i utwórz nowy zbiór uczący  $D_j$
2. Naucz model  $F_j$  z  $D_j$  (dopasowujący się do ujemnych gradientów)  $F = F_1 + \eta \sum \theta_j F_j$
3. Optymalizuj  $\theta_j$  w celu dobrego dopasowania do danych
4.  $j = j + 1$

Ogólny schemat – może być użyty także dla innych postaci funkcji straty

# Wersja dla drzew regresji [za JM]

---

## Gradient Tree Boosting Algorithm

---

1. Initialize  $f_0(x) = \arg \min_{\gamma} \sum_{i=1}^N L(y_i, \gamma)$
2. For  $m = 1$  to  $M$ :
  - (a) For  $i = 1, 2, \dots, N$  compute

$$r_{im} = - \left[ \frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]_{f=f_{m-1}}$$

- (b) Fit a regression tree to the targets  $r_{im}$  giving terminal regions  $R_{jm}$ ,  $j = 1, 2, \dots, J_m$ .
  - (c) For  $j = 1, 2, \dots, J_m$  compute

$$\gamma_{jm} = \arg \min_{\gamma} \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma)$$

- (d) Update  $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$ .
  3. Output  $\hat{f}(x) = f_M(x)$
-

# Inne funkcje straty $L$

Interpretacja statystyczna [FHT – EST 2000]

Odpowiedź złożonego klasyfikatora jako model addytywny

$$F(x) = \sum_{i=1}^T \theta_i f_i(x)$$

Adaboost jako metoda krokkowego poszukiwania minimum funkcji straty

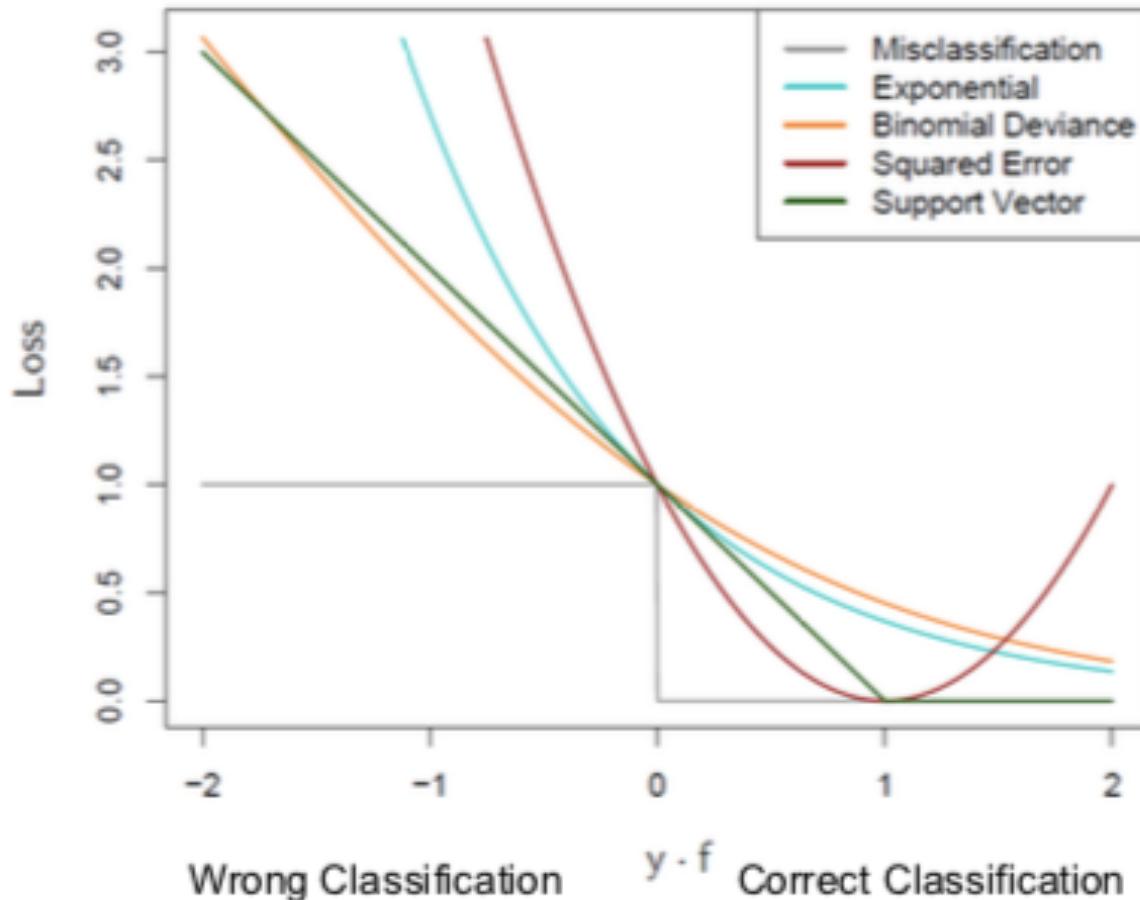
$$\frac{1}{n} \sum_{j=1}^n L(y_j, f(x_j)) \quad \text{gdzie} \quad L(y, f) = e^{-yf}$$

FHT pokazali, że taka funkcja ma własność, że powyższa optymalizacja prowadzi do przybliżenia klasyfikatora

Bayesowskiego

Istnieją alternatywne ciągłe funkcje straty – co prowadzi do tzw. funkcyjnego wzmacniania gradientowego i szerszej klasy metod

# Inne funkcje straty



Misclassification

$$L(y, F) = I[y \neq \text{sign}(F)]$$

Exponential / AdaBoost

$$L(y, F) = \exp(-yF)$$

Binomial Deviance

$$L(y, F) = \log(1 + \exp(-2yF))$$

Quadratic / L2-Boost

$$L(y, F) = (y - F)^2$$

SVM

$$L(y, F) = y \cdot (1 - y \cdot F)$$

# Inne wersje zadania

- Rozwiążanie wzmacniania gradientowego przekształca się dla wersji klasyfikacyjnej oraz uczenia się rankingów
- Bardzo ciekawy przykład rozpoznawania liter dostępny w Cheng Li: A gentle introduction to gradient boosting

# Wersja klasyfikacyjna za wykład IPI PAN

Szukamy funkcji klasyfikacyjnej minimalizującej ryzyko empiryczne:

$$\operatorname{argmin}_f n^{-1} \sum_{i=1}^n L(Y_i, f(X_i))$$

- Inicjalizacja:  $\hat{f}^{[0]}(\cdot) \equiv \operatorname{argmin}_c n^{-1} \sum_{i=1}^n L(Y_i, c)$ .  
Dla  $m = 1, \dots, m_{stop}$  :
- (i) Oblicz

$$U_i = -\frac{\partial}{\partial f} L(Y_i, f)_{|f=\hat{f}^{[m-1]}(X_i)} \quad i = 1, 2, \dots, n.$$

- (ii) Zastosuj wybraną metodę oszacowania funkcji regresji do próby  $(X_i, U_i)$ :

$$(X_i, U_i) \longrightarrow \hat{g}^{[m]}(\cdot)$$

( szacowanie gradientu).

$$(iii) f^{[m]}(\cdot) = f^{[m-1]}(\cdot) + \nu \times \hat{g}^{[m]}(\cdot).$$

$$\text{Albo } \nu_m = \operatorname{argmin}_\nu L(f^{[m-1]}(\cdot) + \nu \times \hat{g}^{[m]}(\cdot))$$

# Uwagi o Funkcjonalnego Wzmacniania Gradientowego

## Wzmacnianie gradientowe (gradient boosting)

- Inspiracja Breiman (1999) o iteracyjnej minimalizacji funkcji straty w boosting
- Friedman, Hastie, Tibshirani (2000) addytywne modele w Adaboost oraz uogólnienia na różne funkcje straty

## Extreme Gradient Boosting

T.Chen (2014/2016) = dodatkowo regularyzacja w celu monitorowania przeuczenia / wymaga specjalnej optymalizacji parametrów

## Implementacja XGBoost (T.Chen w DMLC)

Późniejsza implementacja LightGBM

# Ekstremalne Wzmacnianie Gradientowe

## Extreme Gradient Boosting - EXBoost

Tiangi Chen (2016) wprowadzenie składnika regularyzacji do funkcji straty -> minimalizacja liczby modeli i monitorowanie przeuczenia – lecz trudniejsza do obliczenia

Ponadto wymaga specjalnej optymalizacji parametrów (zwłaszcza dla drzew)

- Opis różnic do GB np. w <https://towardsdatascience.com/boosting-algorithm-xgboost-4d9ec0207d>

Efektywna implementacja biblioteka XGBoost

Z powodzeniem zastosowana w wielu konkursach (np. patrz platforma Kaggle)

Obecnie bardzo popularny

# Odnośniki do literatury

- Intensywny rozwój od lat 90 poprzedniego wieku
- Wiele różnych propozycji:
  - R.Polikar, Ensemble based systems in decision making, IEEE Circuits and Systems Magazine, vol. 6, no. 3, pp. 21–45, 2006.
  - Schapire, Robert E. (1990). The Strength of Weak Learnability . Machine Learning. 5 (2): 197–227.,
  - Yoav Freund and Robert E. Schapire (1997); A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting, Journal of Computer and System Sciences, 55(1):119-139
  - Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine Learning, 36(1/2):105–139, 1999.
  - Leo Breiman (1998). "Arcing classifier (with discussion and a rejoinder by the author)". Ann. Stat. 26 (3): 801–84
  - Tianqi Chen i Carlos Guestrin, 2016. XGBoost: A Scalable Tree Boosting System ,<https://arxiv.org/pdf/1603.02754.pdf>

# **Pytanie i komentarze?**

Dalszy kontakt:

[jerzy.stefanowski@cs.put.poznan.pl](mailto:jerzy.stefanowski@cs.put.poznan.pl)

<http://www.cs.put.poznan.pl/jstefanowski/>



**Fundusze  
Europejskie**  
Polska Cyfrowa



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz  
Rozwoju Regionalnego



# Zespoły modeli predykcyjnych inne podejścia wykład 10

Jerzy Stefanowski

Instytut Informatyki PP

2021

Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI-TECH)  
projekt finansowany z środków Programu Operacyjnego Polska Cyfrowa  
POPC.03.02.00-00-0001/20



Fundusze  
Europejskie  
Polska Cyfrowa



Rzeczpospolita  
Polska

Unia Europejska  
Europejski Fundusz  
Rozwoju Regionalnego



# Plan wykładu

- Zróżnicowanie klasyfikatorów składowych
- Generalizacja stosowa (ang. stacking)
- Podejście tzw. mieszanki ekspertów (ang. mixture of experts)
- Podejścia zespołowe do danych silnie wieloklasowych
- Podsumowanie

# Motywacje dla Stacking [ang.]

- Alternatywne podejścia do budowania złożonych klasyfikatorów
- Klasyfikatory bazowe – często niejednorodne, uczone różnymi algorytmami
  - Lecz może być użyte do zastąpienia głosowania w zespołach bagging lub boosting
- **Struktura wielopoziomowa** – z różnymi podejściami do rozstrzygania niejednoznaczności wskazań klasyfikatorów bazowych
- Koncepcja tzw. meta-uczenia się (wiedza z odpowiedzi innych klasyfikatorów)
- Możliwe dynamiczne modyfikowanie działania zespołu

# Stacking – generalizacja stosowa

Obserwacje – niektóre przykłady mają wysokie prawdopodobieństwo złej klasyfikacji, a inne są częściej dobrze klasyfikowane.

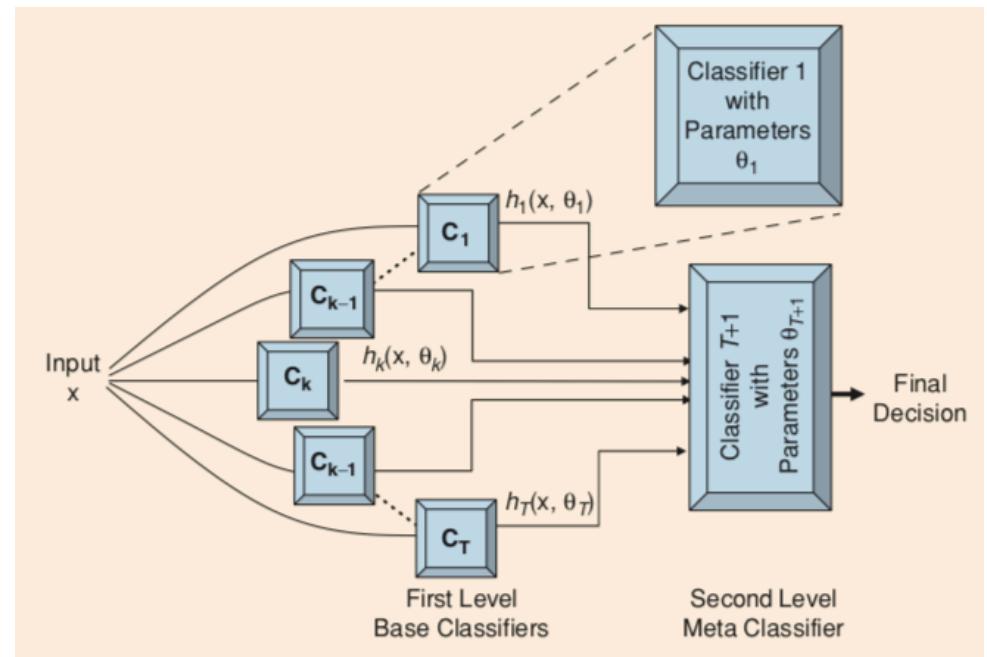
Pytanie – czy można nauczyć się ogólniejszych **meta zasad**, jak dokonać korekty klasyfikacji pewnych modeli (lepiej niż głosowanie większościowe)

Wolpert – **generalizacja stosowa**:

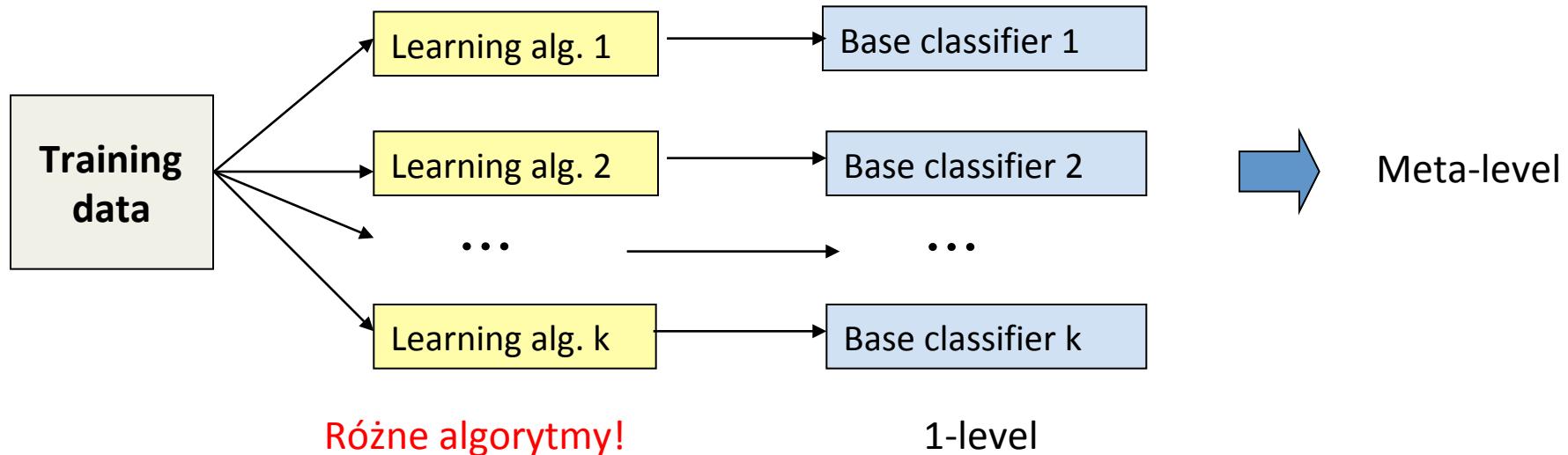
- Wyjścia klasyfikatorów są wejściami dla kolejnych algorytmów (meta-uczących się), w celu nauczenia zasad korekty wcześniejszych predykcji.
- Możliwe jest łączenie wielu warstw algorytmów / klasyfikatorów

# Stacked generalization [Wolpert 1992]

- Wykorzystaj ideę meta-uczenia
  - Predykcje tzw. base learners/model (*level-0 models*) przekazane na wejścia kolejnych tzw. meta learner (*level-1 model*)
- Metody uczenia bazowych modeli (poziom 0) są często różnymi algorytmami (podejścia niejednorodnych modeli)
- Różne rozwiązania idei meta-poziomu (poziom 1), np. tzw. combiner albo arbiter



# Meta-uczenie -> tzw. combiner

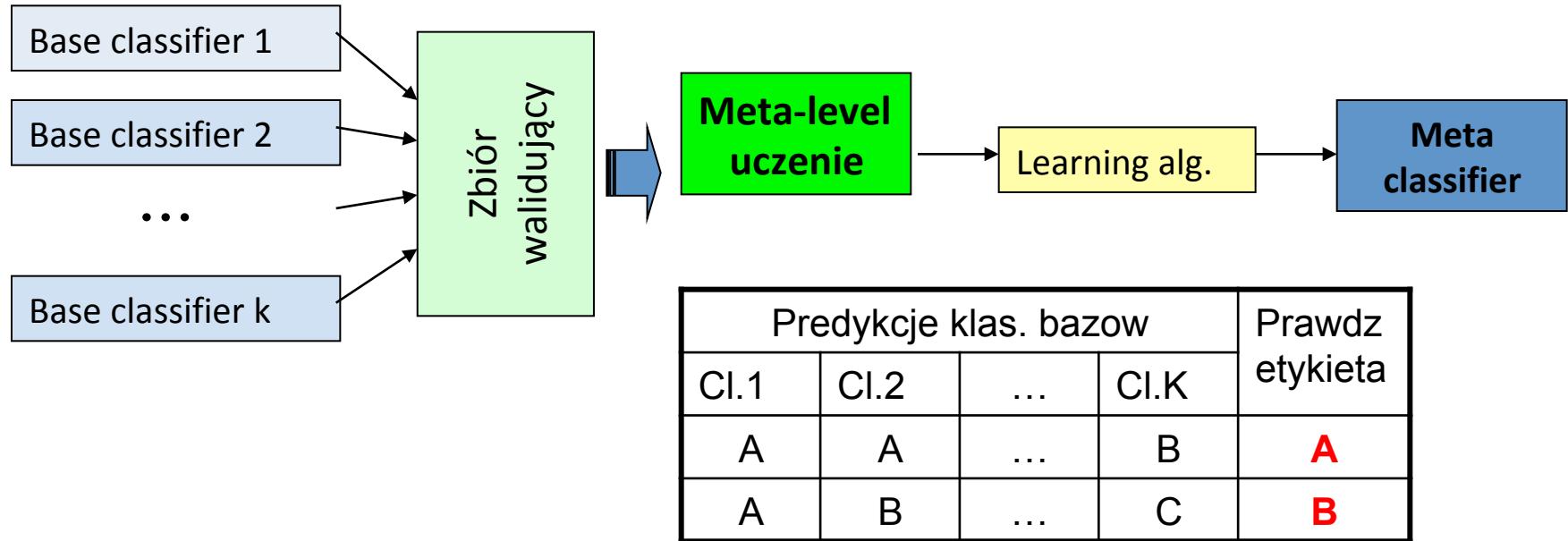


Chan & Stolfo : *Meta-learning* [meta-uczenie]

- Dwa poziomy:
  - 1-level – base classifiers
  - 2-level – meta-classifier
- Różne algorytmy użyte do uczenia klasyfikatorów bazowych (zróżnicowanie klasyfikatorów)

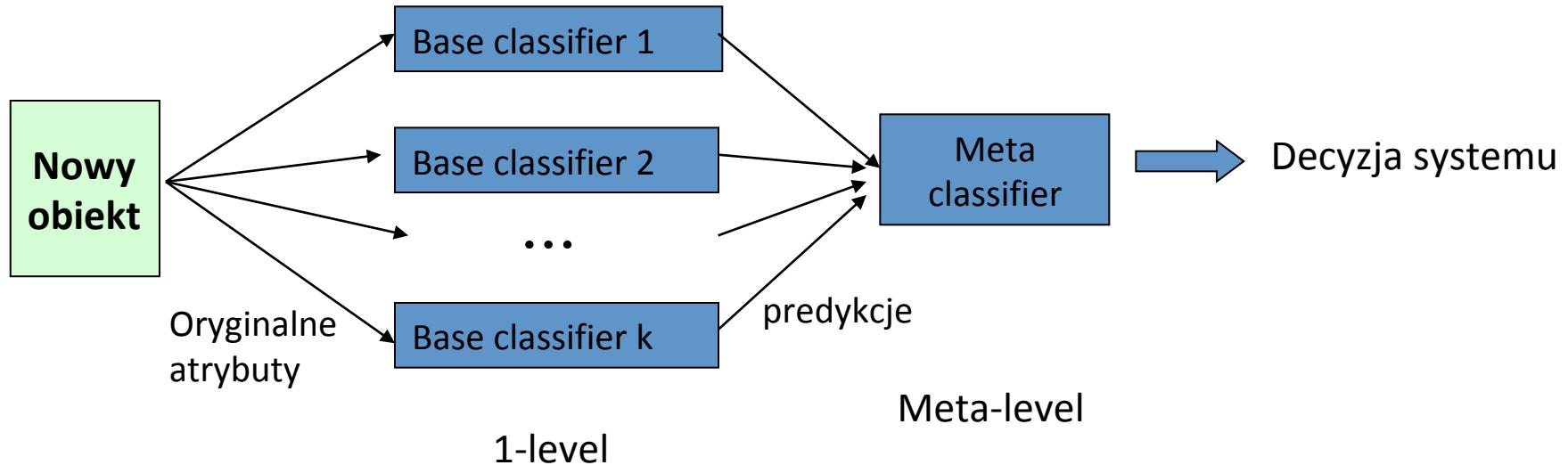
Więcej patrz: Experiments on Multistrategy Learning by Meta-Learning.  
P.Chan, S.Stolfo 1993

# Uczenie meta klasyfikatora



- Predykcje klasyfikatorów bazowych na zbiorze walidującym (ew. wewnętrzna ocena krzyżowa) wraz z właściwą etykietą – zbiór uczący meta-level (może być rozszerzony przez atrybuty z org. danych)
- Niezależny algorytm uczący meta-klasyfikator
- Celem jest wyuczenie korekty predykcji w b. złożony sposób niż głosowanie większościowe

# Predykcja systemu złożonego combiner



Klasyfikacja nowego przykładu

Chan & Stolfo [95/97] : eksperymenty w architekturze ( $\{\text{CART}, \text{ID3}, \text{K-NN}\} \rightarrow \text{NBayes}$ ) / [dane biomedyczne] trafność lepsza niż pojedyncze klasyfikatory i ich złożenia poprzez głosowanie większościowe

## Comparison of classification accuracy (%)

Data set	K-NN	C4.5	MODLEM	Combiner
acl	84.29	85.00	85.00	84.29
bupa	63.19	62.32	68.10	69.12
cleveland	52.10	53.14	54.46	55.66
glass	68.80	65.42	69.63	71.50
hsv	56.56	51.64	55.74	59.02
imidasolium	58.21	58.21	60.70	66.67
...	...	...	...	...
yeast	57.80	52.10	54.30	58.36

More in Nowaczyk, Stefanowski: On Using Rule Induction in  
Multiple Classifiers with a Combiner Aggregation Strategy. ISDA 2005.

Meta-uczenia : Naive Bayes / sprawdzano też inne

Ogólny przyrost trafności + lepiej niż pojedyncze klasyfikatory składowe

# Inne eksperymenty [materiały njt.edu]

Breast Cancer Dataset

Method	Error (%)	Precision (%)	Recall (%)
NB	13.7	86	90
Linear SVM	12.5	92	92
Logistic Regression	10.7	93	95
Random Forest	5.35	95	98
Boosted Trees	3.57	95	95
<b>Stacked Ensemble Classifier</b>	<b>1.78</b>	<b>98</b>	<b>98</b>

Table: Comparison of performance of different classifiers using the Breast Cancer d

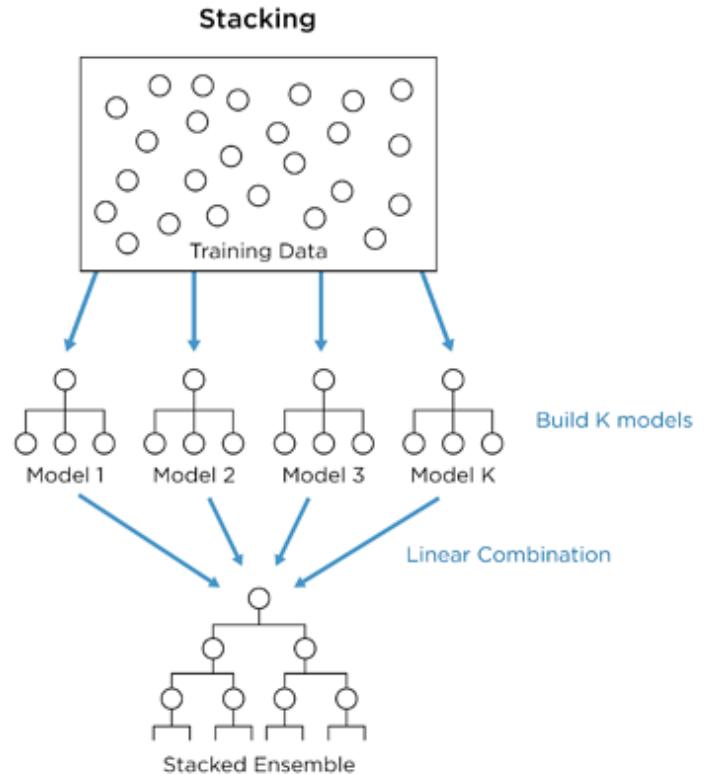
Przyrost miar oceny na 6 różnorodnych zbiorach danych,  
także w porównaniu do standardowych klasyfikatorów  
Więcej na linku podanym w ekursy

# Stacking - rozszerzenia

- Wyjścia klasyfikatorów są rozkładami prawdopodobieństw [Witten 1999] / także meta-uczenie z odmianą regresyjną tzw. model tree [Dzeroski, Zenko 2004] – złożenie 3 klasyfikatorów, ocena eksperymentalna na 30 zbiorach danych z UCI
- StackingC – przeznaczony dla problemów wieloklasowych. Klasyfikatory bazowe ukierunkowane na rozpoznanie jednej z klas
- SCANN – wykorzystanie przekształcenie wyjść poprzez analizę korespondencji i poszukiwanie bliskości w nowej przestrzeni – odmiana kNN jako finalna predykcja [Merz 1999]

# Generalizacja stosowa - więcej

- Literatura naukowa – eksperymentalne doświadczenia z rozszerzaniem klasyfikatorów bagging zamiast niejednorodnych klasyfikatorów badawczych
- Wyższe poziomy mogą być także zespołem klasyfikatorów



# Podejście arbitrażu

Inny sposób rozstrzygania niespójnych decyzji łączonych klasyfikatorów – tzw. arbiter trees - także wprowadzony przez Chan i Stolfo [1999]

- Zbiór uczący podzielony na  $k$  rozłącznych części
- Dla każdej pary klasyfikatorów – uczymy specjalny klasyfikator arbiter do rozstrzygania niezgodności pomiędzy ich decyzjami (specjalne reguła arbitrażu – głosowanie trzech dla przykładu)
- Nowy arbiter (wyższego meta-poziomu) jest uczyony z wyjść arbitrów niższego poziomu aż do finalnej decyzji arbitrażowej
- Konstruuje się strukturę drzewa podjęcia decyzji końcowej

Opis podejścia z algorytmem – książka L.Rokach: Pattern classification using ensemble methods (2009)

# Mixture of experts

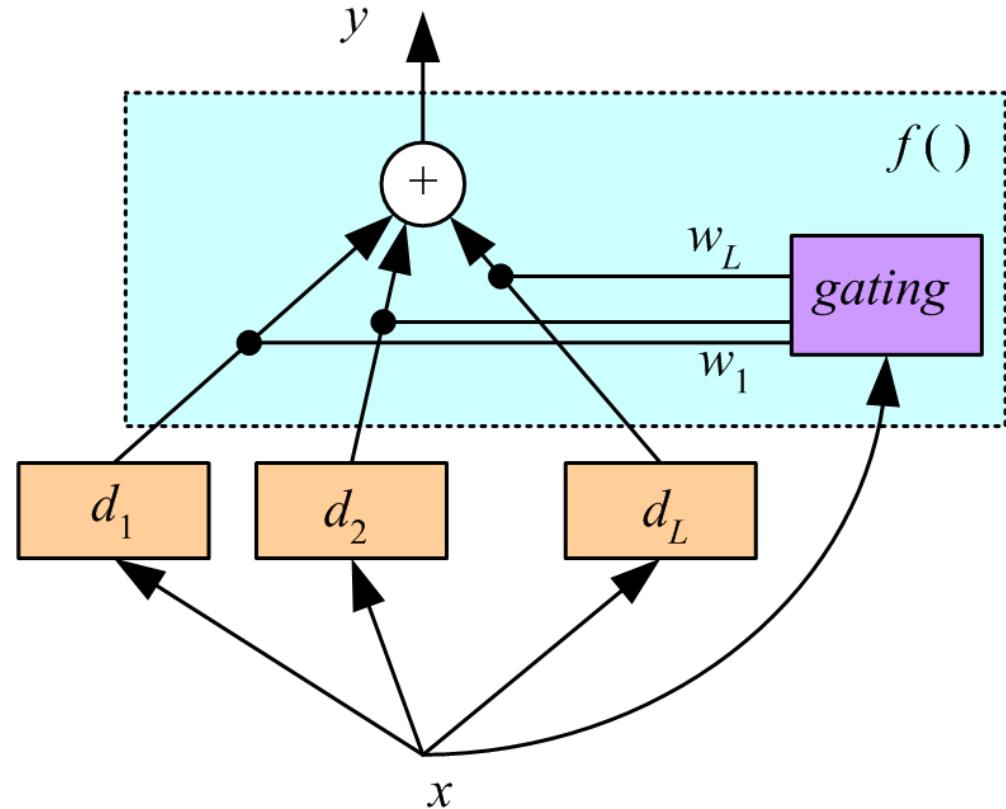
- Motywacja - różne obszary przestrzeni cech pokrywane są przez różne modele / będącymi „ekspertami” od tych pod-problemów
- Ich predykcje (dla nowego / klasyfikowanego przykładu  $x$ ) są “miękko” składane lub specjalny składnik (tzw. ang. gating network) wybiera najbardziej kompetentnych ekspertów dla danego obszaru i przykładu  $x$
- Proces uczenia – obejmuje zarówno modele bazowe jak i tzw. gating network
- Różne rozwiązania:
  - np. probabilistyczne modele generatywne
  - Sieci neuronowe -> soft max element
- Sieć neuronowa RBF może być prostą realizacją idei „mieszanki ekspertów”

# Mieszanka ekspertów z dynamicznym wagowaniem ich predykcji

Tzw. gating network

Przykład – przypisanie wag do predykcji klasyfikatorów dostosowanych do klasyfikowanego przykładu  $x$

$$y = \sum_{j=1}^L w_j(x) d_j(x)$$

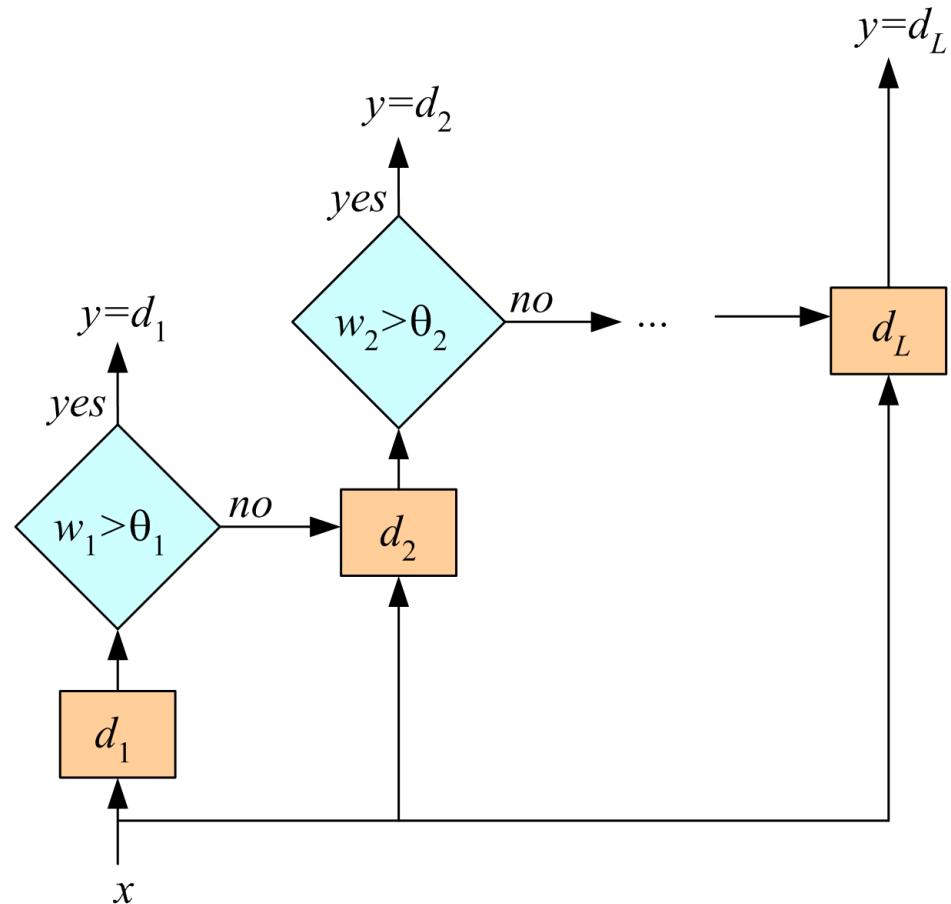


# Rozwiążanie kaskadowe

## Cascade models:

Stwórz kaskadę modeli predykcyjnych (zróżnicowanych) w celu poradzenia sobie ze złożonymi zadaniami / specjalizują się w trudnościach

Wykorzystaj kolejny model i jego predykcję  $d_j$ , jeśli poprzednicy są niepewni (ang. not sufficiently confident)



# Kaskadowa sekwencja

- Bazowe klasyfikatory są dodawane sekwencyjnie
  - jeśli pewność predykcji pierwszego klasyfikatora jest duża, to jego decyzja jest uznawana za ostateczną
  - W innym przypadku decyzja jest przekazywana do kolejnego klasyfikatora, itd.
- Model kaskadowy znajduje zastosowanie w systemach czasu rzeczywistego, jeśli decyzja powinna być szybka, podjęta maksymalnie przez kilka klasyfikatorów

# Miary pewności predykcji zespołu

W przypadku prostych złożen (bagging, stacking, ...) ocenia się tzw. margines decyzji zespołu

- Różnica głosów za zwycięską i drugą klasą
  - Np. dla problemu 3 klasowego bagging z 17 drzewami ma rozkład predykcji: C1 – 10 głosów, C2 – 5 i C3 – 2 głosów -> margines 10-5=5
- Analogiczna różnica dla miar zagregowanych po prawdopodobieństwach lub innych „scores”, także ważonych

Poczekaj do wykładu nt. aktywnego uczenia z techniką “Query by Committee”, gdzie wykorzystuje się takie marginesy

# Miary zróżnicowania

- Wiele propozycji – patrz książka L.Kuncheva *Combining Pattern Classifiers*
- Rozważmy parę klasyfikatorów  $C_i$  oraz  $C_j$  (tzw. pairwise measures) + decyzje binarne (poprawny lub błędny)

	C <sub>j</sub> jest poprawny	C <sub>j</sub> jest błędny
C <sub>i</sub> jest poprawny	a	b
C <sub>i</sub> jest błędny	c	d

- **Q Statistics**  $Q_{ij} = (ad - bc) / (ad + bc)$
- Dodatnie wartości Q: jeśli przykłady są poprawnie klasyfikowane przez oba klasyfikatory, ujemne dla odwrotnych klasyfikacji.  
Maksymalne zróżnicowanie predykcji  $\rightarrow Q=0$

# Miary zróżnicowania klasyfikatorów

- Miary ang. disagreement i double fault
$$D_{ij} = b + c \quad DF_{ij} = d$$
- oraz inne Kappa, korelacja odpowiedzi,...
  - Przegląd w książce Ludmila Kuncheva
- Dla  $T$  klasyfikatorów – mamy  $T(T-1)/2$  miar zróżnicowania par
- Najczęściej uśrednia się je do jednej globalnej wartości

# Niesparowane miary zróżnicowania

Niech dla  $i$ -tego przykładu,  $e_i$  jest liczbą klasyfikatorów z  $T$ , które błędnie klasyfikują ten przykład, wtedy

Entropia (0 – klasyfikatory podejmują te same decyzje, 1 są maksymalnie zróżnicowane):

$$E = \frac{1}{n} \sum_{i=1}^n \frac{1}{T - (T/2)} \min\{e_i, T - e_i\}$$

Kohavi-Wolpert variance

$$KW = \frac{1}{nT^2} \sum_{i=1}^n e_i \cdot (T - e_i)$$

# Dobre i złe zróżnicowanie klasyfikatorów składowych

- **Postulat zróżnicowania** (ang. diversity) łączonych klasyfikatorów ->  
nie mogą być zbyt podobne do siebie (mieć podobnych predykcji) + dostatecznej ich jakości (dla obserwacji z rozkładu przestrzeni cech, rozumianych że wystarczająco dużo klasyfikatorów powinno podjąć prawidłową predykcje)
  - Zbyt wiele miar zróżnicowania i brak uniwersalnej miary
- Tzw. złe zróżnicowanie (ang. bad diversity) – pewna różnorodność klasyfikatorów może wpływać negatywnie na ich połączenie
- Więcej informacji w pozycjach:
  - “Good” and “bad” diversity in majority vote ensembles. G Brown, L Kuncheva MCS 2010

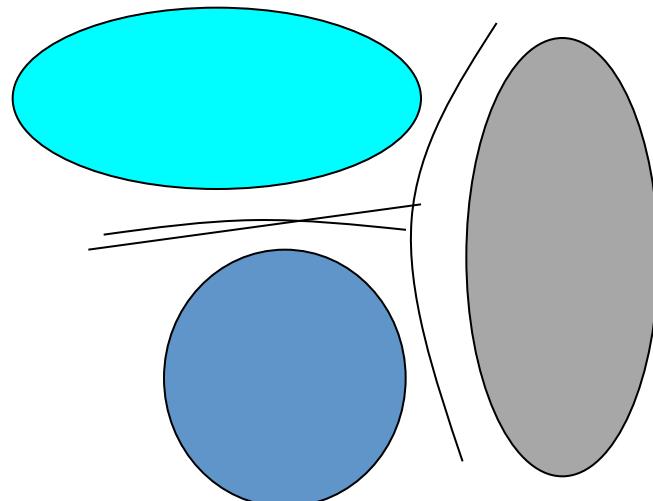
# Wykorzystanie miar zróżnicowania

- Analiza budowy i działania nauczonych już zespołów
- Wykorzystanie do redukcji zbyt licznych zespołów (ang. ensemble pruning), tj. wyboru „najbardziej wartościowych” klasyfikatorów składowych i poprawy predykcji takiego złożenia kombinowanego
- Przegląd podejść [M.Woźniak]:
  1. Rank-based pruning – ustaleniu rankingu klasyfikatorów z wykorzystaniem parowych miar zróżnicowania i wyboru najlepiej ocenianych
  2. Optimization based pruning – sformułowanie problemu optymalizacji (łączone kryteria) i zastosowania podejść ewolucyjnych
  3. Clustering based pruning – grupowanie podobnych klasyfikatorów i wybór ich reprezentantów

Więcej: M.Woźniak: Zespoły klasyfikatorów – aktualne kierunki badań (2015).

# Specjalne zespoły dla problemów wieloklasowych

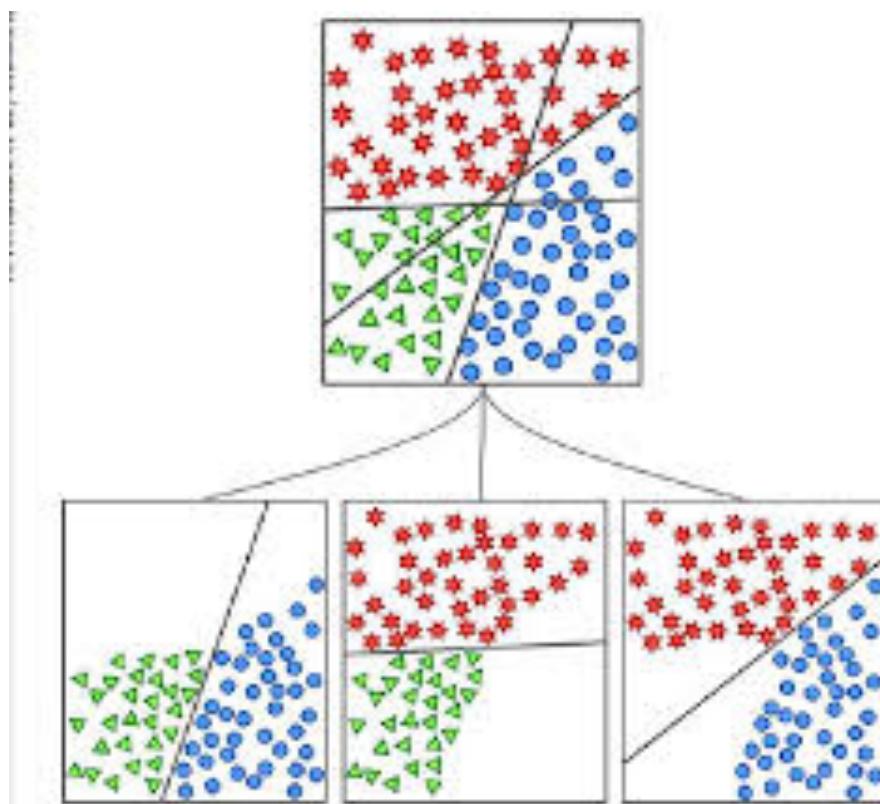
- Zadanie klasyfikacji do wielu klas ( $n >> 2$  kategorii)
- Część rzeczywistych danych – klasy trudne do nauczenia (pojęcia o nieliniowych granicach i trudnych rozkładach w przestrzeni atrybutów)
- Często dekompozycja na podproblemy – łatwiej się nauczyć
- Przykład problemu trzy klasowego (tzw. pairwise decision boundaries between each pairs of classes are simpler)



# Dekompozycja binarna

Podejście z j. ang. one-against-one / pairwise coupling

Rozważ wszystkie kombinacje dwóch klas i wyucz specjalizowane klasyfikatory binarne



# Pairwise coupling - n2-classifier

Zespół złożony z  $(n^2-n)/2$  *binarnych klasyfikatorów* (wszystkie połączenia par z  $n$  klas)

- Każda para klas  $(i,j)$ , gdzie  $i,j \in [1.. n]$ ,  $i \neq j$ , rozróżniana przez niezależny klasyfikator  $C_{ij}$
- Uczenie  $C_{ij}$  – tylko przykłady z klas  $i, j$ ;
- Wszystkie klasyfikatory uczone tym samym algorytmem
- klasyfikator  $C_{ij}$  wskazuje dwie decyzje (1 or 0), klasyfikatory  $C_{ij}$  and  $C_{ji}$  równoważne

$$C_{ji}(\mathbf{x}) = 1 - C_{ij}(\mathbf{x})$$

	1	2	$p$	$\dots$	$q$	$n-1$	$n$
1	0						
2	0						
$p$	1	1	1			1	1
:							
$q$	1	1	1	0		1	1
$n-1$	0						
$n$	0						

Własny rysunek z pracy hab. 2001

# Zasady klasyfikacji w pairwise coupling

- Nowy przykład  $\mathbf{x}$ , przekazany na wszystkie klasyfikatory  $C_{ij}(\mathbf{x})$  w strukturze  $n^2$  (one against one)= konieczna reguła agregacji dla wypracowania decyzji i rozstrzygania konfliktów
- Najczęstsza reguła – wybierz klasę, która wygrała w największej licznie porównać parami klas /odmiana majority voting/.
- Możliwe rozszerzenia:
  - Oszacuj wiarygodność klasyfikatora binarnego  $P_{ij}$  (np. w trakcie uczenia)
  - Reguła - a weighted majority rule:
    - Wybierz klasę „ $i$ ” która maksymalizuje  $\sum_{j=1, i \neq j}^n P_{ij} \cdot C_{ij}(\mathbf{x})$
- Wprowadź odpowiedź „nie wiem” lub zasadę dynamicznego ważenia głosów

# Przykład oceny klasyfikatora $n^2$ z drzewa c4.5

Data set	Classification accuracy $DT$ (%)	Classification accuracy $n^2$ (%)	Improvement $n^2$ vs. $DT$ (%)
Automobile	85.5 $\pm$ 1.9	87.0 $\pm$ 1.9	1.5*
Cooc	54.0 $\pm$ 2.0	59.0 $\pm$ 1.7	5.0
Ecoli	79.7 $\pm$ 0.8	81.0 $\pm$ 1.7	1.3
Glass	70.7 $\pm$ 2.1	74.0 $\pm$ 1.1	3.3
Hist	71.3 $\pm$ 2.3	73.0 $\pm$ 1.8	1.7
Meta-data	47.2 $\pm$ 1.4	49.8 $\pm$ 1.4	2.6
Primary Tumor	40.2 $\pm$ 1.5	45.1 $\pm$ 1.2	4.9
Soybean-large	91.9 $\pm$ 0.7	92.4 $\pm$ 0.5	0.5*
Vowel	81.1 $\pm$ 1.1	83.7 $\pm$ 0.5	2.6
Yeast	49.1 $\pm$ 2.1	52.8 $\pm$ 1.8	3.7

# Inne zagadnienia

Inne warianty podstawowych algorytmów

- Modyfikacje bagging (Pasting small votes)
- Pośrednie modele (Arc-c4), inne boosting, gradient boosting
- Rotation Forest

ECOC – popularna wersja wieloklasowa

Agregacja odpowiedzi liczbowych (ang. numeric predictions)

- Zespoły specjalizowane dla trudnych danych
- ...

# Inne ciekawe wykorzystanie zespołów

Uogólnienia dla trudnych danych, np.

Niebalansowanie klas (zwłaszcza uogólnienia bagging)

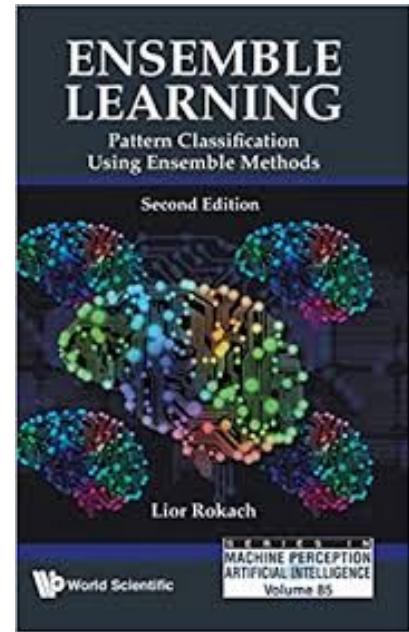
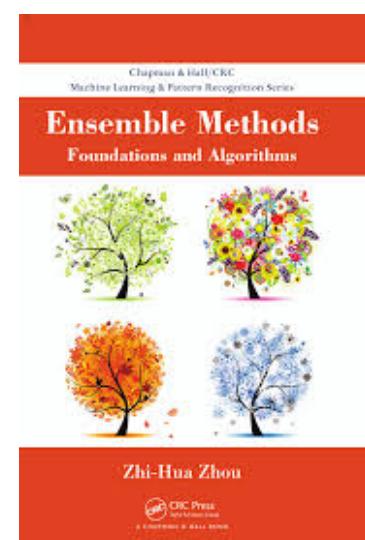
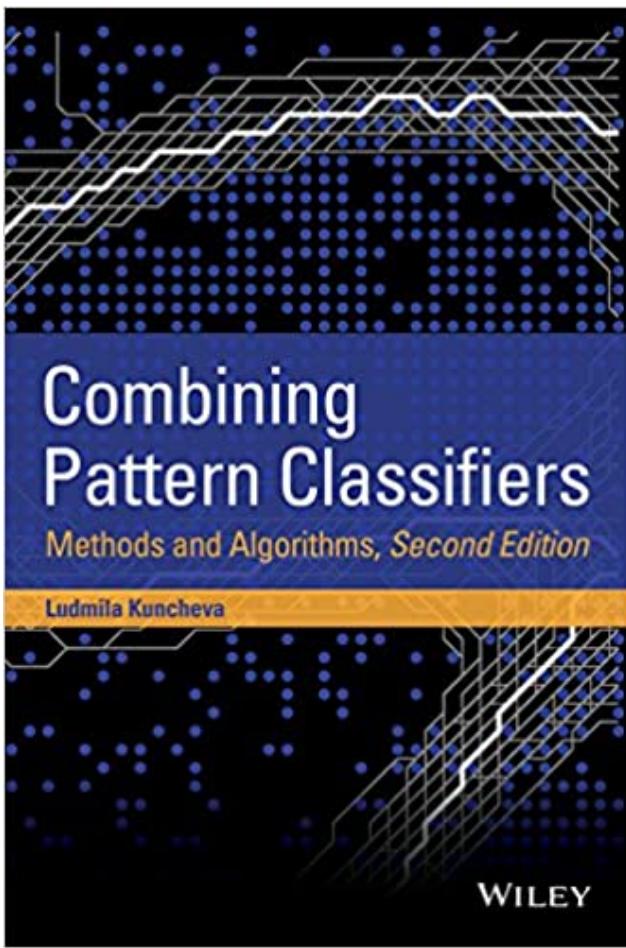
Zespoły klasyfikatorów dla **zmiennych strumieni danych**  
(wiele rozwiązań, nie tylko uogólnienia jak online bagging)

Klasyfikacja wielo-etykietowa (multi-labeled) – tzw. łańcuchy  
klasyfikatorów

Multi-view learning (lub odmiany self-learning) dla trybu  
uczenia częściowo-etykietowanego

Querry by committee w aktywnym uczeniu się

# Więcej odpowiedzi, radzę książki



# Odnośniki do literatury

- Intensywny rozwój od lat 90 poprzedniego wieku
- Wiele różnych propozycji
- Przykładowe pozycje:
  - L.Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, 2004 (large review + list of bibliography).
  - T.Dietterich, Ensemble methods in machine learning, 2000.
  - Using Correspondence Analysis to Combine Classifiers. C.Merz Machine Learning J. (1997)
  - Is Combining Classifiers with Stacking Better than Selecting the Best One? S.Dzeroski, B.Zenko, Machine Learning J. (2004)
  - G.Valentini, F.Masulli, Ensemble of learning machines, 2001 [obszerna lista referencyjna]
  - R.Polikar, Ensemble based systems in decision making, IEEE Circuits and Systems Magazine, vol. 6, no. 3, pp. 21–45, 2006.
  - W Polsce – przykładowo prace M.Woźniak i współpracownicy

# **Pytanie i komentarze?**

Dalszy kontakt:

[jerzy.stefanowski@cs.put.poznan.pl](mailto:jerzy.stefanowski@cs.put.poznan.pl)

<http://www.cs.put.poznan.pl/jstefanowski/>



**Fundusze  
Europejskie**  
Polska Cyfrowa

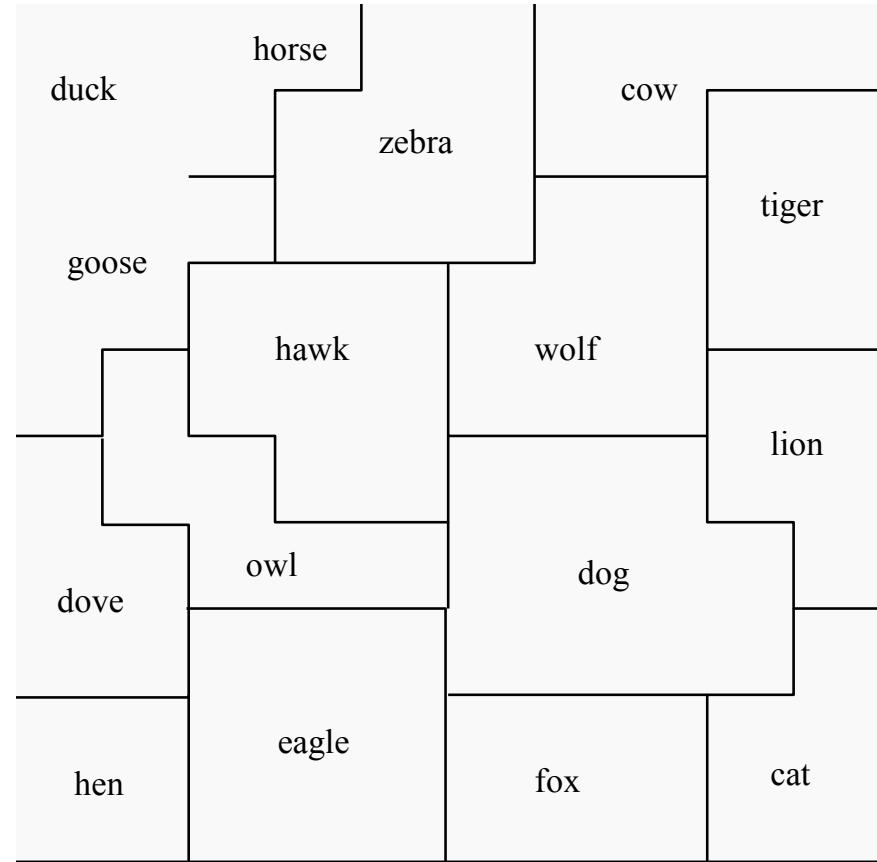
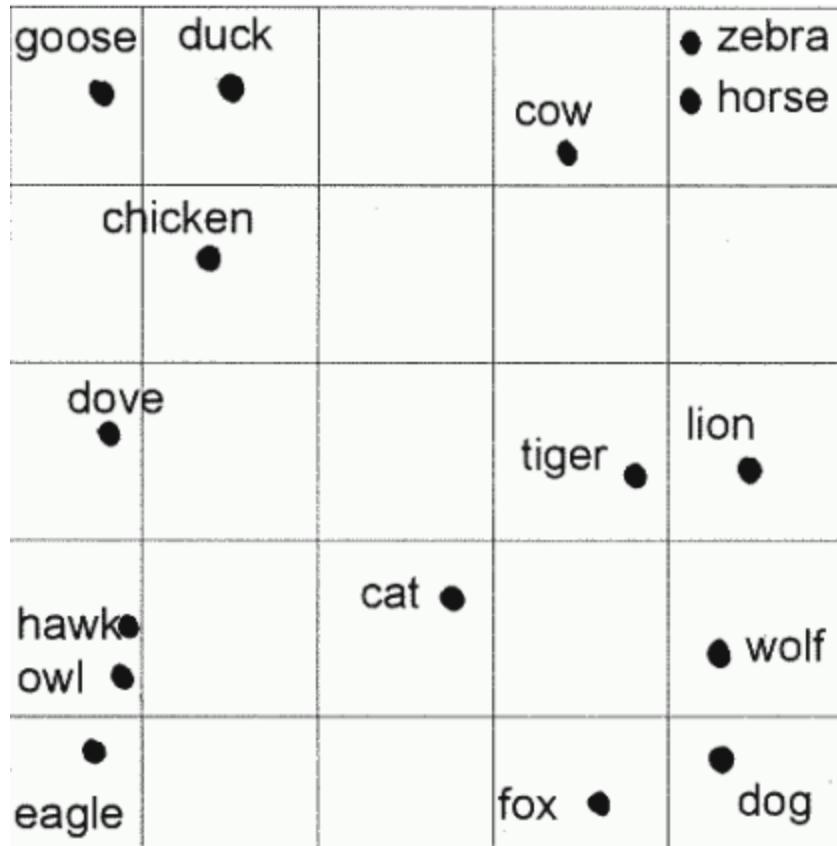


**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz  
Rozwoju Regionalnego



# Porównanie map MDS & SOM



MDS and SOM was used on data vectors from the previous page.

# **Uczenie nienadzorowane**

## **algorytmy grupowania wykład 11**

Jerzy Stefanowski

Instytut Informatyki PP

2021

Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI-TECH)  
projekt finansowany z środków Programu Operacyjnego Polska Cyfrowa  
POPC.03.02.00-00-0001/20



**Fundusze  
Europejskie**  
Polska Cyfrowa



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz  
Rozwoju Regionalnego



# Plan wykładu

- Przypomnienie algorytmów grupowania
  - Algorytm k-średnich
  - Algorytmy hierarchiczne
- Grupowanie z wykorzystaniem sieci neuronowych
- Sieci Kohonena
  - Siec LVQ
  - Siec SOM
- Podsumowanie

# Cele algorytmów grupowania

- Obiekt – przykład uczący - opisany za pomocą  $n$  zmiennych  $X_1, X_2, \dots, X_n$  jest punktem  $x=(x_1, \dots, x_n)$  w  $n$ -wymiarowej przestrzeni  $\Omega$
- Cel podziału na grupy ( $S$ ) → obiekty podobne (reprezentowane przez punkty znajdujące się blisko siebie w przestrzeni) przydzielone do tej samej grupy, a obiekty niepodobne (reprezentowane przez punkty leżące w dużej odległości w przestrzeni) znajdują się w różnych grupach

# Czym jest skupienie?

1. Zbiorem najbardziej podobnych obiektów
2. Podzbiór obiektów, dla których odległość jest mniejsza niż ich odległość od obiektów z innych skupień.
3. Podobszar wielowymiarowej przestrzeni zawierający odpowiednio dużą gęstość obiektów, oddzielony od innych podobszarów o dużej gęstości strefą rzadkiego występowania obiektów

# Przykłady zastosowań analizy skupień

- Zastosowania ekonomiczne:
  - Identyfikacja grup klientów bankowych (np. właścicieli kart kredytowych wg. sposobu wykorzystania kart oraz stylu życia, danych osobowych, demograficznych) → cele marketingowe.
  - Systemy rekommendacji produktów i usług.
  - Rynek usług ubezpieczeniowych (podobne grupy klientów).
  - Analiza sieci sprzedaży (np. czy punkty sprzedaży podobne pod względem społecznego sąsiedztwa liczby personelu, itp., przynoszą podobne obroty).
  - Poszukiwanie wspólnych rynków dla produktów.
  - Planowanie przestrzene, np. analiza nieruchomości
- Badania naukowe (biologia, medycyna, nauki społeczne)
- Analiza zachowań użytkowników serwisów WWW
- Rozpoznawanie obrazów, dźwięku
- Wiele innych

# Podział znanych metod

- Podziałowo-optymalizacyjne: Znajdź podział na zadaną liczbę skupień wg. zadanego kryterium.
- Metody hierarchiczne: Zbuduj drzewiastą strukturę skupień.
- Gęstościowo (Density-based): Poszukuj obszarów o większej gęstości występowania obserwacji
- Grid-based: wykorzystujące wielowymiarowy podział przestrzeni siatką ograniczeń
- Model-based: hipoteza co do własności modelu pewnego skupienia i procedura jego estymacji.

# Inne kryteria podziału [Jain przegląd]

Rodzaj rozwiązań algorytmicznych

- Podziałowo-optymalizacyjne
  - Optymalizacja kryterium, np. k-średnich
  - Mieszaniny rozkładów prawd. (EM)
  - Grafowe
- Hierarchiczne
  - AHC (różne metody łączenia od mniejszych skupień do większych)
  - Deglomeracyjne (podział większych grup)
  - Dostosowane do masywnych danych (BIRCH)
  - Wspierające opis probabilistyczny (COBWEB)
- Inne

Skupienia: jednoznaczny przydział obiektu vs. rozmyty

Tryb przetwarzania danych (pełen dostęp vs. przyrostowy)

Więcej: A. Jain: Data Clustering: 50 Years Beyond K-Means

# Problemy do rozstrzygnięcia przed wyborem metody/algorytmu

- Jak odwzorować obiekty w przestrzeni?
  - Wybór zmiennych
  - Normalizacja zmiennych
- Jak mierzyć odległości między obiektami?
  - Przypomnienie wcześniejszego wykładu z kNN
- Jaką metodę grupowania zastosować?

# Różny zakres danych liczbowych

Normalizacja ma na celu doprowadzenie obiektów lub zmiennych do porównywalnych wielkości. Problem ten dotyczy zmiennych mierzonych w różnych jednostkach (np. sztuki, czas, waluta).

## Przykład

- Rozważmy 3 obiekty i dwie zmienne: wiek osoby mierzony w latach i jej dochód mierzony w złotych lub tys. zł.

Zmienna ->	X	Y1	Y2
Osoba	Wiek	Dochód	Dochód
	(w latach)	(w zł)	( w tys. zł)
A	35	12000	12,0
B	37	6700	6,7
C	45	7000	7,0

# Podobieństwo – podstawa analizy skupień

- Ciągle dyskusyjne – zwłaszcza w nietechnicznych zastosowaniach



- The real meaning of similarity is a philosophical question. We will take a more pragmatic approach
- Depends on representation and algorithm. For many rep./alg., easier to think in terms of a distance (rather than similarity) between vectors.
- Prościej mówić o odległościach między obserwacjami
  - Zwłaszcza jak są matematycznie dobrze zdefiniowane
  - Metryka odległości

# Algorytmy podziałowo – optymalizacyjne

- Zadanie: Podzielenie zbioru obserwacji na  $K$  zbiorów elementów (skupień  $C$ ), które są jak najbardziej jednorodne
- Jednorodność – funkcja oceny
- Intuicja → zmienność wewnętrzskupieniowa  $wc(C)$  i zmienność międzyskupieniowa  $bc(C)$

Mögliwe są różne sposoby zdefiniowania

- np. wybierzmy środki skupień  $\mathbf{r}_k$  (centroidy)  $\mathbf{r}_k = \frac{1}{n_k} \sum_{\mathbf{x} \in C_k} \mathbf{x}$
- Co prowadzi do

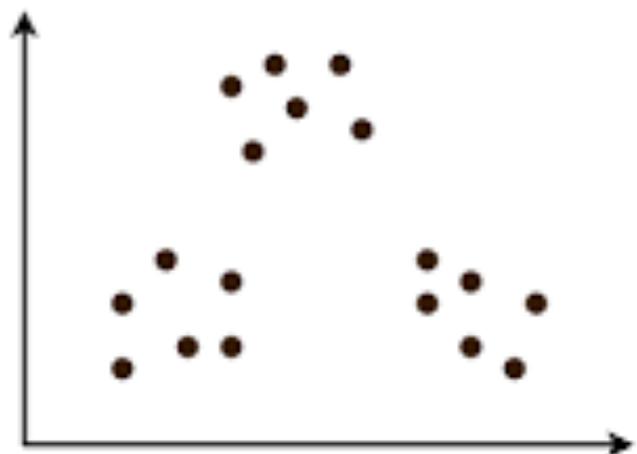
$$wc(C) = \sum_{k=1}^K \sum_{\mathbf{x} \in C_k} d(\mathbf{x}, \mathbf{r}_k)^2$$

$$bc(C) = \sum_{1 \leq j < k \leq K} d(\mathbf{r}_j, \mathbf{r}_k)^2$$

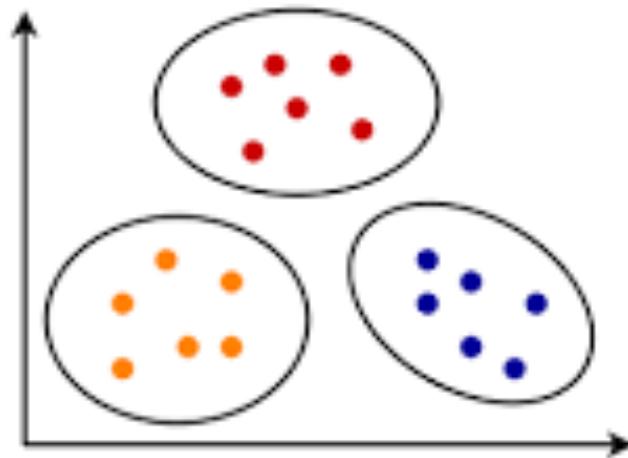
# Algorytm k średnich

- Cel:  $K$  - średnich  $\rightarrow$  minimalizacja  $wc(C)$
- Przeszukiwanie przestrzeni możliwych przypisań  $\rightarrow$  bardzo kosztowne (oszacowanie w ks. Koronackiego)
- Problem optymalizacji kombinatorycznej  $\rightarrow$  systematyczne przeszukiwanie metodą iteracyjnego udoskonalania:
  - Rozpocznij od rozwiązania początkowego (losowego).
  - Ponownie przypisz punkty do skupień tak, aby otrzymać największą zmianę w funkcji oceny.
  - Przelicz zaktualizowane środki skupień, ...
  - Postępuj aż do momentu, w którym nie ma już żadnych zmian w funkcji oceny lub w składzie grup.
- Zachłanne przeszukiwanie  $\rightarrow$  proste i prowadzi do co najmniej lokalnego minimum. Różne modyfikacje, np. rozpoczęnięcia od kilku rozwiązań startowych
- Złożoność algorytmu K - średnich  $\rightarrow O(Knl)$

# Ilustracja k-średnich



Before K-Means



After K-Means

# Ustalanie liczby skupień i startowych centroidów

Liczبę skupień wybiera się na podstawie przesłanek merytorycznych albo szacuje się je metodami hierarchicznymi. Można dokonać obliczeń dla wszystkich wartości  $k$  z ustalonego przedziału:

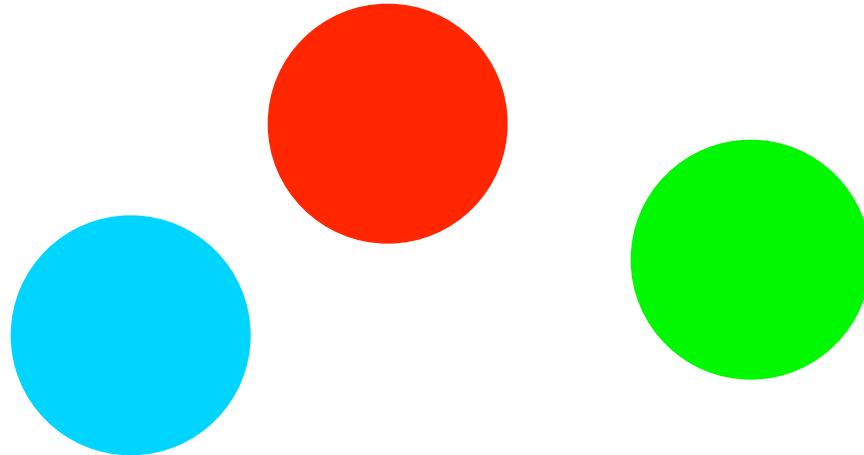
$$k_{\min} \leq k \leq k_{\max}$$

Możliwe są różne podejścia:

1. Arbitralny sposób np. przyjmuje się współrzędne pierwszych  $k$  obiektów jako załączki środków ciężkości
2. Losowy wybór środków ciężkości, przy czym może to być losowy wybór  $k$  obiektów ze zbioru danych albo losowy wybór  $k$  punktów przestrzeni niekoniecznie pokrywających się z położeniem obiektów
3. Wykorzystanie algorytmu optymalizującego w pewien sposób położenie początkowych środków ciężkości np. przez uwzględnianie  $k$  obiektów leżących daleko względem siebie
4. Przyjęcie jako początkowych środków ciężkości uzyskanych na podstawie podziału otrzymanego inną metodą, głównie jedną z metod hierarchicznych

# Pewne ukierunkowanie K-średnich

- Tworzy się „kuliste” kształty skupień



- Co z obserwacjami odstającymi i nieregularnymi kształtami skupień?

# K-means krótkie podsumowanie

## Zalety

- Proste i łatwe do zrozumienia
- Reprezentacja skupień jako centroidy

## Wady

- Jawne podanie liczby skupień
- Wszystkie przykłady muszą być przydzielone do skupień
- Problem z outliers (za duża wrażliwość)
- Ukierunkowanie na jednorodne „sferyczne” kształty skupień

# Dalsze rozszerzenia k-średnich

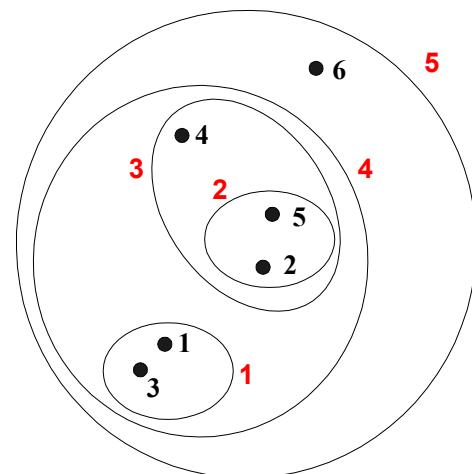
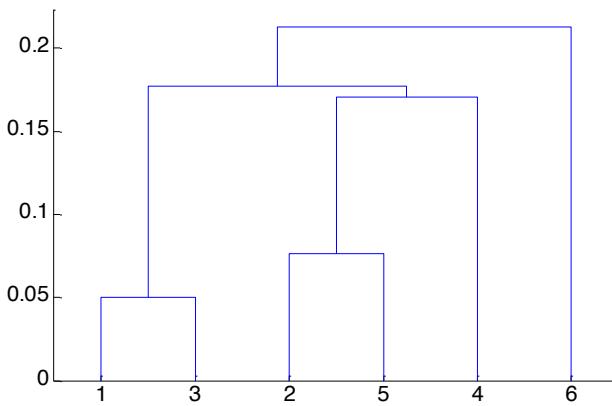
- Rozmyte k-means (Fuzzy ISODATA)
- Wersja k-medoids
- Rozszerzenia dla przetwarzania dużych wolumenów danych, np. PAM
- Inspiracje dla modeli statystycznych (EM)

Kolejny wykład omawia niektóre z nich

Warto zapoznać się z książką S.Wierzchoń, M.Kłopotek:  
Algorytmy analizy skupień. WNT 2015

# Grupowanie hierarchiczne

- Tworzy się stopniowo hierarchię zawierających się skupisk
  - Połączenie lub podział podzbiorów obiektów
- Wizualizacja – struktura drzewa nazwana **dendrogramem**



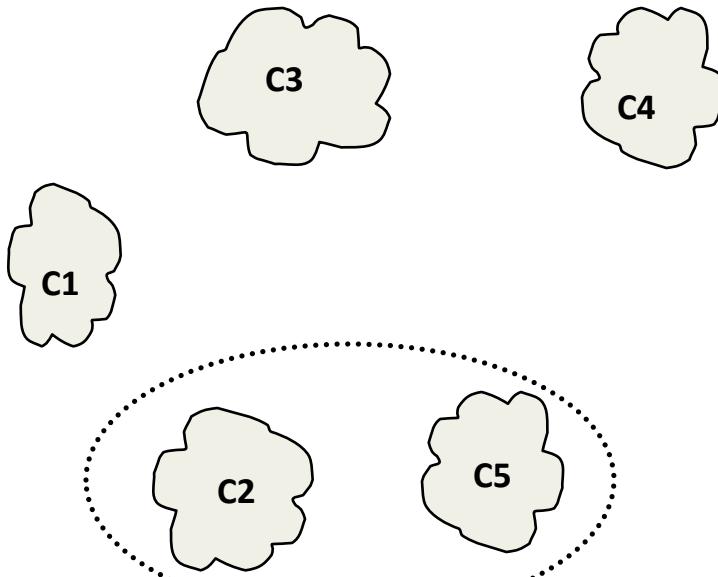
## Hierarchiczne metody aglomeracyjne - algorytm

1. W macierzy odległości znajduje się parę skupień najbliższych sobie.
2. Redukuje się liczbę klas łącząc znalezioną parę
3. Przekształca się macierz odległości metodą wybraną jako kryterium klasyfikacji
4. Powtarza się kroki 1- 3 dopóki nie powstanie jedna klasa zawierająca wszystkie skupienia.

# Jak przeliczać macierz odległości?

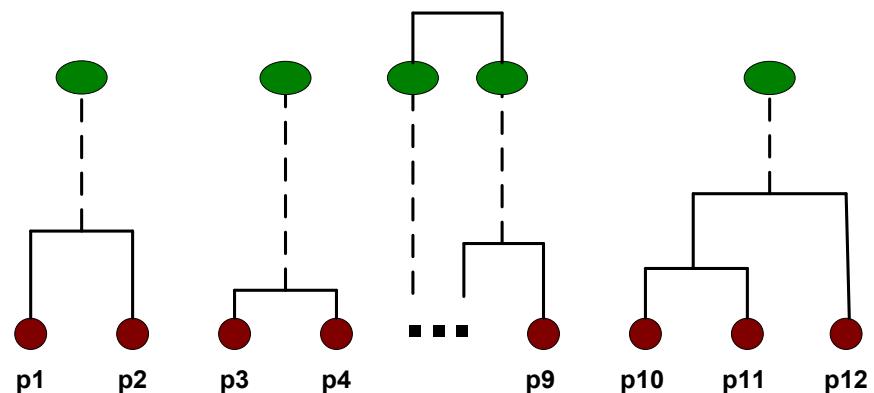
Łączymy dwa skupiska (C2 i C5) i aktualizujemy macierz odległości

Metody hierarchiczne różnią sposobem łączenia skupisk (ang. Linkage method)



	c1	c2	c3	c4	c5
c1					
c2					
c3					
c4					
c5					

Macierz odległości



# Hierarchiczne grupowanie wybór metody łączenia

1. Najbliższego sąsiedztwa (*Single linkage, Nearest neighbor*)
2. Najdalszego sąsiedztwa (*Complete linkage, Furthest neighbor*)
3. Mediany (*Median clustering*)
4. Środka ciężkości (*Centroid clustering*)
5. Średniej odległości wewnętrz skupień  
(*Average linkage within groups*)
6. Średniej odległości między skupieniami  
(*Average linkage between groups*)
7. Minimalnej wariancji Warda (*Ward's method*)

# Odległości między skupieniami

Single linkage  
minimum distance:

$$d_{\min}(C_i, C_j) = \min_{p \in C_i, p' \in C_j} \|p - p'\|$$

Complete linkage  
maximum distance:

$$d_{\max}(C_i, C_j) = \max_{p \in C_i, p' \in C_j} \|p - p'\|$$

mean distance:

$$d_{mean}(C_i, C_j) = \|m_i - m_j\|$$

average distance:

$$d_{ave}(C_i, C_j) = 1 / (n_i n_j) \sum_{p \in C_i} \sum_{p' \in C_j} \|p - p'\|$$

$m_i$  Jest średnią obiektów z  $C_i$      $n_i$  Jest liczbą obiektów w skupisku  $C_i$

# Single Link Agglomerative Clustering

- Użyj maksymalnego podobieństwa dwóch obiektów:

$$sim(c_i, c_j) = \max_{x \in c_i, y \in c_j} sim(x, y)$$

- Prowadzi do „(long and thin) clusters due to *chaining effect*” (efekt łańcuchowy); prowadzić do formowania grup niejednorodnych (heterogenicznych);
  - Dogodne w specyficznych zastosowaniach
- Pozwala na wykrycie **obserwacji odstających**, nie należących do żadnej z grup, i warto przeprowadzić klasyfikację za jej pomocą na samym początku, aby wyeliminować takie obserwacje i przejść bez nich do właściwej części analizy

# Complete Link Agglomerative Clustering

- Użyj maksymalnej odległości – minimalnego podobieństwa

$$sim(c_i, c_j) = \min_{x \in c_i, y \in c_j} sim(x, y)$$

- Ukierunkowana do “tight,” spherical clusters
- Metoda zalecana gdy, kiedy obiekty faktycznie formują naturalnie oddzielone "kępki". Metoda ta nie jest odpowiednia, jeśli skupienia są w jakiś sposób wydłużone lub mają naturę "łańcucha".

# Wrażliwość na dobór metod łączenia skupień

Diagram dla 22 przyp.

Pojedyncze wiązanie

Odległości euklidesowe

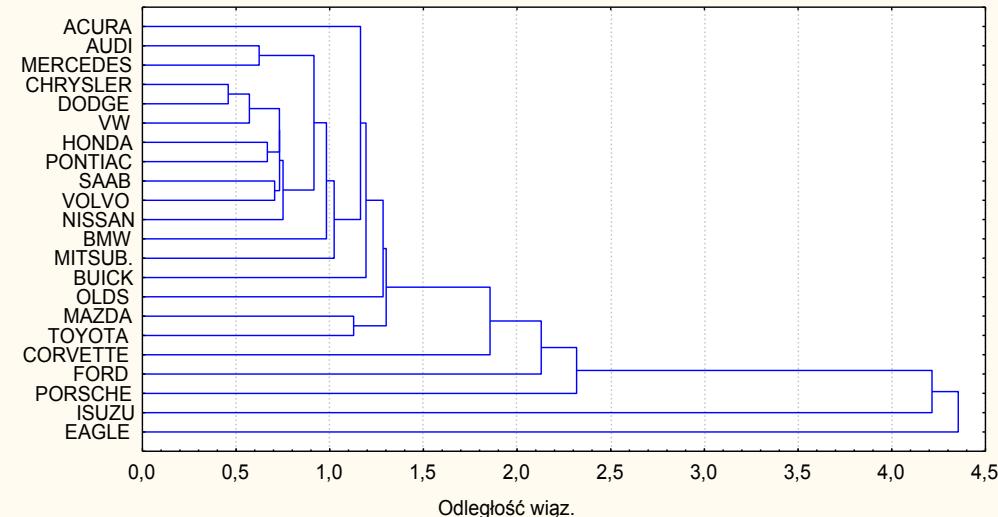
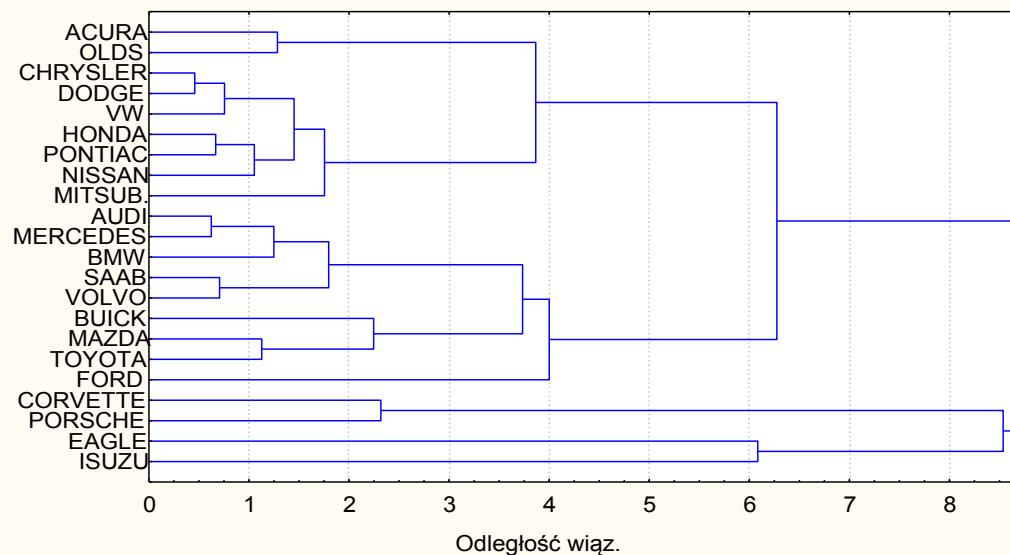


Diagram dla 22 przyp.

Metoda Warda

Odległości euklidesowe



Rysunki – z własnego  
uruchomienia Statsoft Statistica

# Metoda średnich połączeń [Unweighted pair-group average]

- W metodzie tej odległość między dwoma skupieniami oblicza się jako średnią odległość między wszystkimi parami obiektów należących do dwóch różnych skupień
- Metoda ta jest efektywna, gdy obiekty formują naturalnie oddzielone "kępki", ale zdaje także egzamin w przypadku skupień wydłużonych, mających charakter "łańcucha"

# Metoda ważonych środków ciężkości (mediany) [Weighted pair-group centroid]

- Jest to metoda podobna jak poprzednia, z tym wyjątkiem, że w obliczeniach wprowadza się „ważenie”, aby uwzględnić różnice między wielkościami skupień (tzn. liczbą zawartych w nich obiektów).
- Zatem, metoda ta jest lepsza od poprzedniej w sytuacji, gdy istnieją (lub podejrzewamy, że istnieją) znaczne różnice w rozmiarach (liczności) skupień

# Metody łączenia – Ward method

- Gdy powiększamy jedno ze skupień  $C_k$ , wariancja wewnętrzgrupowa (liczona przez kwadraty odchyлеń od średnich w zbiorach  $C_k$ ) rośnie.
- Metoda polega na takim powiększaniu zbiorów  $C_k$ , która zapewnia **najmniejszy przyrost tej wariancji** dla danej iteracji.
- Kryterium grupowania jednostek: minimum zróżnicowania wektorów cech  $x_j$  tworzących zbiór  $C_k$  ( $k = 1, \dots, K$ ) względem wartości średnich w tych zbiorach.
- Ogólnie, metoda ta jest traktowana jako bardzo efektywna, chociaż zmierza do tworzenia skupień o małej wielkości → zrównoważone drzewa o wielu elementach
- Ważne – powiązanie z miarą odległości między obiektami (Pearson vs. inne)

# Przykłady użycia metody Warda

## Cars data

Diagram dla 22 przyp.

Metoda Warda

Odległości euklidesowe

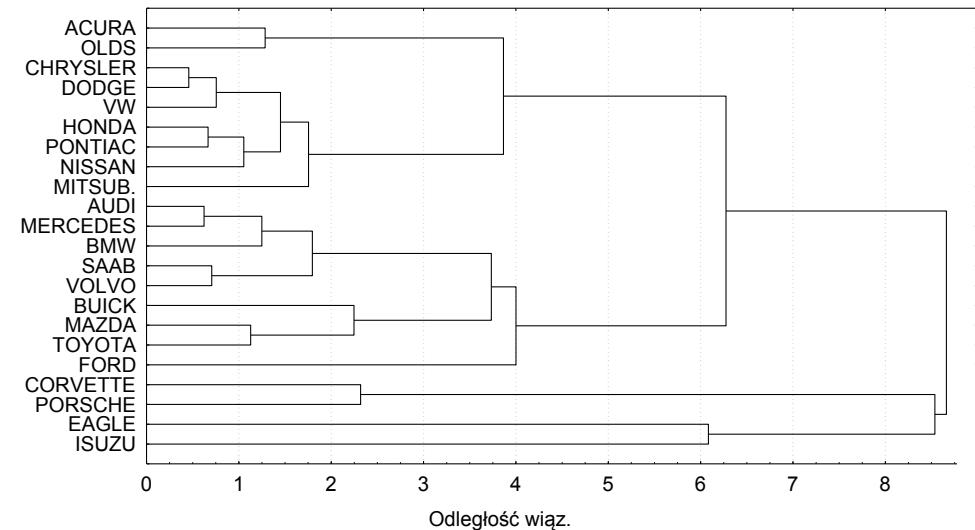
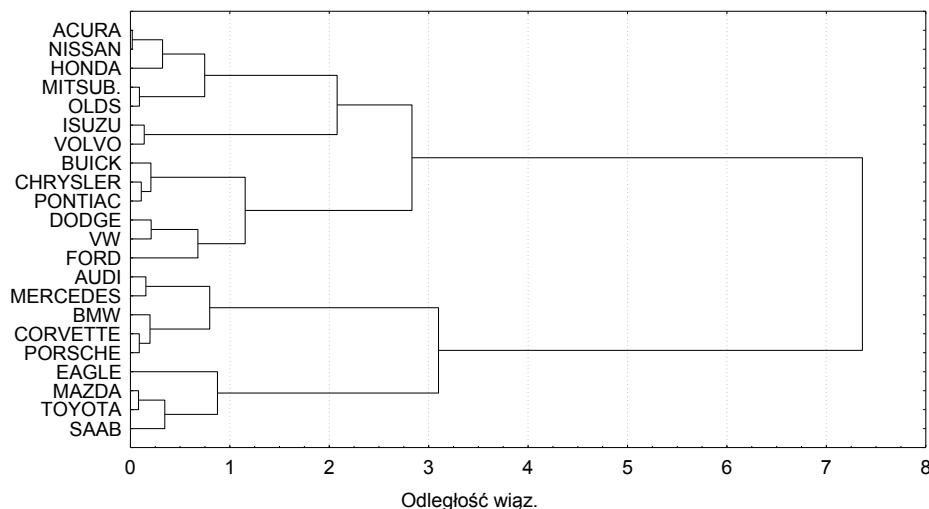


Diagram dla 22 przyp.

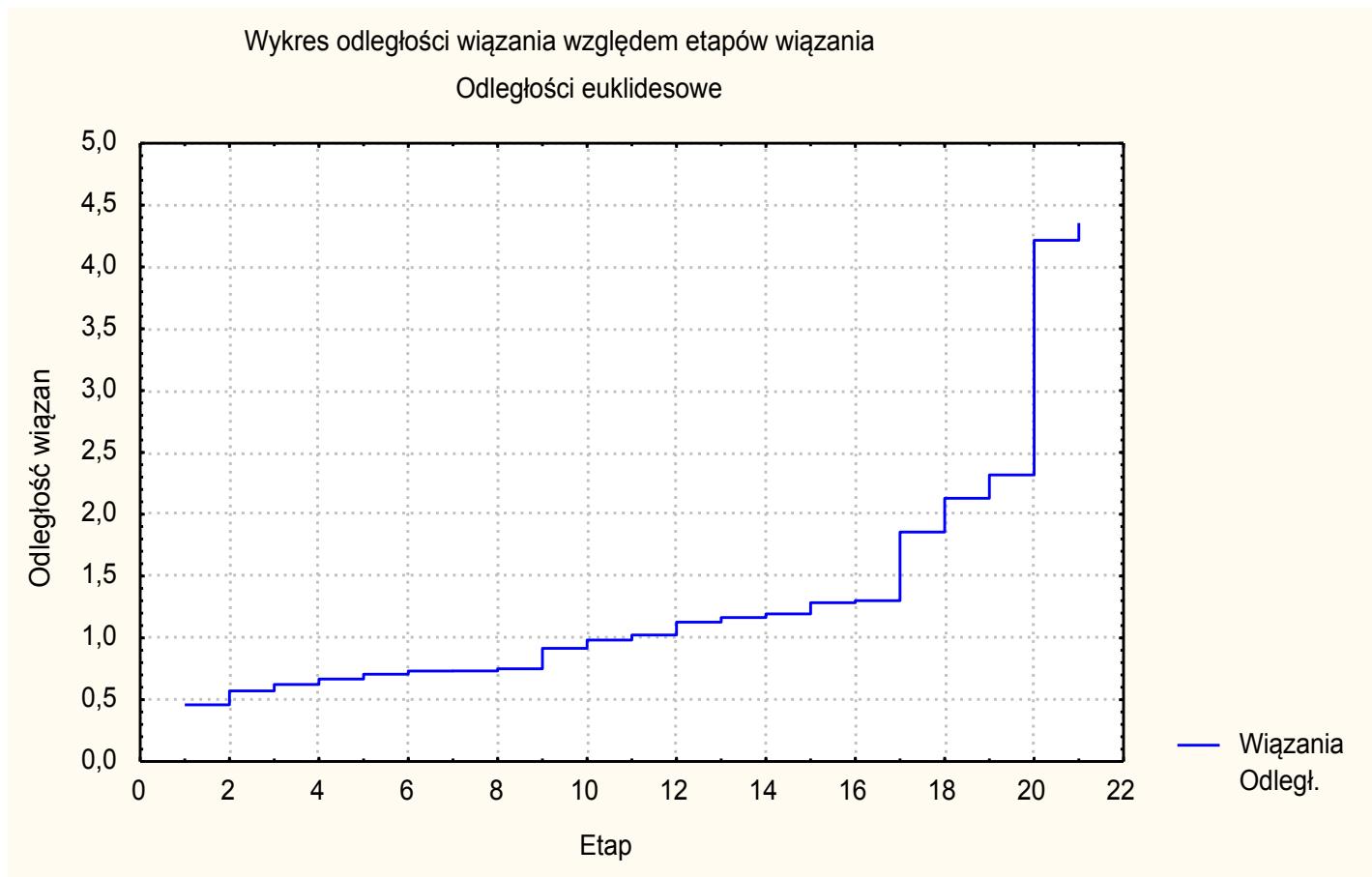
Metoda Warda

1-r Pearsona



# AHC – jak odnaleźć liczbę skupień?

Znajdź punkt przegięcia („kolanko”) wykresu



# Sieci neuronowe - samoorganizujące

Propozycja T.Kohonena

# Typowe zadania dla sieci

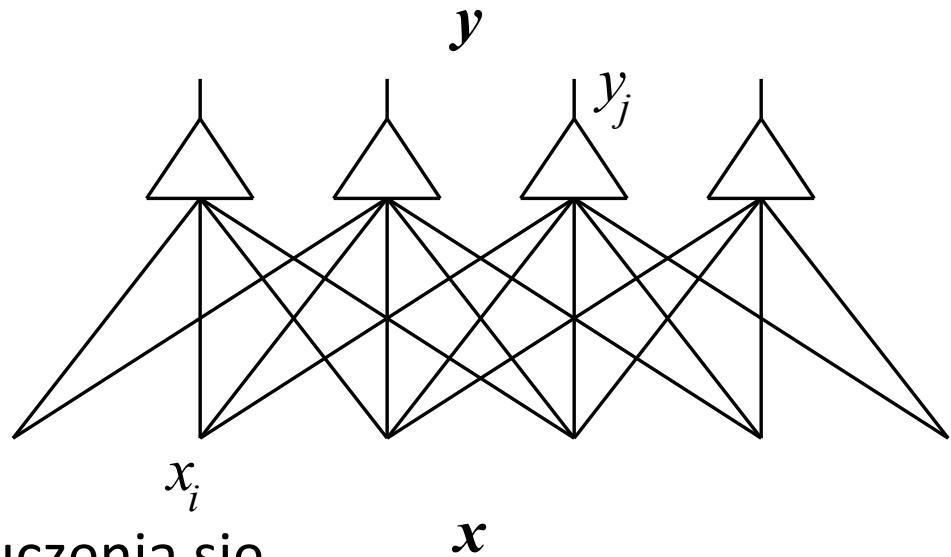
- **Grupowanie obserwacji:** Sieć w wyniku procesu uczenia dokonuje podziału przykładów uczących na klasy (grupy) przykładów podobnych do siebie i przyporządkowuje każdej klasie różne elementy wyjściowe sieci – **sieci LVQ**.
- **Tworzenie mapy cech:** Dane wejściowe transformowane są z wielowymiarowej przestrzeni przykładów w „małowymiarową” przestrzeń ich cech charakterystycznych. Elementy warstwy wyjściowej są geometrycznie uporządkowane. Wymaga się, aby podobne przykłady wejściowe generowały aktywność bliskich geometrycznie elementów wyjściowych - **Sieci SOM Kohonena**.
- **Analiza czynników głównych:** Sieć posiada wyjście wieloelementowe, a każdy z elementów odpowiada za jeden z tzw. czynników głównych, według których określone jest podobieństwo sygnałów wejściowych.

# Wprowadzenie do sieci Kohonena

- Inny tryb uczenia się:
  - Bez nadzoru (brak informacji  $y$  o zadanym wyjściu; tylko opis przykładów  $x$ )
  - Sieć sama powinna wykrywać istotne zależności w danych wejściowych, badać podobieństwo wektorów  $x$ , rozpoznawania cech istotnych czy regularności bez „nadzoru”
  - Typowe zastosowanie → grupowanie, kodowanie i kompresja, projekcja wielowymiarowa, wykrywanie cech istotnych.
  - Kluczowe jest badanie podobieństwa wektorów (wejścia, wag),
    - Miara iloczynu skalarnego wektora wag i wektora wejściowego
- **Zasady uczenia się konkurencyjnego** (przez współzawodnictwo)
- Tylko zwycięskie neurony lub ich sąsiedzi są nauczani (modyfikacja wag)
- Na ogólną prostszą topologię sieci

# Podstawowa sieć Kohonena LVQ

- Celem jest grupowanie wektorów wejściowych  $\mathbf{x}$
- Istota działania → podobieństwo wektorów
- Podobne wektory powinny pobudzać te same neurony
- Prosta topologia



- gdzie  $y_j = \mathbf{w}_j \mathbf{x} = \sum_i w_{ij} x_i$
- Reguła konkurencyjnego uczenia się

# Wektory i miary podobieństwa

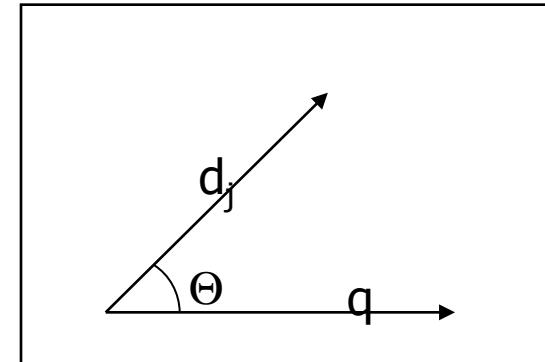
- Dany jest zbiór uczący  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$
- Podobieństwo dwóch wektorów – odległość Euklidesowa

$$\|\mathbf{x}_i - \mathbf{x}_j\| = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T \cdot (\mathbf{x}_i - \mathbf{x}_j)}$$

- Równoważna miara cosinusowa (kątowa)

$$\cos(\theta) = \frac{\mathbf{x}_i^T \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \cdot \|\mathbf{x}_j\|}$$

$$\cos \theta_j = \frac{\mathbf{a}_j^T \mathbf{q}}{\|\mathbf{a}_j\|_2 \|\mathbf{q}\|_2} = \frac{\sum_{i=1}^t a_{ij} q_i}{\sqrt{\sum_{i=1}^t a_{ij}^2} \sqrt{\sum_{i=1}^t q_i^2}}$$



# Sieć Kohonena - LVQ

- Przetwarza się kolejne wektory  $x$  poszukując  $p$  grup – odpowiadając im wybrane neurony
- Przed rozpoczęciem uczenia wektory wag są inicjowane losowo (małe liczby z przedziału  $-0.5;0.5$ )
- Wektory wag są normalizowane dla kolejnych  $p$  neuronów

$$\hat{\mathbf{w}}_i = \frac{\mathbf{w}_i}{\|\mathbf{w}_i\|}$$

- Stosuję się regułą „zwycięzca bierze wszystko” w celu identyfikacji neuronu zwycięzcy.
- Wagi TYLKO neuronu zwycięskiego podlegają modyfikacji

# Reguła „zwycięzca bierze wszystko”

- Określenie zwycięzcy:

$$\| \mathbf{x} - \hat{\mathbf{w}}_m \| = \min_{i=1,\dots,p} \| \mathbf{x} - \hat{\mathbf{w}}_i \|$$

- Alternatywnie iloczyn skalarny

$$\hat{\mathbf{w}}_m^T \mathbf{x} = \max_{i=1,\dots,p} \hat{\mathbf{w}}_i^T \mathbf{x}$$

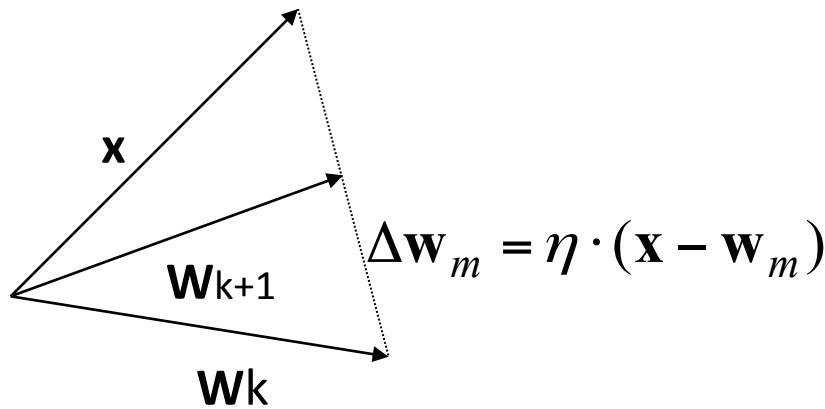
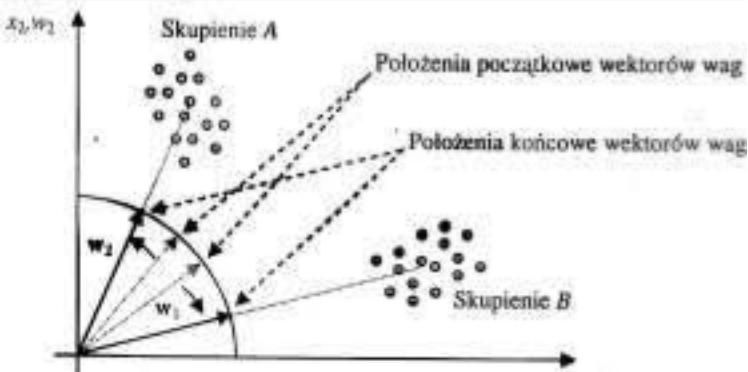
- Zwycięzcą jest jeden neuron  $m$ . Korekcja wag  $\mathbf{w}_m$  odbywa się wyłącznie dla neuronu zwycięzcy według reguły:
- $\eta$  - stała uczenia (na ogólny między 0.1 a 0.7)  $\Delta \mathbf{w}_m = \eta \cdot (\mathbf{x} - \mathbf{w}_m)$
- Przykład – interpretacja geometryczna

# Uczenie zwycięskiego neuronu

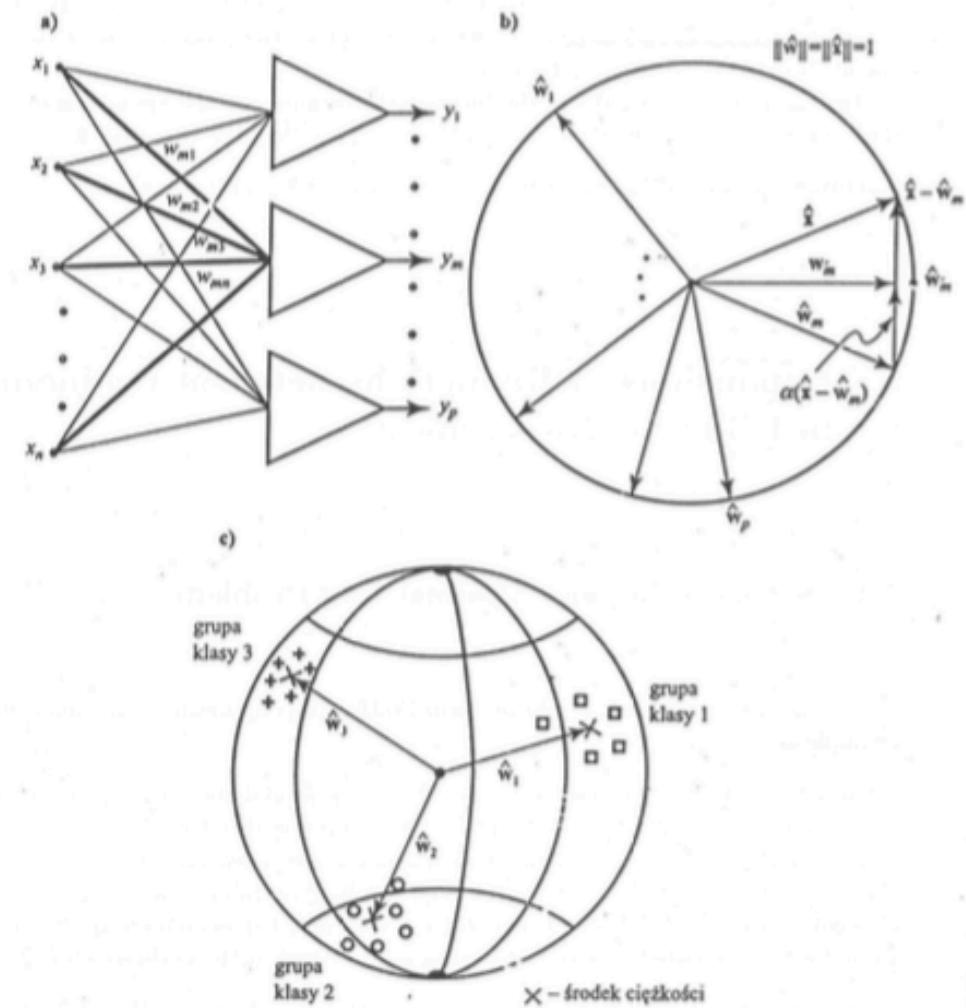
Po odnalezieniu zwycięskiego neuronu dokonuje się aktualizacji wag (k numer kolejnego kroku)

$$\mathbf{w}_m^{k+1} = \mathbf{w}_m^k + \eta \cdot (\mathbf{x} - \hat{\mathbf{w}}_m^k)$$

$$\hat{\mathbf{w}}_m^{k+1} = \frac{\mathbf{w}_m^{k+1}}{\|\mathbf{w}_m^{k+1}\|}$$

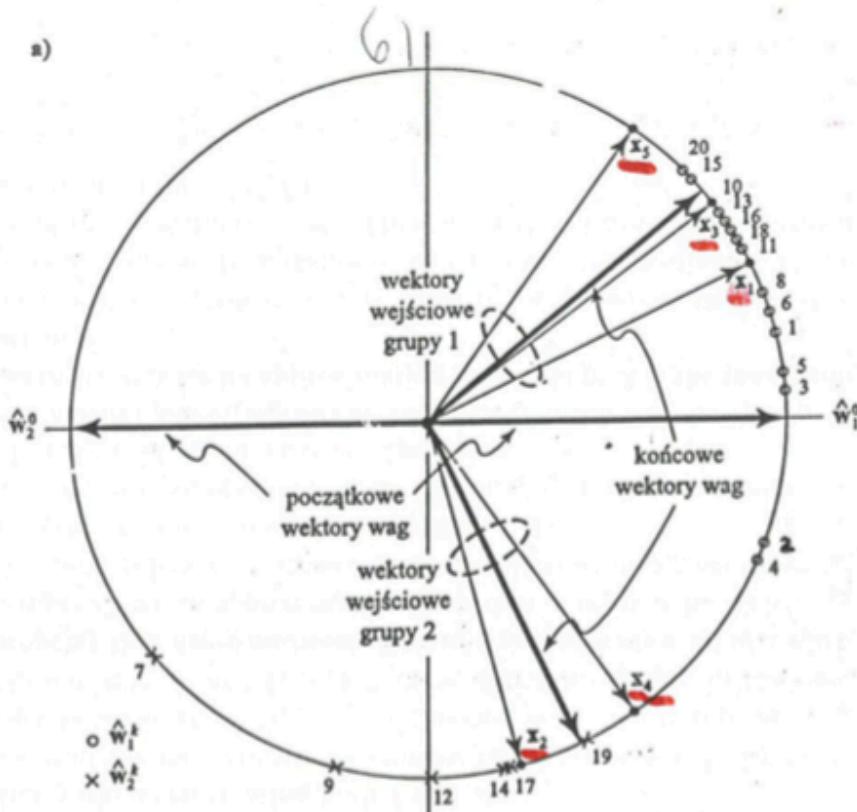


# Ilustracja uczenia sieci LVQ



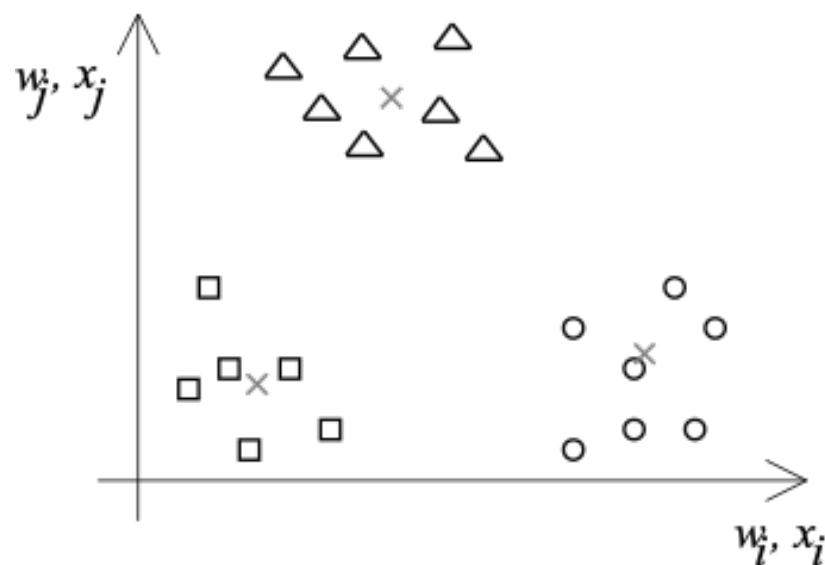
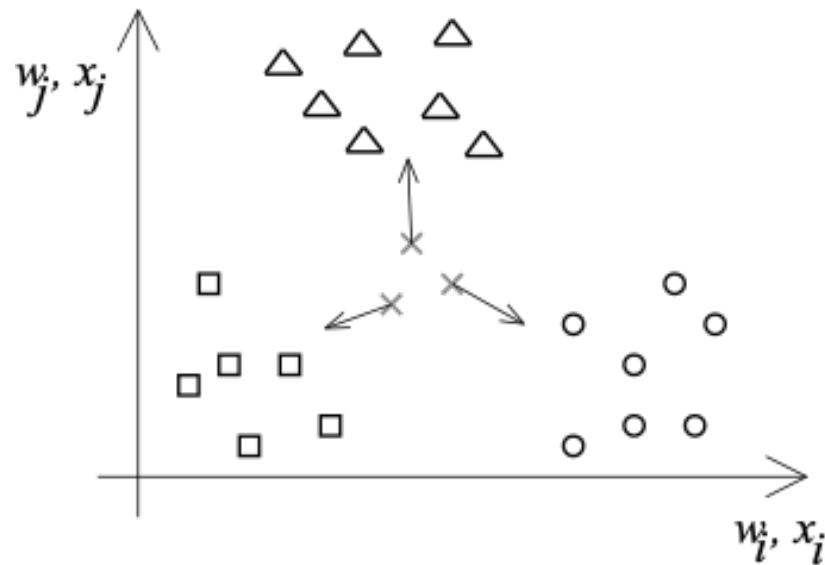
Rys. 7.6. Uczenie się z rywalizacją: a) warstwa ucząca się, b) interpretacja geometryczna kroku uczenia, c) wektory wag na kuli jednostkowej ( $p = n = 3$ ) (grubszego linie odpowiadają modyfikowanym wagom)

# Ilustracja procesu uczenia dwóch skupisk



Krok $k$	$\hat{w}_1^k$	$\hat{w}_2^k$
1	18,46	-180,00
2	-30,77	
3	7,11	
4	-31,45	
5	7,11	
6	31,45	
7		-130,22
8	34,43	
9		-100,00
10	43,78	
11	40,33	
12		-90,00
13	42,67	
14		-80,02
15	47,90	
16	42,39	
17		-80,01
18	43,69	
19		-75,01
20	48,42	

# Wspólna wizualizacja wag i przykładów

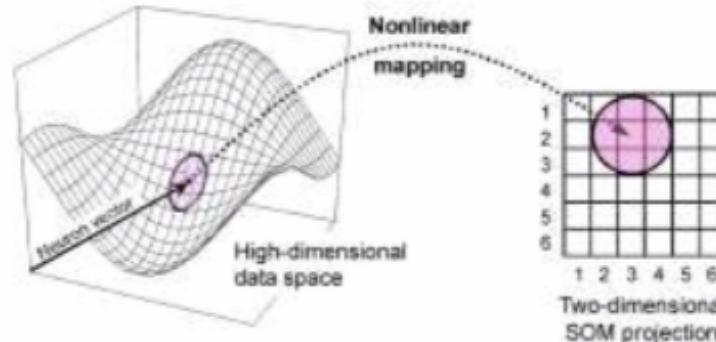


# Kilka uwag o uczeniu sieci

- Po zakończeniu uczenia znormalizowane wektory wag (neuronów) wskazują środki ciężkości wykrytych grup obserwacji → analogia do k-srednich.
- Dobór wag początkowych – rozrzucenie po przestrzeni (hiperkuli)
- Problem doboru liczby neuronów (martwe neurony)
- Tzw. techniki sumienia → „sumienie” ten sam neuron nie zwycięża zbyt często
- Stopniowe zmniejszanie prędkości uczenia
- Iteracyjne powtarzanie prezentacji przykładów
- W ostatnim kroku – „kalibracja” sieci

# Odzwierowanie cech istotnych

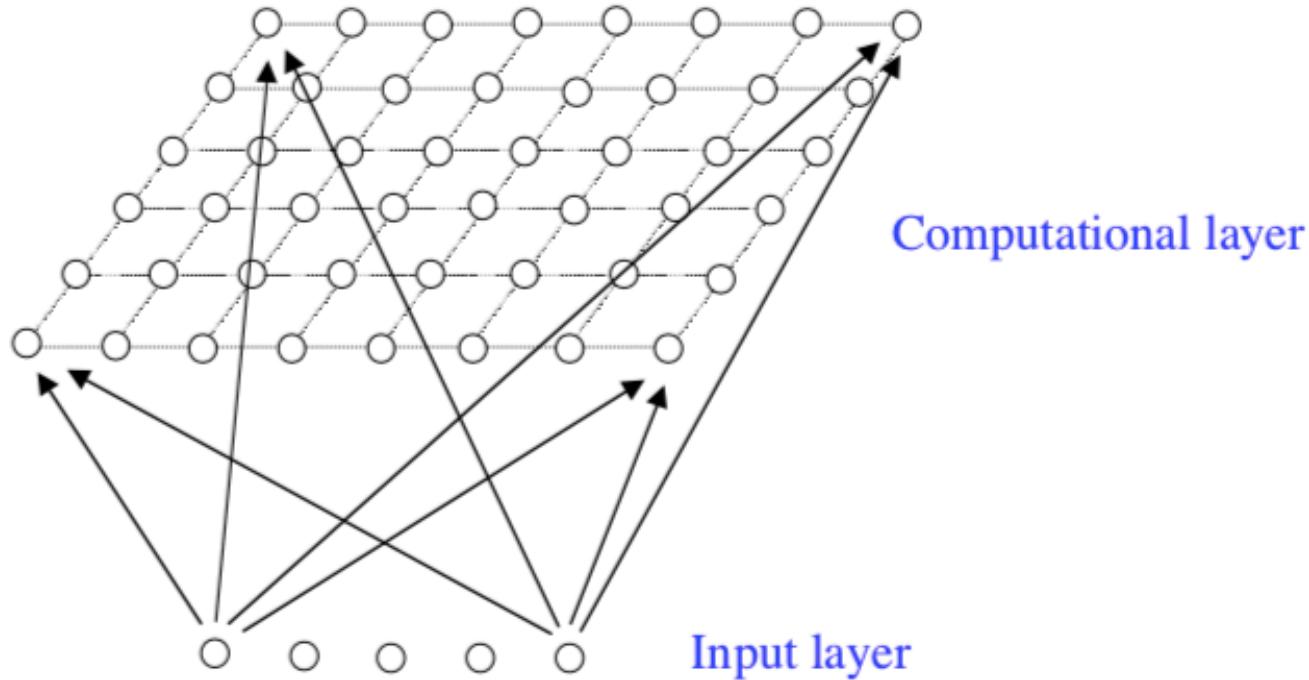
- W eksploracji danych duże znaczenie ma transformacja wysoce-wielowymiarowych danych wejściowych w małowymiarowe przestrzenie tak aby zauważać pewną harmonijną strukturę danych
- Podejścia analityczne (projekcje w statystyce), np. PCA, Skalowanie wielowymiarowe MDS -> patrz niezależny wykład dr Susmagi
- SOM – projekcja nieliniowa z zachowaniem bliskości położenia przykładów na warstwie wizualizacyjnej



# Sieci SOM

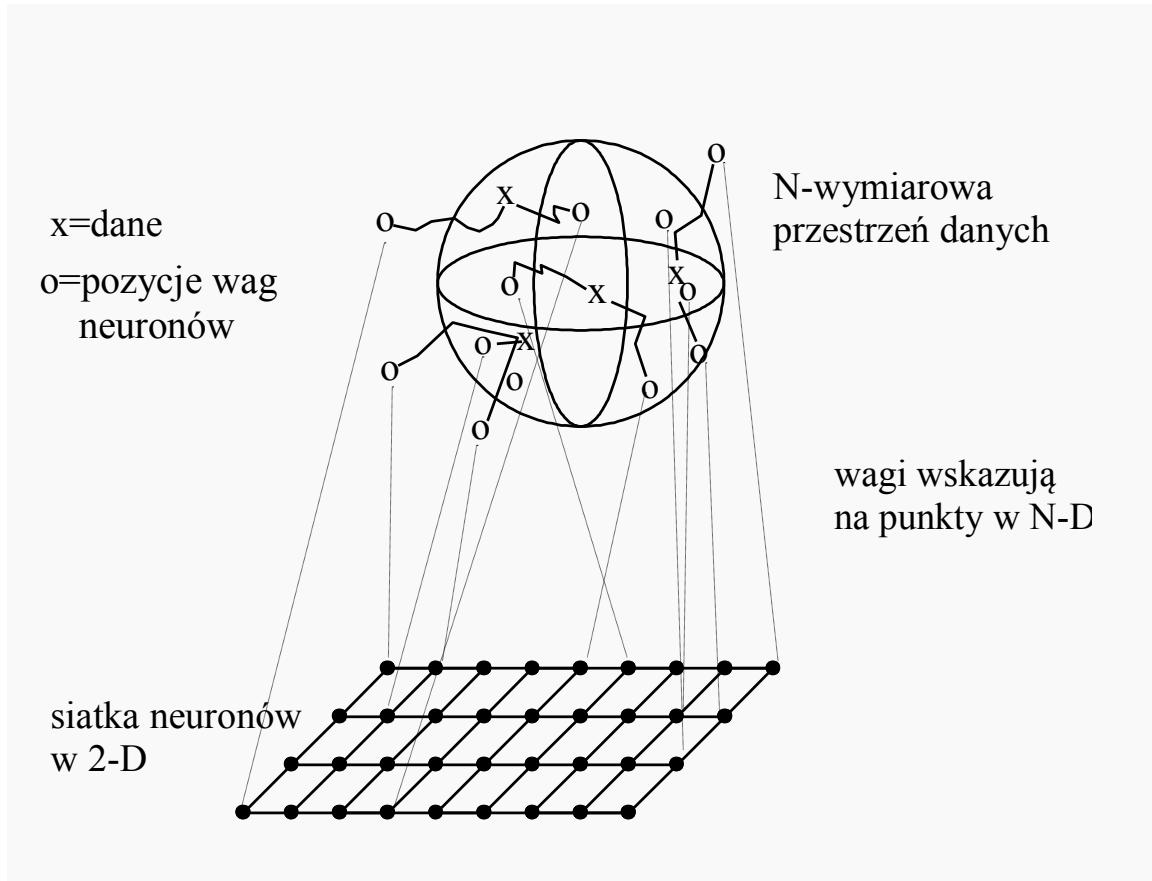
- Podstawą odwzorowania takie uporządkowanie neuronów, takie że położenie zwycięskich neuronów niesie informacje
- Topologia → relacja sąsiedztwa
- Podobne przykłady wejściowe  $\times$  powinny aktywizować sąsiednie neurony
- „Gęstość” wzorców w zredukowanej przestrzeni musi odpowiadać gęstości przykładów w oryginalnej przestrzeni

# Typowa architektura sieci SOM



Wielowymiarowe wejścia  $x$  podane na neurony w warstwie wyjściowej, która może być specjalnie wizualizowana

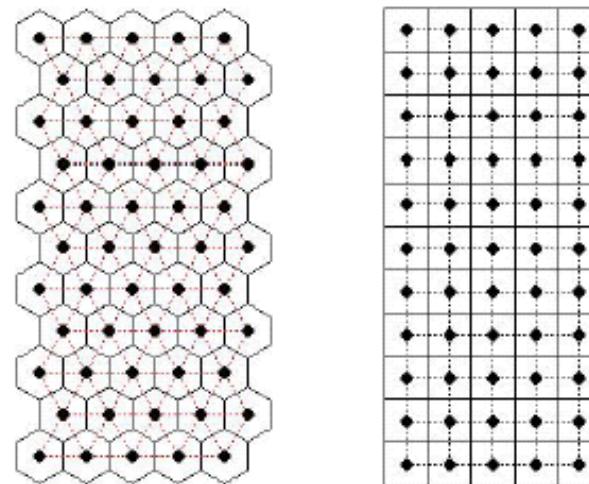
# Idea uczenia sieci SOM



Rysunek za książką A.Żurada

# Typowe topologie sieci SOM

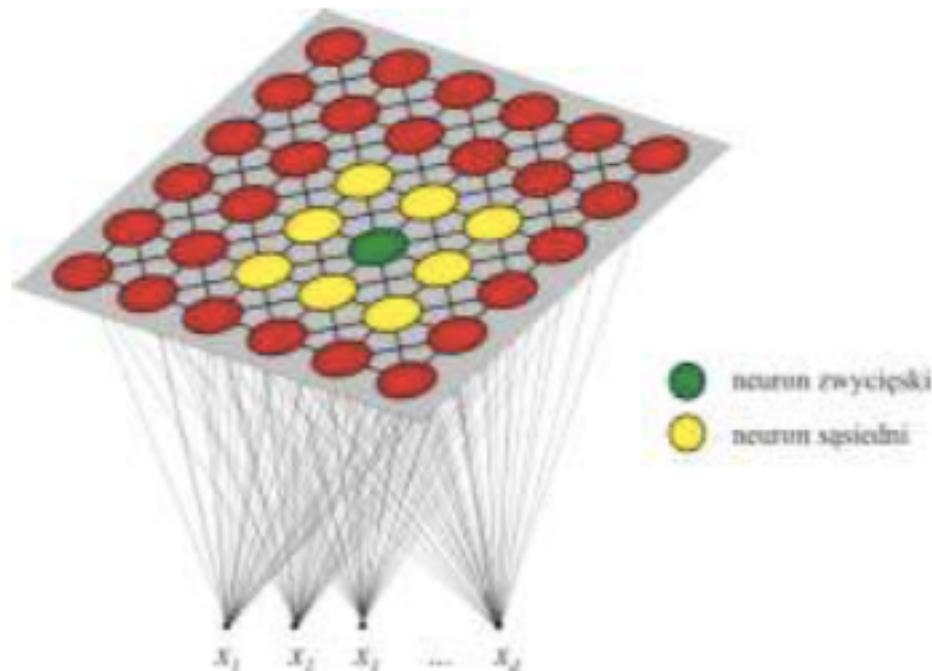
Dwie typowe topologie /odmiana ułożenia neuronów w plaster miodu lub macierz kwadratowa / i sąsiedztwo najbliższych neuronów



Rysunek 7.2: Sąsiedztwo na mapach Kohonena: Neurony ułożone w siatkę (strona lewa) hexagonalną i (strona prawa) prostokątną. {gridhr.JPG}

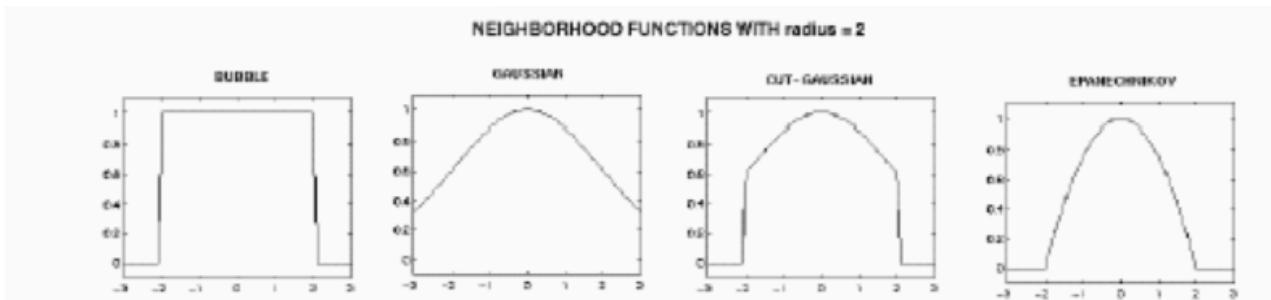
# Identyfikacja neuronu zwycięzcy

Uczymy neuron zwycięzcy  $x_m$  oraz jego najbliższych sąsiadów zgodnie z wybraną funkcją sąsiedztwa

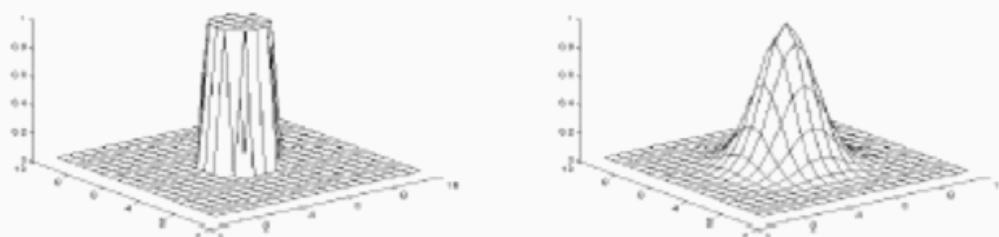


# Funkcje lokalnego sąsiedztwa

Uczymy neuron zwycięzcy  $x_m$  oraz jego najbliższych sąsiadów zgodnie z wybraną funkcją sąsiedztwa



Rysunek 7.3: Jednowymiarowe funkcje sąsiedztwa dla promienia  $R=2$ : bubble, gaussian, cut-gaussian, Epanechnikov {figs7/neigh4.ps}.



Rysunek 7.4: Funkcje sąsiedztwa: bubble i gaussian określone na płaszczyźnie. Funkcja bubble wyróżnia sąsiedztwo w sposób ostry: 1 - tak, 0 - nie; funkcja gaussian w sposób łagodny jako liczbę z przedziału  $(0, 1]$ . {bubble2.ps, gauss2.ps}

# Algorytm SOM

Siatka neuronów  $i = 1 \dots K$  w 1D-3D, każdy neuron z  $N$  wagami.

Neurony z wagami  $\mathbf{W}_i(t) = \{W_{i1} W_{i2} \dots W_{iN}\}$ , wektory  $\mathbf{X}=\{X_1, X_2 \dots X_N\}$ .

$t$  - dyskretny czas; nie ma połączeń pomiędzy neuronami!

1. Inicjalizacja: przypadkowe  $\mathbf{W}_i(0)$  dla wszystkich  $i=1..K$ .  
Funkcja sąsiedztwa  $h(|r-r_c|/\sigma(t), t)$  definiuje wokół neuronu położonego w miejscu  $r_c$  siatki obszar  $O_s(r_c)$ .
2. Oblicz odległości  $d(\mathbf{X}, \mathbf{W})$ , znajdź neuron z wagami  $W_c$  najbardziej podobnymi do  $\mathbf{X}$  (neuron-zwycięzcę).
3. Zmień wagi wszystkich neuronów w sąsiedztwie  $O_s(r_c)$
4. Powoli zmniejszaj siłę  $h_o(t)$  i promień  $\sigma(t)$ .
5. Iteruj aż ustaną zmiany lub wyczerpiesz liczbę epok.

Efekt: podział na wieloboki Voronoia.

# Zastosowanie SOM do analizy alfabetu

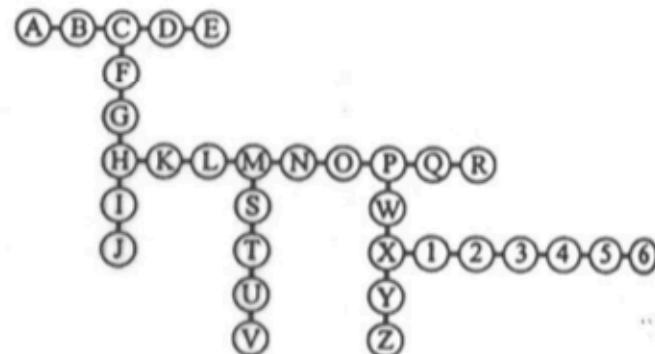
a)

	symbol wektora																															
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	1	2	3	4	5	6
składowe																																
$x_1$	1	2	3	4	5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3			
$x_2$	0	0	0	0	0	1	2	3	4	5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3			
$x_3$	0	0	0	0	0	0	0	0	0	0	1	2	3	4	5	6	7	8	3	3	3	3	6	6	6	6	6	6				
$x_4$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	3	4	1	2	3	4	2	2			
$x_5$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	3			

b)

B	C	D	E	*	Q	R	*	Y	Z
A	*	*	*	*	P	*	*	X	*
*	F	*	N	O	*	W	*	*	1
*	G	*	M	*	*	*	*	2	*
H	K	L	*	T	U	*	3	*	*
*	I	*	*	*	*	*	*	4	*
*	J	*	S	*	*	V	*	5	6

c)



Rys. 7.15. Przykład samoorganizującej się mapy cech: a) zbiór obrazów uczących  
 b) mapa powstała po cyklu uczenia, c) drzewo o minimalnej rozpiętości (z pracy Kohonen (1984), za zgodą ©Springer Verlag)  
 Lecz: właściwe kodowanie symboli – "sąsiedztwo"

# Analiza mapy fonemów języka fińskiego

SOM Toolbox: Intro to SOM by Teuvo Kohonen - Windows Internet Explorer - [Praca w trybie offline]

E:\nowki\RBF\SOM Toolbox Intro to SOM by Teuvo Kohonen.htm

Live Search

Plik Edycja Widok Ulubione Narzędzia Pomoc

SOM Toolbox: Intro to SOM by Teuvo Kohonen

LABORATORY OF COMPUTER AND INFORMATION SCIENCE ADAPTIVE INFORMATICS RESEARCH CENTRE CIS

CIS Toolbox Home About Docs Download Links

## The Self-Organizing Map (SOM)

by Teuvo Kohonen

### Introduction

The SOM is a new, effective software tool for the visualization of high-dimensional data. It converts complex, nonlinear statistical relationships between high-dimensional data items into simple geometric relationships on a low-dimensional display. As it thereby compresses information while preserving the most important topological and metric relationships of the primary data items on the display, it may also be thought to produce some kind of abstractions. These two aspects, visualization and abstraction, can be utilized in a number of ways in complex tasks such as process analysis, machine perception, control, and communication.

The SOM usually consists of a two-dimensional regular grid of nodes. A model of some observation is associated with each node (cf. Fig. 1).

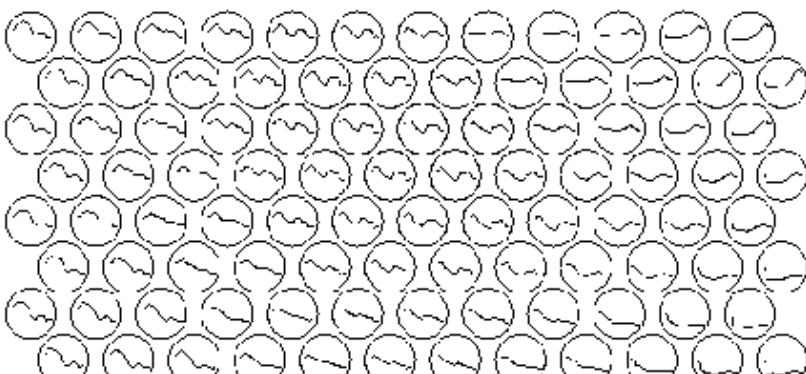
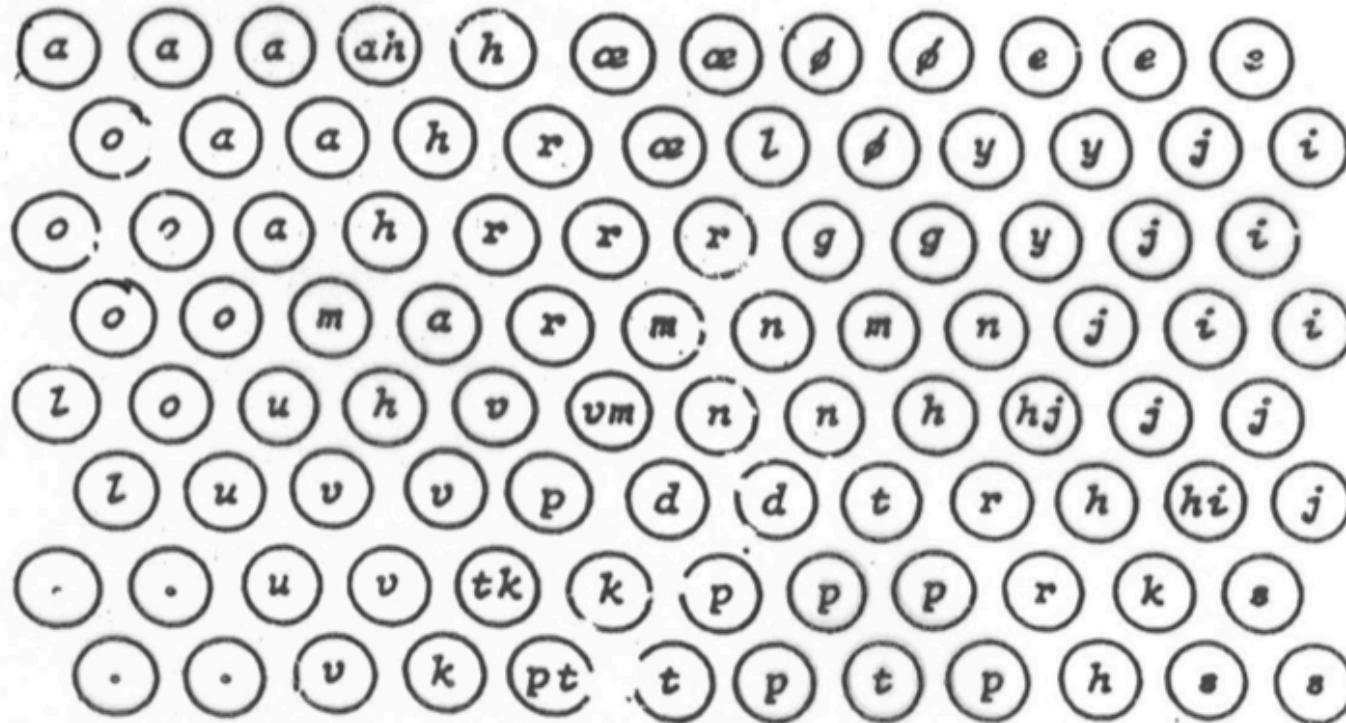


Figure 1: In this exemplary application, each processing element in the hexagonal grid holds a model of a short-time spectrum of natural speech

# Neuronowa mapa - typewriter

- Dane wejściowe: Sygnał mowy (pojedyncze słowa)
- Przetwarzanie danych: 1) 12-bitowy przetwornik analogowo-cyfrowy próbuje sygnał wejściowy co ~10ms; 2) 256-punktowa szybka analiza Fouriera (FFT); 3) wyjście: 15-punktowe spektrum sygnału wejściowego.
- Sieć: SOM: 15 wejść, warstwa wyjściowa:  $8 \times 12 = 96$  neuronów.
- Uczenie: Sygnał mowy (pojedyncze słowa).
- Kalibracja: Po nauczeniu sieci podawano na jej wejścia "wzorcowe" fonemy, opatrując zwycięzców odpowiednimi etykietami. Prawie każdej klasie fonemów odpowiada jeden zwycięzca lub grupa blisko położonych zwycięzców.

# Analiza mapy fonemów języka fińskiego



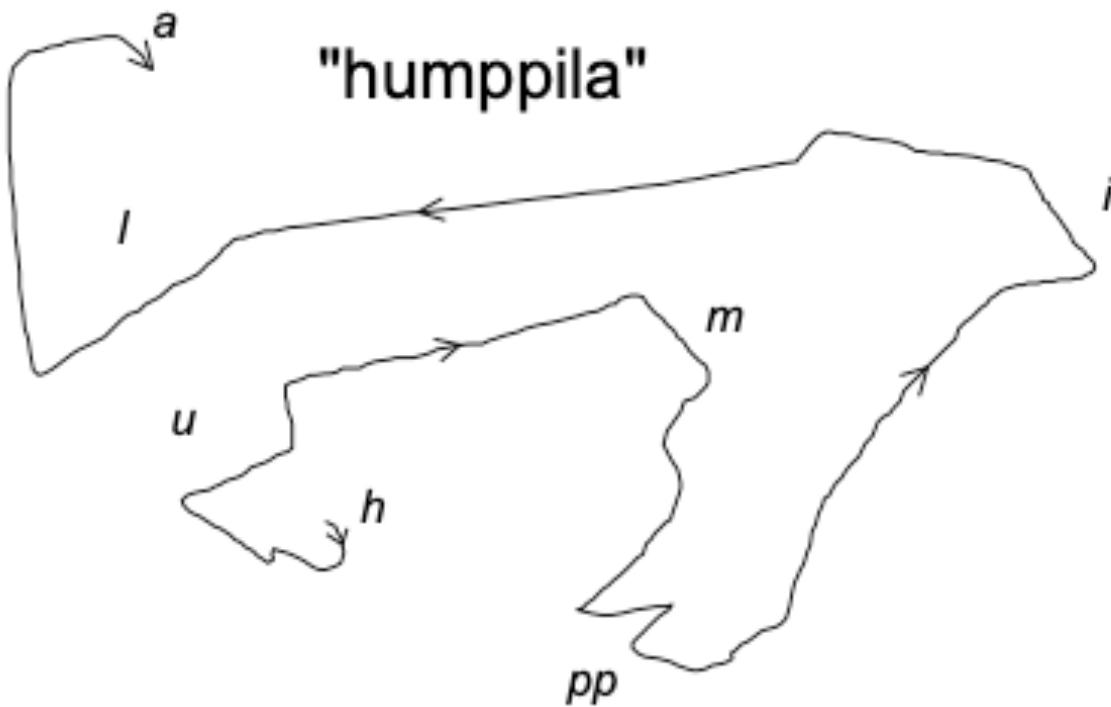
Rys. 7.18. Mapa fonemów języka fińskiego [z pracy Kohonena (1988), za zgodą ©IEEE]

Podobne fonemy – zachowują bliskość  
Możliwość analizy online wypowiedzi i tworzenie obrazów

# SOM neural typewriter

- Podczas wypowiadania słowa notowane jest położenie zwycięzcy, które zmieniając się tworzy obraz - marszrutę. Otrzymana w ten sposób transkrypcja fonetyczna słowa może być dalej przetwarzane w innym systemie
- Rodzaj automatycznej maszyny do pisania, rozpoznaje słowa ich “nie rozumiejąc”
- trafność rozpoznawania fonemów: 92-97%,

"humppila"

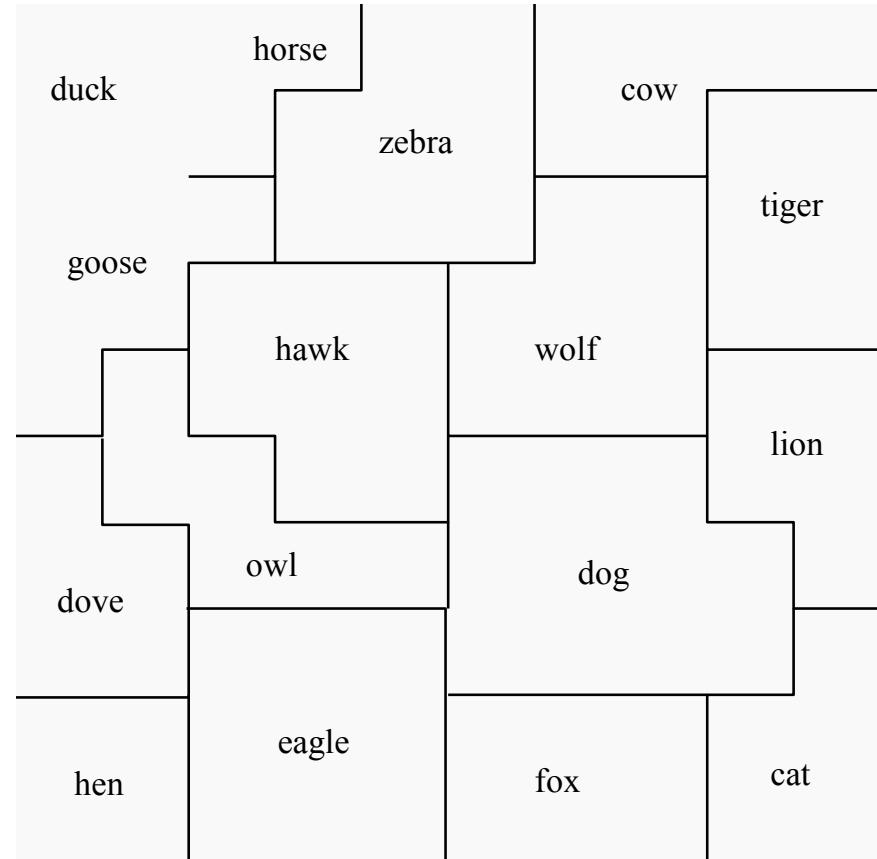
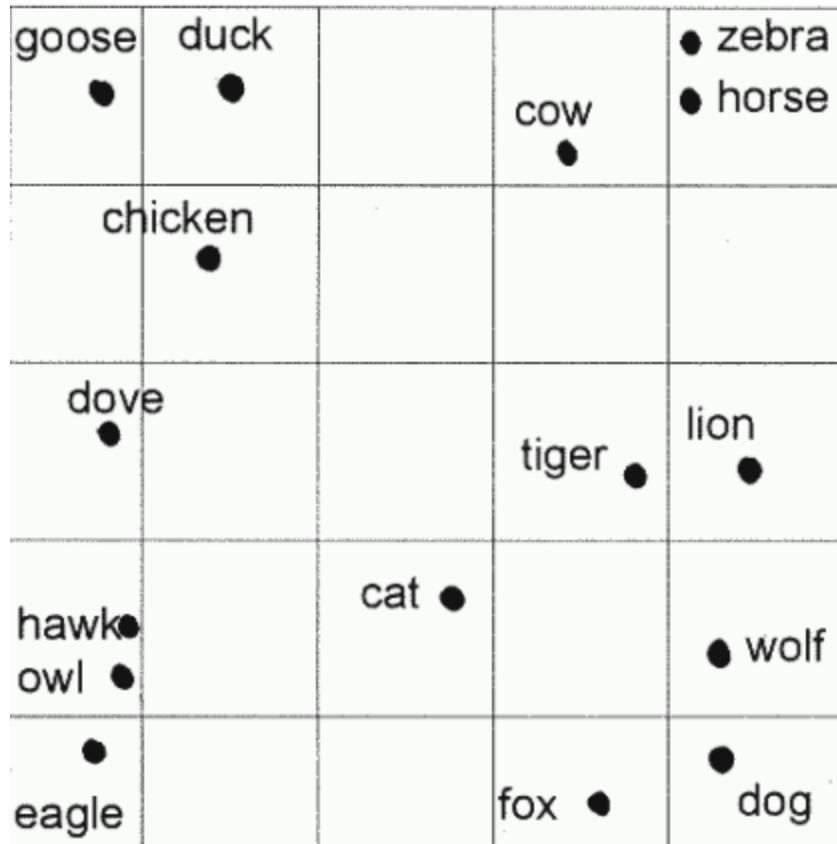


# Przykład mapy semantycznej

		d	d	g		e		w	t	h	z		
animal		o	h	u	o	a	g	f	d	o	c	g	i
		v	e	c	s	w	w	l	o	o	l	a	e
		e	n	k	e	l	k	e	x	g	f	t	r
is	small	1	1	1	1	1	1	0	0	0	1	0	0
	medium	0	0	0	0	0	0	1	1	1	0	0	0
	big	0	0	0	0	0	0	0	0	0	1	1	1
has	2 legs	1	1	1	1	1	1	1	0	0	0	0	0
	4 legs	0	0	0	0	0	0	0	1	1	1	1	1
	hair	0	0	0	0	0	0	0	1	1	1	1	1
	hooves	0	0	0	0	0	0	0	0	0	0	0	1
	mane	0	0	0	0	0	0	0	0	1	0	0	1
	feathers	1	1	1	1	1	1	1	0	0	0	0	0
likes	hunt	0	0	0	0	1	1	1	1	0	1	1	1
	run	0	0	0	0	0	0	0	1	1	0	1	1
	to fly	1	0	0	1	1	1	1	0	0	0	0	0
	swim	0	0	1	1	0	0	0	0	0	0	0	0

SOM was used on such data by Ritter and Kohonen 1989,  
MDS by Naud & Duch (1996) = dalsza analiza w wykładzie W.Ducha nt.  
Wizualizacji wielowymiarowych -> patrz strona WWW

# Porównanie map MDS & SOM



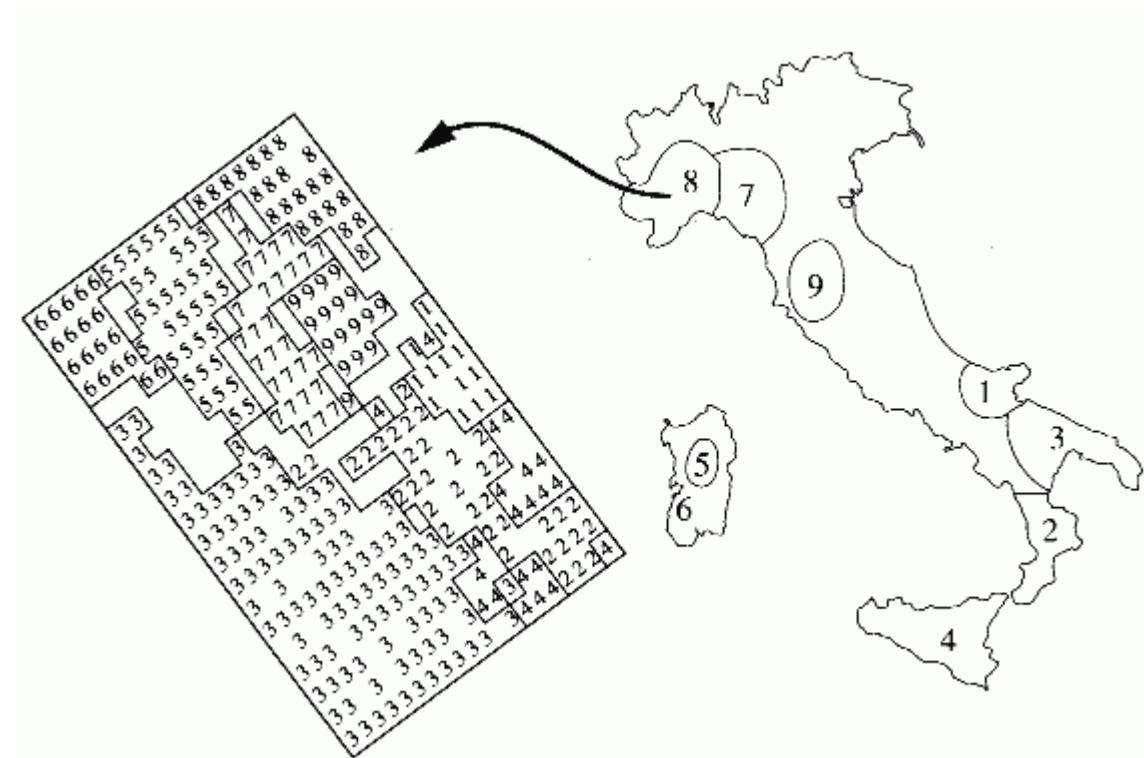
Użycie SOM i MDS na danych opisujących zwierzęta – zwróć uwagę na bliskość semantyczną rodzin zwierząc, np. kotowate po prawej stronie mapy; ptaki tworzą także podobne grupy

# Analiza jakości oliwy w różnych rejonach Włoch

572 próbek oliwy pochodzącej z 9 prowincji Włoch

Badania laboratoryjne – 8 podstawowych składników  
SOM 20 x 20 neuronów,  
Odwzorowanie 8D => 2D.  
Kalibracja zwycięskich neuronów z etykietą prowincji

Analiza sąsiedztwa i bliskości innych rejonów



Zauważ że przekształcenie zachowało nieznane uczeniu sąsiedztwo topograficzne , tylko prowincja #3 jest lekko rozproszona

# Ocena wpływu cech na grupy

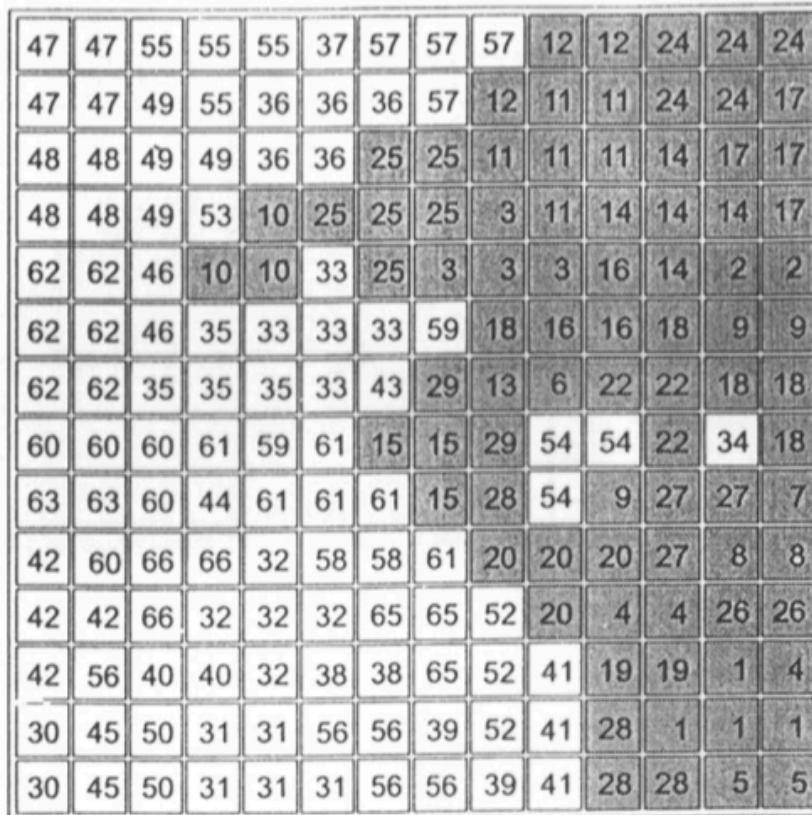
Analizie poddano dane dotyczące 66 banków hiszpańskich, przy czym blisko połowa z nich (29) zbankrutowała w czasie kryzysu. Do reprezentacji profilu każdego z banków, na podstawie analiz statystycznych, wybrano dziewięć wskaźników finansowych (tablica 5.7). Ich wartości zostały, podobnie jak w poprzednim przypadku, znormalizowane do zerowej średniej i jednostkowej wariancji.

Tablica 5.7. Wskaźniki finansowe wykorzystane w badaniu banków [37]

Symbol	Opis
R1	Aktywa bieżące do aktywów całkowitych
R2	(Aktywa bieżące - środki pieniężne) do aktywów całkowitych
R3	Aktywa bieżące do zobowiązań
R4	Rezerwy do zobowiązań
R5	Przychody netto do aktywów
R6	Przychody netto do kapitału obrotowego
R7	Przychody netto do zobowiązań
R8	Koszty sprzedaży do sprzedaży
R9	Przepływy pieniężne do zobowiązań

Do analizy danych wykorzystano sieć 196 neuronów, uformowanych w macierz

# Analiza banków hiszpańskich



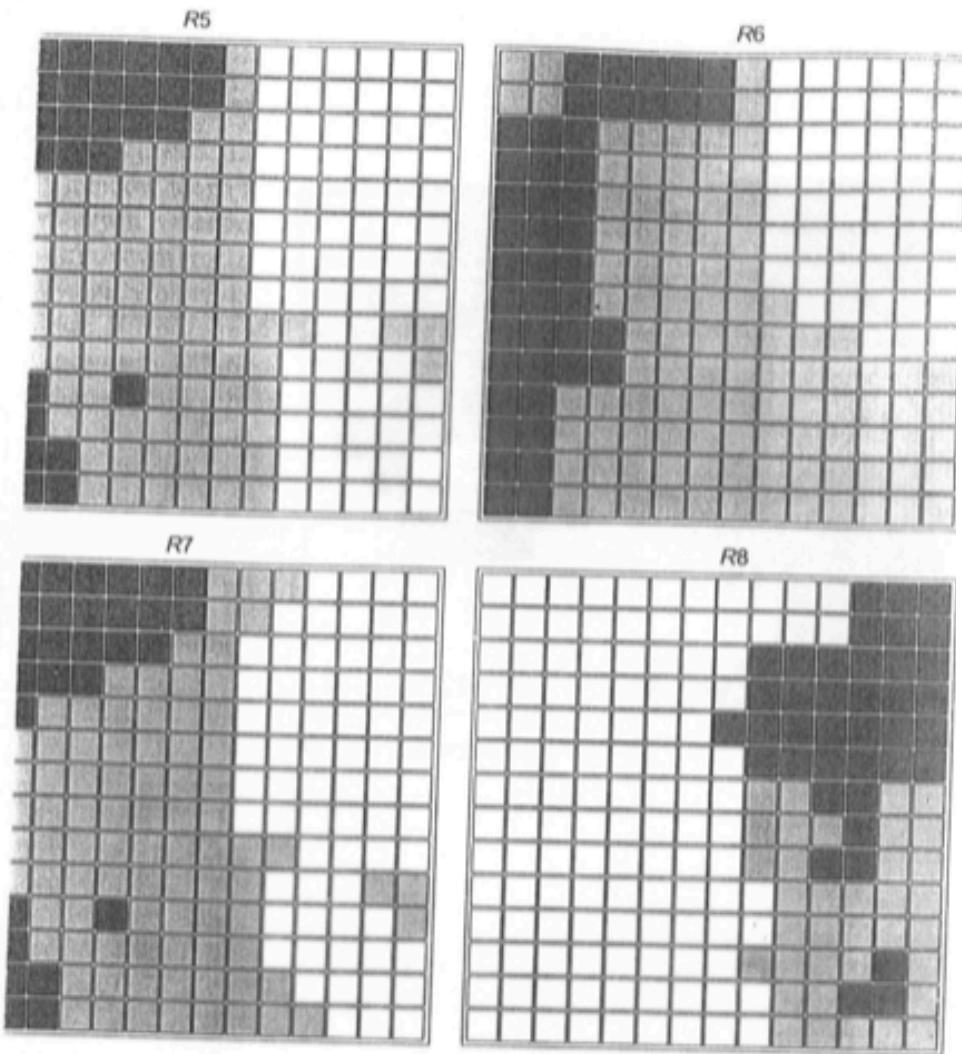
obszar bezpieczny



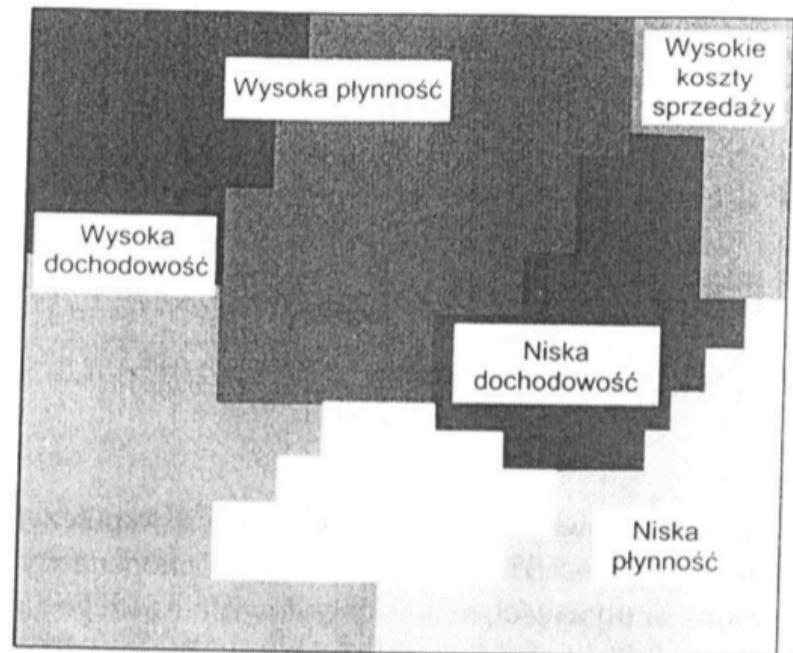
obszar bankructwa

Rys. 5.34. Mapa cech analizowanych banków [37]

# Indywidualna analiza znaczenia cech



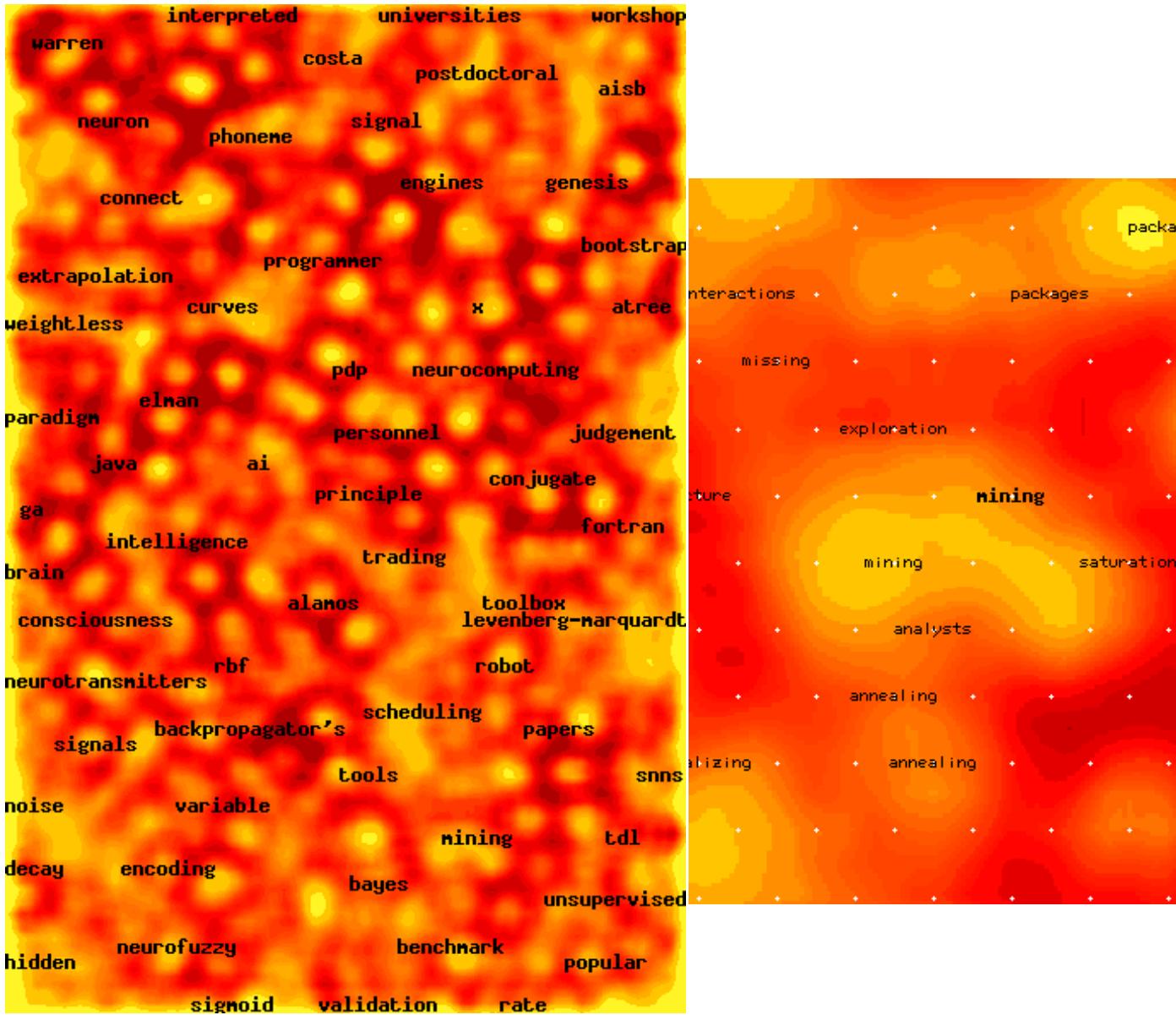
Rys. 5.36. Mapy wag sieci Kohonena dla wskaźników od R5 do R8 [37]



Rys. 5.38. Regiony na mapie cech wyznaczone przez mapy wag [37]

# Web SOM

- SOM do pogrupowania 12088 artykułów z internetu
- Aplikacja do stopniowego (zoom) przeglądania zawartości skupisk
- Spójrz na [websom.hut.fi](http://websom.hut.fi) Web page



# Odnośniki do literatury

SOM intensywny rozwój od lat 80 poprzedniego wieku

- Wiele różnych zastosowań

Analiza skupień – obszerna literatura:

- Kohonen, Teuvo; Honkela, Timo (2007). "Kohonen Network". Scholarpedia WWW
- Kohonen, Teuvo (1982). "Self-Organized Formation of Topologically Correct Feature Maps"
- T. Kohonen, Self-Organization and Associative Memory. Springer, Berlin, 1984
- Żurada J., Barski M., Jędruch W.: Sztuczne sieci neuronowe. PWN 1996.
- Stapor K. Automatyczna klasyfikacja obiektów. Akademicka Oficyna Wydawnicza EXIT, Warszawa 2005

# **Pytanie i komentarze?**

Dalszy kontakt:

[jerzy.stefanowski@cs.put.poznan.pl](mailto:jerzy.stefanowski@cs.put.poznan.pl)

<http://www.cs.put.poznan.pl/jstefanowski/>



**Fundusze  
Europejskie**  
Polska Cyfrowa



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz  
Rozwoju Regionalnego



# **Uczenie nienadzorowane**

## **algorytmy grupowania wykład 12 cz II**

Jerzy Stefanowski

Instytut Informatyki PP

2021

Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI-TECH)  
projekt finansowany z środków Programu Operacyjnego Polska Cyfrowa  
POPC.03.02.00-00-0001/20



**Fundusze  
Europejskie**  
Polska Cyfrowa



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz  
Rozwoju Regionalnego



# Plan wykładu

- Rozszerzenia klasycznych algorytmów grupowania
  - Algorytm k-średnich
    - K-medoid, PAM, ...
  - Algorytmy hierarchiczne
    - Podstawowe AHC
    - BIRCH
- Algorytmy gęstościowe
  - DBSCAN
- Podejścia wykorzystujące modele statystyczne
  - Algorytm mieszanin rozkładów (EM)
- Inne algorytmy grupowania dla trudnych danych
- Ocena jakości grupowania
- Podsumowanie

# Grupowanie z wykorzystaniem modeli prawdopodobieństwa

- Podejścia oparte na założeniu, że danych są generowanie w wyniku realizacji pewnego procesu statystycznego
- Zakłada się pewien model rozkładu prawdopodobieństwa występowanie obserwacji
- Każdemu potencjalnemu **skupisku** odpowiada **model**, w postępowaniu (algorytmie) weryfikuje się stopień dobrego dopasowania oryginalnych danych do przyjętego modelu
- Celem grupowania jest znalezienie zbioru (mieszaniny) modeli (rozkładów) opisujących skupiska oraz estymacja parametrów tych modeli
- Obiekty przydziela się do skupisk zgodnie ze sparametryzowanymi modelami i zasadą klasyfikacji Bayesowskiej

# Mieszaniny rozkładów (1)

- Typowe podejście do grupowania wykorzystującego modele statystyczne – przyjęcie założenia mieszaniny wielowymiarowych rozkładów prawdopodobieństwa (przykład algorytm EM)
- Założenia: Podział danych  $X=\{x_1, \dots, x_m\}$  na  $K$  skupisk jest równoznaczny z łącznym rozkładem prawdopodobieństwa zbudowanym z  $K$  składowych rozkładów o parametrach  $\theta_j$ . Łączny rozkład ze zbiorem parametrów  $\theta=\{\theta_1, \dots, \theta_K\}$ :

$$P(x|\theta) = \sum_{j=1}^K p(j) \cdot p_j(x|\theta_j)$$

- gdzie  $p(j)$  jest prawdopodobieństwem przydziału obiektu  $x$  do  $j$ -tego skupiska (modelu);  $\sum p(j)=1$

# Mieszaniny rozkładów (2)

- Łączny rozkład prawdopodobieństwa dla obiektu  $x$

$$P(x|\theta) = \sum_{j=1}^K p(j) \cdot p_j(x|\theta_j)$$

- Interpretacja statystyczna (modele generatywne) - przykłady (obiekty) uczące otrzymywane są dwustopniowo:
  - Losowanie jednego z  $K$  źródeł – które generuje przykłady z swojej grupy :  $p(j)$  prawdopodobieństwo wylosowania j-tego źródła
  - Sam przykład jest generowany zgodnie z funkcją gęstości prawdopodobieństwa  $f_j(x|\theta_j)$  wynikającą z przyjętego modelu

# Mieszaniny rozkładów (3)

- Mając rozkład prawdopodobieństwa dla obiektu  $x$

$$P(x | \theta) = \sum_{j=1}^K p(j) \cdot p_j(x | \theta_j)$$

- Jeśli wszystkie obiekty w  $X$  są generowane niezależnie, to łączne prawdopodobieństwo otrzymania / wygenerowania obiektów  $X=\{x_1, \dots, x_m\}$  jest iloczynem prawdopodobieństw dla indywidualnych obiektów

$$P(X | \theta) = \prod_{i=1}^m P(x_i | \theta) = \prod_{i=1}^m \sum_{j=1}^K p(j) \cdot p_j(x_i | \theta_j)$$

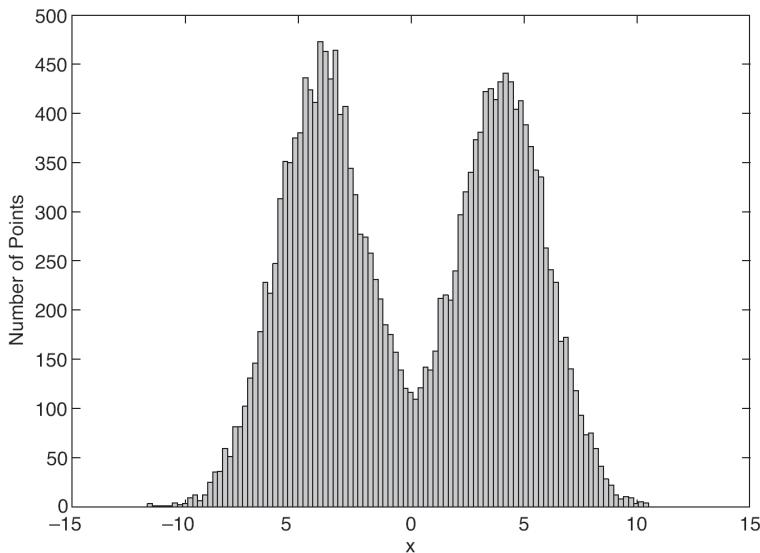
- Najczęściej zakłada się rozkłady normalne. Nazywa się to mieszaniną rozkładów Gaussowskich (ang. mixture of Gaussians)

# Przykład analizy mieszaniny rozkładów normalnych

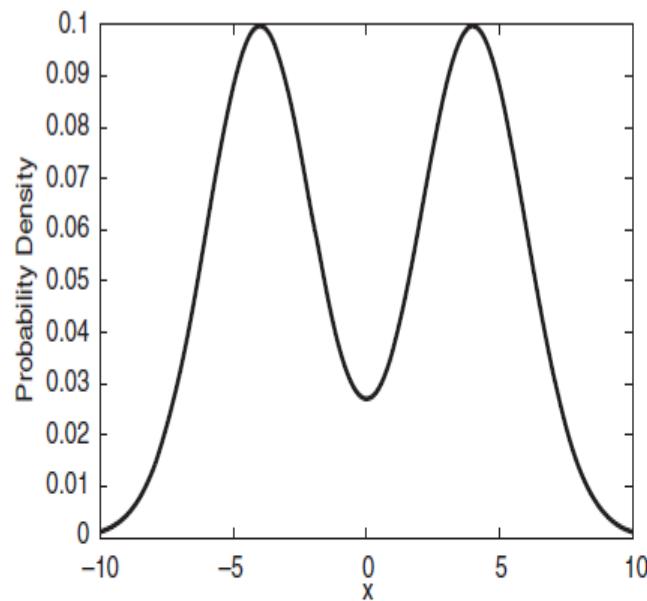
- **Prosty przykład** [Kumar et al]: rozważ modelowanie obiektów tworzących histogram – patrz rys.
- Model może być mieszaniną dwóch rozkładów normalnych (każdy sparametryzowany wartością oczekiwana oraz odchyleniem standardowym  $\sigma$ 
  - patrz wzór

$$prob(x_i|\Theta) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

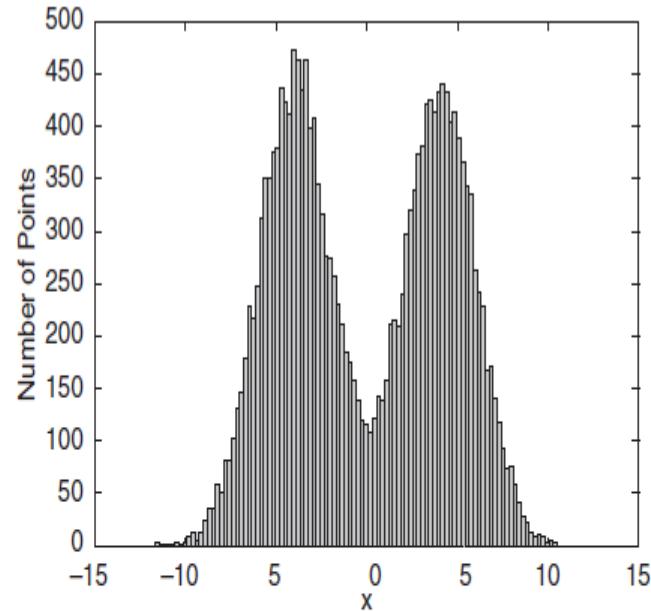
- Jeżeli estymuje oba parametry – to przy założeniu równych prawdopodobieństw komponentów  $p(1)=p(2)=0.5$ :
  - Można w pełni opisać oba skupiska
  - Można obliczyć prawdopodobieństwo przydziału dowolnego obiektu do skupiska 1 oraz skupiska 2
  - Przypisać obiekt do bardziej prawdopodobnego skupiska



# Model – mieszanina gausowska



(a) Probability density function for the mixture model.



(b) 20,000 points generated from the mixture model.

**Figure 8.2.** Mixture model consisting of two normal distributions with means of -4 and 4, respectively. Both distributions have a standard deviation of 2.

# Złożenie rozkładów normalnych

W przykładzie

$$\text{Prob} = p(1)N(\mu_1, \sigma_1) + p(2)N(\mu_2, \sigma_2)$$

Lecz ogólnie rozważamy rozkłady wielowymiarowe  
zależne od wektora  $\mu$  oraz macierzy kowariancji  $\Sigma$

Funkcja gęstości n-wymiarowego rozkładu normalnego wektora losowego  $X$  jest wzorem:

$$f_{\mu, \Sigma}(X) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(X - \mu)^T \Sigma^{-1} (X - \mu)\right).$$

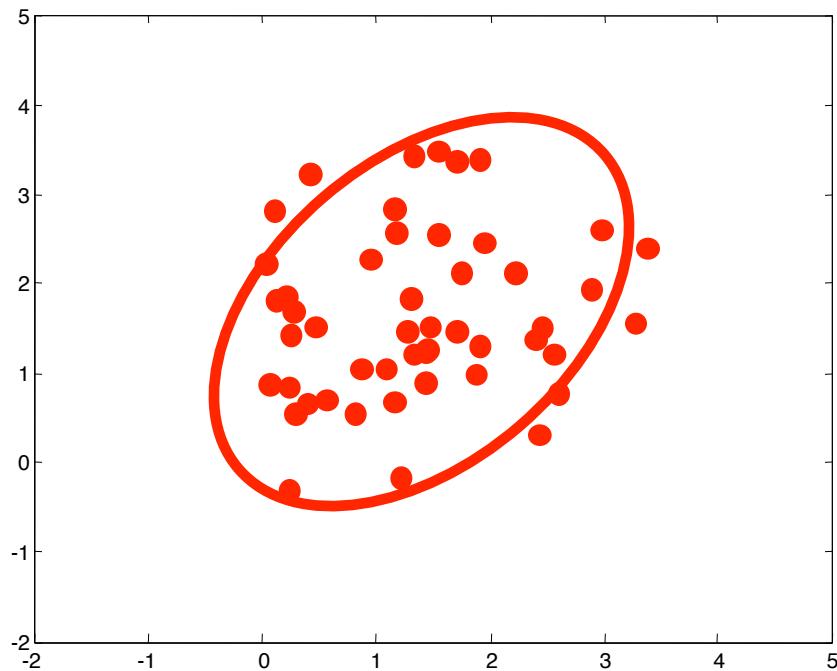
$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^N X_i.$$

Estymator macierzy kowariancji o największej wiarygodności:

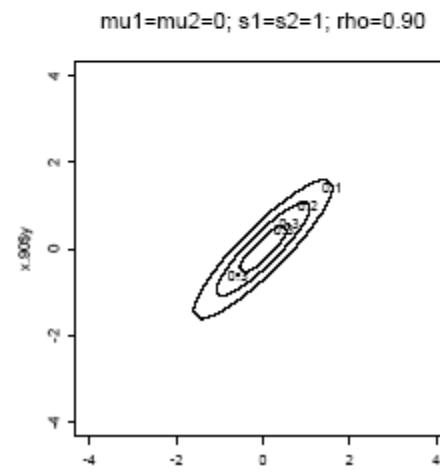
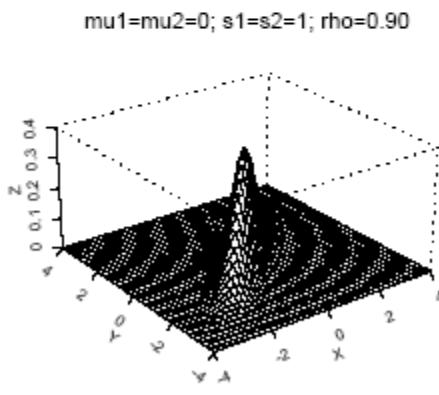
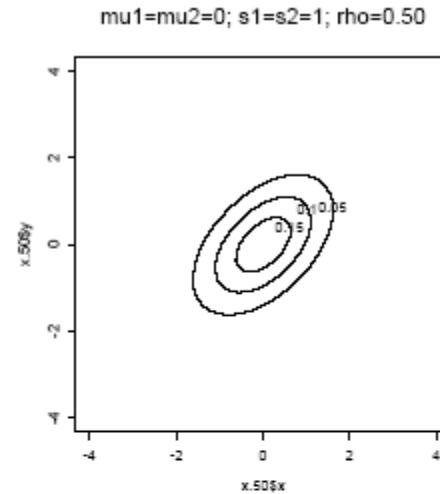
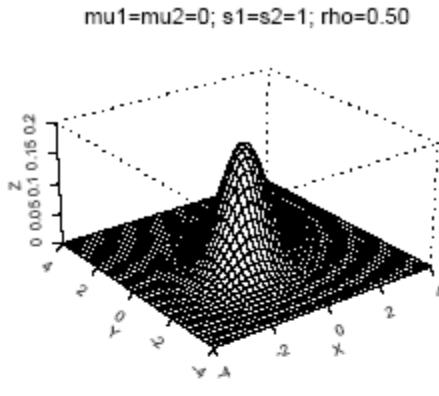
$$\hat{\Sigma} = \frac{1}{N} \sum_{i=1}^N (X_i - \hat{\mu})(X_i - \hat{\mu})^T.$$

# Modelowanie dwuwymiarowego rozkładu normalnego ( $d=2$ )

$$\mathcal{N}(\underline{x} ; \underline{\mu}, \Sigma) = \frac{1}{(2\pi)^{d/2}} |\Sigma|^{-1/2} \exp \left\{ -\frac{1}{2} (\underline{x} - \underline{\mu})^T \Sigma^{-1} (\underline{x} - \underline{\mu}) \right\}$$



# Dwuwymiarowe rozkłady prawdopodobieństwa



Więcej w literaturze nt. rozkładów prawdopodobieństwa

# Funkcja wiarygodności i MLE

- Dla wybranego modelu statystycznych danych – należy oszacować jego parametry  $\Theta$  na podstawie  $m$  prób (tutaj przykładów uczących)
- **Metoda największej wiarygodności** (ang. Maximum likelihood method)
  - Na podstawie wybranego rozkładu określamy prawdopodobieństwo a posteriori obiektu  $x_i$
  - Parametry rozkładu dobiera się tak, aby maksymalizować prawdopodobieństwa a posteriori rozkładu dla obiektów z danych uczących  $X=\{x_1, \dots, x_m\}$
- Definiujemy funkcję wiarygodności ( $L$  – z ang. **Likelihood function**)

# Funkcja wiarygodności i MLE

- Prawd. a posteriori obiektów z X – iloczyn indywidualnych prawd. dla obiektów
- Definiujemy funkcję wiarygodności (L – z ang. Likelihood function)

$$L(X; \theta) = \prod_{i=1}^m p(x_i | \theta) = \prod_{i=1}^m \prod_{j=1}^k p(x_i | \theta) \cdot p(j)$$

- Cel – wybierz parametry  $\Theta$  maksymalizujące powyższą funkcję wiarygodności
- Najczęściej wykorzystuje się logarytmiczne przekształcenie funkcji wiarygodności  $\text{LogL}(X; \theta)$

$$\text{LogL}(X; \theta) = \sum_{i=1}^m \log p(x_i | \theta)$$

# Funkcja wiarygodności dla $N(\mu, \Sigma)$

W algorytmie EM wykorzystuje się mieszany rozkładów Gaussowskich – rozważmy przykład jednowymiarowy

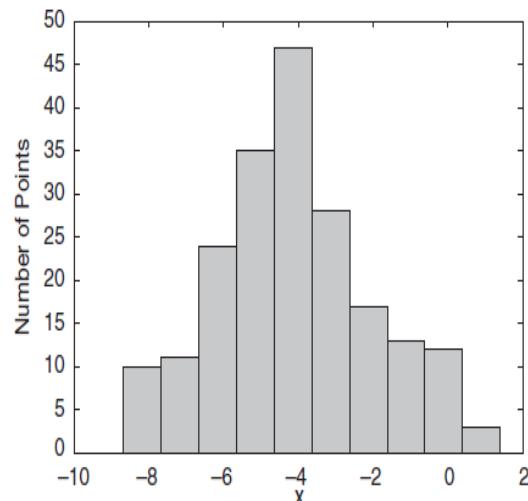
$$L(X; \theta) = \prod_{i=1}^m \frac{1}{\sqrt{2\pi} \cdot \sigma} e^{-\frac{(x_i - \mu)^2}{2\sigma^2}}$$

Logarytmiczna funkcja wiarygodności

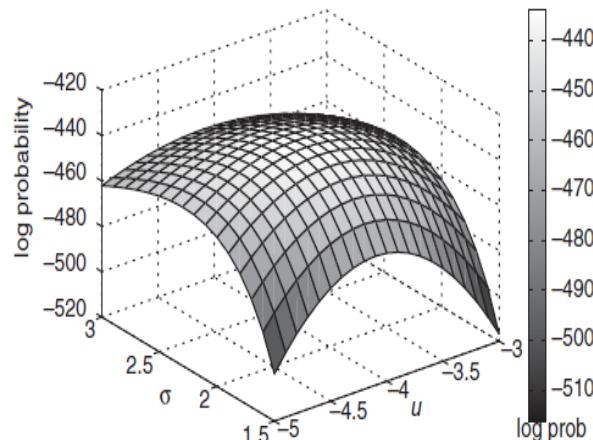
$$\text{Log}L(X; \theta) = -\sum_{i=1}^m \frac{(x_i - \mu)^2}{2\sigma^2} - 0.5m \log 2\pi - m \log \sigma$$

Przykład ilustracyjny kolejny slajd – prosty wybór  
Procedura wyznaczania estymatorów dla rozkładów ciągłych – pochodne cząstkowe logarytmicznej funkcji L względem nieznanego parametru

# MLE – poszukiwanie parametrów maksymalizujących $\text{LogL}(X;\Theta)$



(a) Histogram of 200 points from a Gaussian distribution.



(b) Log likelihood plot of the 200 points for different values of the mean and standard deviation.

**Figure 8.3.** 200 points from a Gaussian distribution and their log probability for different parameter values.

Analiza wykresu :  $\mu=-4.1$  oraz  $\theta=2.1$

# Algorytm EM

**EM** nazwa ang. **Expectation-Maximization**

- Inicjalizacja początkowych wartości parametrów rozkładów

**Repeat**

1. (*Expectation step*) Dla każdego obiektu z  $X$  oblicz jego przynależność do skupiska (rozkładu)
2. (*Maximization step*) Użyj tych prawdopodobieństw do iteracyjnej aktualizacji parametrów rozkładu

**Until** (zmiany parametrów nie są znaczące)

# Zapis algorytmu EM

---

## Algorithm 9.2 EM algorithm.

- 1: Select an initial set of model parameters.  
(As with K-means, this can be done randomly or in a variety of ways.)
  - 2: **repeat**
  - 3:   **Expectation Step** For each object, calculate the probability that each object belongs to each distribution, i.e., calculate  $\text{prob}(\text{distribution } j | \mathbf{x}_i, \Theta)$ .
  - 4:   **Maximization Step** Given the probabilities from the expectation step, find the new estimates of the parameters that maximize the expected likelihood.
  - 5: **until** The parameters do not change.  
(Alternatively, stop if the change in the parameters is below a specified threshold.)
-

# Krok oczekiwanej przynależności do skupiska (expectation)

Mając oszacowanie parametrów j-tego rozkładu normalnego  $\mu_j$  oraz  $\Sigma_j$  ( $\sigma_j$ ) oraz wstępne  $p(j)$  oblicz przynależności każdego obiektu z X do odpowiedniego skupiska ( $j=1,\dots,K$ )

$$t_{ij}^{(h)} = \frac{p_j^{(h)} \cdot p(x_i | \mu_j^{(h)}, \Sigma_j^{(h)})}{\sum_{l=1}^K p_l^{(h)} \cdot p(x_i | \mu_l^{(h)}, \Sigma_l^{(h)})}$$

gdzie  $t_{ij}^{(h)}$  stopień przynależności obiektu  $x_i$  do j-tego skupiska w h-tej iteracji (prawdopodobieństwa z reguły Bayesowskiej)

W przypadku rozkładu normalnego używamy funkcji gęstości prawdopodobieństwa f jako  $p(x|\theta)$

# Krok estymacji parametrów maksymalizujących logL

Na podstawie wyliczonych przynależności  $t_{ij}^h$  poszuge się nowych estymatorów parametrów rozkładów (MLE – maksymalizujących log funkcji wiarygodności)

$$p_j^{(h+1)} = \frac{1}{m} \sum_{i=1}^m t_{ij}^{(h)}$$

$$\mu_j^{(h+1)} = \frac{\sum_{i=1}^m t_{ij}^{(h)} \cdot x_i}{m \cdot p_j^{(h+1)}}$$

$$\Sigma_j^{(h+1)} = \frac{\sum_{i=1}^m t_{ij}^{(h)} \cdot (x_i - \mu_j^{(h)}) \cdot (x_i - \mu_j^{(h)})^T}{m \cdot p_j^{(h+1)}}$$

# Expectation-Maximization

- Kolejne iteracje algorytmu powinny polepszać (maksymalizować) oszacowanie log funkcji wiarygodności (the log-likelihood  $L$ ) modeli
- Iteruj procedurę aż do zbieżności (warunku zatrzymania)

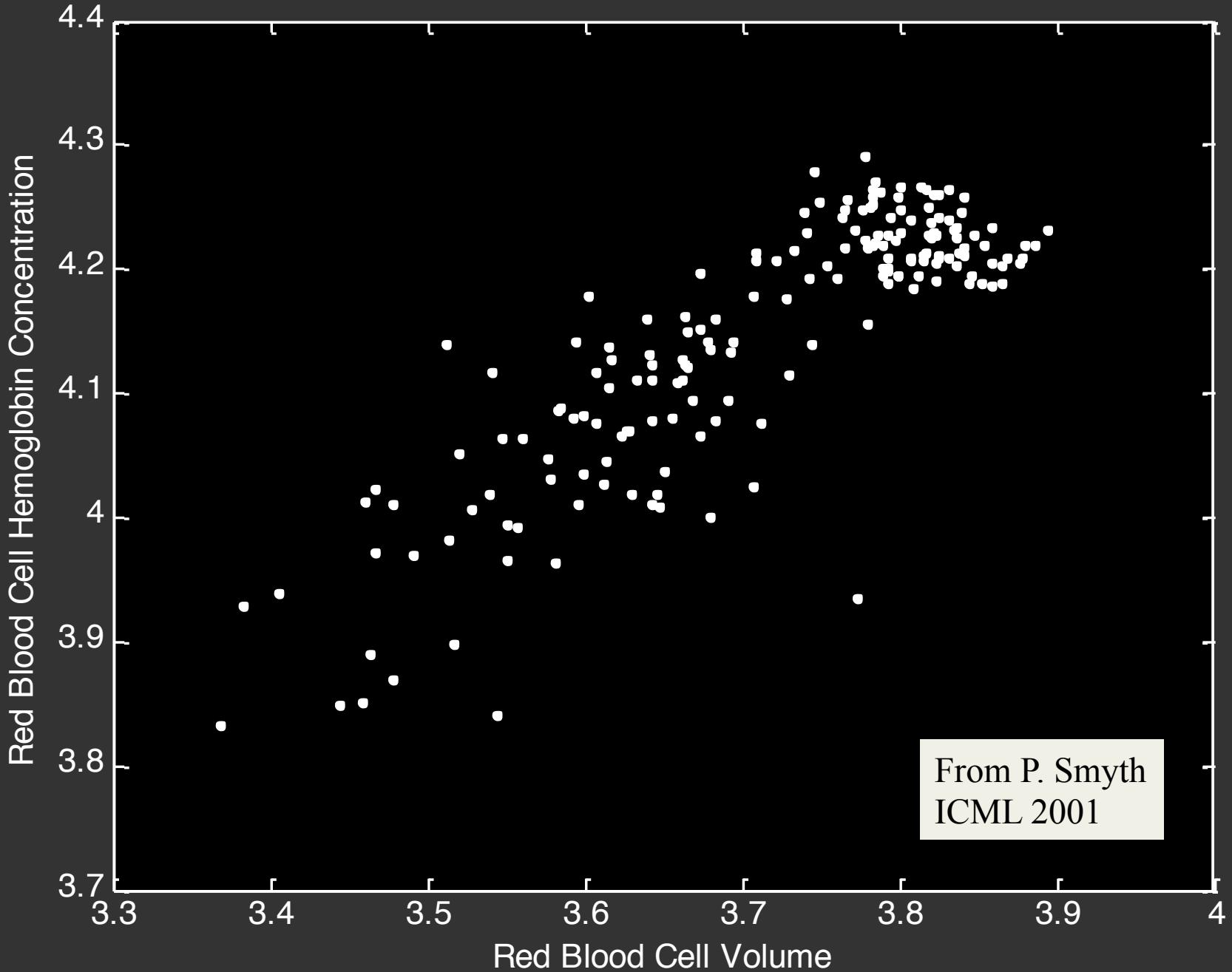
Możliwość uwzględniania niekompletnych danych

Autorzy:

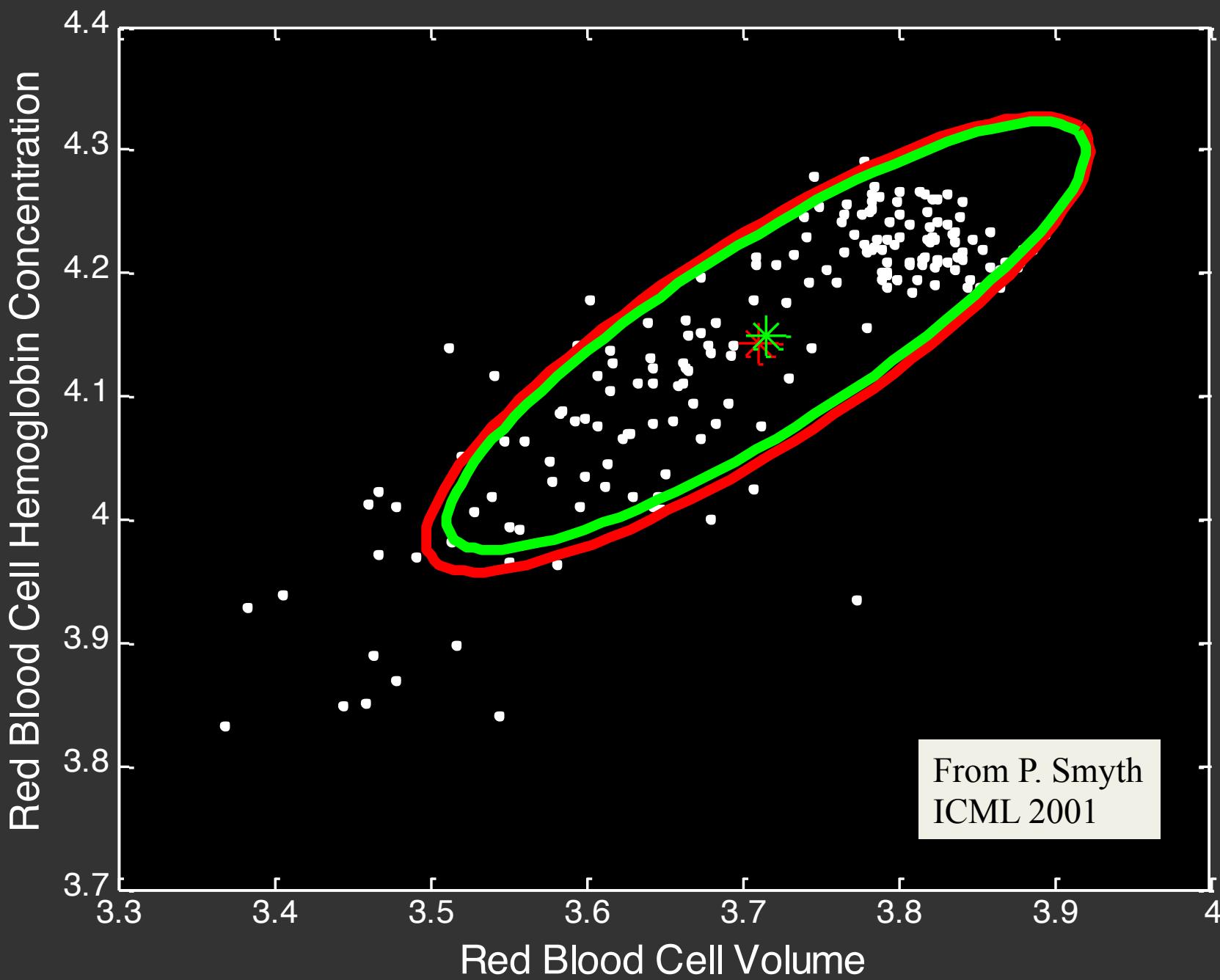
Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society. Series B (Methodological), 39(1), 1–38.

# Przykład działania algorytmu EM

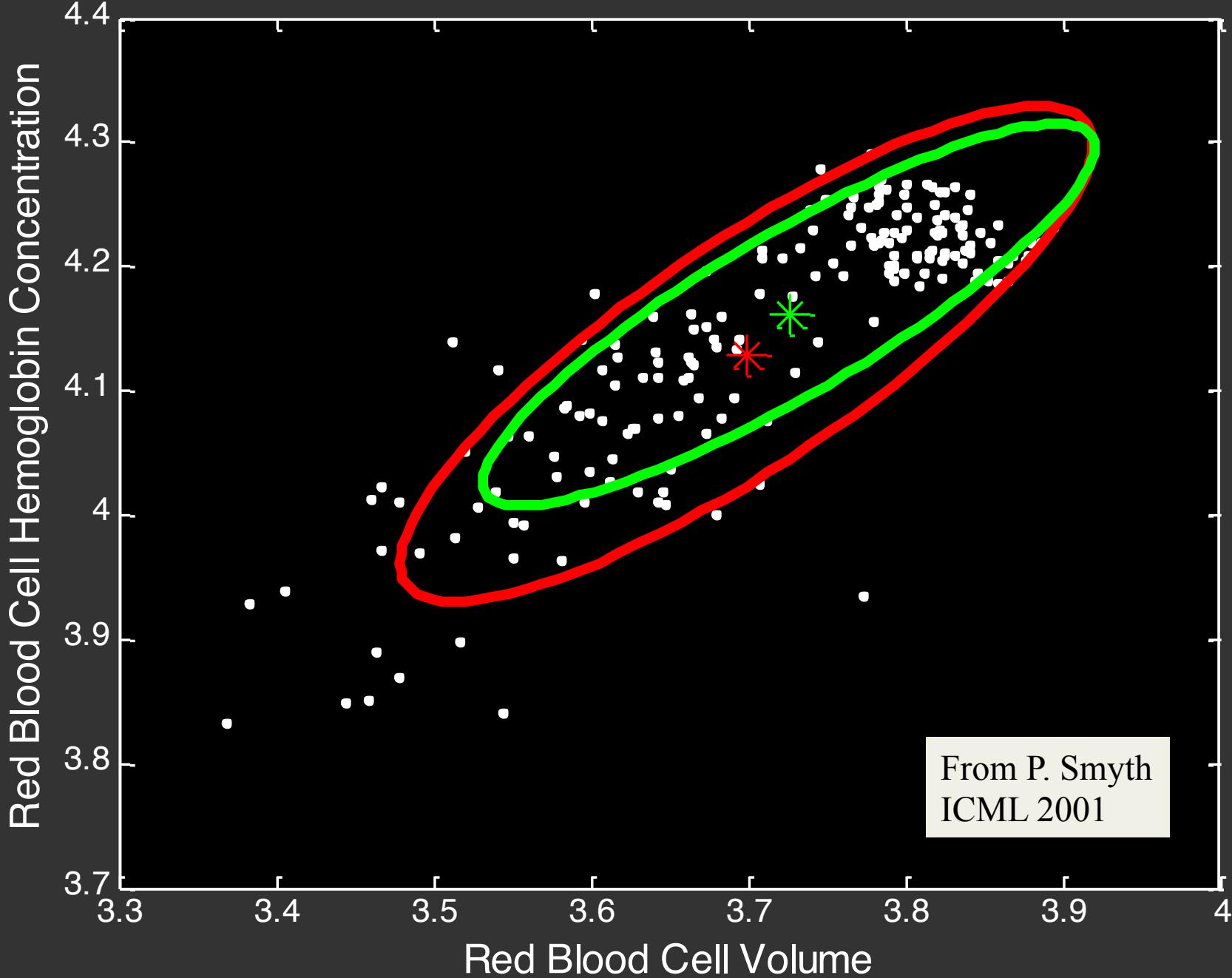
# ANEMIA PATIENTS AND CONTROLS



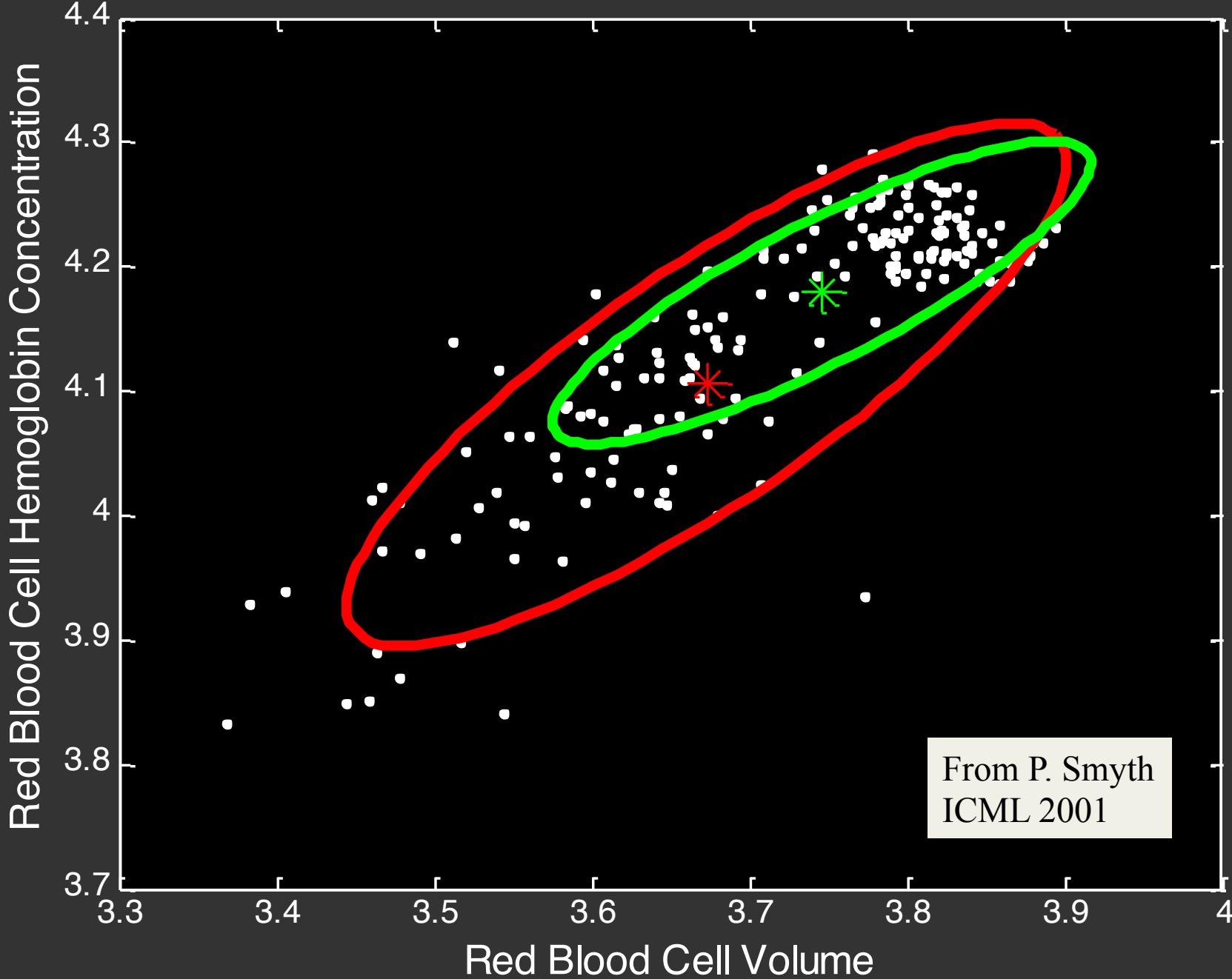
# EM ITERATION 1



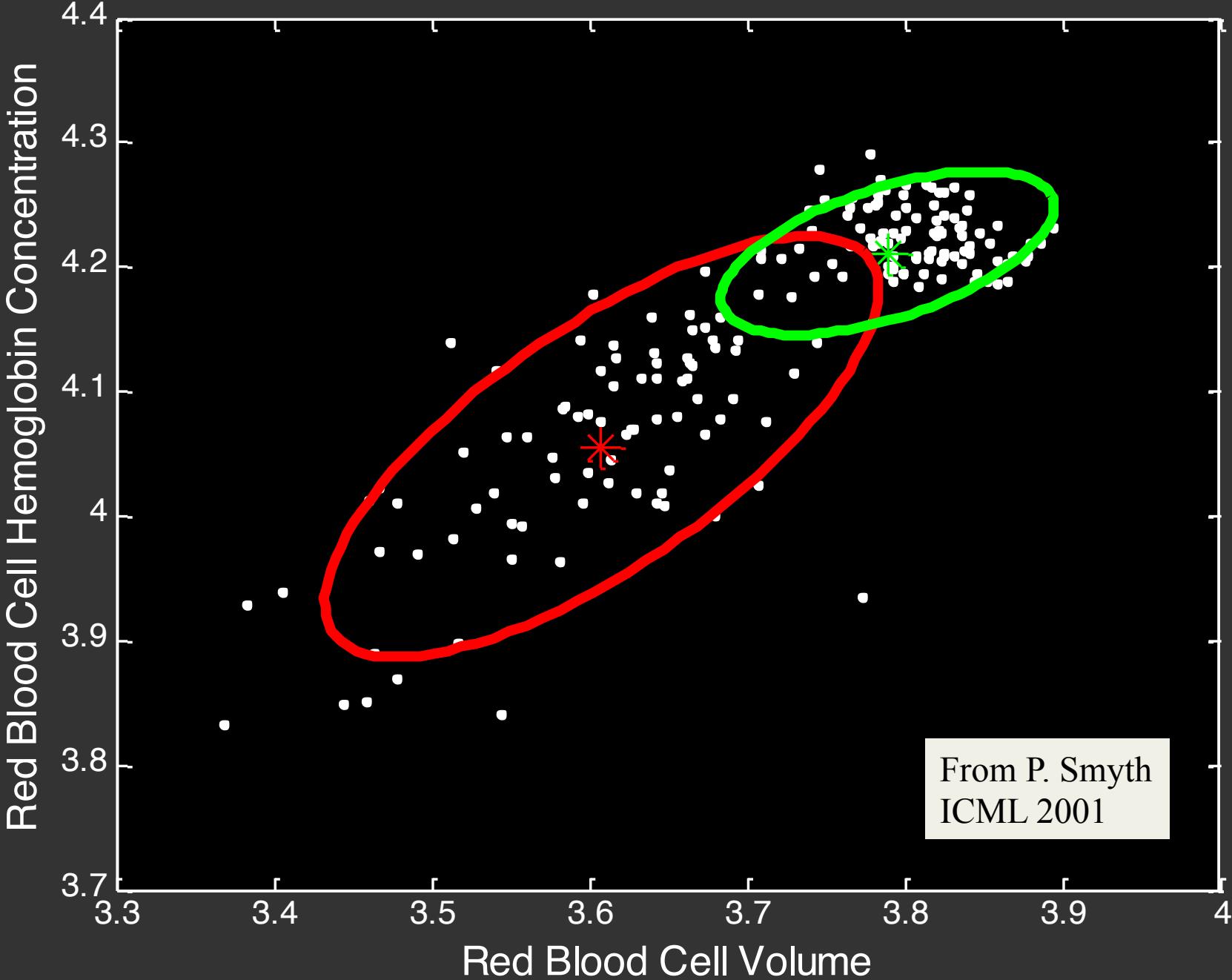
### EM ITERATION 3



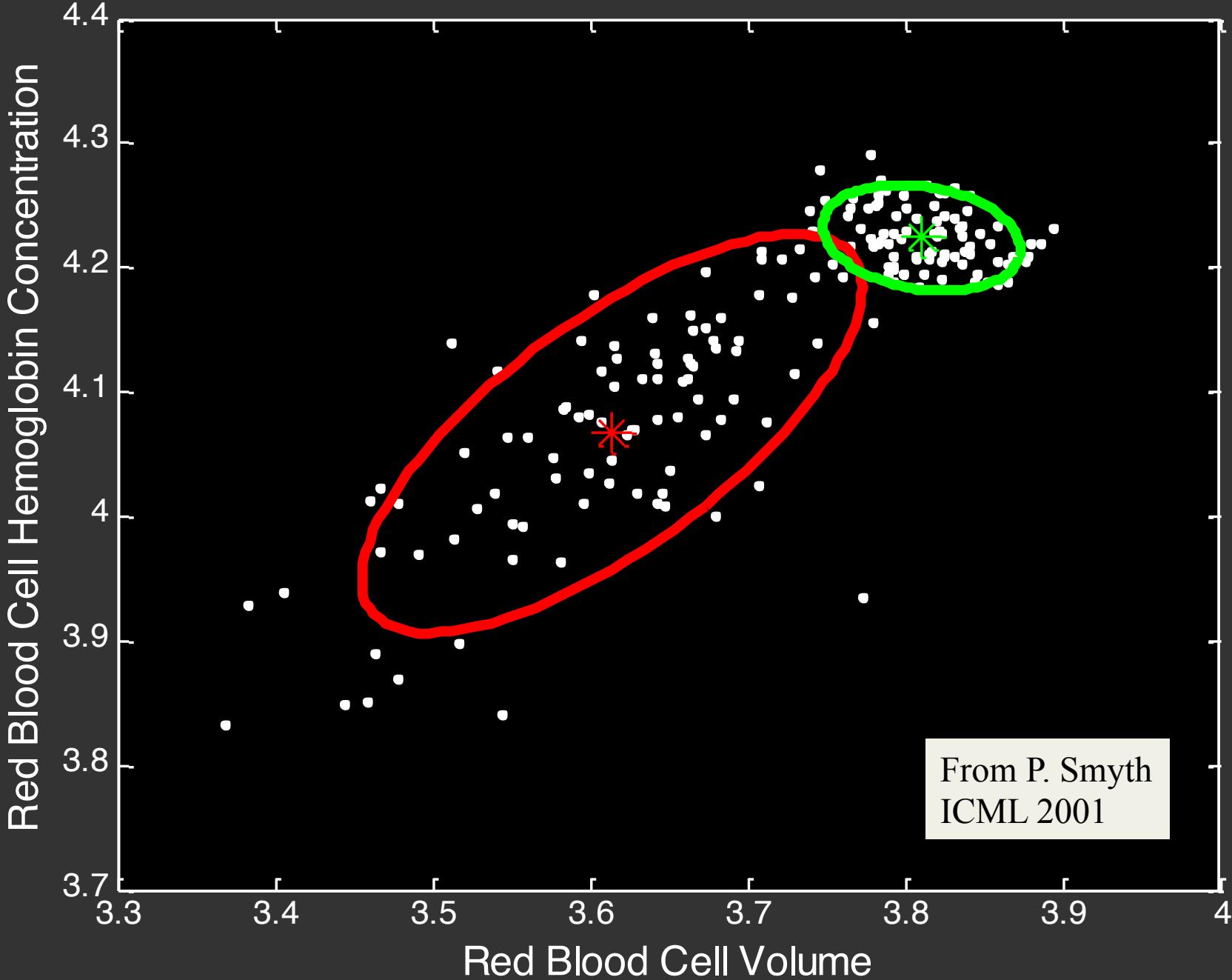
# EM ITERATION 5



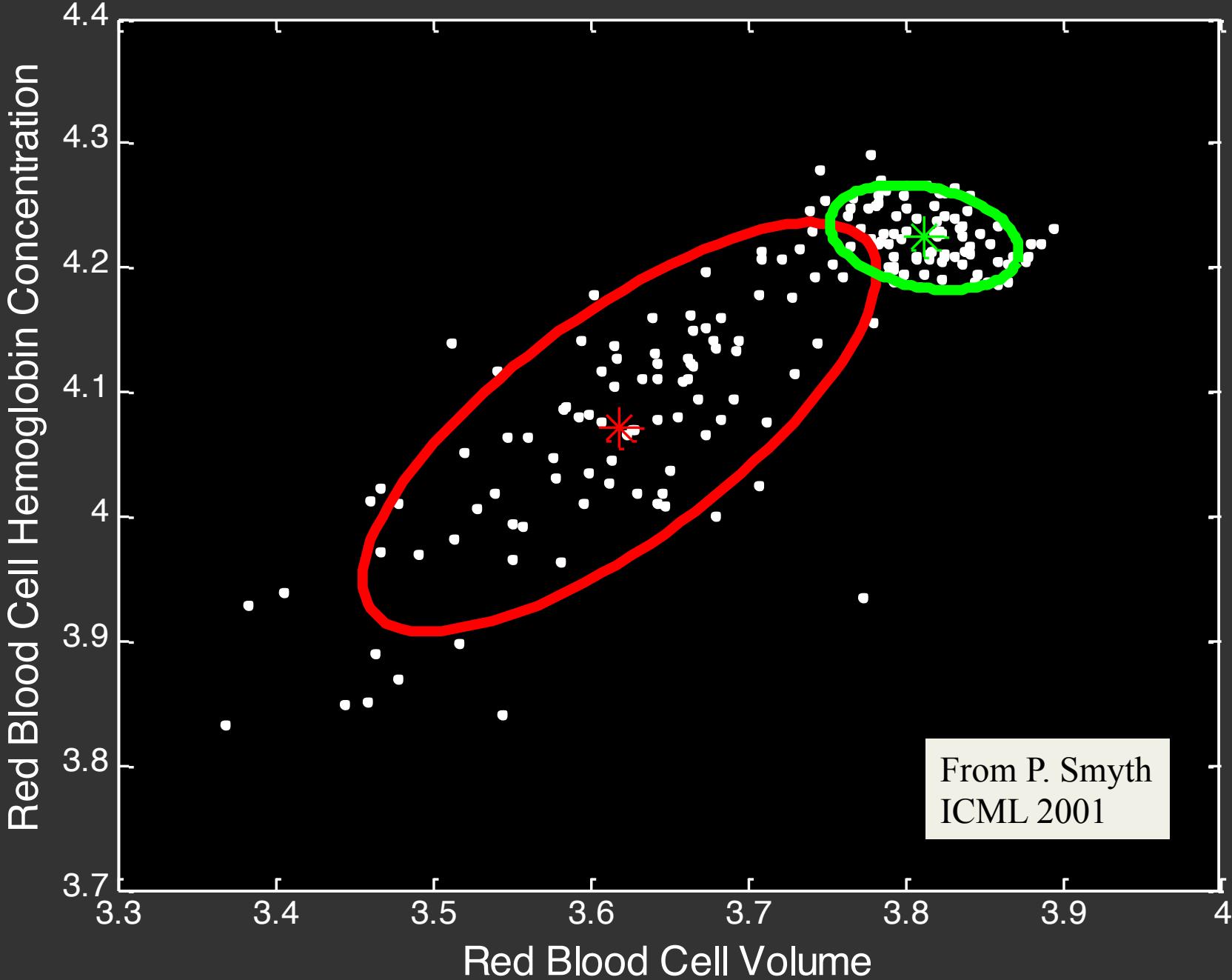
# EM ITERATION 10



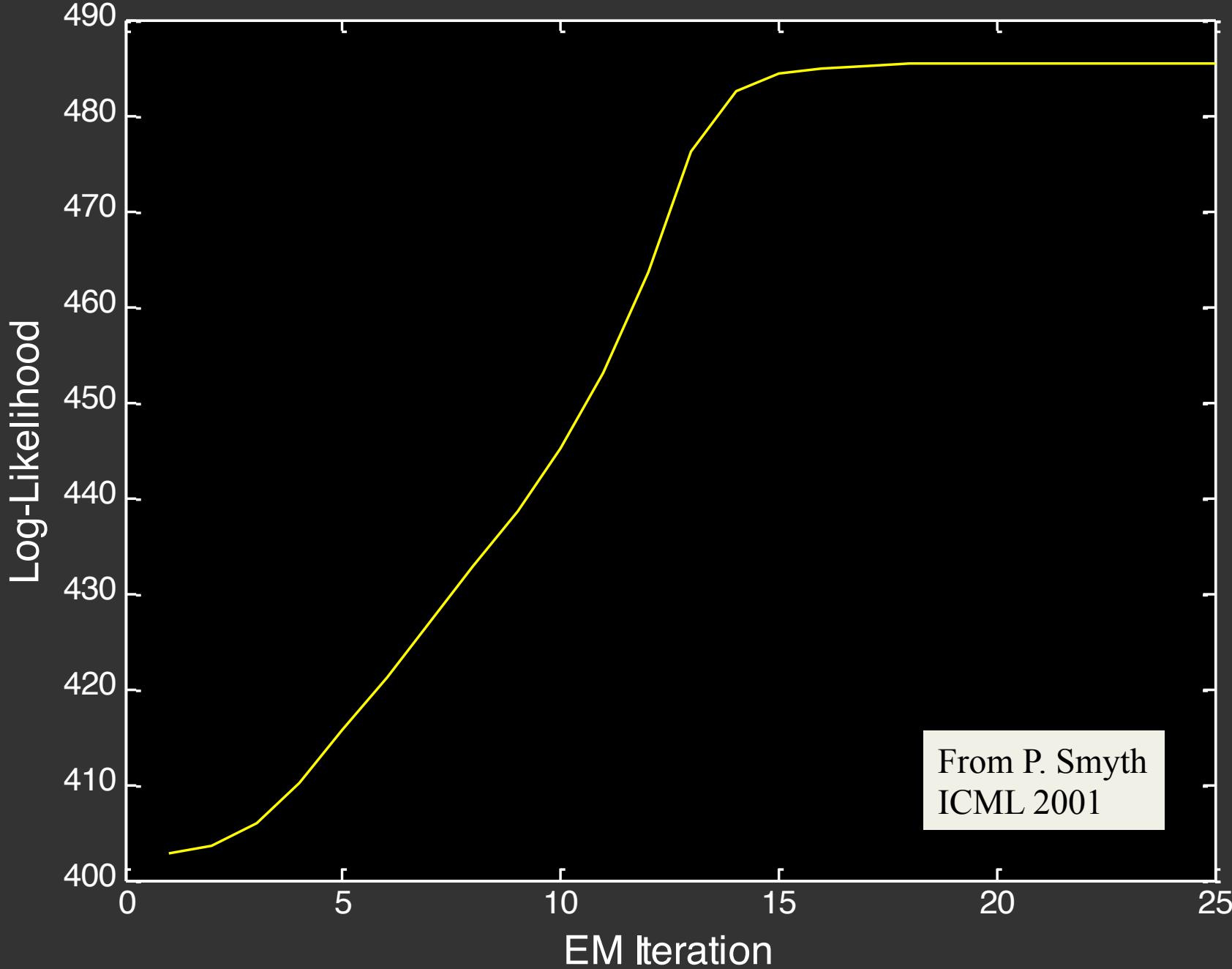
# EM ITERATION 15



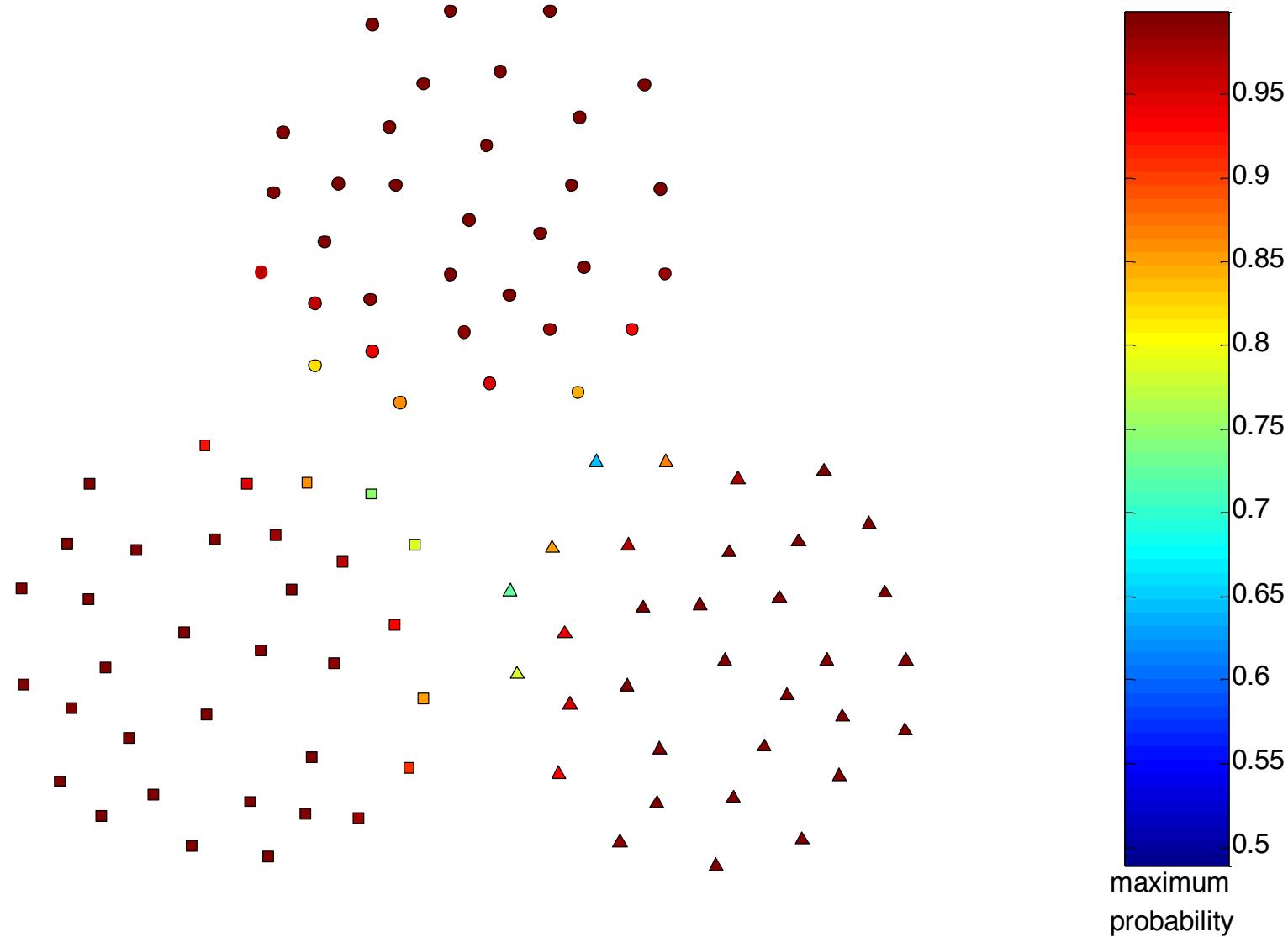
# EM ITERATION 25



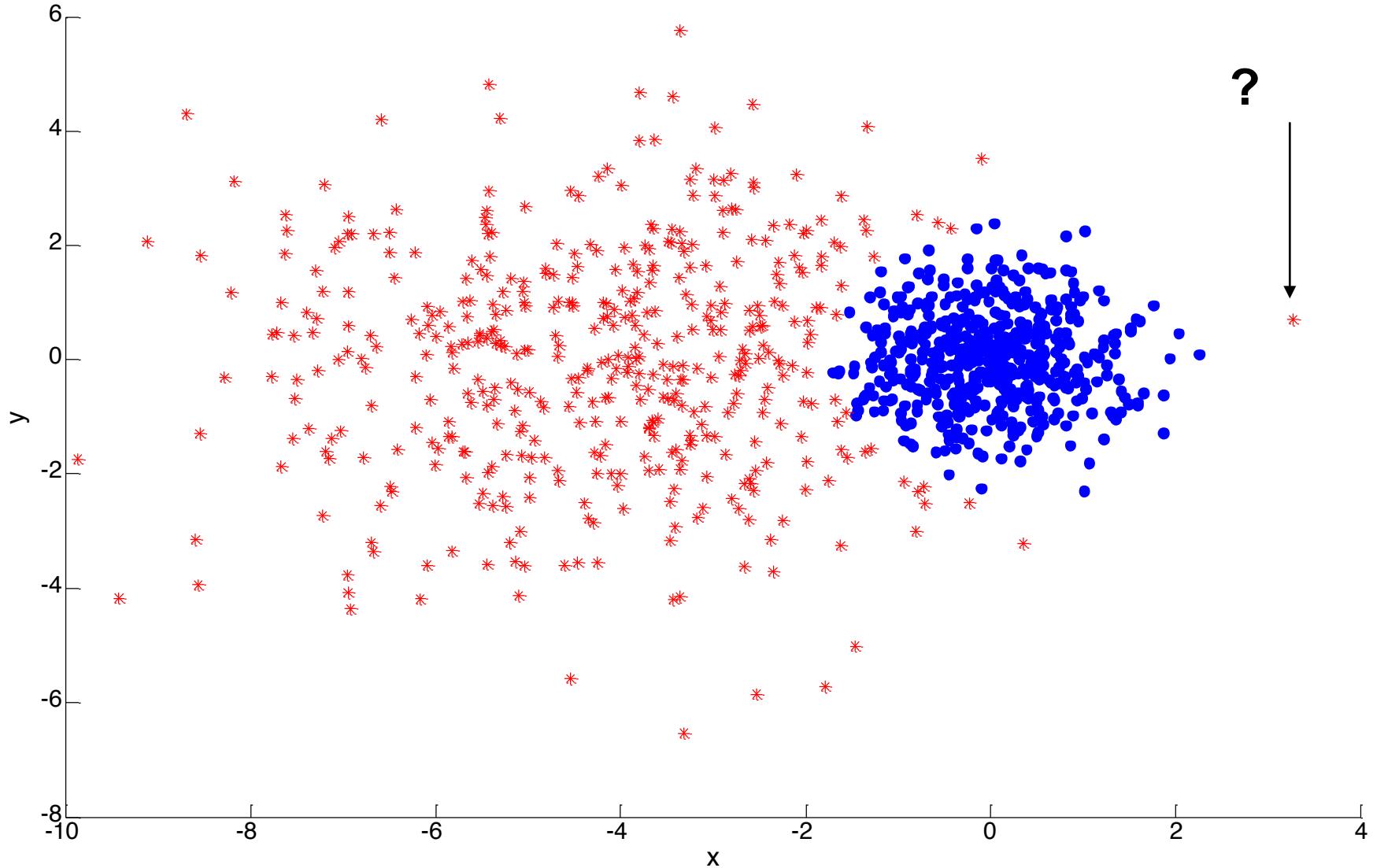
# LOG-LIKELIHOOD AS A FUNCTION OF EM ITERATIONS



# Trzy skupiska odkryte algorytmem EM



# Probabilistic Clustering: Dense and Sparse Clusters



# EM w porównaniu do algorytmu k-średnich

- EM łączy podejście probabilistyczne oraz zasadę oszacowanie MLE z paradrygmatem algorytmów iteracyjno-optymalizacyjnych
- Podobieństwa do algorytmu k-średnich, gdyż
  - podział obiektów także do k-skupisk (choć może być „miękki - soft” przydział)
  - Krok oczekiwania (E) odpowiednik przydziału odległościowego do najbliższego centroidu
  - Krok maksymalizacji (M) aktualizacja oszacowań parametrów odpowiednik przeliczania położenia centroidów, lecz z wykorzystaniem maksymalizacji funkcji wiarygodności LogL

# EM w porównaniu do algorytmu k-średnich

- EM jest bardziej ogólny niż k-średnich, z uwagi na różne rozkłady można modelować skupiska o innych kształtach niż sferyczne (np. eliptyczne)
- Silniejsze założenia i podstawy statystyczne
- W literaturze dalsze rozszerzanie lub wykorzystanie modeli probabilistycznych (np. CEM, SNOB, AUTOCLASS), także w wersji hierarchicznej – przegląd książki T.Morzy Eksploracja danych

# Ograniczenia algorytmu EM

- Zbieżność może być powolna
- Poszukuje tzw. lokalne minimum
- Dobór liczby skupisk  $k$  – nie jest łatwy (są propozycje automatyzacji – patrz książka K.Stąpor)
- Trudność dopasowania do potencjalnych skupisk będących b. rzadkie (z małą ilością obiektów)
  - Nieodporny na obecność obserwacji samotniczych (outliers) lub szumu (noise points)
- Liczba parametrów modeli wzrasta  $O(d^2)$ , gdzie  $d$  jest liczbą cech
  - Zwłaszcza l. parametrów dla macierzy kowariancji

# Grupowanie pojęciowe

- Tworzenie skupisk, które modeluje potencjalne pojęcia ukryte w danych – oraz automatycznie wspiera opisy skupisk w języku potencjalnie interpretowalnym przez człowieka
- Pierwsze algorytmy symboliczne (dla atrybutów jakościowych, oraz wykorzystanie zmodyfikowanych algorytmów odkrywania reguł)
- R.Michalski, R. Stepp: Learning from observation. Conceptual Clustering (1983)

# Cluster (Michalski, Stepp)

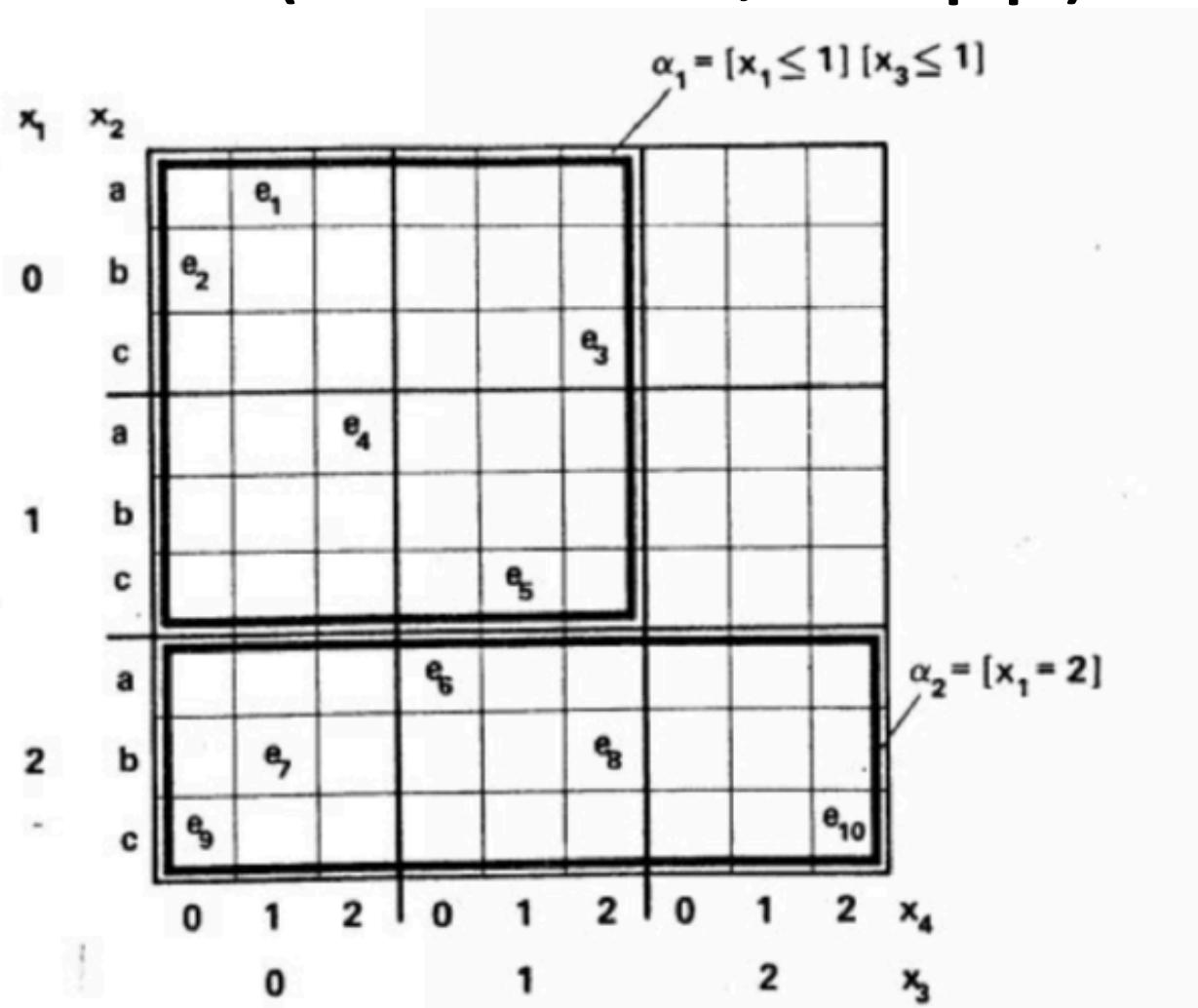


Figure 11-11: A diagrammatic representation of the clustering  $\{\alpha_1, \alpha_2\}$ .

# Cluster (Michalski, Stepp)

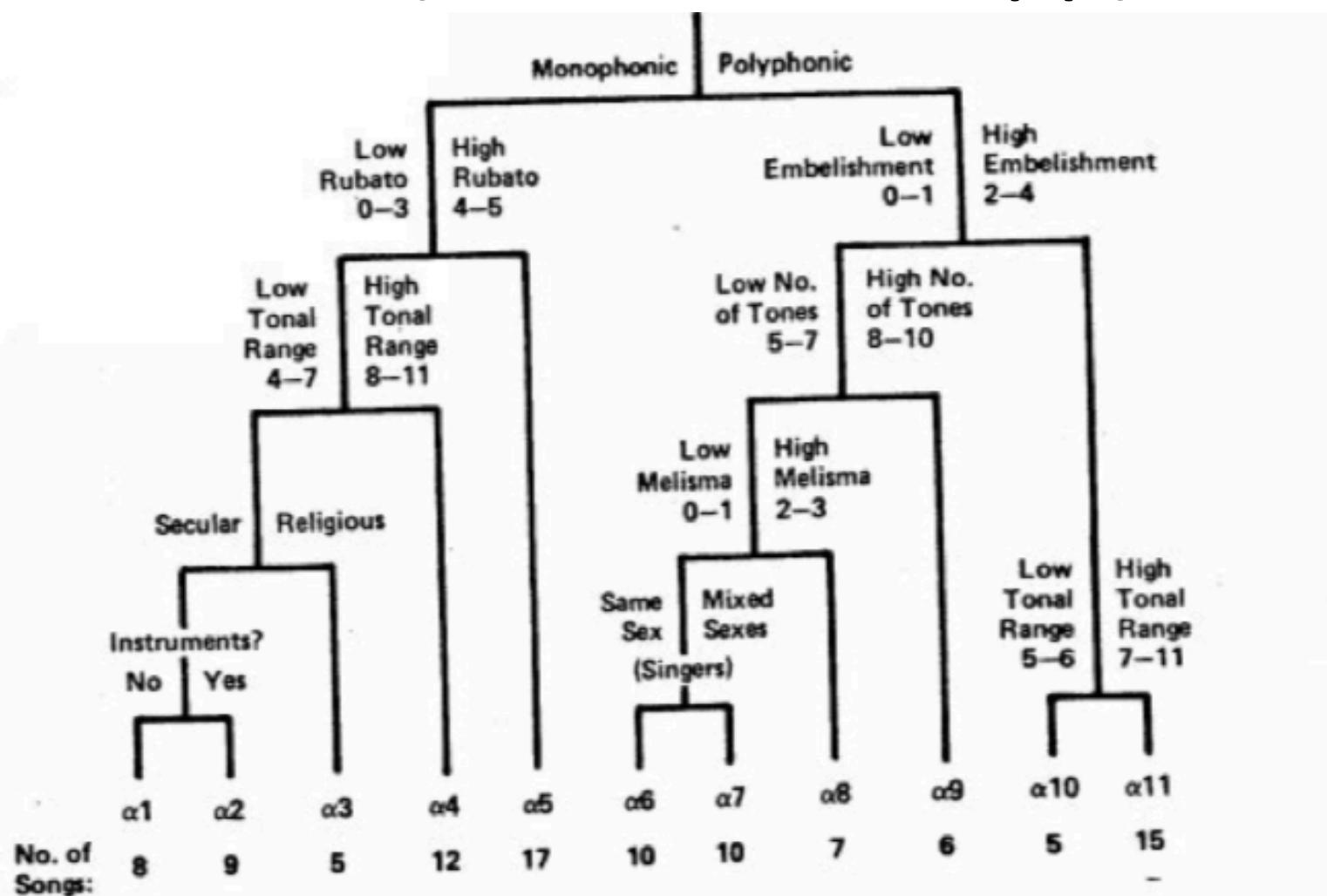


Figure 11-13: A classification hierarchy of Spanish folk songs produced by CLUSTER/2.

# COBWEB – różne elementy w jednym

- Podejście hierarchicznego grupowania
- Wsparcie dla tworzenia opisów pojęć z wykorzystaniem prawdopodobieństw
- Elementy probabilistycznego modelowania
- Uczenie przyrostowe z danych

Ogólny schemat postępowania:

Start:

tree consists of empty root node

Then:

add instances one by one

update tree appropriately at each stage

to update, find the right leaf for an instance

May involve restructuring the tree (split leaf or join to cluster)

Base update decisions on category utility

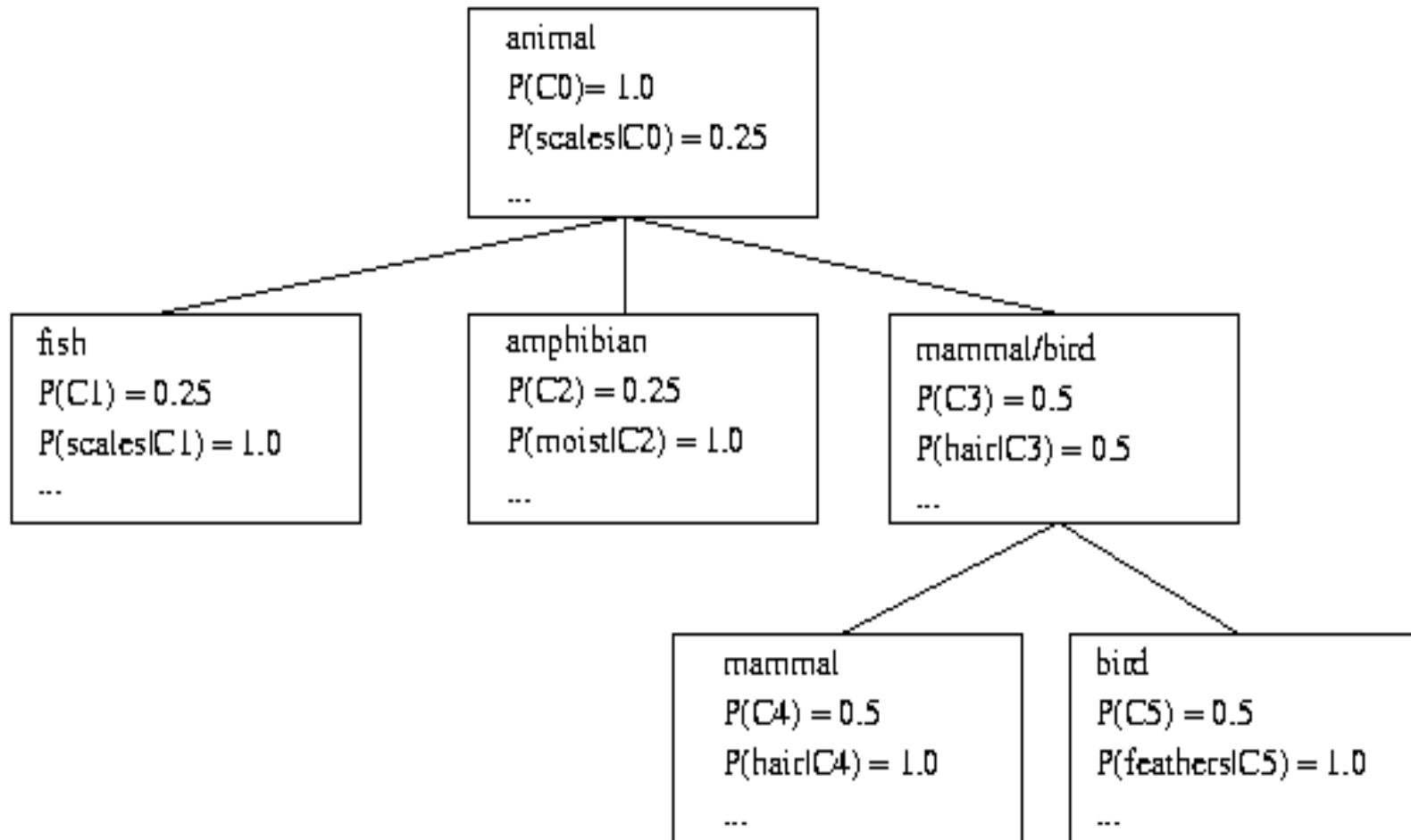
# COBWEB - D.Fisher 1986

- Podział zbioru obiektów tak, aby znaleźć taką strukturę kategorii (klas), która prowadzi do maksymalizacji informacji, jaką można przewidzieć znając kategorie przykładu (pot. klasyfikacja)
- heurystyczna funkcja oceny grupowania
  - Inspiracja wnioskowaniem Bayesowskim

$$\frac{1}{|C|} \sum_{d \in C} P(c(x) = d) \left[ \sum_{a_i} \sum_{v_{ij}} P(a_i(x) = v_{ij} | c(x) = d)^2 - \sum_{a_i} \sum_{v_{ij}} P(a_i(x) = v_{ij})^2 \right]$$

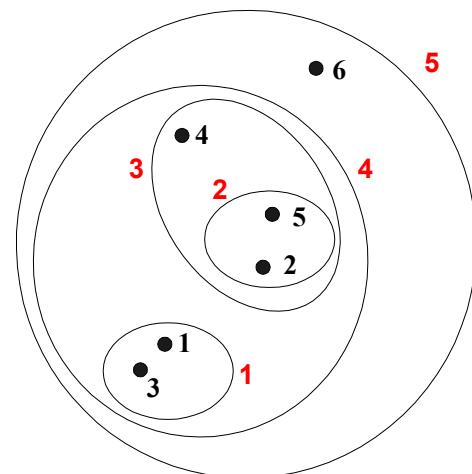
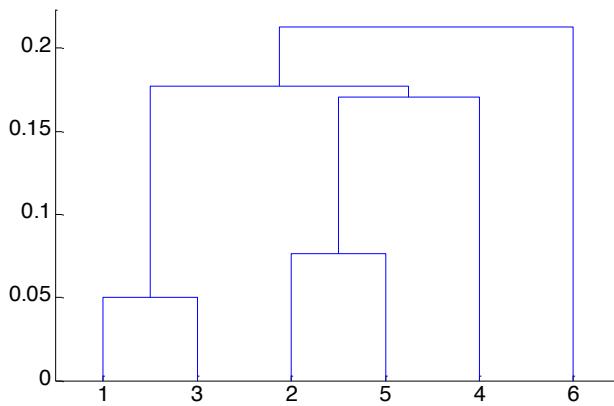
# COBWEB – dane zoo

Dynamiczna struktura drzewa + opis skupisk zestaw prawdop.



# Grupowanie hierarchiczne

- Tworzy się stopniowo hierarchię zawierających się skupisk
  - Połączenie lub podział podzbiorów obiektów
- Wizualizacja – struktura drzewa nazwana **dendrogramem**



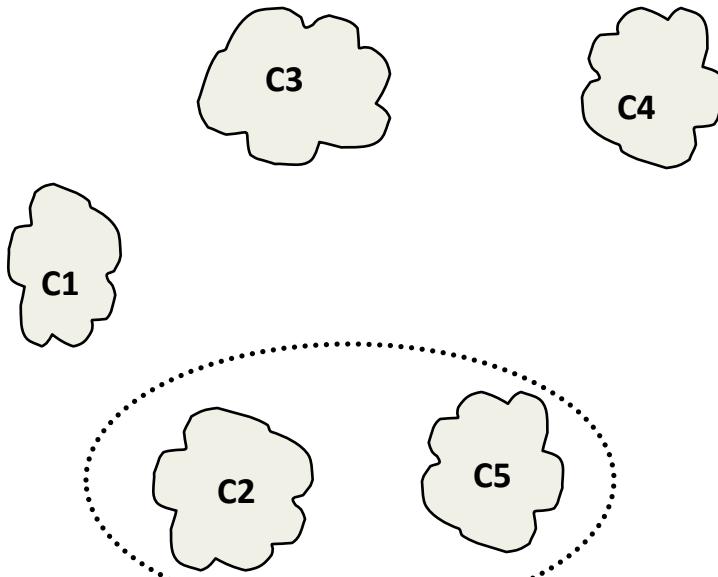
## Hierarchiczne metody aglomeracyjne - algorytm

1. W macierzy odległości znajduje się parę skupień najbliższych sobie.
2. Redukuje się liczbę klas łącząc znalezioną parę
3. Przekształca się macierz odległości metodą wybraną jako kryterium klasyfikacji
4. Powtarza się kroki 1- 3 dopóki nie powstanie jedna klasa zawierająca wszystkie skupienia.

# Jak przeliczać macierz odległości?

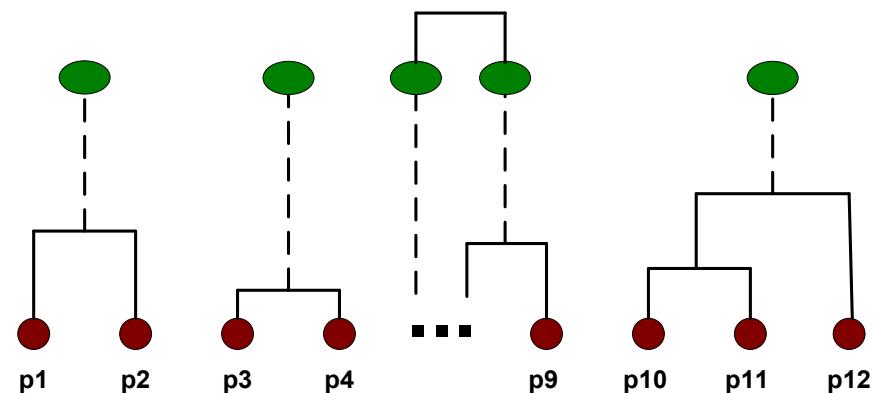
Łączymy dwa skupiska (C2 i C5) i aktualizujemy macierz odległości

Metody hierarchiczne różnią sposobem łączenia skupisk (ang. Linkage method)



	c1	c2	c3	c4	c5
c1					
c2					
c3					
c4					
c5					

Macierz odległości



# Hierarchiczne grupowanie wybór metody łączenia

1. Najbliższego sąsiedztwa (*Single linkage, Nearest neighbor*)
2. Najdalszego sąsiedztwa (*Complete linkage, Furthest neighbor*)
3. Mediany (*Median clustering*)
4. Środka ciężkości (*Centroid clustering*)
5. Średniej odległości wewnątrz skupień  
(*Average linkage within groups*)
6. Średniej odległości między skupieniami  
(*Average linkage between groups*)
7. Minimalnej wariancji Warda (*Ward's method*)

# Odległości między skupieniami

Single linkage  
minimum distance:

$$d_{\min}(C_i, C_j) = \min_{p \in C_i, p' \in C_j} \|p - p'\|$$

Complete linkage  
maximum distance:

$$d_{\max}(C_i, C_j) = \max_{p \in C_i, p' \in C_j} \|p - p'\|$$

mean distance:

$$d_{mean}(C_i, C_j) = \|m_i - m_j\|$$

average distance:

$$d_{ave}(C_i, C_j) = 1 / (n_i n_j) \sum_{p \in C_i} \sum_{p' \in C_j} \|p - p'\|$$

$m_i$  Jest średnią obiektów z  $C_i$      $n_i$  Jest liczbą obiektów w skupisku  $C_i$

# Single Link Agglomerative Clustering

- Użyj maksymalnego podobieństwa dwóch obiektów:

$$sim(c_i, c_j) = \max_{x \in c_i, y \in c_j} sim(x, y)$$

- Prowadzi do „(long and thin) clusters due to *chaining effect*” (efekt łańcuchowy); prowadzić do formowania grup niejednorodnych (heterogenicznych);
  - Dogodne w specyficznych zastosowaniach
- Pozwala na wykrycie **obserwacji odstających**, nie należących do żadnej z grup, i warto przeprowadzić klasyfikację za jej pomocą na samym początku, aby wyeliminować takie obserwacje i przejść bez nich do właściwej części analizy

# Complete Link Agglomerative Clustering

- Użyj maksymalnej odległości – minimalnego podobieństwa

$$sim(c_i, c_j) = \min_{x \in c_i, y \in c_j} sim(x, y)$$

- Ukierunkowana do “tight,” spherical clusters
- Metoda zalecana gdy, kiedy obiekty faktycznie formują naturalnie oddzielone "kępki". Metoda ta nie jest odpowiednia, jeśli skupienia są w jakiś sposób wydłużone lub mają naturę "łańcucha".

# Wrażliwość na dobór metod łączenia skupień

Diagram dla 22 przyp.

Pojedyncze wiązanie

Odległości euklidesowe

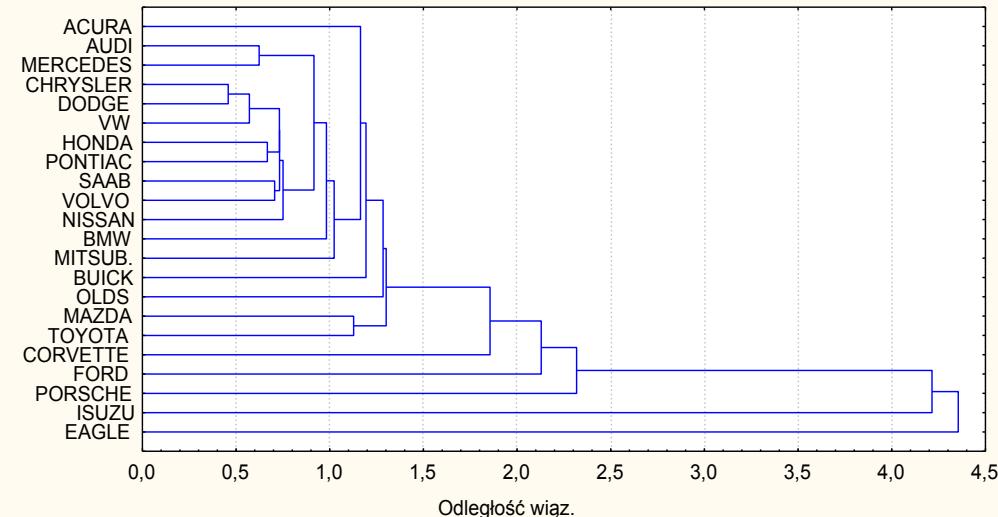
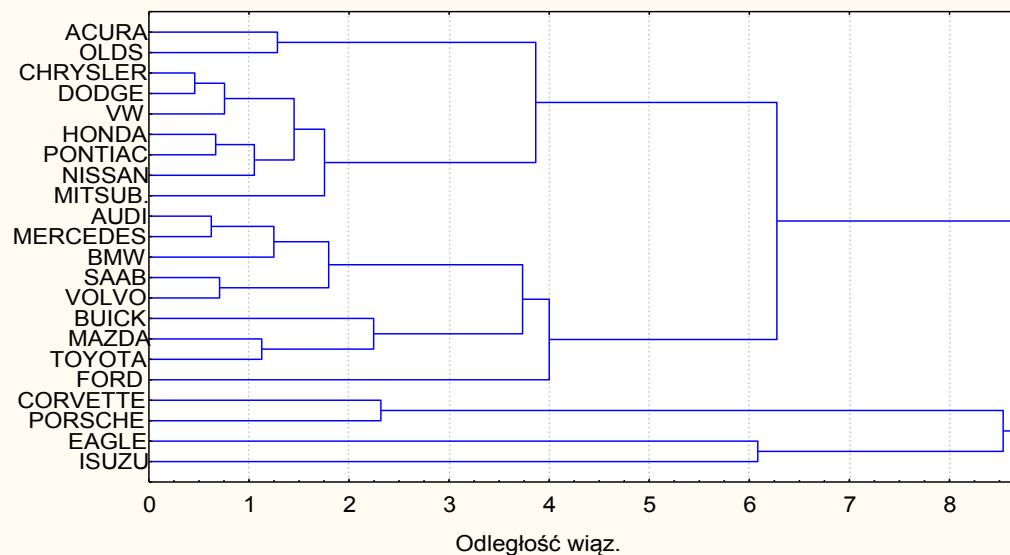


Diagram dla 22 przyp.

Metoda Warda

Odległości euklidesowe



Rysunki – z własnego  
uruchomienia Statsoft Statistica

# Metoda średnich połączeń [Unweighted pair-group average]

- W metodzie tej odległość między dwoma skupieniami oblicza się jako średnią odległość między wszystkimi parami obiektów należących do dwóch różnych skupień
- Metoda ta jest efektywna, gdy obiekty formują naturalnie oddzielone "kępki", ale zdaje także egzamin w przypadku skupień wydłużonych, mających charakter "łańcucha"

# Metoda ważonych środków ciężkości (mediany) [Weighted pair-group centroid]

- Jest to metoda podobna jak poprzednia, z tym wyjątkiem, że w obliczeniach wprowadza się „ważenie”, aby uwzględnić różnice między wielkościami skupień (tzn. liczbą zawartych w nich obiektów).
- Zatem, metoda ta jest lepsza od poprzedniej w sytuacji, gdy istnieją (lub podejrzewamy, że istnieją) znaczne różnice w rozmiarach (liczności) skupień

# Metody łączenia – Ward method

- Gdy powiększamy jedno ze skupień  $C_k$ , wariancja wewnętrzgrupowa (liczona przez kwadraty odchyлеń od średnich w zbiorach  $C_k$ ) rośnie.
- Metoda polega na takim powiększaniu zbiorów  $C_k$ , która zapewnia **najmniejszy przyrost tej wariancji** dla danej iteracji.
- Kryterium grupowania jednostek: minimum zróżnicowania wektorów cech  $x_j$  tworzących zbiór  $C_k$  ( $k = 1, \dots, K$ ) względem wartości średnich w tych zbiorach.
- Ogólnie, metoda ta jest traktowana jako bardzo efektywna, chociaż zmierza do tworzenia skupień o małej wielkości → zrównoważone drzewa o wielu elementach
- Ważne – powiązanie z miarą odległości między obiektami (Pearson vs. inne)

# Przykłady użycia metody Warda

## Cars data

Diagram dla 22 przyp.

Metoda Warda

Odległości euklidesowe

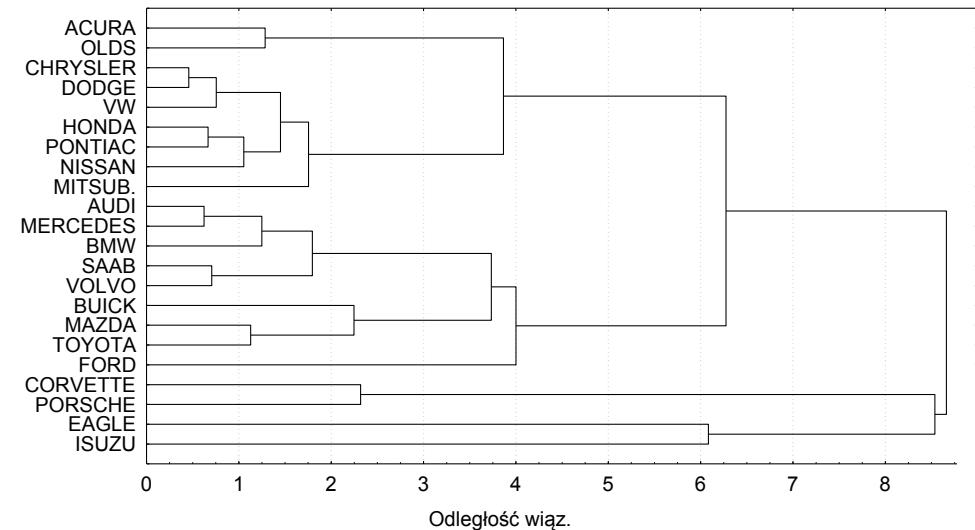
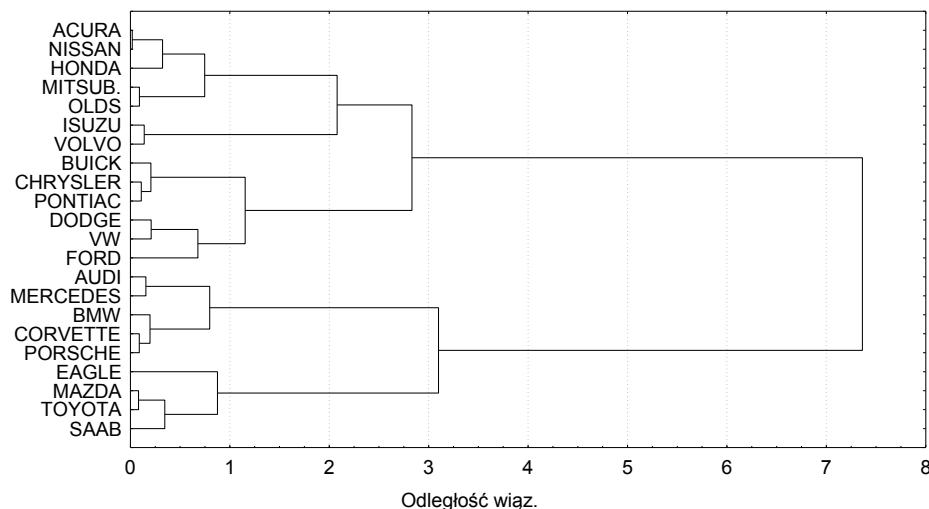


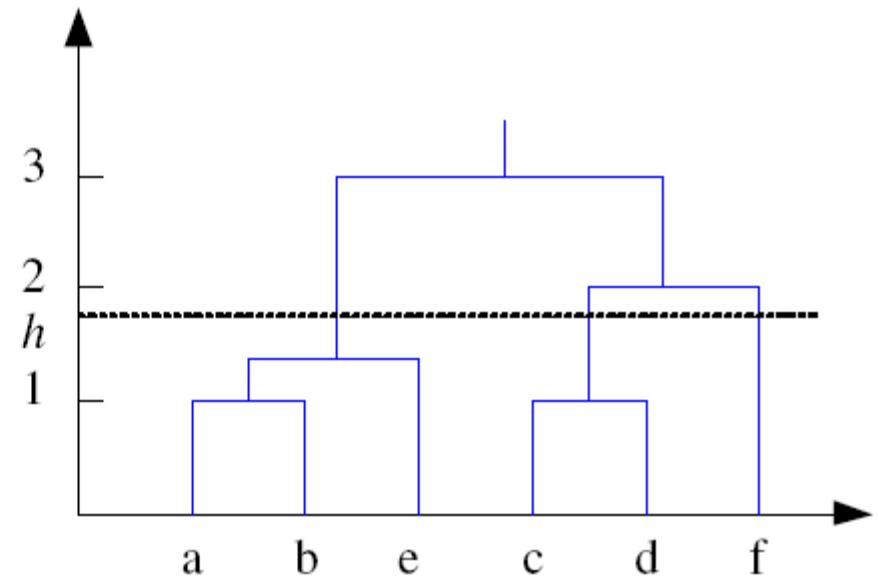
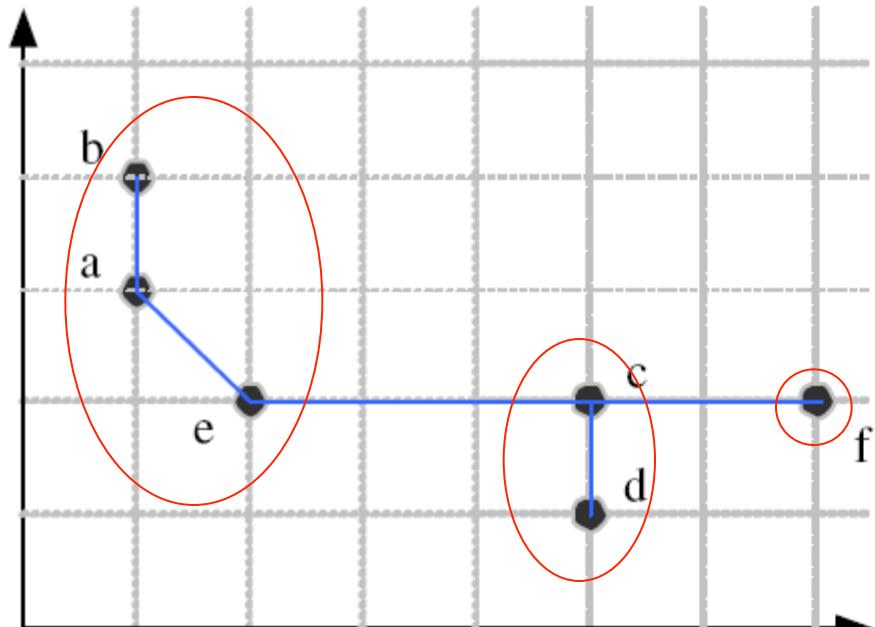
Diagram dla 22 przyp.

Metoda Warda

1-r Pearsona



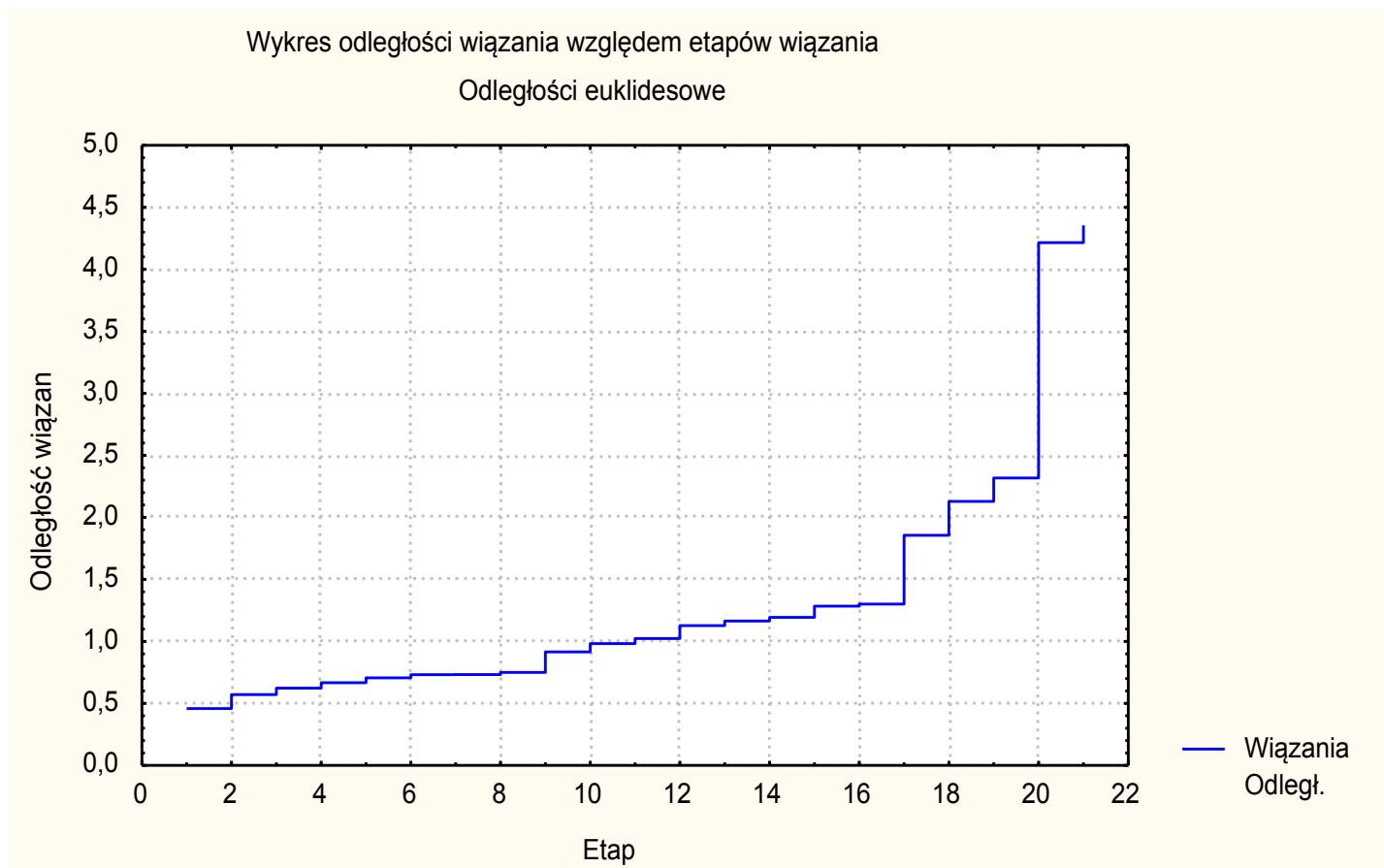
# Jak wykorzystać AHC do oszacowania potencjalnej liczby skupisk w danych?



*Dendrogram*

# AHC – jak odnaleźć liczbę skupień?

Znajdź punkt przegięcia („kolanko”) wykresu odległości wiązania względem kolejnych etapów łączenia obiektów / skupisk



# Grupowanie Dużych Repozytoriów

- Skalowalność (względem liczby przykładów lecz także wysokiej wymiarowości atrybutów)
- Uwzględnianie złożonych typów danych
- Konstruowanie dowolnych kształtów skupień
- Wspomaganie parametryzacji (jak dobrać  $k$ , parametry DBSCAN, itp.) = postulaty tzw. AUTOML
- Odporność na szum i obserwacje nietypowe
- Grupowanie przyrostowe
- Przetwarzanie strumieni danych
- Zmiana położenia definicji pojęć (data shift / w przypadku strumieni tzw. concept drift)

# Problemy i wyzwania

- Od lat 90tych widoczny postęp w zakresie skalowalnych algorytmów (zwłaszcza ja dane nie mieszczą się w PAO):
  - Partitioning:  $k$ -means,  $k$ -medoids, PAM, CLARANS
  - Hierarchical: BIRCH, CURE
  - Density-based: CLIQUE, OPTICS
  - Grid-based: STING, WaveCluster.
  - Model-based: Autoclass, Denclue, Cobweb.

Lecz

- Obecne techniki ciągle nie spełniają wystarczająco dobrze stawianych wymagań
- Otwarte problemy i wyzwania badawcze; zwłaszcza dla nietypowych i złożonych danych

# Metody hierarchiczne dla dużych zbiorów danych

- Niektóre z ograniczeń metod aglomeracyjnych:
  - słaba skalowalność: złożoność czasowa przynajmniej  $O(n^2)$ , gdzie  $n$  jest liczbą obiektów,
  - „krytyczne” znaczenie decyzji o wyborze punktu połączenia kolejnych skupień w trakcie budowania drzewa hierarchii,
  - algorytmy nie zmieniają, ani nie poprawiają, wcześniej podjętych decyzji.
- Rozwinięcia algorytmów hierarchicznych oraz ich integracja z metodami gęstościowymi:
  - BIRCH (1996): użycie drzew o strukturze „CF-tree”, uczenie przyrostowe i stopniowa poprawa jakości pod-skupień.
  - CURE (1998): wybór losowy odpowiednio rozproszonych punktów, wstępne grupowanie z określeniem ich punktów reprezentatywnych, łączenie grup w nowe skupienia wraz z przesuwaniem punktów reprezentatywnych w stronę środków tworzonego skupienia zgodnie z „shrinking factor  $\alpha'$ ”; eliminacja wpływu „outliers”.

# BIRCH – efektywne rozszerzenie grupowanie hierarchicznego

- Działa efektywnie: decyzja dla jednej grupy (dzielenie czy połączenie z inną grupą) nie wymaga przeglądania całego zbioru danych
- I/O koszt jest liniowy względem rozmiaru danych: przeglądanie zbioru danych raz
- Ukierunkowany na tworzenie zrównoważonego drzewa hierarchii skupisk

## BIRCH – ang. Balanced Iterative Reducing and Clustering using Hierarchies – Zhang et al. (1996)

- Wykorzystuje hierarchiczne drzewo CF (ang. Clustering Feature)
- Działanie algorytmu:
  - **Faza 1**: przyrostowo przeczytaj raz DB (zew. Baza danych) w celu zbudowania w pamięci początkowej struktury drzewa CF (rodzaj wielopoziomowej kompresji danych zachowującej wewnętrzną strukturę zgrupowań danych).
  - **Faza 2**: zastosuj wybrany (inny) algorytm skupień dla lepszego pogrupowania obiektów w liściach drzewa CF.
- *Dobra skalowalność*: znajduje zadawalające grupowanie po jednokrotnym przeczytaniu bazy danych i ulepsza je wykorzystując niewielu dodatkowych operacji odczytu DB.
- *Ograniczenia*: zaproponowany dla danych liczbowych, wrażliwość wyników na kolejność prezentacji przykładów.

# Informacje o danych skupiskach

$CF$  - struktura wykorzystywana w konstrukcji drzewa

Podstawowe parametry BIRCH:

$B$  – maksymalna liczba rozgałęzień w drzewie

$L$  – maksymalna liczba obiektów w liściu

$T$  – maksymalny promień (grup) w liściu

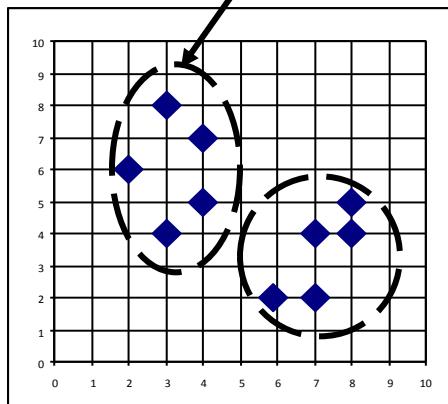
**Clustering Feature:**  $CF = \overrightarrow{(N, LS, SS)}$

$N$ : Number of data points

$$LS: \sum_{i=1}^N \overrightarrow{X_i}$$

$$SS: \sum_{i=1}^N \overrightarrow{X_i^2}$$

$$CF = (5, (16,30),(54,190))$$



(3, 4)

(2, 6)

(4, 5)

(4, 7)

(3, 8)

# Przykładowa struktura CF Tree

Korzeń -Root

$B = 7$

$L = 6$

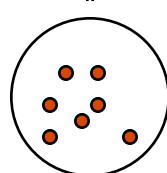
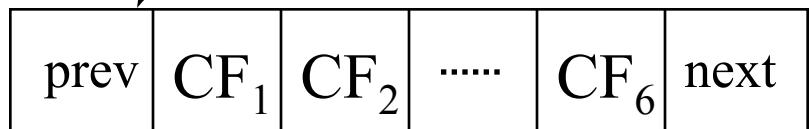
$CF_1$	$CF_2$	$CF_3$	.....	$CF_6$
$child_1$	$child_2$	$child_3$		$child_6$

Węzły pośrednie

$CF_1$	$CF_2$	$CF_3$	.....	$CF_5$
$child_1$	$child_2$	$child_3$		$child_5$

Końcowy liść

Końcowy liść



Dalszy podział na skupiska innym “szymbkim” algorytmem

# Przetwarzanie kolejnego przykładu

- Wstawienie przykładu do struktury drzewa

**Krok 1.** Wybierz liść  $l$  do wstawiania. Użyj jednej z funkcji odległości do wyznaczenia najbliższej grupy do badanego punktu

**Krok 2.** Jeśli w liściu  $l$  jest miejsce to wstaw  $x$ ,

Jeśli nie Podziel liść  $l$  na dwa liście i popraw ścieżkę od  $l$  do korzenia.

**Krok 3.** Rekonstrukcja drzewo przez połączenie dwa najbliższe węzły i podzielić na dwa (w razie potrzeby): merge i resplite

# Inne algorytmy grupowania

Środowisko data mining:

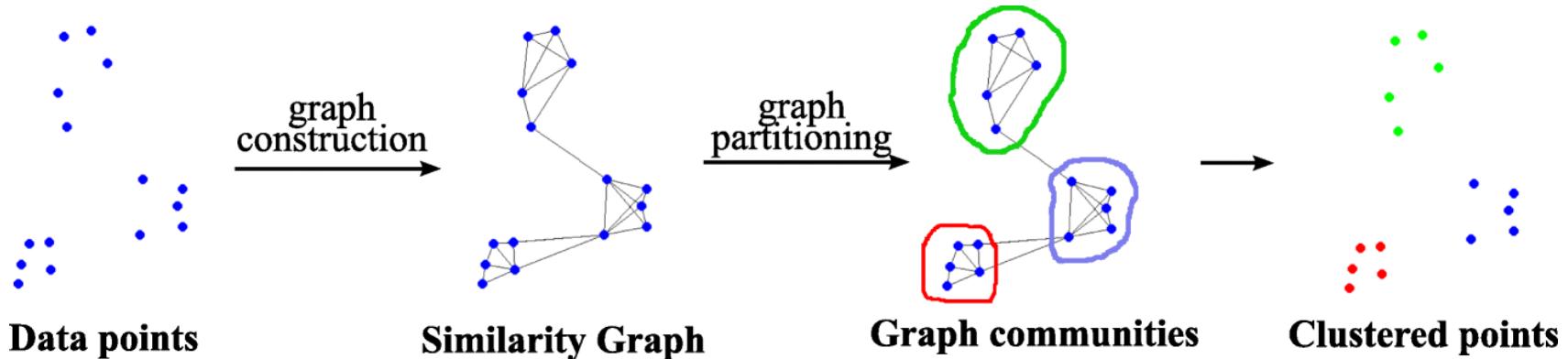
- Algorytmy “gridowe”
  - STING, CLIQUE, WaveCluster
- Grafowe algorytmy
  - Chameleon
  - Jarvis-Patrick
  - Based on Nearest Neighbor (SNN)
- Tzw. Sub-space clustering
- Online stream clustering

# Grafowe algorytmy

Przeznaczone do przetwarzania danych jako grafów (często dużych, np. w odniesieniu do Internetu lub sieci społecznych, lub dotyczących naturalnie występujących powiązań, np. związki chemiczne)

Także grupujące standardowe reprezentacje danych, wykorzystując ich strukturę wewnętrzną modelowaną jako graf.

- Grafy najbliższych sąsiadów (np.  $\epsilon$ -ball, kNN and CkNN graphs)
- Drzewa rozpinające w grafach



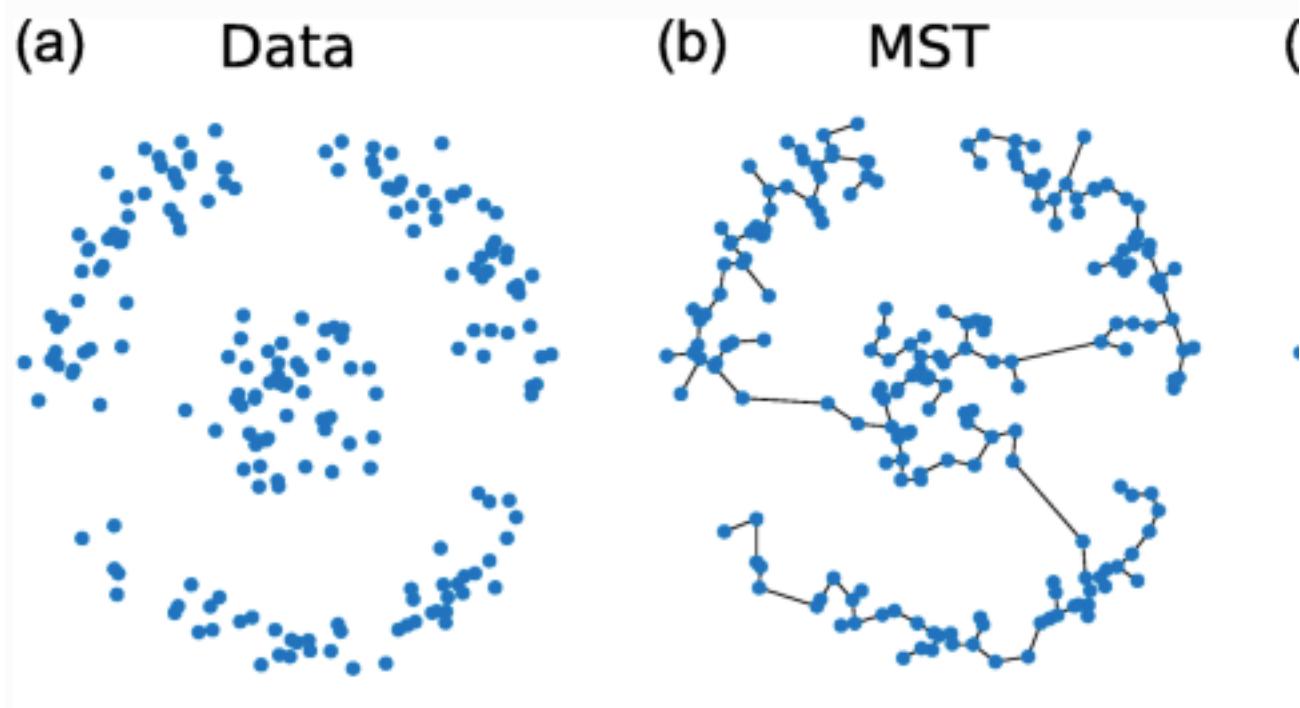
# Minimalne drzewa rozpinające (MST)

Inne podejścia do grupowania wykorzystujące globalne własności geometrii połączeń pomiędzy punktami

Minimalne drzewo rozpinające może to modelować. Później połączone z analizą k-sąsiadów lub tzw. dendrytów (metoda wrocławska) można zidentyfikować skupiska

Definicja formalna: Niech  $G = (V, E, f)$  będzie nieskierowanym grafem spójnym nieskierowanym ważonym (Wagi odległości lub podobieństwa między wierzchołkami). Minimalnym drzewem rozpinającym (minimum spanning tree, MST) w  $G$  nazwiemy takie drzewo rozpinające (graf bez cykli, który łączy wszystkie rozważane wierzchołki), w którym suma wag krawędzi jest najmniejsza możliwa.

# Minimalne drzewa rozpinające (MST)



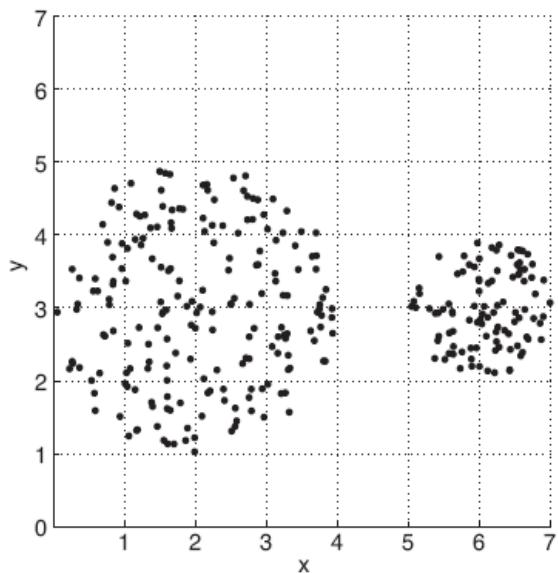
# Grid-based Clustering

---

## Algorithm 9.4 Basic grid-based clustering algorithm.

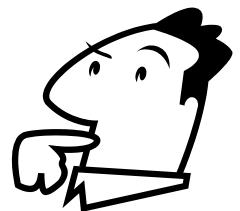
---

- 1: Define a set of grid cells.
  - 2: Assign objects to the appropriate cells and compute the density of each cell.
  - 3: Eliminate cells having a density below a specified threshold,  $\tau$ .
  - 4: Form clusters from contiguous (adjacent) groups of dense cells.
- 



0	0	0	0	0	0	0
0	0	0	0	0	0	0
4	17	18	6	0	0	0
14	14	13	13	0	18	27
11	18	10	21	0	24	31
3	20	14	4	0	0	0
0	0	0	0	0	0	0

# Ocena jakości skupień



# Czy można poszukiwać pojedynczej miary?

- Pewne „trudne” rady

*“The problem of how to judge the quality of a clustering is difficult and there seems to be no universal answer to it.”*

*“The nature of processes leading to useful classifications remains little understood, despite considerable effort in this direction.”*  
— R. Michalski, R. Stepp [MS83]

*“How do you know the resulting classifications are any good?”*  
— D. Fisher [Fis87]

Ocena wyników algorytmów grupowania – wielokryterialna / brak pojedynczej dominującej miary takiej jak w uczeniu nadzorowanym oraz zależna od rodzaju grupowania (płaska, hierarchiczna, ...)

# Różne spojrzenia na ocenę grupowania

- Wewnętrzne (ocena tylko charakterystyki skupień i rozkładem przykładów)
  - Brak dodatkowych źródeł informacji, np. zbioru odniesienia etykiet
  - Miary oceny oparte na danych (internal measures)
- Zewnętrzne
  - „Benchmarking on existing labels”
  - Porównanie skupień z tzw. ground-truth categories / zadanym podziałem
- Ocena ekspercka

# Różne spojrzenia na ocenę grupowania

Ponadto miary są przeznaczone do określonego rodzaju grupowania:

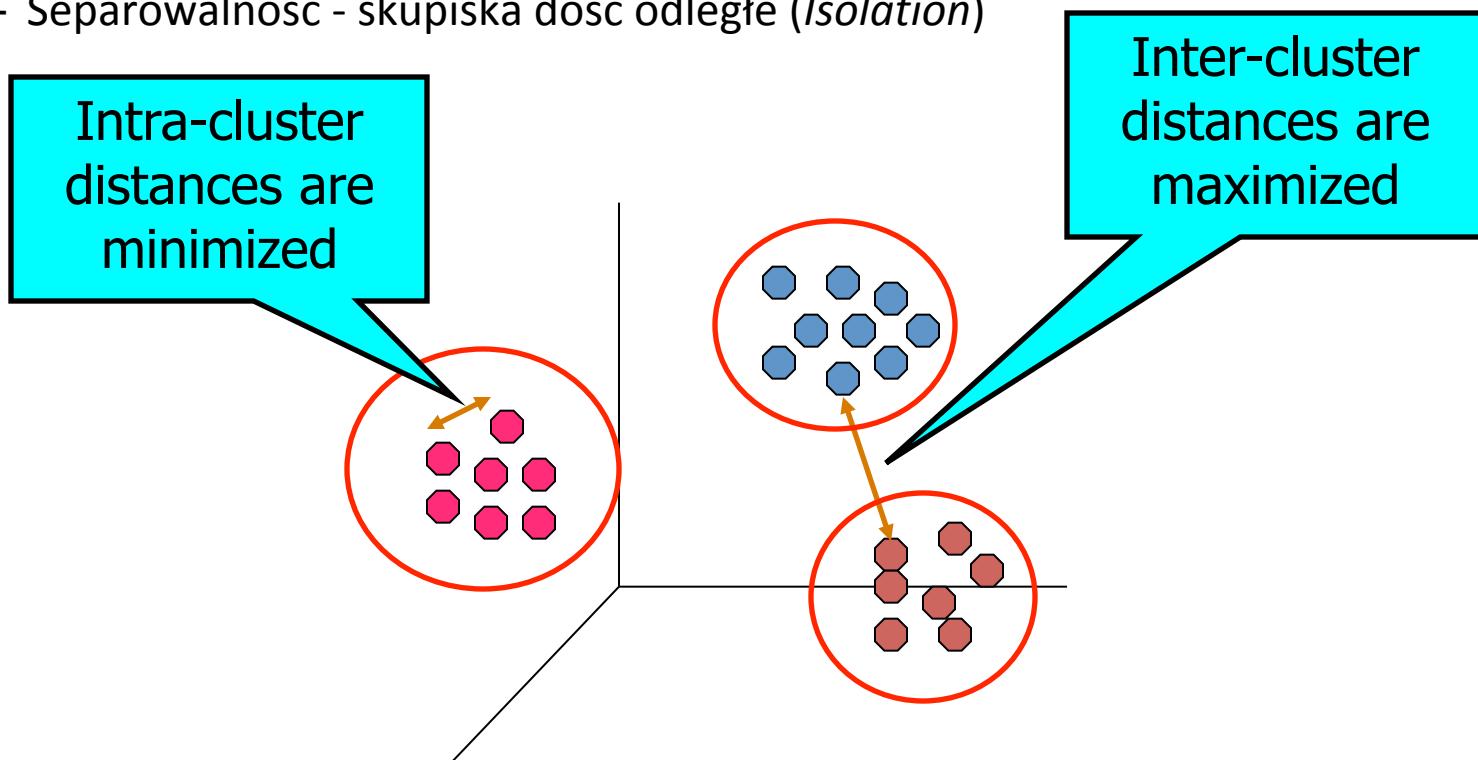
- Płaska struktura skupisk vs. hierarchiczna
- Jednoznaczne (ang. Crisp) przydziały do skupisk vs. rozmyte przynależności

Rozpoczniemy do płaskiej struktury (jak k-means) z jednoznaczymi przydziałami = literatura ponad 30 różnych propozycji

# Ocena wewnętrzna jakości skupień

## Miary oceny oparte na danych (internal measures)

- Oparte na odległościach
- Duże podobieństwo obiektów wewnętrz skupienia (*Compactness*) / zwartość skupiska
- Separowalność - skupiska dość odległe (*Isolation*)



# Podstawowe miary wewnętrzne

- Zwartość skupisk (możliwe bliskie obserwacje  
– mała średnica skupisk lub odległość od centroidu)
- Separowalność skupisk (powinny być maksymalnie odróżnialne od siebie; średnie odległości pomiędzy parami punktów lub środków skupień)

# Najprostsze miary

$K$  – skupisk  $C_k$ , każde o liczności  $n_k$

Skupiska  $C_k$  charakteryzowane przez centroidy - średnie obiekty w skupieniu

$$\mathbf{r}_k = \frac{1}{n_k} \sum_{\mathbf{x} \in C_k} \mathbf{x}$$

- Błąd zmienności wewnętrz skupieniowej

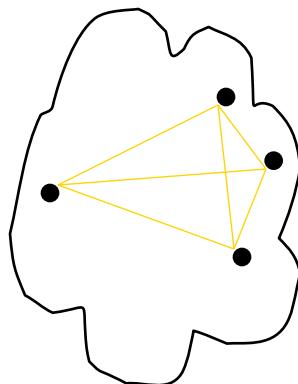
$$wc(C) = \frac{1}{K} \sum_{k=1}^K \frac{1}{n_k} \sum_{\mathbf{x} \in C_k} d(\mathbf{x}, \mathbf{r}_k)$$

- Separowalność (odległości między centroidami)

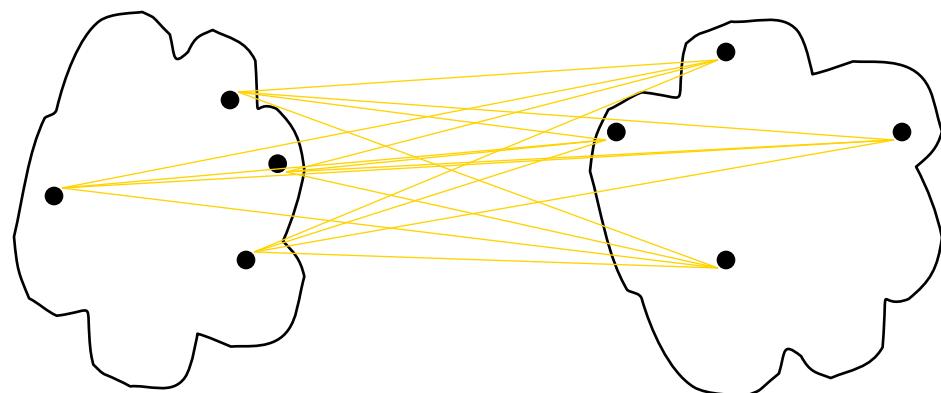
$$bc(C) = \frac{1}{K(K-1)/2} \sum_{1 \leq j < k \leq K} d(\mathbf{r}_j, \mathbf{r}_k)$$

# Inne miary: Cohesion and Separation

- A proximity graph – graf podobieństwa można wykorzystać
  - Cluster cohesion is the sum of the weight (odległości) of all links within a cluster.
  - Cluster separation is the sum of the weights (odległości) between nodes in the cluster and nodes outside the cluster.



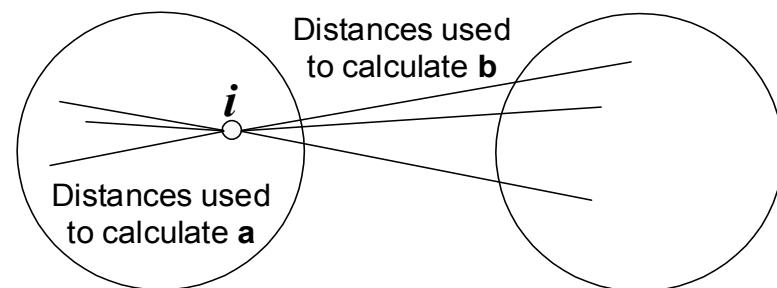
cohesion



separation

# Silhouette Coefficient / współczynnik sylwetkowy lub zarysu

- Współczynnik Silhouette wykorzystuje podobieństwo obiektu do innych obiektów w tym samym skupisku z odniesieniem do podobieństwa do obiektów z innych skupisk
- Dla pojedynczego obiektu  $i$ 
  - Oblicz  $a$  = średnia odległość obiektu  $i$  do innych obiektów w tym samym skupisku
  - Oblicz  $b$  = min (średnia odległość  $i$  punktów z innego skupiska)
  - The silhouette coefficient jest zdefiniowany jako
$$s = (b - a) / \max(a, b)$$
  - Zakres od -1 do 1
  - Na ogół pomiędzy 0 i 1.
  - Im bliższe wartości 1 tym lepsze skupisko
- Można dalej obliczać średnie wartości dla całego skupiska lub zbioru skupisk

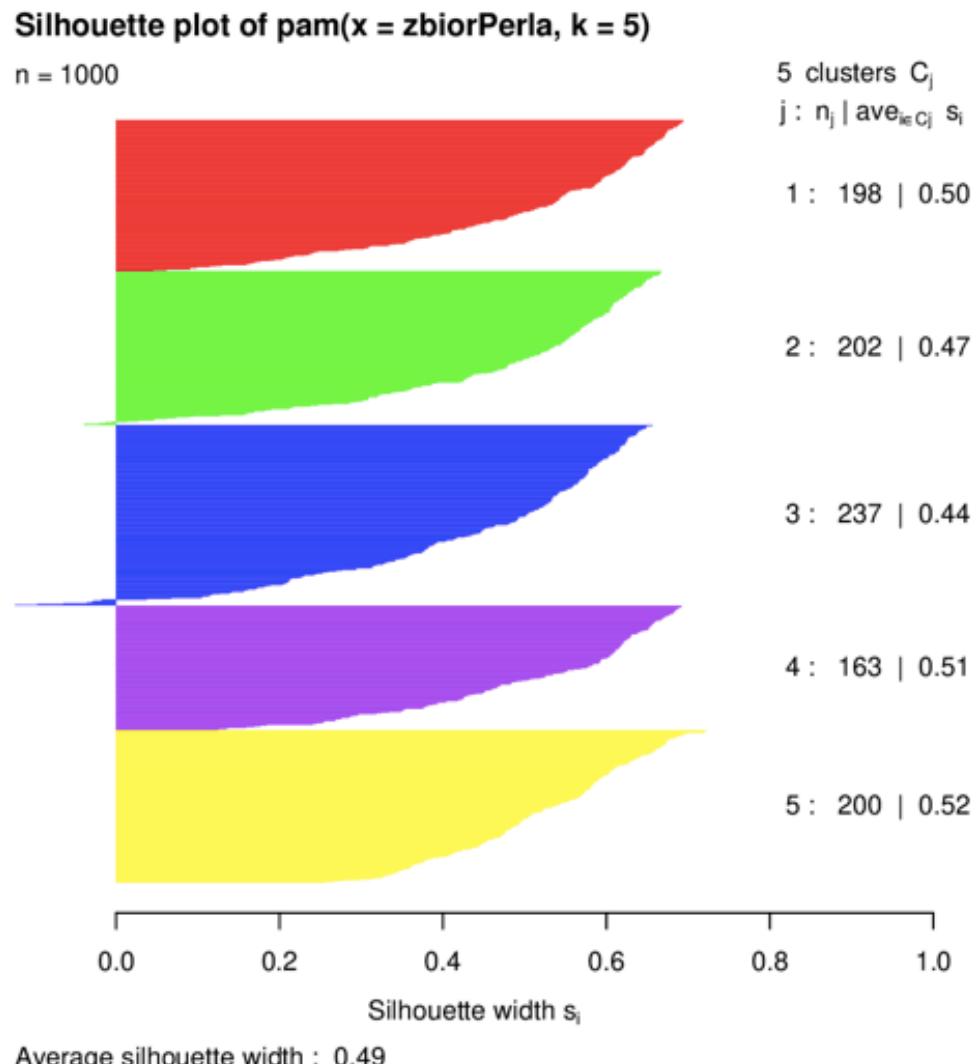


Średnia wartość zarysu  $\bar{s}_k$ , dla każdego skupienia mówi o tym jak dobrze dane są przydzielone do tego skupienia. Z tego względu średni zarys dla całego zbioru danych może służyć jako miara jakości podziału. **Współczynnik zarysu** ma postać  $SC = \max_k \bar{s}_k$ , i jego interpretacja została zawarta w poniższej tabeli.

SC	Interpretacja
0,71-1,00	Silna struktura
0,51-0,70	Istotna struktura
0,26-0,50	Słaba struktura
$\leq 0,25$	Brak struktury

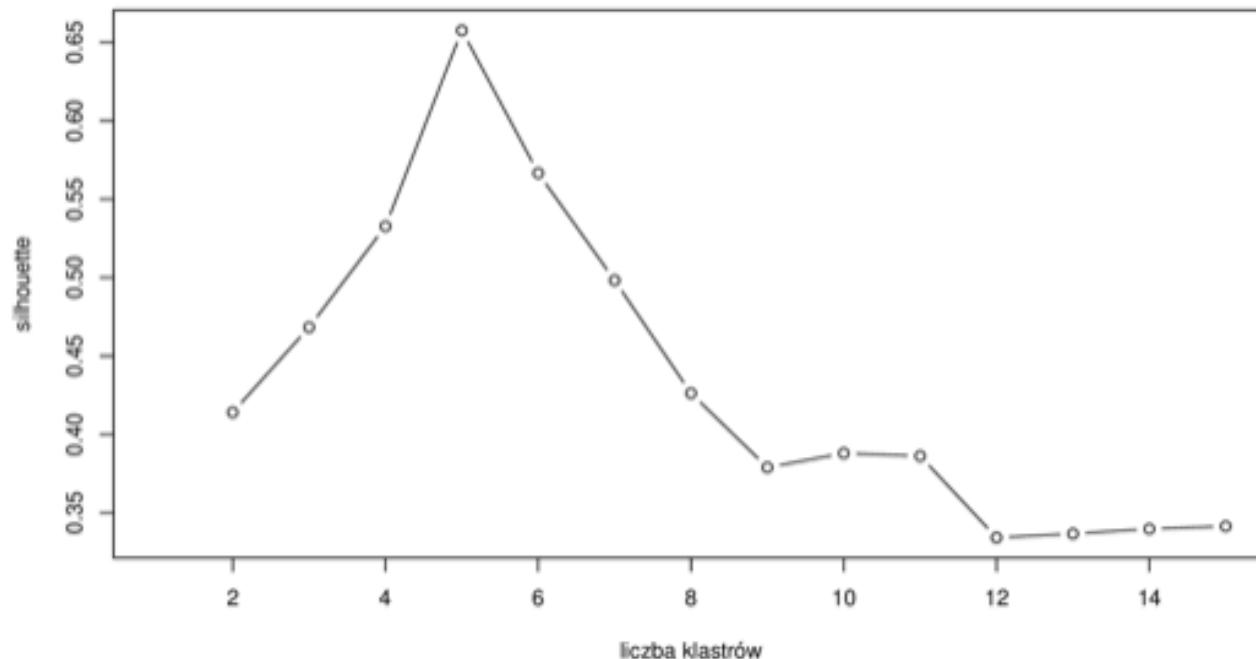
Zarys podlega wizualizacji za pomocą **wykresu zarysu**.

# Przykład użycia współczynnika Silhouette



Rysunek 3.6: Wykres dopasowania punktów do poszczególnych klastrów z użyciem miary silhouette.

# Wykorzystanie Silhouette do wyboru liczby skupisk w algorytmie PAM



## Inne kryteria wewnętrznej jakości skupień

- Compactness → determining the weakest connection within the cluster, i.e., the largest distance between two objects  $R_i$  and  $R_k$  within the cluster.
- Isolation → determining the strongest connection of a cluster to another cluster, i.e., the smallest distance between a cluster centroid and another cluster centroid

$$\left( \sum_{C_j} \left( \frac{\max(D(R_i, R_k)) \text{ where } (R_i, R_k) \in C_j}{\min(D(C_j, C_m)) \text{ where } C_m \neq C_j} \right) \right)^{-1}$$

- Object positioning → the quality of clustering is determined by the extent to which each object  $R_j$  has been correctly positioned in given clusters

$$\sum_{R_i} (\max(D((R_i, R_k))) - \min(D(R_i, R_m)))$$

where  $(R_i, R_k) \in C_j$  and  $R_m \notin C_j$ .

# Inne współczynniki

- Współczynnik Dunna
- Wskaźnik Daviera-Bouldina
- Indeks **Calińskiego** i Harabasza / także b. przydatny do wyboru liczby skupień w k-średnich

- Indeks DAVIESA-BOULDINA (ang. *Davies-Bouldin index*):

$$DB = \frac{1}{n} \sum_{i=1}^K \max_{j \neq i} \frac{\sigma_i + \sigma_j}{d(c_i, c_j)},$$

gdzie  $\sigma_i$  jest średnią odległością wszystkich punktów ze skupienia  $i$  do jego środka, a  $d(c_i, c_j)$  jest odlegością pomiędzy środkami skupień  $i$  oraz  $j$ .

- Indeks DUNNA (ang. *Dunn index*):

$$D = \frac{\min_{1 \leq i < j \leq K} d(i, j)}{\max_{1 \leq k \leq K} d'(k)},$$

gdzie  $d(i, j)$  jest odlegością pomiędzy skupieniami  $i$  oraz  $j$ , a  $d'(k)$  odlegością wewnątrz skupienia  $k$ .

Caliński i Harabasz (1974) zaproponowali aby końcową liczbę skupień wybierać w oparciu o wartości indeksu postaci:

$$CH(K) = \frac{\text{tr}(B(C_K))/(K-1)}{\text{tr}(W(C_K))/(n-K)}.$$

Optymalną wartość  $K$  dobieramy tak, aby ją zmaksymalizować.

## Literatura



Caliński, T., Harabasz, J. (1974). *A dendrite method for cluster analysis*. Communications in Statistics 3(1):1–27.

tr – ślad macierzy :

Niech  $A$  będzie macierzą kwadratową stopnia  $n$ . Śladem macierzy  $A$  nazywamy wielkość

$$\text{tr}(A) = \sum_{i=1}^n a_{ii} = a_{11} + a_{22} + \cdots + a_{nn}.$$

## Rozkład całkowitej sumy kwadratów

Założymy, że dokonaliśmy podziału na  $K$  skupień.

$C(i) = k$  – gdy  $x_i$  należy do  $k$ -tego skupienia.

$T$  – suma kwadratów odległości między elementami tego samego skupienia i różnych skupień

$$T = W + B$$

$$W = \frac{1}{2} \sum_{k=1}^K \sum_{C(i)=k} \sum_{C(i')=k} d_{ii'}$$

$$B = \frac{1}{2} \sum_{k=1}^K \sum_{C(i)=k} \sum_{C(i') \neq k} d_{ii'}$$

(T – Total, W – Within, B – Between)

Minimalizacja  $W$ , czyli minimalizacja rozrzutu punktów wewnątrz skupień  $\equiv$  maksymalizacji rozrzutu punktów między skupieniami.

# Zewnętrzne miary

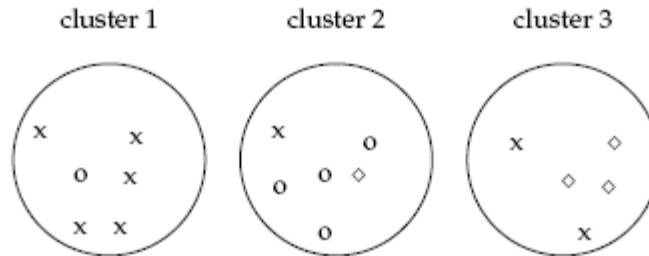
- Porównuje się strukturę skupisk vs. znany podział z zbioru testowego
- Część miar wykorzystuje specjalną tablicę dwudzielczą
- Na ogół stosowany do oceny algorytmu i porównania go do innych
- Popularne miary
  - Wskaźnik Randa
  - Purity
  - Odmiany miary F
- Przegląd miar – dostępny na [https://en.wikipedia.org/wiki/Cluster\\_analysis](https://en.wikipedia.org/wiki/Cluster_analysis)

# Ocena jakości algorytmu gdy znany jest właściwy przydział do klas (ang. ground truth)

## Jain's example

16.3 *Evaluation of clustering*

357



► **Figure 16.4** Purity as an external evaluation criterion for cluster quality. Majority class and number of members of the majority class for the three clusters are: x, 5 (cluster 1); o, 4 (cluster 2); and  $\diamond$ , 3 (cluster 3). Purity is  $(1/17) \times (5 + 4 + 3) \approx 0.71$ .

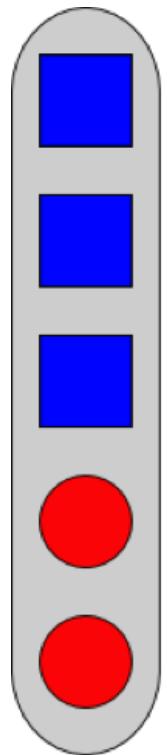
	purity	NMI	RI	$F_5$
minimum	0.0	0.0	0.0	0.0
maximum	1	1	1	1
value for Figure 16.4	0.71	0.36	0.68	0.46

► **Table 16.2** The four external evaluation measures applied to the clustering in Figure 16.4.

# Odniesienie do zewnętrznego podziału

- Dostępne referencyjne etykiety (manually labeled data)
  - Ekspert etykietuje w zależności od własności danych
  - Istnieją benchmarki TREC, Reuters, itp..
  - Tryb sem-supervised
- Różne podejścia:
  - „Accuracy of clustering: Percentage of pairs of tuples in the same cluster that share common label” (etykiety w skupieniach)
    - Faworyzujemy małe „czyste” skupienia
    - Czy powinniśmy mieć zgodność liczby skupień i etykiet

# Pewne problemy w ocenie odwzorowań



F-measure: 0.6

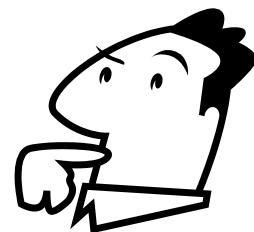
F-measure: 0.6

# Ogólne zasady

- Homogeneity - Jednorodności
  - Każde skupienie zawiera przykłady z jak najmniejszej liczby etykiet klas
  - Idealnie – tylko jedna klasa
- „Completeness”
  - Każda klasy reprezentowana w możliwe najmniejszej liczbie skupień
- Typowe miary
  - Purity
  - F-miara

# Ocena grupowania

- Inna niż w przypadku uczenia nadzorowanego (predykcji wartości)
- Poprawność grupowania zależna od oceny obserwatora / analityka
- Różne metody AS są skuteczne przy różnych rodzajach skupień i założeniach, co do danych:
  - Co rozumie się przez skupienie, jaki ma kształt, dobór miary odległości → sferyczne vs. inne
- Dla pewnych metod i zastosowań:
  - Miary zmienności wewnętrz i między – skupieniowych
  - Idea zbiorów kategorii odniesienia (np. TREC)



# Analiza skupień - podsumowanie

- Liczne i ważne zastosowanie praktyczne analizy skupień (AS).
- AS używana „samodzielnie” w zgłębianiu danych, lub jako jedno z narzędzi podczas wstępnego przetwarzania w procesie KDD.
- Jakość skupień i działanie wielu algorytmów związane są określeniem miary odległości obiektów.
- Podstawowe klasy metod:
  - hierarchiczne,
  - podziałowo/optymalizacyjne,
  - gęstościowe,
  - „grid-based”,
  - wykorzystujące modele matematyczne (np. probabilistyczne lub neuronowe).
- Ważne zagadnienie to także wykrywanie obiektów nietypowych (outliers discovery).

# Więcej w książkach, artykułach

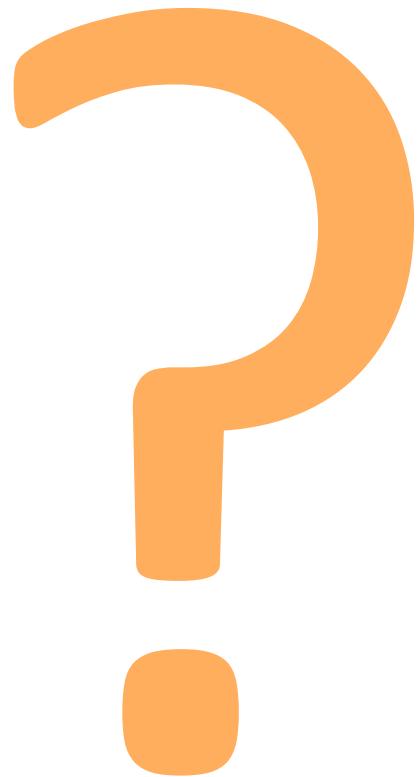
Szukaj też samodziałnie



# Wybrane źródła literaturowe

- A. D. Gordon: Classification. Chapman & Hall 1999
- B. S. Everitt, S. Landau, M. Leese, Cluster analysis, Oxford University Press, 2001

Może pytanie lub komentarze?



# Time is going on ...



# **Pytanie i komentarze?**

Dalszy kontakt:

[jerzy.stefanowski@cs.put.poznan.pl](mailto:jerzy.stefanowski@cs.put.poznan.pl)

<http://www.cs.put.poznan.pl/jstefanowski/>



**Fundusze  
Europejskie**  
Polska Cyfrowa



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz  
Rozwoju Regionalnego



# **Uczenie nienadzorowane**

## **algorytmy grupowania wykład 12**

Jerzy Stefanowski

Instytut Informatyki PP

2021

Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI-TECH)  
projekt finansowany z środków Programu Operacyjnego Polska Cyfrowa  
POPC.03.02.00-00-0001/20



**Fundusze  
Europejskie**  
Polska Cyfrowa



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz  
Rozwoju Regionalnego



# Plan wykładu

- Rozszerzenia klasycznych algorytmów grupowania
  - Algorytm k-srednich
    - K-medoid, PAM, ...
  - Algorytmy hierarchiczne
    - BIRCH
- Algorytmy gęstościowe
  - DBSCAN

----- druga część wykładu -----

- Podejścia wykorzystujące modele statystyczne
  - Algorytm mieszanin rozkładów (EM)
- Ocena jakości grupowania
- Podsumowanie

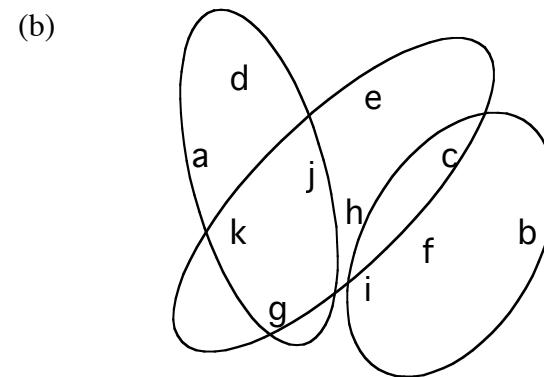
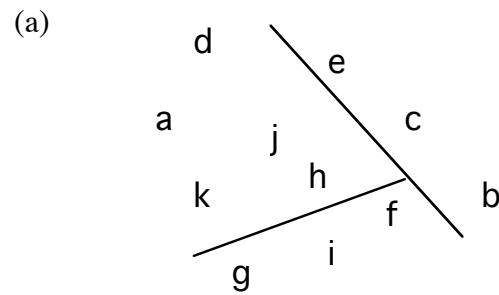
# Przypomnienie podziału metod

- Podziałowo-optymalizacyjne: Znajdź podział na zadaną liczbę skupień wg. zadanego kryterium.
- Metody hierarchiczne: Zbuduj drzewiastą strukturę skupień.
- Gęstościowo (Density-based): Poszukuj obszarów o większej gęstości występowania obserwacji
- Grid-based: wykorzystujące wielowymiarowy podział przestrzeni siatką ograniczeń
- Model-based: hipoteza co do własności modelu pewnego skupienia i procedura jego estymacji.

# Czym jest skupienie?

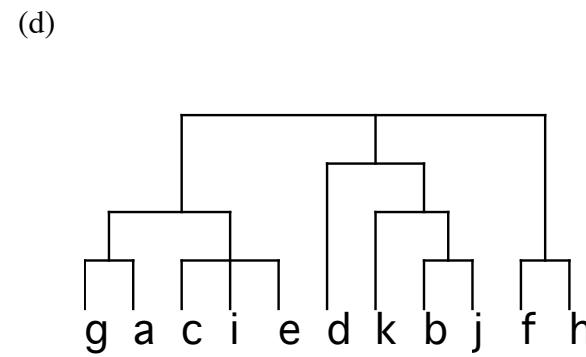
1. Zbiorem najbardziej podobnych obiektów
2. Podzbiór obiektów, dla których odległość jest mniejsza niż ich odległość od obiektów z innych skupień.
3. Podobszar wielowymiarowej przestrzeni zawierający odpowiednio dużą gęstość obiektów, oddzielony od innych podobszarów o dużej gęstości strefą rzadkiego występowania obiektów

# Różne sposoby reprezentowania skupisk



(c)

	1	2	3
a	0.4	0.1	0.5
b	0.1	0.8	0.1
c	0.3	0.3	0.4
d	0.1	0.1	0.8
e	0.4	0.2	0.4
f	0.1	0.4	0.5
g	0.7	0.2	0.1
h	0.5	0.4	0.1
...			



# Algorytmy podziałowo – optymalizacyjne

- Zadanie: Podzielenie zbioru obserwacji na  $K$  zbiorów elementów (skupień  $C$ ), które są jak najbardziej jednorodne
- Jednorodność – funkcja oceny
- Intuicja → zmienność wewnętrzskupieniowa  $wc(C)$  i zmienność międzyskupieniowa  $bc(C)$

Mögliwe są różne sposoby zdefiniowania

- np. wybierzmy środki skupień  $\mathbf{r}_k$  (centroidy)  $\mathbf{r}_k = \frac{1}{n_k} \sum_{\mathbf{x} \in C_k} \mathbf{x}$
- Co prowadzi do

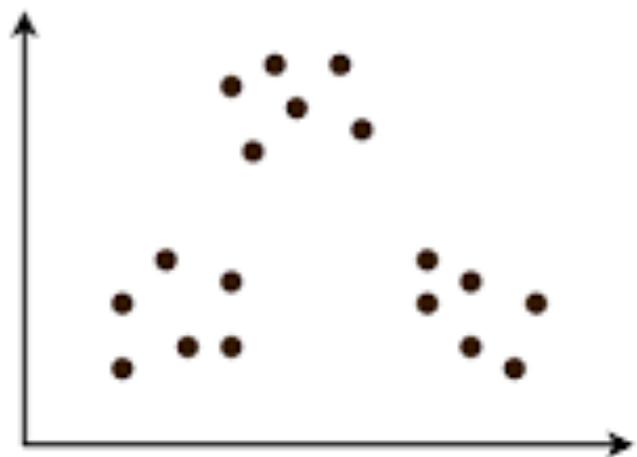
$$wc(C) = \sum_{k=1}^K \sum_{\mathbf{x} \in C_k} d(\mathbf{x}, \mathbf{r}_k)^2$$

$$bc(C) = \sum_{1 \leq j < k \leq K} d(\mathbf{r}_j, \mathbf{r}_k)^2$$

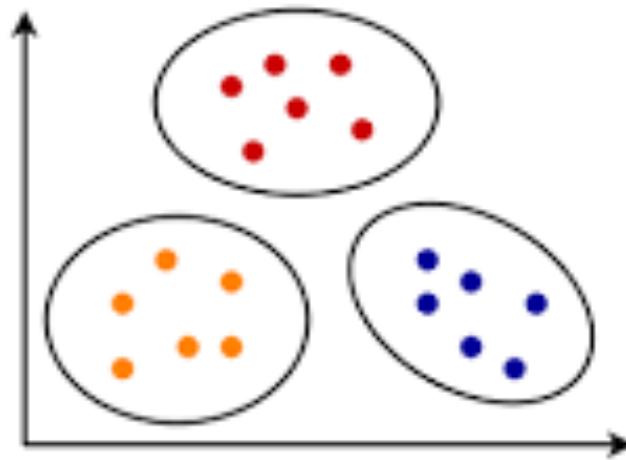
# Algorytm $k$ średnich ( $k$ – means)

- Cel:  $k$ - średnich → minimalizacja  $wc(C)$
- Przeszukiwanie przestrzeni możliwych przypisań → bardzo kosztowne (oszacowanie w ks. Koronackiego)
- Problem optymalizacji kombinatorycznej → systematyczne przeszukiwanie metodą iteracyjnego udoskonalania:
  - Rozpocznij od rozwiązania początkowego (losowego).
  - Ponownie przypisz punkty do skupień tak, aby otrzymać największą zmianę w funkcji oceny.
  - Przelicz zaktualizowane środki skupień, ...
  - Postępuj aż do momentu, w którym nie ma już żadnych zmian w funkcji oceny lub w składzie grup.
- Zachłanne przeszukiwanie → proste i prowadzi do co najmniej lokalnego minimum. Różne modyfikacje, np. rozpoczęnięcia od kilku rozwiązań startowych
- Złożoność algorytmy K - średnich →  $O(Knl)$

# Ilustracja k-średnich

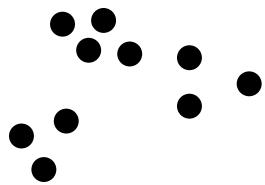


Before K-Means

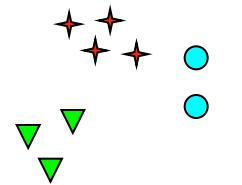
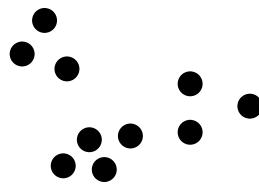


After K-Means

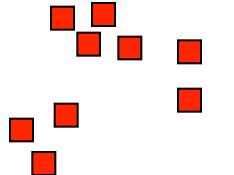
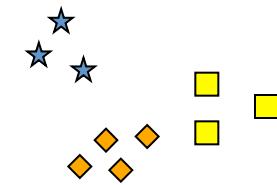
# Trudność określenia liczby skupisk, ...



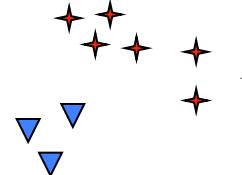
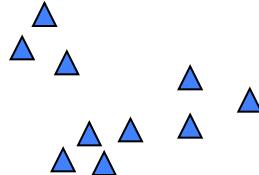
Ile jest naturalnych skupisk?



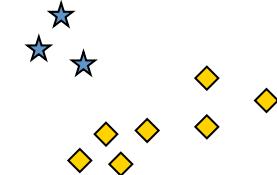
Sześć skupisk



Dwa skupiska



Cztery skupiska



# Ustalanie liczby skupień

Liczbę skupień wybiera się na podstawie przesłanek merytorycznych albo szacuje się je metodami hierarchicznymi. Można dokonać obliczeń dla wszystkich wartości  $k$  z ustalonego przedziału:

$$k_{\min} \leq k \leq k_{\max}$$

Możliwe są różne podejścia:

1. Wybór kryterium oceny skupisk
2. Uruchomienie algorytmu k-średnich dla różnych  $k$
3. Propozycja podziału dla najlepszego  $k$

Alternatywnie: - użyj innego algorytmu (np. hierarchicznego) do identyfikacji możliwej liczby skupisk na podstawie dendrogramu

Niezależnie:

K-średnich dość czułe na obecność obserwacji samotniczych (ang. outliers) – mogą tworzyć pojedyncze skupiska i zakłócać grupowanie pozostały przykładów -> warto wykryć i „odłożyć” ze zbioru do niezależnej

Preferencja dla tworzenia sferycznych kształtów skupisk

# Dobór $k$ w algorytmie k-średnich

- X-means popularne w implementacjach (WEKA, Python)
- Stosowane kryteria oceny podziału na skupiska:
  - Bayesian Information Criterion (BIC)
  - Akaike Information Criterion (AIC)
- Operacja “improve structure” – podział wybranego skupiska na dwa.
- D. Pelleg and A. Moore (2000) X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In Proceedings of the 17th International Conf. on Machine Learning, 727--734.

# X-means

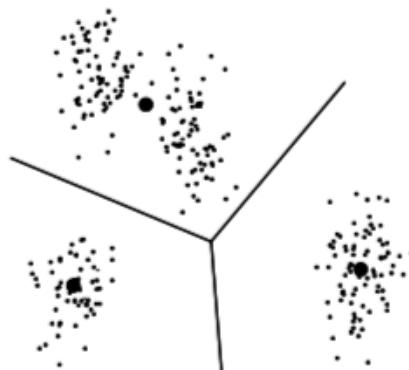


Figure 1. The result of running K-means with three centroids.

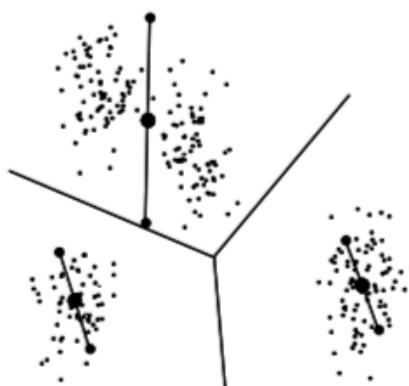
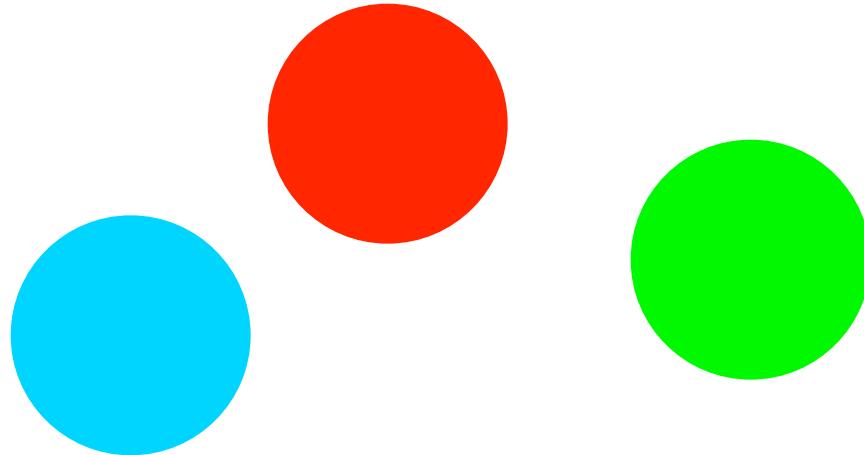


Figure 2. Each original centroid splits into two children.

# Pewne ukierunkowanie K-średnich

- Tworzy się „kuliste” kształty skupień



- Co z obserwacjami odstającymi i nieregularnymi kształtami skupień?

# K-means krótkie podsumowanie

## Zalety

- Proste i łatwe do zrozumienia
- Reprezentacja skupień jako centroidy

## Wady

- Jawne podanie liczby skupień
- Wszystkie przykłady muszą być przydzielone do skupień
- Problem z outliers (za duża wrażliwość)
- Ukierunkowanie na jednorodne „sferyczne” kształty skupień

# Dalsze rozszerzenia $k$ -średnich

- Rozmyte k-means (Fuzzy ISODATA)
- Wersja k-medoids
- Rozszerzenia dla przetwarzania dużych wolumenów danych, np. PAM
- Inspiracje dla modeli statystycznych (EM)
- Odniesienia do grupowania spektralnego

Obszerne omówienie w pracy:

Warto zapoznać się z książką S.Wierzchoń, M.Kłopotek:  
Algorytmy analizy skupień. WNT 2015

# Inne spojrzenie na skupiska

Crisp vs. soft clusters (ang. fuzzy clustering)

Obiekt  $x_i$  może należeć do wielu skupisk  $C_j$  w różnym stopniu przynależności z zakresu [0;1]

Najbardziej znany algorytm Fuzzy c-mean [Bezdek]

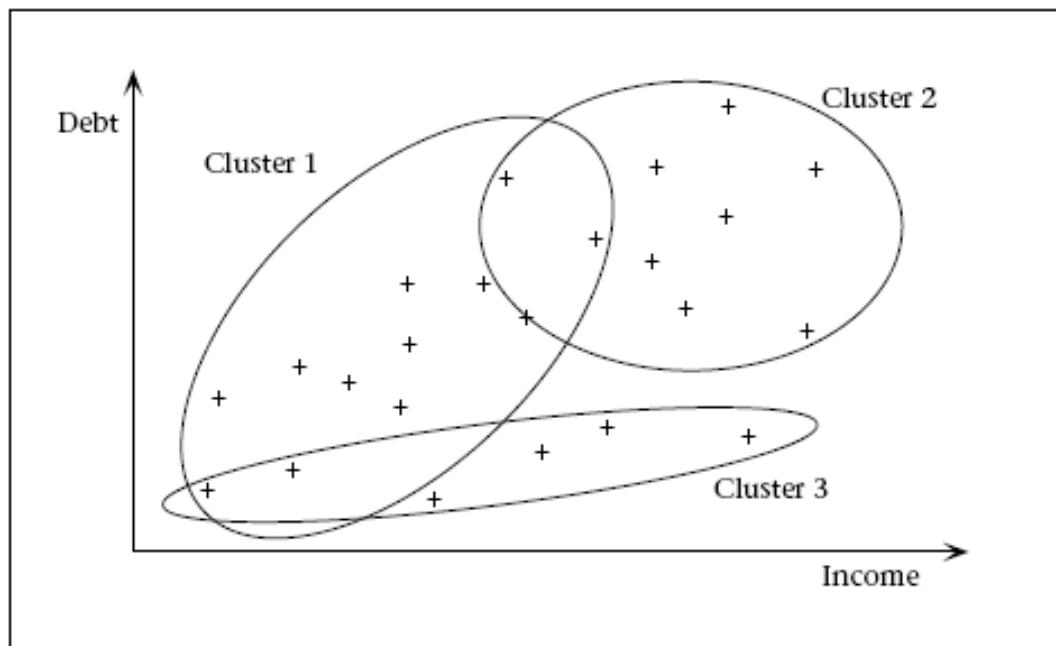


Figure 5. A Simple Clustering of the Loan Data Set into Three Clusters.

Note that original labels are replaced by a +.

# Algorytm $k$ -medoids

- Ograniczenie standardowego  $k$ -średnich, m.in.:
  - Czuły na obserwacje odstające i tzw. szum
  - Konieczność przeliczania macierzy odległości w każdej iteracji
- Algorytm  $k$ -medoids (PAM)
  - Zastąpienie reprezentanta skupiska - średniego obiektu (na ogół sztuczne położonego) poprzez rzeczywisty obiekt z danych położony najbliżej centrum
  - Zmiany w algorytmie – inny sposób oceny wymiany obiektów (**medoidów**) w kolejnych iteracjach -> najbardziej znana wersja PAM (Partitioning Around Medoid) - Kaufman i Rousseeuw 1987

# Przykład wpływu outliers

- Dla standardowego k-średnich i jednego atrybutu x rozważ skupisko:
  - średnia z obserwacji o wartościach 1,3,5,7,9 wynosi 5
  - Jeśli ostatnia obserwacja ulegnie zmianie na 1,3,5,7, 1009 to średnia będzie 205

Dla k-medoid (obiekt najbliższy centrum) – dla 1,3,5,7, 1009 będzie to 7, a w kolejnych iteracjach przesunie się na 5

# PAM – algorytm k-medoids

1. Wybierz (losowo)  $k$  rzeczywistych obiektów jako załączki skupisk
2. Przydziel każdy obiekt do tego skupiska, gdzie jest najbliższy medoid
3. W kolejnym kroku aktualizuje się położenia centroidów – medoidów i powtarzany jest przydział obiektów do najbliższego skupiska
  - Wymiana obiektów z medoidami na podstawie oszacowanie specjalnej funkcji kosztu wymiany (SWAP) – uwaga analizuje się wszystkie pary (obiekt vs. medoid)
  - Zaproponowano specjalne funkcje niepodobieństwa dla atrybutów jakościowych

# K-medoids przykład

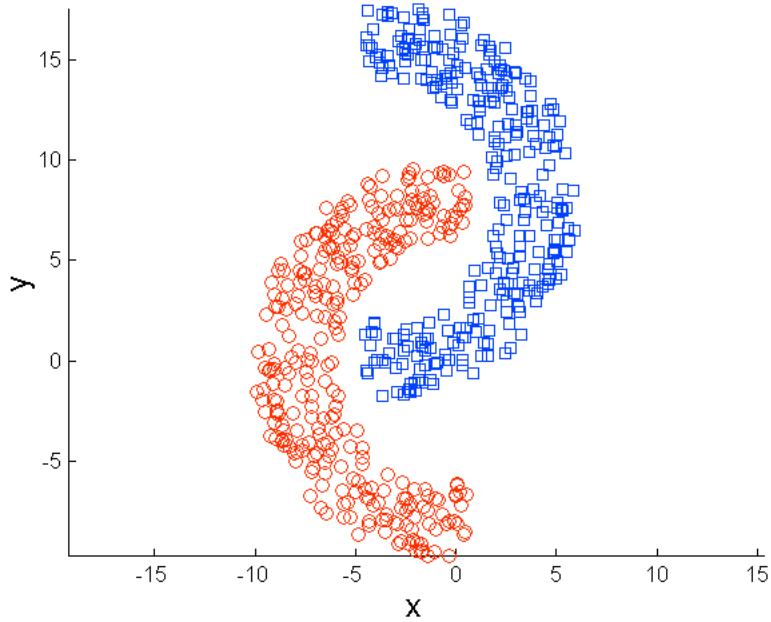
- Można przeanalizować prosty przykład dwuwymiarowy opisany na blogu <https://www.geeksforgeeks.org/ml-k-medoids-clustering-with-example/>
- Lub podręczniki J.Han i inni Data Mining.

# PAM – ograniczenia dla zbyt masywnych danych

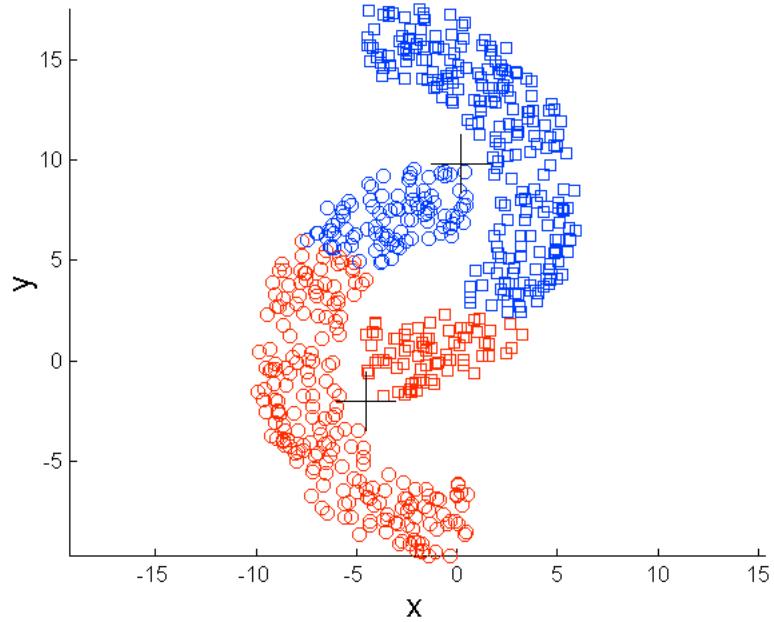
- Wada PAM – słaba skalowalność dla relatywnie dużych wolumenów danych
  - Ocena złożoność PAM  $O(k(n-k)^2)$  , gdzie n – liczba obiektów, k – liczba skupisk
- **CLARA** (ang. Clustering Large Applications)– próba poprawy skalowalności
  - Losowana reprezentatywna próba danych z całych danych
  - PAM poszukuje zbioru dobrych medoidów
  - Jeśli dobrze dobrana próba, to będą także odzwierciedlać rozkład całych danych
  - Możliwe powtórzenie losowania i wybór najlepszego zbioru medoidów
    - Przydział wszystkich obiektów do skupisk wg. wybranych medoidów
- Inne próby modyfikacji dla przyspieszenia obliczeń kosztów wymiany  $O(k)$  – Schubert i in. Fast and eager k-medoids clustering 2020.

# Ograniczenia k-mean i motywacje dla innych algorytmów

# Ograniczenia K-średnich: Niesferyczne kształty



Oryginalny zestaw  
danych



K-means (2 skupienia)

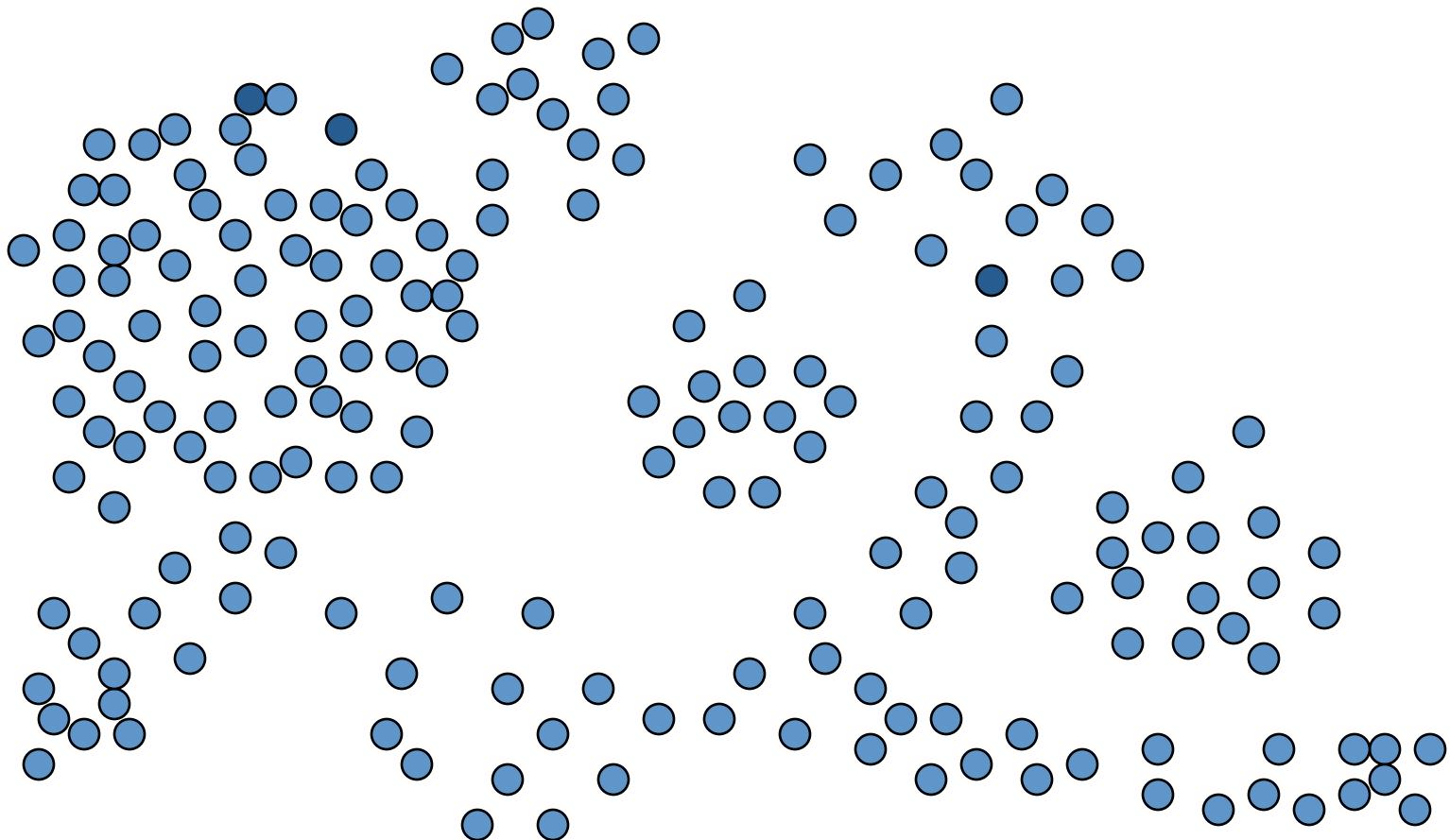
# Inne algorytmy grupowania

Wykorzystują inne paradygmaty, np.

- **Gęstość** obiektów w przestrzeni cech
- Strukturę sieci komórek – tzw. **grid**

Skupisko – obszar charakteryzujący się dużą gęstością obiektów. Skupiska obiektów odseparowane od siebie obszarami o małej gęstości występowania obiektów

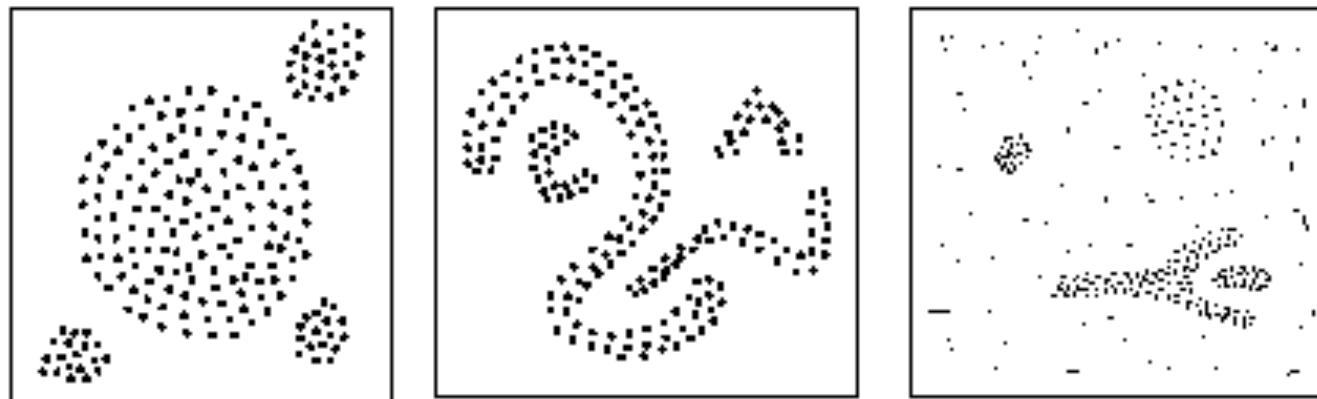
# Modelowanie dowolnych kształtów



# Metody gęstościowe

- Wykorzystują pojęcie gęstości (ang. density) – **lokalne sąsiedztwo** punktu/skupienia, a także „gęsto” połączonych punktów
- Podstawowy pomysł – przyrostowe tworzenie skupiska poprzez dołączanie obiektów należących do najbliższego sąsiedztwa tego skupiska pod warunkiem, że spełniają pewne minimalne parametry
- Właściwości metod gęstościowych:
  - Wykrywanie skupień o dowolnych kształtach (niesferycznych)
  - Odporność na „szum informacyjny” i obs. „outliers”
  - Samodzielne określanie liczby potrzebnych skupisk
  - Potrzebna parametryzacja oceny gęstości i warunków zatrzymania
- Znane algorytmy:
  - DBSCAN: Ester, et al. (KDD'96)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98)
  - Laczne rozszerzanie

# Metody gęstościowe – ang. Density-Based Clustering



- Skupiska grupują gęste punkty, obszary i są odległe od innych gęstych obszarów lub „rzadkich” punktów
- Jak oceniać gęstość i jakość skupiska?
- DBSCAN - grupowanie wykorzystujące ocenę gęstości rozkładu (lokalne kryterium – sąsiedztwo punktu), parametry oczekiwanej gęstości (minimalna liczba punktów) oraz wielkości sąsiedztwa

# DBSCAN: Algorytm gęstościowy

- DBSCAN: Density Based Spatial Clustering of Applications with Noise (Ester et al.'96)
  - Wprowadza pojęcie „*density-based cluster*”: Skupienie będące największym zbiorem punktów gęsto połączonych „*density-connected points*” (ze względu na parametry sąsiedztwa punktów)
    - Skupienie to zbiór obiektów wzajemnie osiągalnych lub połączonych z pewną zadaną gęstością
  - Wykorzystuje  $\epsilon$ -sąsiedztwo punktu i możliwość podziału obiektów na typy (wg. liczby sąsiadów) oraz zasady przemieszczania się pomiędzy nimi (idea osiągalności)
  - Możliwość wykrywania skupień o dowolnym kształcie w obecności szumu informacyjnego (ang noise) i obserwacji samotniczych

# DBSCAN: Podstawowe pojęcia

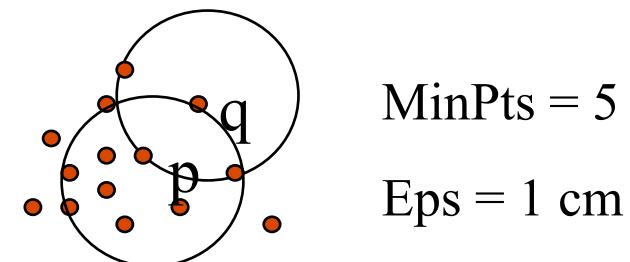
- Parametry:
  - **Eps** ( $\epsilon$ ): Maksymalny promień sąsiedztwa
  - **MinPts**: minimalna liczba punktów (obiektów) w Eps-sąsiedztwie badanego punktu
- D – dany zbiór obiektów do pogrupowania; wybrana miara odległości  $dist(p,q)$
- $\epsilon$  sąsiedztwo obiektu p to zbiór innych punktów q spełniających
$$N_{Eps}(p) : \{ \text{punkt } q \text{ należy do } D \mid dist(p,q) \leq Eps \}$$

W zależności od liczby sąsiadów w otoczeniu obiektu p:

Obiekt rdzenia

Obiekt graniczny

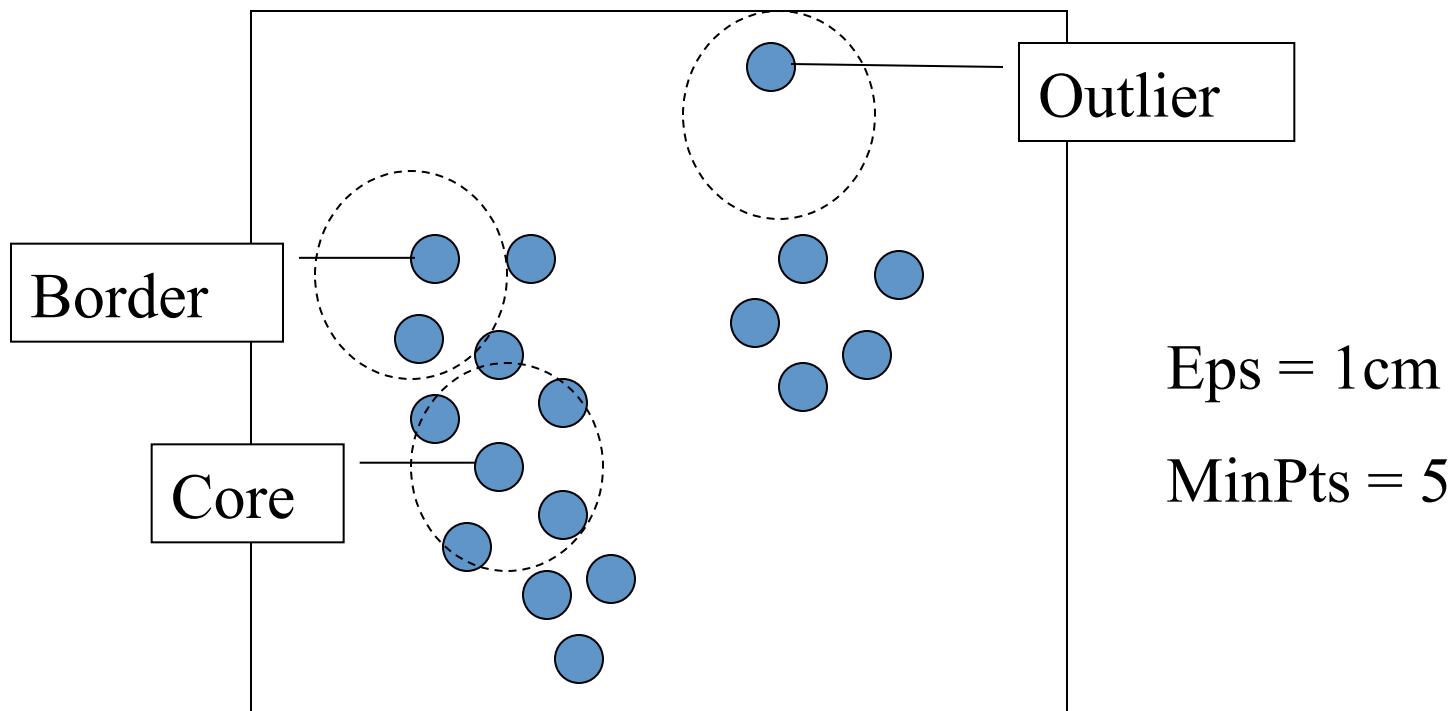
Obiekt oddalony / Szum (ang. noise point)



# Typy obiektów

- **Obiekt centralny – rdzeń** (ang. core point) = obiekt, który ma co najmniej MinPts sąsiednich obiektów w swoim  $\epsilon$  sąsiedztwie (są to zalaźki do budowy gęstych skupisk)
- **Obiekt brzegowy** (ang. border point) = obiekt mający mniej niż MinPts sąsiednich obiektów w swoim  $\epsilon$  sąsiedztwie, lecz należący do sąsiedztwa co najmniej jednego punktu rdzenia
- **Obiekt oddalony / szum** (ang. noise point) = obiekt z liczbą sąsiadów niż MinPts nie należący do sąsiedztwa innych punktów rdzeniowych (odległy o więcej niż  $\epsilon$  od innych potencjalnych skupisk)

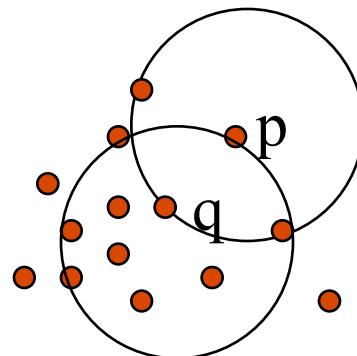
# DBSCAN – ilustracja typów obiektów



# DBSCAN: Podstawowe pojęcia (2)

- Osiągalność obiektów (niedbędna do tworzenia skupisk)
- $N_{eps}(q)$ : {punkt  $p$  należy do  $D$  |  $dist(p,q) \leq Eps$ }
- **Bezpośrednia osiągalność gęstościowa**
- Mówimy, że obiekt  $p$  jest bezpośrednio osiągalny z punktu  $q$  (ze względu na parametry  $Eps$ ,  $MinPts$  ), jeśli
  - 1)  $p$  należy do  $\epsilon$  sąsiedztwa  $N_{Eps}(q)$
  - 2) Obiekt  $q$  jest centralny:

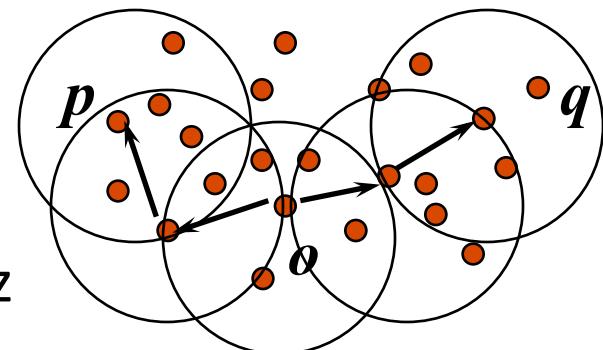
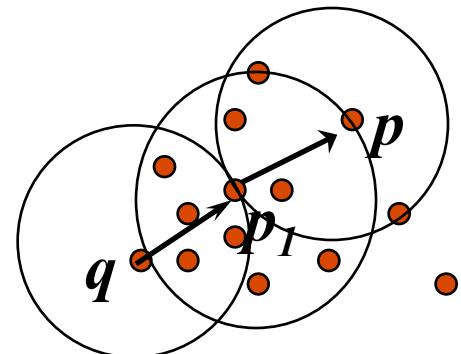
$$|N_{Eps}(q)| \geq MinPts$$



$$\begin{aligned}MinPts &= 5 \\Eps &= 1 \text{ cm}\end{aligned}$$

# DBSCAN: Podstawowe pojęcia (3)

- Gęstościowa osiągalność (Density-reachable):
  - Obiekt  $p$  jest gęstościowo osiągalny z punktu  $q$  jeśli istnieje łańcuch punktów pośrednich  $p_1, \dots, p_n, p_1 = q, p_n = p$ , takich że  $p_{i+1}$  jest bezpośrednio osiągalny z  $p_i$ ,
- Połączniowa gęstości (Density-connected)
  - Obiekt  $p$  jest gęstościowo połączony z obiektem  $q$  (wrt.  $Eps, MinPts$ ) jeśli istnieje punkty o taki, że obiekty  $p$  oraz  $q$  są z niego gęstościowo osiągalne
- Połączenia / osiągalność pozwalają na określenie skupiska zaczynając z jednego z obiektów centralnych



# DBSCAN: Zarys algorytmu

- Wybierz punkt startowy  $p$
- Odnajdź wszystkie punkty do gęstościowego osiągnięcia z  $p$  (density-reachable from  $p$  wrt  $Eps$  and  $MinPts$ ).
- Jeśli  $p$  jest rdzeniem (*core point*), utwórz skupienie.
- Jeśli  $p$  jest punktem granicznym (border point) i żadne punkty nie są z niego gęstościowo osiągalne, DBSCAN wybiera następny punkt z bazy danych
- Proces jest kontuowany dopóki żaden nowy punkt nie może być dodany do dowolnego skupienia
- Punkty które nie są rdzeniem lub graniczne i nie mogą być zaliczone do skupisk – stają się punktami oddalonymi (noise)
- Złożoność:  $O(n \log n)$  w przypadku użycia specjalnego „spatial index”, w przeciwnym razie  $O(n^2)$ .

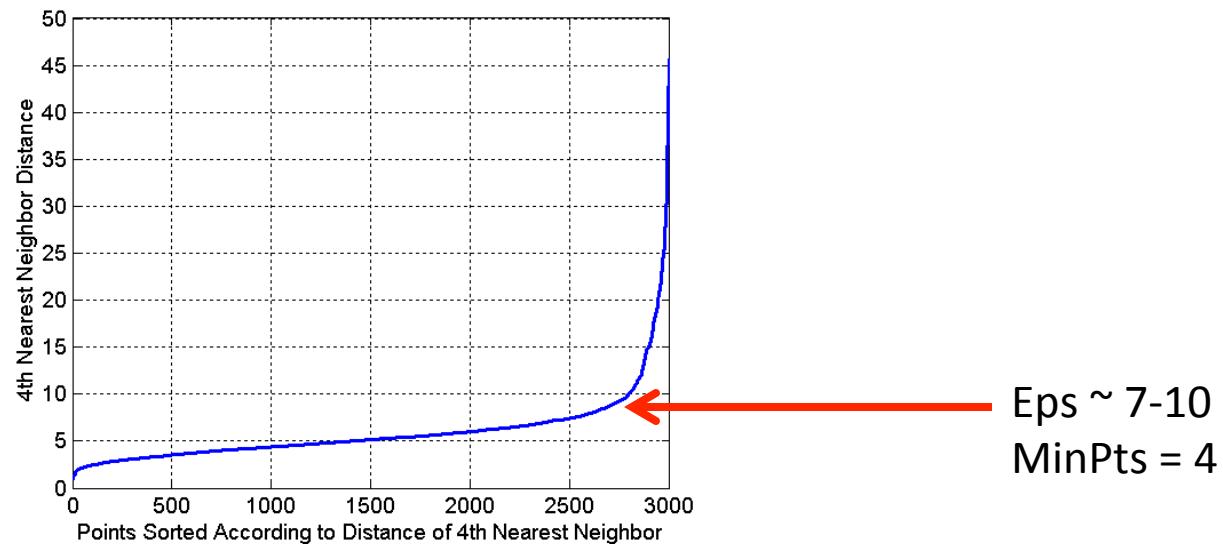
# DBSCAN trudności parametryzacji

Algorytm jest dość czuły na dobór parametrów Eps ( $\varepsilon$ ) i MinPts  
Autorzy zaproponowali heurystykę:

- Niech  $d$  będzie odlegością obiektu  $p$  do jego  $k$ -tego najbliższego sąsiada, sąsiedztwem obiektu  $p$  będzie dokładnie  $k + 1$  obiektów
- Należy dobrać  $k$  dla danych  $D$
- Określić funkcję  $k$ -dist, która odwzorowuje każdy obiekt  $p$  w danych  $D$  na odległość do jego  $k$ -tego najbliższego sąsiada
- Uporządkuj wartości  $k$ -dist dla obiektów (wykres gęstości)
  - Dla obiektu  $p$  - ustawiając wartość parametru  $Eps$  na  $k$ -dist( $p$ ), a wartość parametru  $MinPts$  na  $k$ , wszystkie obiekty z mniejszą lub równą wartością  $k$ -dist staną się obiektami wewnętrznymi sąsiedztwa/ skupiska / inne są kandydatami na punkty oddalone
- Znajdź punkt progowy przegięcia wykresu

# DBSCAN: Wykres k-sasiedztwa

- Punkty w gęstym skupisku, większość ich  $k^{\text{th}}$  najbliższych sąsiadów ma podobną wartość odległości
- Obiekty oddalone oraz szum (ang. noise points) i sch  $k^{\text{th}}$  najbliższy sąsiad jest dużo bardziej odległy (odległość wyraźnie rośnie)
- Posortuj obiekty wg. ich odległości od  $k^{\text{th}}$  najbliższego sąsiada i zrób wykres
- Znajdź odległość  $d$  odpowiadającej przegięciu kształtu w kresu (ang. “**knee**” in the curve)
  - $\text{Eps} = d$ ,  $\text{MinPts} = k$



# Przykłady porównania algorytmów

DBSCAN

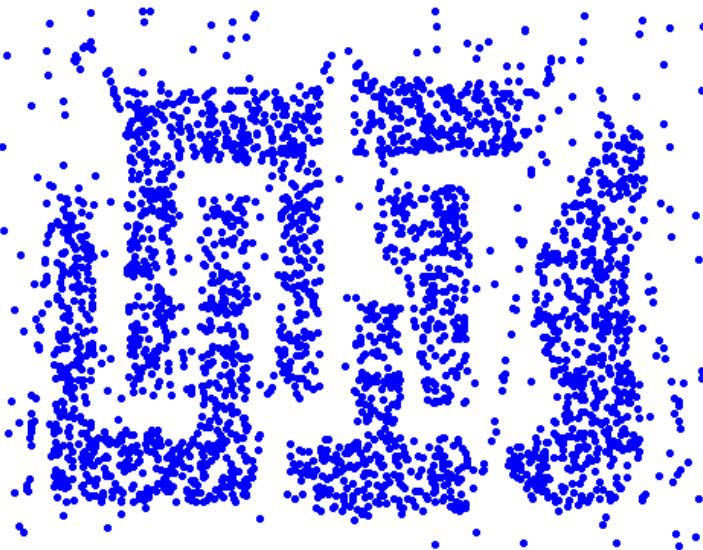


k-means

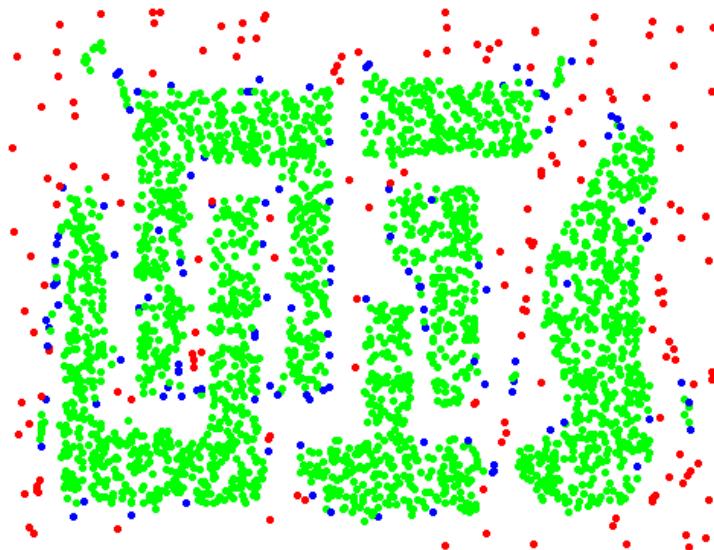


Więcej w blogu pt. An introduction to the DBSCAN algorithm and its Implementation in Python [Nagesh Singh Chauhan] KDnuggets

# DBSCAN: przykład użycia



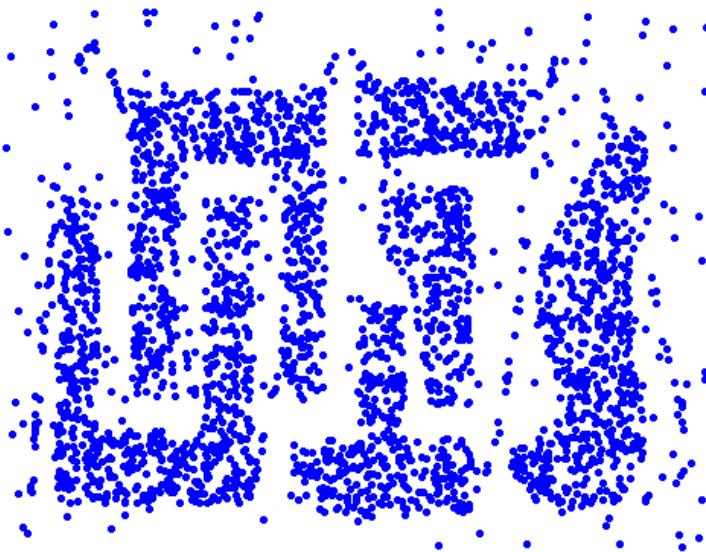
Oryginalne dane



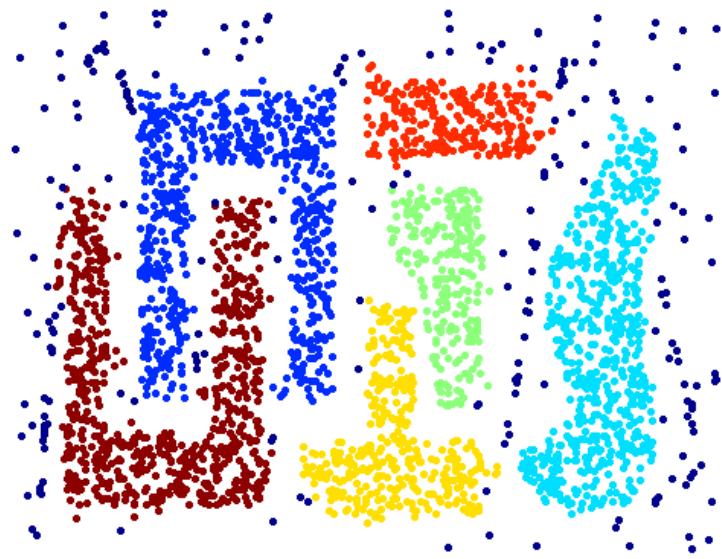
Typy obiektów: **core**,  
**border** and **noise**

Eps = 10, MinPts = 4

# Skupiska DBSCAN



Original Points

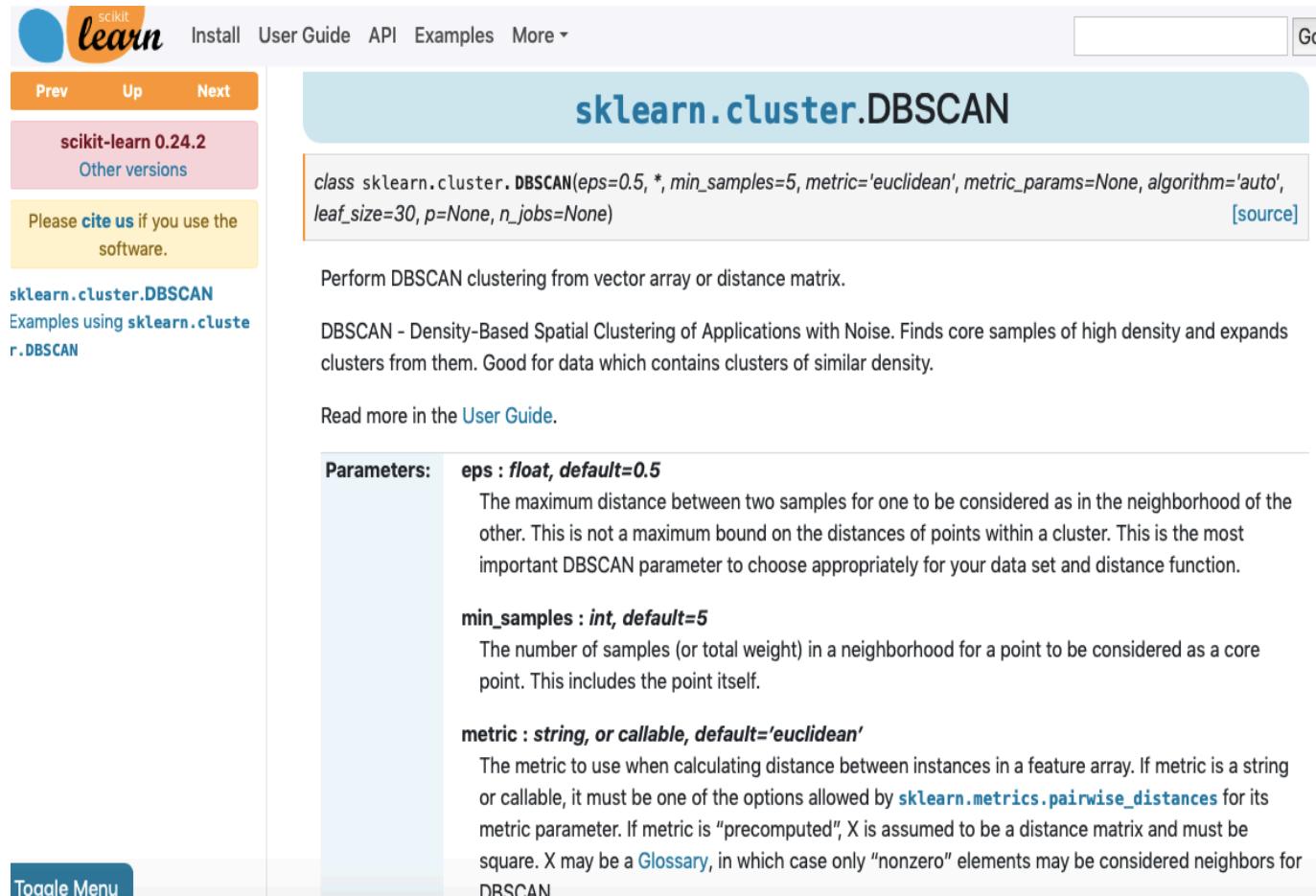


Clusters

- Radzi sobie z szumem informacyjnym
- Tworzy skupiska o różnych kształtach

# Liczne implementacje DBSCAN

## Scikit learn - cluster



The screenshot shows the scikit-learn documentation page for the `DBSCAN` class. At the top, there's a navigation bar with links for "Install", "User Guide", "API", "Examples", and "More". Below the navigation bar, there's a search bar and a "Go" button. On the left, there's a sidebar with links for "Prev", "Up", "Next", "scikit-learn 0.24.2", "Other versions", and a "Please cite us" section. The main content area has a title "sklearn.cluster.DBSCAN" and a code snippet:

```
class sklearn.cluster.DBSCAN(eps=0.5, *, min_samples=5, metric='euclidean', metric_params=None, algorithm='auto', leaf_size=30, p=None, n_jobs=None)
```

With a "[source]" link. Below the code snippet, there's a brief description: "Perform DBSCAN clustering from vector array or distance matrix." and a detailed explanation: "DBSCAN - Density-Based Spatial Clustering of Applications with Noise. Finds core samples of high density and expands clusters from them. Good for data which contains clusters of similar density." There's also a link to "Read more in the User Guide." At the bottom of the content area, there's a table for parameters:

Także inne języki i środowiska,  
WEKA +  
DBSCAN. Lightweight Java

# Przykład DBSCAN -sklearn

---

```
1 import numpy as np
2 from sklearn.cluster import DBSCAN
3 from sklearn import metrics
4 from sklearn.datasets import make_blobs
5 from sklearn.preprocessing import StandardScaler
6
7 # Generate sample data
8 centers = [[1, 1], [-1, -1], [1, -1]]
9 X, labels_true = make_blobs(n_samples=750, centers=centers, cluster_std=0.4,
10                             random_state=0)
11
12 X = StandardScaler().fit_transform(X)
13
14 # Compute DBSCAN
15 db = DBSCAN(eps=0.3, min_samples=10).fit(X)
16 core_samples_mask = np.zeros_like(db.labels_, dtype=bool)
17 core_samples_mask[db.core_sample_indices_] = True
18 labels = db.labels_
19
20 # Number of clusters in labels, ignoring noise if present.
21 n_clusters_ = len(set(labels)) - (1 if -1 in labels else 0)
```

# Grupowanie z wykorzystaniem modeli prawdopodobieństwa

- Podejścia oparte na założeniu, że danych są generowanie w wyniku realizacji pewnego procesu statystycznego
- Zakłada się pewien model rozkładu prawdopodobieństwa występowanie obserwacji
- Każdemu potencjalnemu skupisku odpowiada model, w postępowaniu (algorytmie) weryfikuje się stopień dobrego dopasowania oryginalnych danych do przyjętego modelu
- Celem grupowania jest znalezienie zbioru (mieszaniny) modeli opisujących skupiska oraz estymacja parametrów tych modeli
- Obiekty przydziela się do skupisk zgodnie ze sparametryzowanymi modelami i zasadą klasyfikacji Bayesowskiej
- Patrz kolejny wykład

# **Pytanie i komentarze?**

Dalszy kontakt:

[jerzy.stefanowski@cs.put.poznan.pl](mailto:jerzy.stefanowski@cs.put.poznan.pl)

<http://www.cs.put.poznan.pl/jstefanowski/>



**Fundusze  
Europejskie**  
Polska Cyfrowa



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz  
Rozwoju Regionalnego



# **Uczenie częściowo nadzorowane**

## wykład 13

Jerzy Stefanowski

Instytut Informatyki PP

2021

Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI-TECH)  
projekt finansowany z środków Programu Operacyjnego Polska Cyfrowa  
POPC.03.02.00-00-0001/20



**Fundusze  
Europejskie**  
Polska Cyfrowa



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz  
Rozwoju Regionalnego



# Plan wykładu

- Motywacje do uczenia częściowo nadzorowanego (ang. semi – supervised learning; SSL)
- Wybrane metody SSL
  - Cluster-and-label
  - Self training
  - Co-training oraz multi-view learning
  - Podejścia grafowe z propagacją etykiet
  - Rozszerzania SVM
- Podsumowanie

# Różne podejścia do uczenia się z przykładów

- Uczenie w pełni nadzorowane (ang. supervised) – etykietowane przykłady

$$L = \left\{ (x_i, y_i) \right\}_{i=1}^n$$

- Uczenie nienadzorowane (unsupervised) – nieetykietowane  $U = \left\{ (x_i) \right\}_{i=1}^m$

- Uczenie częściowo etykietowane (ang. semi-supervised)  
Etykietowane dane  $L$  oraz (część) nieetykietowanych  $U$ , na ogólnie  $m >> n$

**Nie możemy pozyskać etykiet dla choć wybranych przykładów z  $U$**  – różnica wobec aktywnego uczenia się!

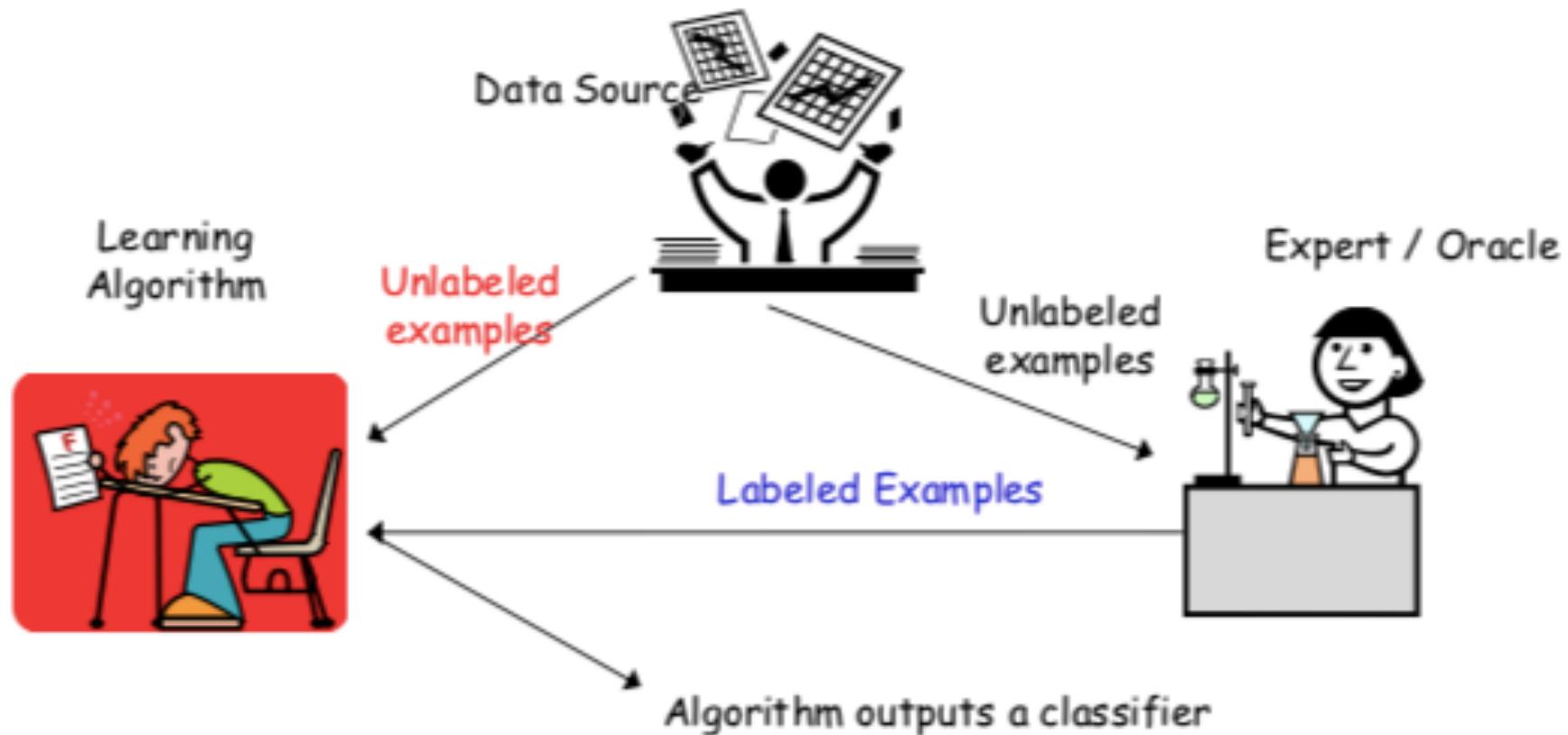
Cel: skonstruowanie lepszego klasyfikatora z  $L$  i  $U$  niż z samych przykładów etykietowanych  $L$

# Różne podejścia do uczenia się z przykładów

## Obserwacje motywacyjne:

- Uczenie nadzorowane wymaga wielu etykietowanych przykładów
  - Etykietowanie = kosztowne, czasochłonne i męczące, nierealne dla wielkich kolekcji przykładów
  - Ograniczony dostęp do osób adnotujących przykłady
  - Wymóg wiarygodności (ekspert lepszy niż ochotnicy; niektóre dane trudne do etykietyzacji, obciążenie – bias osób)
- Nieetykietowane przykłady
  - Potencjalnie łatwo dostępne w dużej ilości
  - Często możliwe do automatycznej rejestracji, np.
    - Web, pomiary sensorów, pozyskiwane przykładów z repozytoriów danych
  - „Tańsze” w przygotowaniu

# Ogólne motywacje – skorzystajmy z nieetykietowanych przykładów do ulepszenia klasyfikatora



za wykładem N. Balcan nt semi-supervised learning

# Przykład anotacji zapisu mowy

- Tzw. Switchboard data – analiza mowy dla zapisu rozmów między ludźmi
- System rozpoznawania mowy -> narzędzia do tworzenia surowej transkrypcji
- Oszacowanie 400 godzin pracy specjalisty anotującego dla 1 godziny zapisu i transkrypcji

# Przykłady L i U

## Strony WWW, obrazy, dokumenty

### Etykietowanie:

- Konieczność czytania lub oglądu
- Wymaga namysłu osoby
- Czasami wspomagane dodatkowymi źródłami

### Nieetykietowane:

Potencjalne dostępne w dużej ilości, automatycznie pozyskiwane z minimalnym kosztem

## Dane bio-medyczne

### Przykładowo predykcja struktur genetycznych

Adnotacja wymaga wysiłku eksperta, dodatkowej wiedzy, niekiedy potwierdzenia (po długim czasie)

Nowe urządzenia diagnostyczne (np. sekwencjonowanie DNA) może generować dużo danych eksperymentalnych

# Związek SSL z ludzkim poznawaniem świata

Ludzie często poznają obiekty ze świata z ograniczonym wskazywaniem kategorii, a wiele obiektów później sami klasyfikują :

Przykład uczenia dzieci pojęć

X=zwierzę (cechy) y = gatunek, np. pies

Nauczyciel-rodzic wskazuje i wyjaśnia “to jest pies”

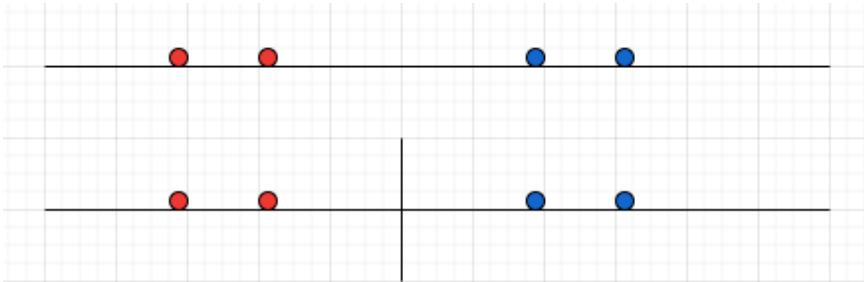
Dzieci obserwują inne zwierzęta (nie objaszone) i samodzielnie klasyfikują i nazywają



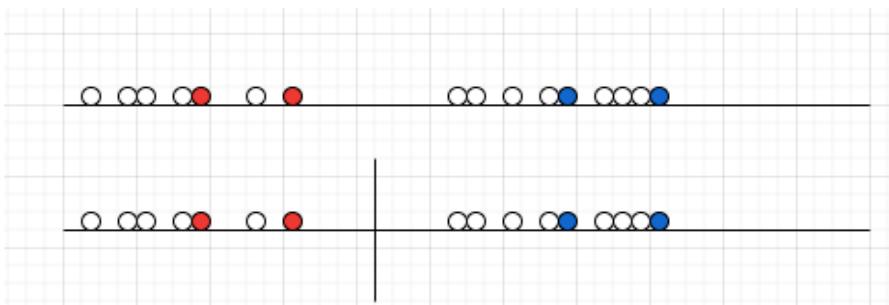
Za Barnabas Poczos

# Kiedy SSL może pomóc?

Etykiety : Czerwone klasa 1, niebieskie klasa -1



Dodajmy rozkład przykładowów nieetykietowanych (szare)

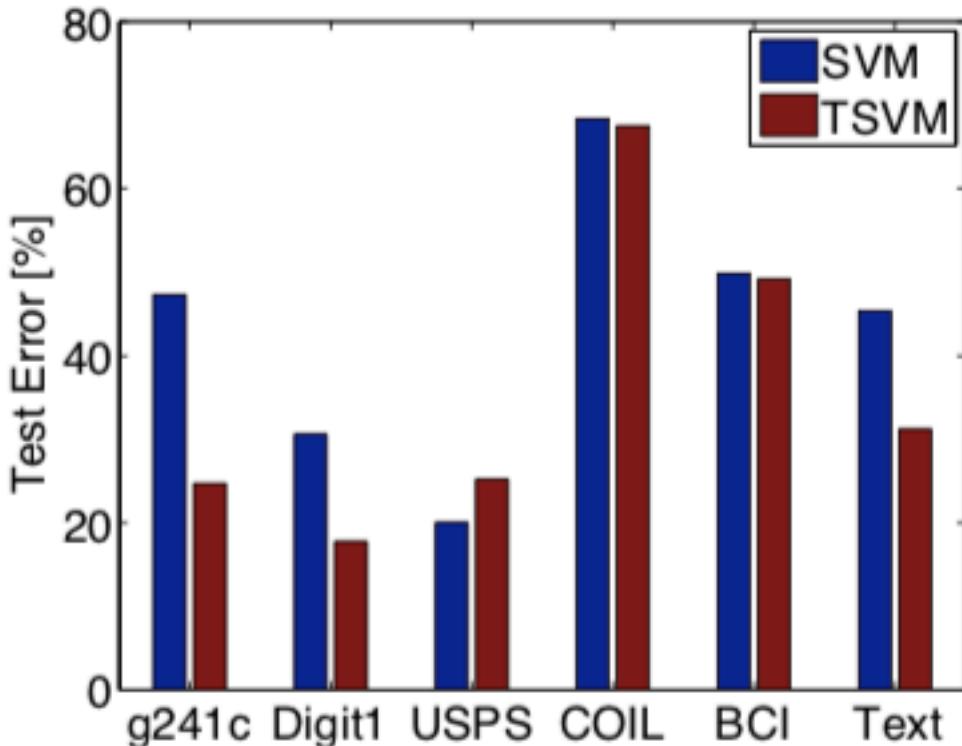


Zmiana granicy decyzyjnej na dokładniejszą!

Założenie: przykłady z różnych klas tworzą spójne rozkłady

Za wykładem Zhu

# Inny przykład użycia Transductive SVM dla różnych zbiorów danych



10 labeled points  
~1400 unlabeled  
points

SVM: supervised  
TSVM:  
semi-supervised

# Wzrost zainteresowania SSL

W “świecie naukowym – konferencyjnym” widoczny od początku wieku, nowe metody i zastosowania, często w przykładach / obszarach:

- Rozpoznawania obrazów i wizji komputerowej
- Analizy zapisów mowy i przetwarzania dokumentów tekstowych
- Eksploracji zasobów internetu (strony WWW, sieci użytkowników, itd.)

Także widoczne w „przemysłowym” ML i DM / np. patrz wystąpienia na GHOSTDay

# Podstawowe metody SSL

Zgodnie z podziałem Zhu:

- Cluster and label (wykorzystane algorytmów grupowania)
  - Generatywne podejścia probabilistyczne (EM, ...)
- Self-learning / samo uczenie się
- Co-training / uczenie się wzajemne
- Multi-view ensembles
- Transductive SVM
- Graph based models

Więcej w pracy przeglądowej po ang. : Zhu, Xiaojin, Semi-Supervised Learning University of Wisconsin-Madison [online pdf].

# Pewne założenia i hipotezy o danych

W celu efektywnego wykorzystania nieetykietowanych przykładów U – powinno się poczynić pewne założenia i oczekiwania

A. Blum pisał o potrzebie przekonania, że:

Unlabeled data useful if we have beliefs not only about the form of the target, but also about its relationship with the underlying distribution.

# Wstępne hipotezy badawcze dla SSL

W celu wykorzystania nietykietowanych przykładów w procesie uczenia konieczne są założenia co do rozkładów danych. Najczęstsze dwa to:

- **Założenie ciągłości** (Continuity): Najbliższe punkty najprawdopodobniej mają taką samą etykietę / powiązane z preferencją do granicy decyzyjnej w rzadszych obszarach
- **Zgrupowanie w skupiskach** (cluster assumption): dane mogą tworzyć skupiska i punkty wewnętrz skupisk najprawdopodobniej mają te same etykiety (lecz przykłady z tej samej klasy mogą tworzyć wiele skupisk).
- **Manifolds** – punkty często należą do tzw. kształtu geometrii rozmaitości punktów w nowej przestrzeni a nie w oryginalnej reprezentacji.

# Ilustracja założenia w przypadku skupisk

Punkty położone bardzo blisko siebie mają podobne etykiety

Założenie nt. skupień w danych

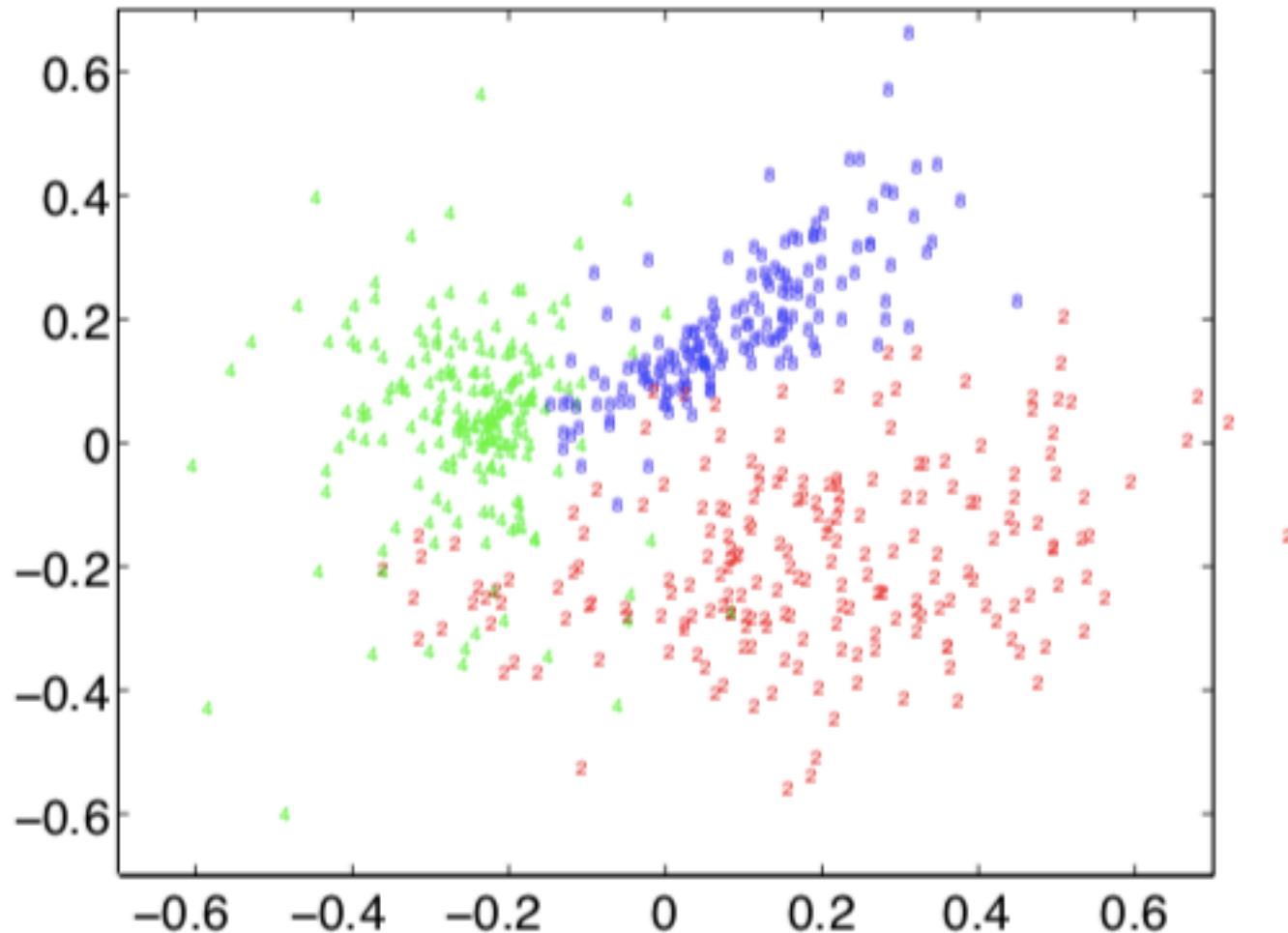
- Dane naturalnie tworzą skupiska
- Punkty w jednym skupisku powinny mieć tą samą etykietę

Rozważmy zbiory nieetykowanych punktów w dwóch skupiskach wokół niewielu etykietowanych:



# Przykład rozpoznawania liter

Example: 2D view on **handwritten digits 2, 4, 8**



[non-linear 2D-embedding with "Stochastic Neighbor Embedding"]

# Grupuj i etykietuj /Cluster and label

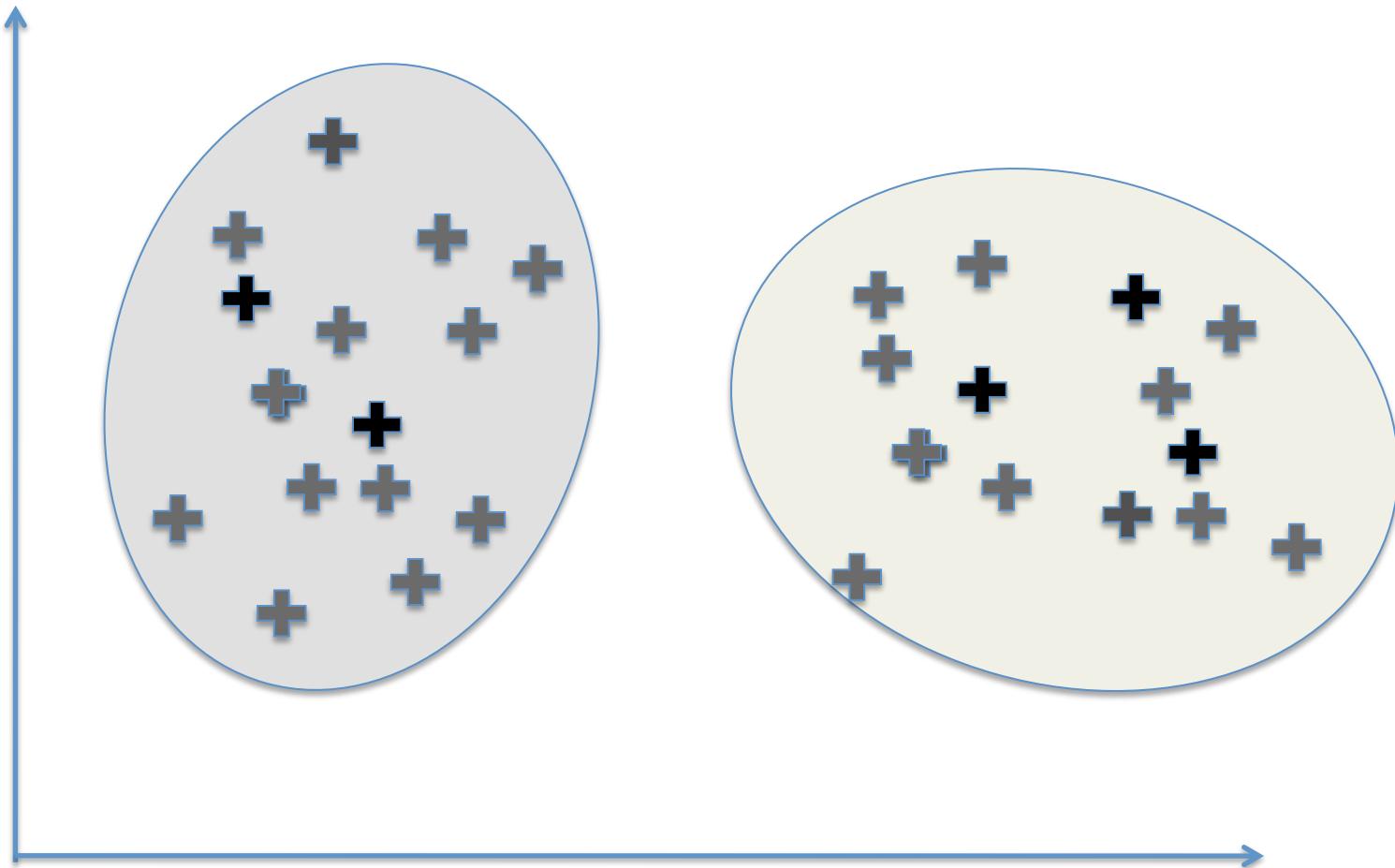
**Input:** etykietowane przykłady  $L=\{x,y\} /n/$  oraz nieetykietowane  $U=\{x\} /m/$  oraz algorytmy A grupowania i uczenia klasyfikatora LK

1. Znajdź skupienia w zbiorze nienadzorowanych x ( $m+n$ )
2. Dla każdego ze skupień, znajdź zbiór wszystkich etykietowanych przykładów S
3. Naucz algorytmem LK klasyfikator f na podstawie zbioru S
4. Zastosuj f do predykcji etykiet na pozostałych nieetykietowanych przykładów w skupieniu

Na koniec nauczyć klasyfikator na całym zbiorze przykładów (obejmującym zaetykietowane przykłady ze skupisk) -> **output**

Obserwacja: zakłada się, że granice skupisk są zgodne z granicami decyzyjnymi

SSL – grupowanie = założenia nt.  
“kształtu” rozkładów



# Pytanie o dobór algorytmu

- Liczba skupisk nie powinna być mniejsza niż liczba klas (ta jest znana z uwagi na zbiór L)
- Regularne sferyczne rozkłady = niektóre podejścia wykorzystując k-means / typowe cluster and label; w niektórych propozycjach także algorytmy gęstościowe
- Bardziej złożone kształty rozkładów = adaptacja modeli generatywnych, w szczególności **algorytmu EM** (przykłady etykietowane wykorzystane do początkowej identyfikacji składników mieszaniny rozkładów)

# Basic Algorithm

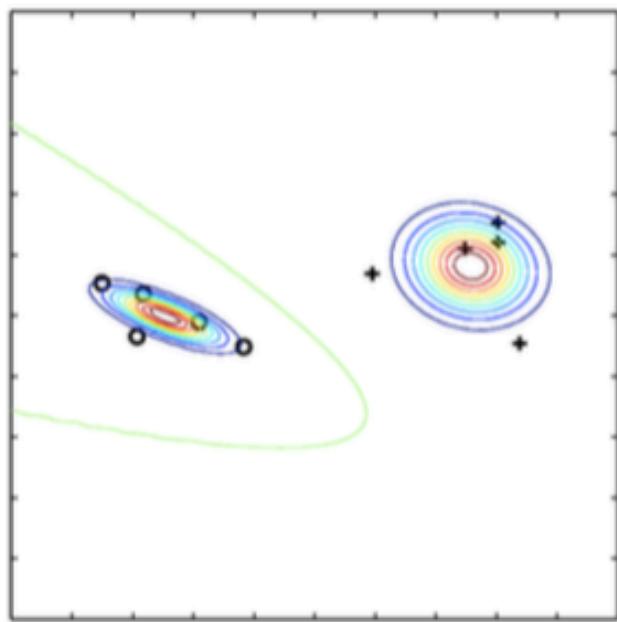
**Algorithm** EM( $L, U$ )

- 1 Learn an initial naïve Bayesian classifier  $f$  from only the labeled set  $L$  (using Equations (27) and (28) in Chap. 3);
- 2 **repeat**
  - // E-Step
  - 3   **for** each example  $d_i$  in  $U$  **do**
  - 4     Using the current classifier  $f$  to compute  $\Pr(c_j|d_i)$  (using Equation (29) in Chap. 3).
  - 5   **end**
  - 6     // M-Step
  - 7   learn a new naïve Bayesian classifier  $f$  from  $L \cup U$  by computing  $\Pr(c_j)$  and  $\Pr(w_t|c_j)$  (using Equations (27) and (28) in Chap. 3).
- 7 **until** the classifier parameters stabilize  
Return the classifier  $f$  from the last iteration.

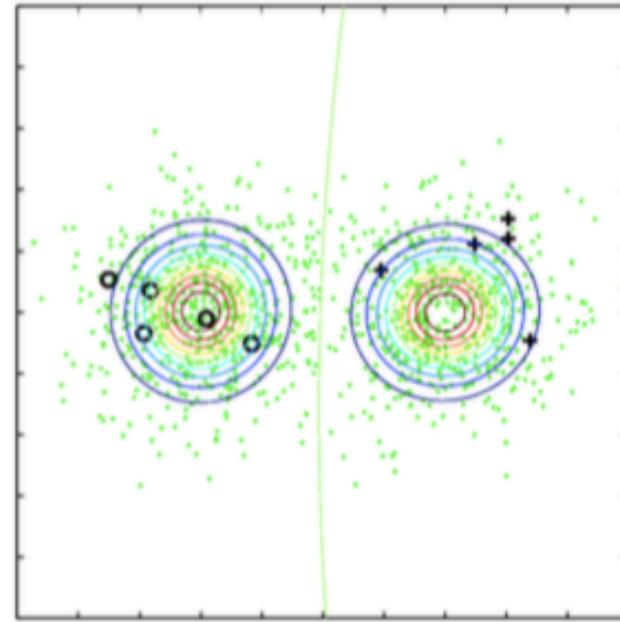
**Fig. 5.1.** The EM algorithm with naïve Bayesian classification

# Wykorzystanie modeli generatywnych - algorytm EM w SSL z nieetykietowanymi danymi

only labeled data



with unlabeled data



from [Semi-Supervised Learning, ICML 2007 Tutorial; Xiaojin Zhu]

# Podejścia Samo-uczenie się (ang. Self-learning)

Ucz się  $f$  z  $L$  i klasyfikuj przykłady  $U$  oceniając pewność predykcji

Wykorzystaj część najpewniejszych predykcji do rozszerzania zbioru uczącego:

---

## Algorithm 1 Self-training

---

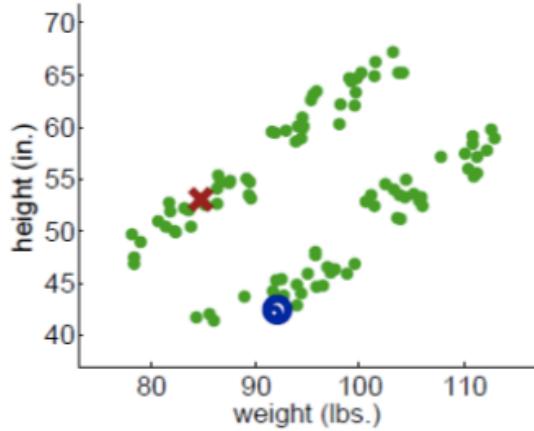
```
1: repeat
2:    $m \leftarrow \text{train\_model}(L)$ 
3:   for  $x \in U$  do
4:     if  $\max m(x) > \tau$  then
5:        $L \leftarrow L \cup \{(x, p(x))\}$ 
6: until no more predictions are confident
```

---

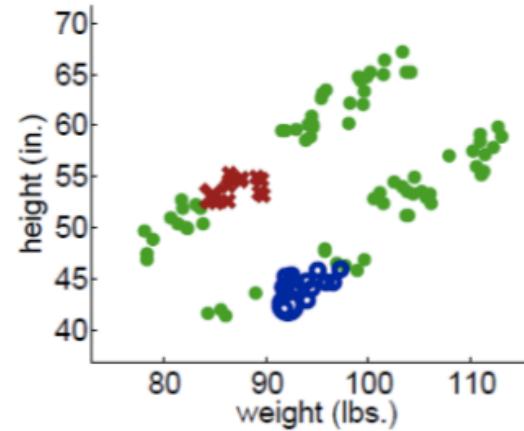
Podejście dość ogólne (typu wrapper) stosowalne wokół klasycznych algorytmów / Yarowsky, 1995; McClosky et al., 2006

Pomimo ciekawych zastosowań, złożone dane i niewłaściwe predykcje mogą źle ukierunkować kolejne iteracje

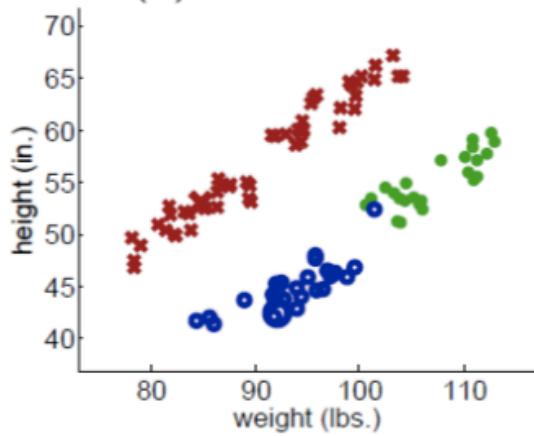
# Przykład self-learning



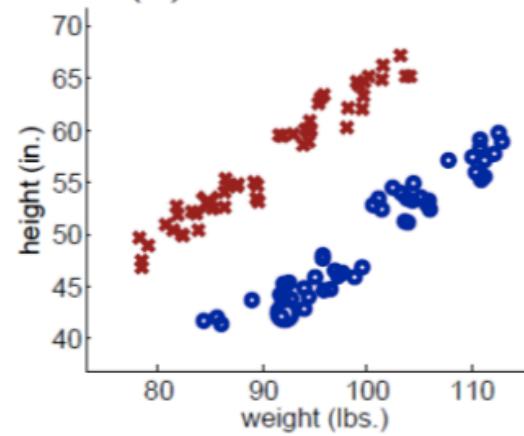
(a) Iteration 1



(b) Iteration 25



(c) Iteration 74



(d) Final labeling of all instances

Propagowanie etykiet z wykorzystaniem 1-NN

# Samo-uczenie, cd.

Otwarte pytania:

Q1: tzw. Wrapper approach – zastanów się dlaczego

Q2: Parametryzacja – co i jak dobieramy?

Q3: Kiedy ma szanse na działanie (tzw. bootstrap classifier)?

Przygody barona Munchausena ->



# Podejścia Self-learning - pytania

Podejście dość ogólne (typu wrapper) stosowalne wokół różnych klasyfikatorów – lecz w miarę skutecznych (mały błąd) z tzw. dobrym oszacowaniem marginesu pewności

Wybór miary pewności predykcji (wiele możliwości), także wybór warunku zatrzymania

Złożone dane i niewłaściwe predykcje mogą źle ukierunkować kolejne iteracje

“The main downside of self-training is that the model is unable to correct its own mistakes. If the model's predictions on unlabelled data are confident but wrong, the erroneous data is nevertheless incorporated into training and the model's errors are amplified.” S.Ruder

# Inne znane zastosowania

Przetwarzanie języka naturalnego, np.

- Yarowsky - ujednoznacznienie sensu słowa w zależności od kontekstu / np. jakie jest znaczenia słowa ang. plant
- Rilof – identyfikacja tzw. subiektywnych rzeczowników w tekście
- Maeirezio i in: ocena emocjonalnego wydźwieku rejestrów dialogów pomiędzy ludźmi

Rozpoznawanie obrazów, np.

Rosenberg i in.: rozpoznawanie obiektów na zdjęciach

# Inny paradymat -> co-training

Co-training wprowadzony przez (Blum & Mitchell, 1998) (Mitchell, 1999) wykorzystuje:

- tzw. **dwa różne, komplementarne spojrzenia (*views*)** na dane uczące, które są przydatne do budowy klasyfikatorów
- zbiór atrybutów  $X$  podzielony na **dwa rozłączne** podzbiory  $\{X_1, X_2\}$ , każdy z nich dostarcza wystarczająco dużo informacji aby wytrenować poprawny klasyfikator przy odpowiednio dużej ilości danych

Hipoteza zgodności -> klasyfikatory wyuczone z części  $c_1, c_2$  s.t.  $c_1(X_1)=c_2(X_2)=C^*(X)$  mogą być potencjalnie zgodne dla wystarczająco dużej liczby przykładów

# Dwa niezależnie zbiory atrybutów

# Naturalne w części zastosowań:

Obrazy: - różne metody przetwarzania, np. view 1 - pixel features; view 2 - Fourier coefficients

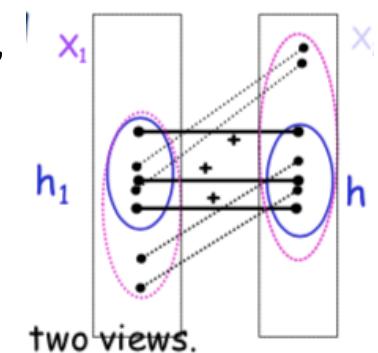
Teksty/emaile: nagłówek (header) vs. wnętrze (tekst)

Strony WWW = oryginalne zadania Blum & Mitchell (zawartość tekstowa strony vs. linki i ich opisy anchor text of hyperlinks pointing to the webpage)

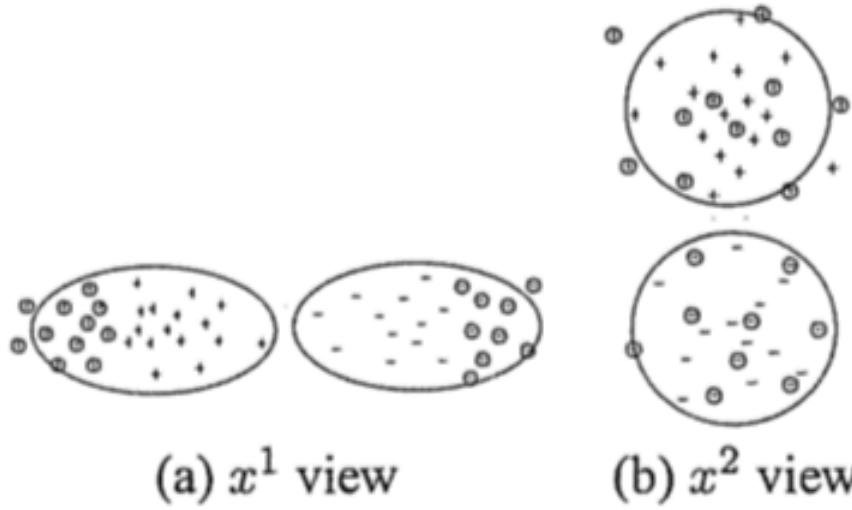
Prof. Avrim Blum    My Advisor

# Idea iteracyjnego uczenia się wzajemnego

- Każdy klasyfikator jest uczony niezależnie na etykietowanych przykładach ze swoim zbiorem atrybutów ( $L_1$  oddzielnie  $L_2$ )
- Następnie każdy klasyfikator jest użyty do predykcji klasyfikacji puli nieetykietowanych przykładów  $U$
- Jeden klasyfikator przekazuje kilka najpewniejszych “swoich predykcji” dla nieetykietowanych przykładów drugiemu klasyfikatorowi
- Zbiory uczące  $L_1$  i  $L_2$  są poszerzone i można ponownie wyuczyć klasyfikatory (z oddzielnymi zbiorami atrybutów)
- Idea: “Each view teaching (training) the other view”



Klasyfikatory powinny być odpowiednio skuteczne i pewne w etykietowaniu



## Rysunek z pracy Zhu

Figure 3: Co-Training: Conditional independent assumption on feature split. With this assumption the high confident data points in  $x^1$  view, represented by circled labels, will be randomly scattered in  $x^2$  view. This is advantageous if they are to be used to teach the classifier in  $x^2$  view.

Pozyteczne – jeśli najbardziej pewne predykcje jednego klasyfikatora odpowiadają tym przykładom dla których drugi klasyfikator nie był zbyt pewien

# Co-training – uwagi metodyczne

Uwagi badaczy:

- Views – niezależne zbiory atrybutów [formalnie pełna niezależność w sensie statystycznym]
- Klasyfikatory - weakly-useful predictor, tzn. ma większe prawdopodobieństwo predykcji klasy pozytywnej na rzeczywistym przykładzie pozytywnym niż analogicznej predykcji pozytywnej na rzeczywistym przykładzie negatywnym.
- Posiadamy wystarczającą liczbę przykładów startowych – z poprawnymi etykietami (Labeled)
- Ciekawe powiązania z modelem teoretycznym tzw. PAC learning [Arvin Blum]

Więcej w badaniach M.Balcan praca nt : Direct Optimization of Agreement between  $h_1$  and  $h_2$

# Przykłady eksperymentów [Blum, Mitchell]

- Startowy zbiór  $L$  zawierający 12 etykietowanych stron WWW (Faculty vs not)
- Ponadto 1,000 nie zaetykietowanych przykładów
- average error: learning from labeled data 11.1%;
- average error: co-training 5.0%

	Page-base classifier	Link-based classifier	Combined classifier
Supervised training	12.9	12.4	11.1
Co-training	6.2	11.6	5.0

# Inne zastosowania - obrazy

## Results: images [Levin-Viola-Freund '03]:

- Visual detectors with different kinds of processing

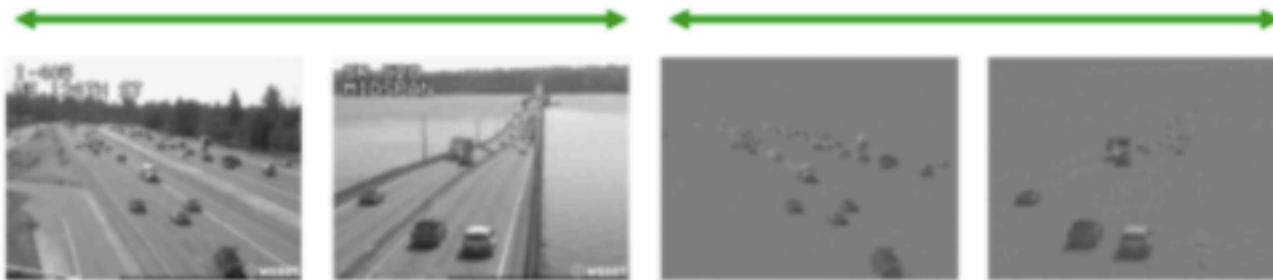
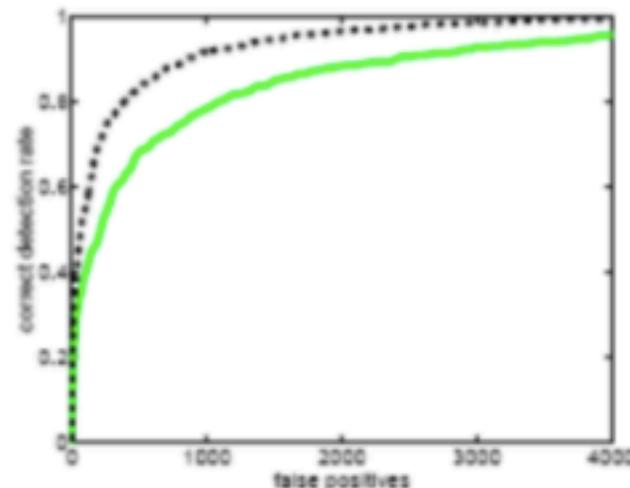


Figure 1: Example images used to test and train the car detection system. On the left are the original images. On the right are background subtracted images.

- Images with 50 labeled cars.  
22,000 unlabeled images.
- Factor 2-3+ improvement.



From [LVF03]

# Multi-view learning- rozszerzenie

- Więcej niż dwa spojrzenia na dane (czyli niezależne zbiory atrybutów)
- Naucz niezależne klasyfikatory na oddzielnych zbiorach (views – zbiory atrybutów)
- Modus operandi: predykcje klasyfikatorów wystarczająco zgodne na przykładach nieetykietowanych
- Dla przykładów testowych – agregacja **głosowania** jak w zespole klasyfikatorów
- Co-training może być widziany specjalny pod-przypadek

Spojrzeć do: J.Zhao et al: Multi-view Learning Overview: Recent Progress and New Challenges.

# Podejścia inspirowane

Spójrz do bloga Sebastian Ruder'a An overview of proxy-label approaches for semi-supervised learning

Powiązane z multi-view training

- Democratic co-training
- Tri-training
- Tri-training with disagreement
- Asymmetric tri-training
- Multi-task tri-training
- ....

# Democratic co-training

Wielokrotne uczenie zróżnicowanych klasyfikatorów (z tzw. innymi inductive bias) z L i ocena ich predykcji na U

M – zbiór klasyfikatorów z tą samą predykcją j-tej klasy dla przykładu x

w – ocena pewności predykcji klasyfikatora

Przykłady nieetykietowane dla których większość klasyfikatorów osiąga wysoką zgodność mogą być dodane do zbioru uczącego

---

### Algorithm 3 Democratic Co-learning

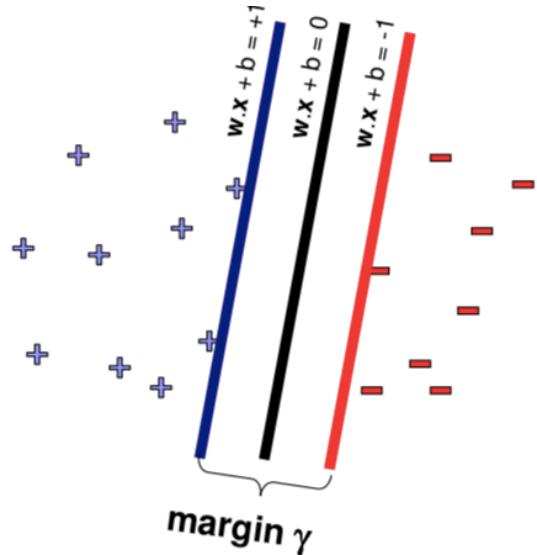
---

```
1: repeat
2:   for  $i \in \{1..n\}$  do
3:      $m_i \leftarrow \text{train\_model}(L)$ 
4:   for  $x \in U$  do
5:     for  $j \in \{1..C\}$  do
6:        $M \leftarrow \{i \mid p_i(x) = j\}$ 
7:       if  $|M| > n/2$  and  $\sum_{i \in M} w_i > \sum_{i \notin M} w_i$  then
8:          $L \leftarrow L \cup \{(x, j)\}$ 
9:   until none of  $m_i$  changes
10:  apply weighted majority vote over  $m_i$ 
```

---

# SVM w obecności częściowo etykietowanych

Przypomnijmy zasady klasycznego SVM



W trudniejszych nakładających się rozkładach przykładów element regularizacyjny ze zmiennymi osłabiającymi

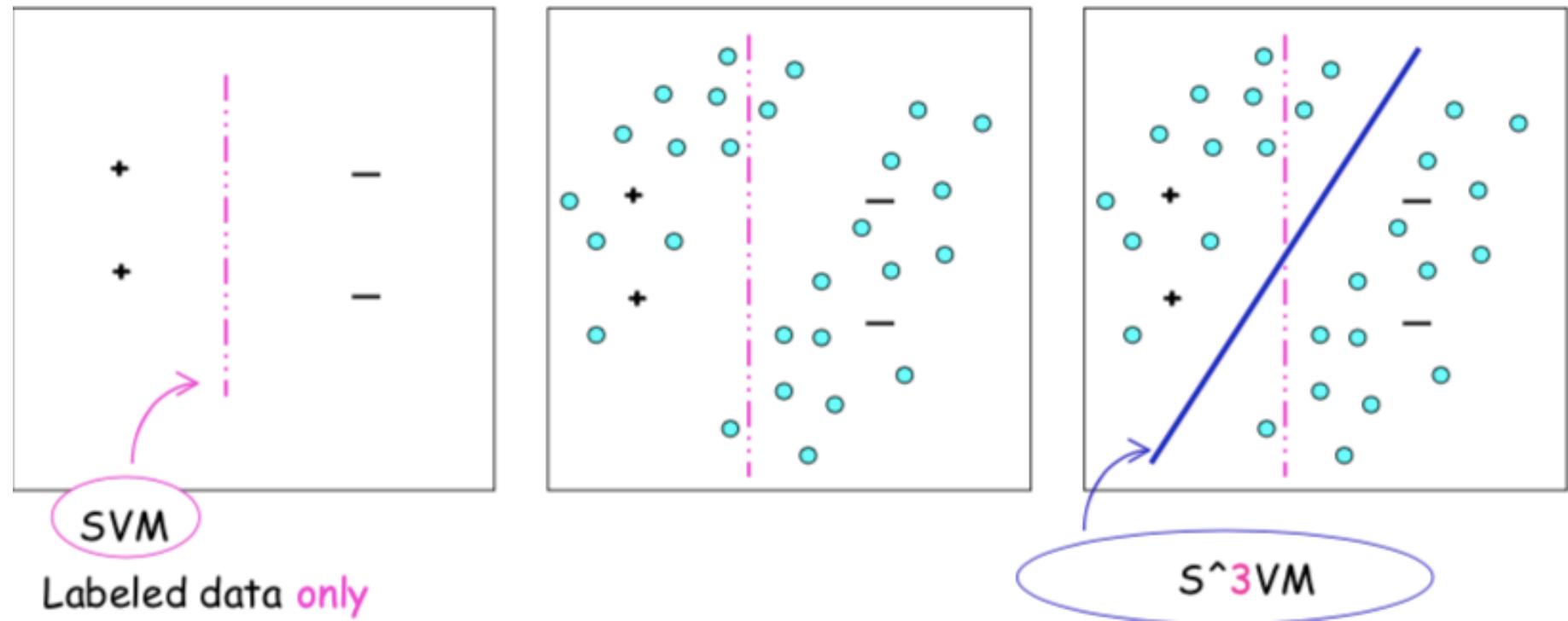
$$L(w) = \frac{\|\vec{w}\|^2}{2} + C \left( \sum_{i=1}^N \xi_i^k \right)$$

$$f(\vec{x}_i) = \begin{cases} 1 & \text{if } \vec{w} \cdot \vec{x}_i + b \geq 1 - \xi_i \\ -1 & \text{if } \vec{w} \cdot \vec{x}_i + b \leq -1 + \xi_i \end{cases}$$

# Semi-supervised SVM

Granica decyzyjna powinna przejść przez **obszar o małej gęstości przykładów** (etykietowanych i nieetykietowanych) pomiędzy dwoma klasami (-1;1)

Rozszerzenie sformułowania zadania programowania mat. SVM, aby uwzględnić obecność nieetykietowanych przykładów



# SVM -> S3VM

## Zadanie optymalizacji SVM

$$L(w) = \frac{\|\vec{w}\|^2}{2} + C \left( \sum_{i=1}^l \xi_i^k \right)$$

$$\gamma_i (wx_i + b) \geq 1 - \xi_i \quad i = 1, \dots, l$$

Funkcja celu jest przetworzona z wykorzystaniem funkcji straty (Hinge loss), która zastępują zmienną osłabiającą ( $\xi$ )

$$\min \left( \frac{1}{2} \|W\|^2 + C \sum_{i=1}^l \max(1 - \gamma_i |w \cdot x_i + b|, 0) \right)$$

za: Ding. S. et al. An overview on semi-supervised support vector machine

# S3VM

Przykłady nieetykietowane – analogiczna funkcja straty

$$\max(1 - |f(x)|, 0)$$

- ma wartość dodatnią, gdy  $-1 < f(x) < 1$  oraz 0 poza tym.

Odpowiednik kary za naruszenie separowalności marginesu

Dla wszystkich przykładów nieetykietowanych w funkcji celu pojawi się element

$$\frac{1}{u} \sum_{i=l+1}^{l+u} \max(1 - |f(x)|, 0)$$

Co prowadzi do re-definicji zadania programowania matematycznego

Trudne do rozwiązania – nieciągłe zadanie programowania kwadratowego

Przeformułowanie S3VM w pracy [Bennett and Demiriz]

# Transductive Reasoning

## Komentarz:

"Transductive" means here "reasoning from particular to particular" as opposed to "Inductive" - "reasoning from particular to general". So the motivation is that the performance of the model outside the current set of labeled and unlabeled points is not of interest

## lub [Wikipedia]

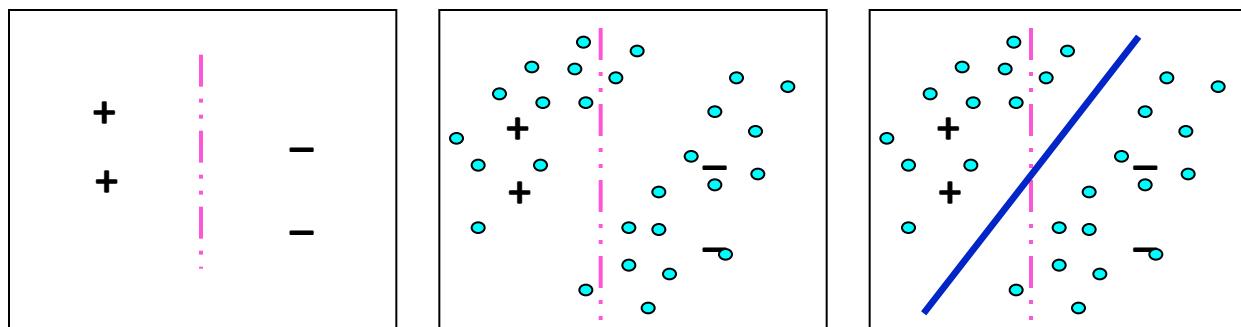
Transduction or transductive inference is reasoning from observed, specific (training) cases to specific (test) cases.

## jako przeciwnieństwo klasycznej zasady indukcji :

induction is reasoning from observed training cases to general rules, which are then applied to the test cases

# SVM [Joachims]

- Hiperpłaszczyzna powinna przejść przez obszar małej gęstości przykładów ( $L+U$ ) starając się utrzymać możliwie szeroki margines
- W ogólności trudny problem do optymalizacji
- Alternatywne podejście (z transdukcją)
  - Zbuduj szeroki margines z etykietowanymi przykładami L. Etykietuj przykłady z U.
  - Spróbuj przełączyć etykiety niektórych z przykładów z U jeśli jest to korzystane.



# Transductive SVM [Joachims]

Wykorzystuje optymalizacje na podzbiorze przykładów  $L$  i  $U$  i próbuje zaetykietować część przykładów  $U$  (tak jakby były testowe)

Następnie przełącza się etykiety niektórych przykładów, jeśli powiększają margines

1. Train SVM on labeled data
2. Classify the unlabeled data using the SVM, assign  $q$  examples with highest  $f(x)$  to the positive class, rest to negative class ( $q$  is user-set parameter)
3. Initialize  $C^* = 1E - 5$  (small value, almost disregarding unlabeled data points)
4. Iterate while increasing slack coefficient  $C^*$  of unlabeled data points:
  - 4.1 Find two unlabeled examples that have different predicted labels, and a total Hinge loss  $> 2$
  - 4.2 Swap the predicted labels of the examples  $\implies$  decrease Hinge loss
  - 4.3 Increase slack parameter  $C^*$  and retrain SVM

# Transductive SVM

Formalizacja Joachims [99] wykorzystuje założenia etykietyzacji podobnych obiektów i podobny problem optymalizacyjny

Input:  $S_l = \{(x_1, y_1), \dots, (x_{m_l}, y_{m_l})\}$

$S_u = \{x_1, \dots, x_{m_u}\}$

$$\operatorname{argmin}_w \|w\|^2 + C \sum_i \xi_i + C \sum_u \widehat{\xi}_u$$

- $y_i w \cdot x_i \geq 1 - \xi_i$ , for all  $i \in \{1, \dots, m_l\}$
- $\widehat{y_u} w \cdot x_u \geq 1 - \widehat{\xi}_u$ , for all  $u \in \{1, \dots, m_u\}$
- $\widehat{y_u} \in \{-1, 1\}$  for all  $u \in \{1, \dots, m_u\}$

Zmienne osłabiające  $\xi$  – funkcje straty Hinge loss – jak poprzednio

# Pareż uwag nt. implementacji SVM

Sklearn Python – tylko wersje label propagation (grafowe)

Indywidualne projekty – semisup-learn for Python - patrz listy dla mat. dodatkowe

SVM – dostępny w nowych bibliotekach R -> RSSL Package

## Package ‘RSSL’

February 4, 2020

**Version** 0.9.1

**Title** Implementations of Semi-Supervised Learning Approaches for Classification

**Depends** R(>= 2.10.0)

**Imports** methods, Rcpp, MASS, kernlab, quadprog, Matrix, dplyr, tidyr, ggplot2, reshape2, scales, cluster

**LinkingTo** Rcpp, RcppArmadillo

**Suggests** testthat, rmarkdown, SparseM, numDeriv, LiblineaR

**Description** A collection of implementations of semi-supervised classifiers and methods to evaluate their performance. The package includes implementations of, among others, Implicitly Constrained Learning, Moment Constrained Learning, the Transductive SVM, Manifold regularization, Maximum Contrastive Pessimistic Likelihood estimation, S4VM and WellSVM.

**License** GPL (>= 2)

**URL** <http://www.github.com/jkrijthe/RSSL>

**BugReports** <http://www.github.com/jkrijthe/RSSL>

# Inne zagadnienia

Podejścia grafowe i propagacja etykiet – pomijamy w tym wykładzie

Inne metody, nieomawiane w tej edycji przedmiotu

- Zaawansowane metody generatywne (np. wykorzystujące Hidden Markov Models)
- Sieci neuronowe
  - w szczególności: GAN - Generative Adversarial Networks - w uczeniu częściowo nadzorowanym (z małą liczbą etykietowych obrazów)
  - oraz inne DANN, i tzw. Restricted Boltzmann Machines

# Literatura i materiały dodatkowe

## Artykuły:

J.Engelen, H.Hoos: A survey on semi-supervised learning. Machine learning 2020.

Xiaojin Zhu: Semi-Supervised Learning Literature Survey.

Jing Zhao, Xijiong Xie, Xin Xu, Shiliang Sun: Multi-view Learning Overview: Recent Progress and New Challenges

Avrim Blum, Tom Mitchell: Combining Labeled and Unlabeled Data with Co-Training. COLT 1998

Książka: Chapelle, Olivier; Schölkopf, Bernhard; Zien, Alexander (2006). Semi-supervised learning. Cambridge, Mass.: MIT Press

## Wykłady:

P.Rai: CS5350 / 6350 Semi-supervised Learning

N.Balcan: Semi-supervised Learning [inspiracja dla obecnego wykładu)

A.Zien: Semi-supervised learning: Tutorial at ANN summer school

Barnabas Poczos: Semi-supervised learning

S.Ruder: An overview of proxy-label approaches for semi-supervised learning [blog]

## Software:

Głównie indywidualne projekty

np. semisup-learn for Python, RSSL dla R

# Przeglądowa strona WWW

## Soft Computing and Intelligent Information Systems

A University of Granada research group

s

Research

Publications

Teaching

Thematic Sites

Software

Awards

In the Press

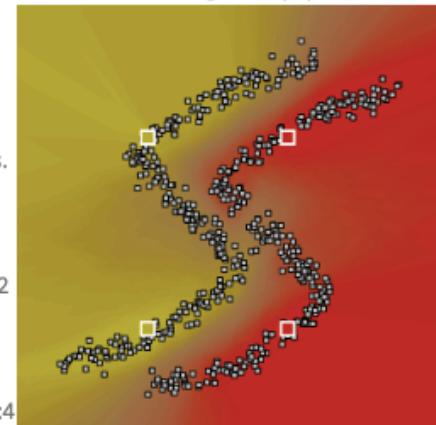
5

[Home](#) » [Thematic Sites](#) » Semi-supervised Classification: An Insight into Self-Labeling Approaches

### Semi-supervised Classification: An Insight into Self-Labeling Approaches

This Website contains SCI<sup>2</sup>S research material on Semi-Supervised Classification. This research is related to the following SCI<sup>2</sup>S papers published recently:

- **I. Triguero, S. García, F. Herrera**, *Self-Labeled Techniques for Semi-Supervised Learning: Taxonomy, Software and Empirical Study*. Knowledge and Information Systems, COMPLEMENTARY MATERIAL to the paper [here](#): datasets, experimental results and source codes. doi: [10.1007/s10115-013-0706-y](https://doi.org/10.1007/s10115-013-0706-y), in press (2014). 
- **I. Triguero, José A. Sáez, J. Luengo, S. García, F. Herrera**, *On the Characterization of Noise Filters for Self-Training Semi-Supervised in Nearest Neighbor Classification*. Neurocomputing 132 (2014) 30-41, doi: [10.1016/j.neucom.2013.05.055](https://doi.org/10.1016/j.neucom.2013.05.055) 
- **I. Triguero, S. García, F. Herrera**, *SEG-SSC: A Framework based on Synthetic Examples Generation for Self-Labeled Semi-Supervised Classification*. IEEE Transactions on Cybernetics 45:4 (2015) 622-634, doi: [10.1109/TCYB.2014.2332003](https://doi.org/10.1109/TCYB.2014.2332003). 

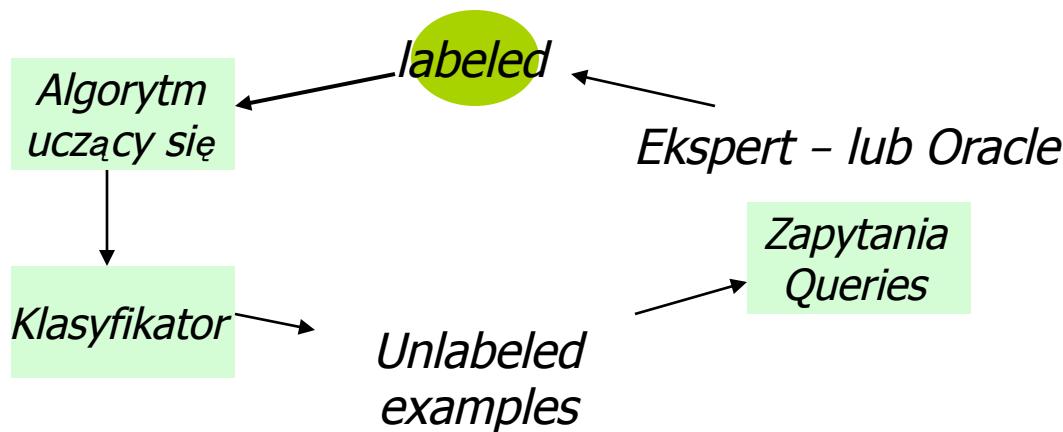


The web is organized according to the following Summary:

1. [Introduction to Semi-Supervised Classification](#)
2. [Self-Labeled Classification](#)
  - A. [Background](#)
  - B. [Taxonomy](#)

# SSL nie jest aktywnym uczeniem się

- Także zbiór etykietowanych i niezaetykietowanych przykładów
- **Active Learning:**
  - Algorytm na wpływ na etykietowanie przykładów
- Ma możliwości wyboru części niezaetykietowanych przykładów (najbardziej informatycznych dla klasyfikacji)
- Jest w stanie zadać pytanie ekspertowi / wyroczni o tzw. prawdziwą wartość etykiety wybranych przykładów
- Jak wybierać przykłady?



# **Pytanie i komentarze?**

Dalszy kontakt:

[jerzy.stefanowski@cs.put.poznan.pl](mailto:jerzy.stefanowski@cs.put.poznan.pl)

<http://www.cs.put.poznan.pl/jstefanowski/>



**Fundusze  
Europejskie**  
Polska Cyfrowa



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz  
Rozwoju Regionalnego



# **Wyjaśnialność systemów uczących się**

## wykład 15

Jerzy Stefanowski

Instytut Informatyki PP

2021 – update 2024

Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI-TECH)  
projekt finansowany z środków Programu Operacyjnego Polska Cyfrowa  
POPC.03.02.00-00-0001/20



**Fundusze  
Europejskie**  
Polska Cyfrowa



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz  
Rozwoju Regionalnego



# Plan wykładu

- Motywacje do wyjaśnialnej sztucznej inteligencji
- Terminologia
- Klasyfikacja rodzajów wyjaśnień
- Klasyfikacja podstawowych metod
- Wybrane metody XAI
  - Ocena ważności atrybutów
  - LIME
  - SHAP
  - Kontrfakty
- Podsumowanie

# Spojrzenie na rozwój AI / ML

Obserwacje z ostatnich kilkudziesięciu lat

- Gwałtowny rozwój metod ML, w szczególności głębokiego uczenia
- Przejście z środowiska akademickiego do fazy technologicznej produktów/ środowisk wykorzystywanych praktycznie / zainteresowanie przemysłu i biznesu
- Spektakularne zastosowania systemów ML/DANN z wysoką zdolnością predykcyjną (obrazy, wizja komputerowa, teksty, tłumaczenie języka, rozpoznawania mowy,...)
- Lecz są to b. złożone systemy typu „black box” w zakresie oferowania informacji jak wewnętrz działają, czego się nauczyły i jak doszły do konkretnej decyzji

# Wyjaśnialność inteligentnych systemów

Wzrost zainteresowania tzw. XAI – ang. explainable AI

Próba definicji – cyt ang. [F.Giannotti et al]:

Explainable-AI explores and investigates methods to produce or complement AI models to **make accessible and interpretable** the internal logic and the outcome of the algorithms, making such process **understandable by humans.**

Działania w celu wyjaśnienie zasad działania systemu AI oraz rezultatów ich działania

Zarówno dla ekspertów, jak nie wyspecjalizowanych użytkowników

# Motywacje – ograniczenia automatycznej predykcji

- Predykcja - to jedna ze możliwych perspektyw ML, w rzeczywistych zastosowaniach inteligentnych systemów oczekujemy także innych kryteriów oceny
- Czy ludzie ufają tzw. black box models oferowanym przez obecne ML?
- Cytaty:
  - If a machine learning model performs well, why do not we just trust the model and ignore why it made a certain decision? “The problem is that a single metric, such as classification accuracy, is an incomplete description of most real-world tasks.” (Doshi-Velez and Kim 2017)
  - Do you just want to know what is predicted? For example, the probability that a customer will churn or how effective some drug will be for a patient. Or do you want to know why the prediction was made and possibly pay for the interpretability with a drop in predictive performance? (Molnar book)

# Motywacje – ograniczenia automatycznej predykcji

Ponadto

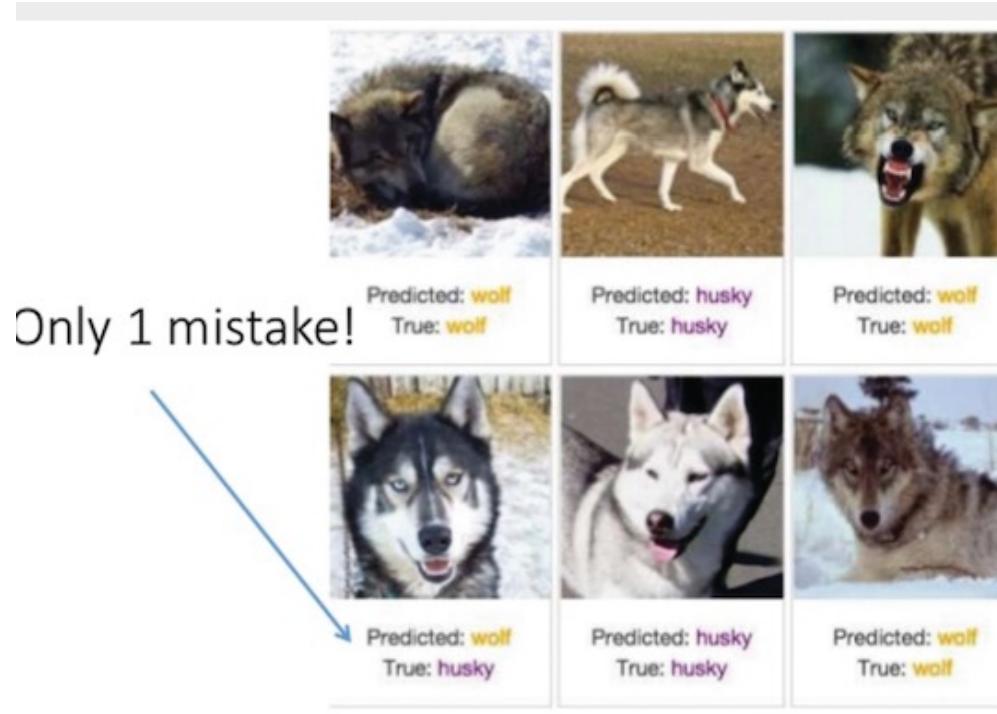
- Inne miary określają inne perspektywy spojrzenia na systemy uczące się oraz ich praktyczne wykorzystanie
- Zamknięty vs. otwarty świat – czy dysponujemy wystarczającymi danymi oraz przygotowaliśmy dogłębne testy?
- Jak radzić sobie z krytycznymi błędami systemu oraz zaskakującymi nietypowymi sytuacjami.
- Podatność na tzw. bias w danych i procesie uczenia się oraz niesprawiedliwość (ang. unfair) decyzji
- Mała odporność na tzw. ataki zewn. „hacking and adversarial attacks”

# Motywujące przykłady

- Cynthia Rudin et al – analiza działania systemu prawnego predykcji możliwości bycia w przyszłości recydywistą i zwolnień warunkowych w USA -> tzw. COMPAS system : błędne predykcje – przetrzymywania w więzieniu

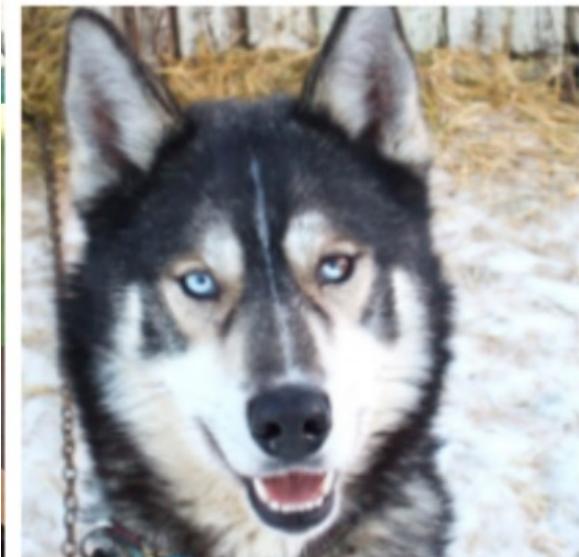
Detale: <https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm>

# Błędne działanie sieci CNN w rozpoznawaniu obrazów

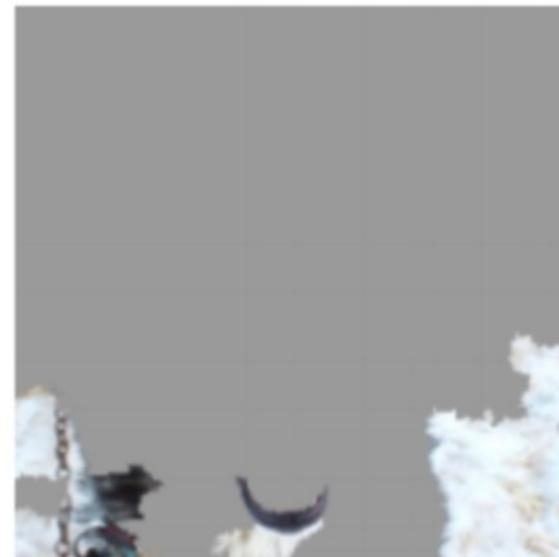


Czy Syberian Husky jest podobny do wilka?

# Efekt tła obrazu



(a) Husky classified as wolf



(b) Explanation

Sieć nauczyła się tła – śniegu na którym występowały wilki

Literatura: liczne inne przykłady -DNN nie uczą się właściwych cech

# „Krytyczne” systemy

- Systemy sterujące urządzeniami, pojazdami, oraz o bardzo ważnym znaczeniu dla sprawnego działania ważnych organizacji
  - System musi być wysoce niezawodny i zachowywać tę niezawodność w różnorodnych i częściowo nieprzewidywalnych sytuacjach.



- Rygorystyczne wymagania wobec ich analizy i weryfikacji poprawności działania
  - Wyjaśnialność może pomóc w analizie błędów lub sytuacji odpowiadającym „adversarial attacks”

# Potrzeby udzielania wyjaśnień – podejście regulacyjne

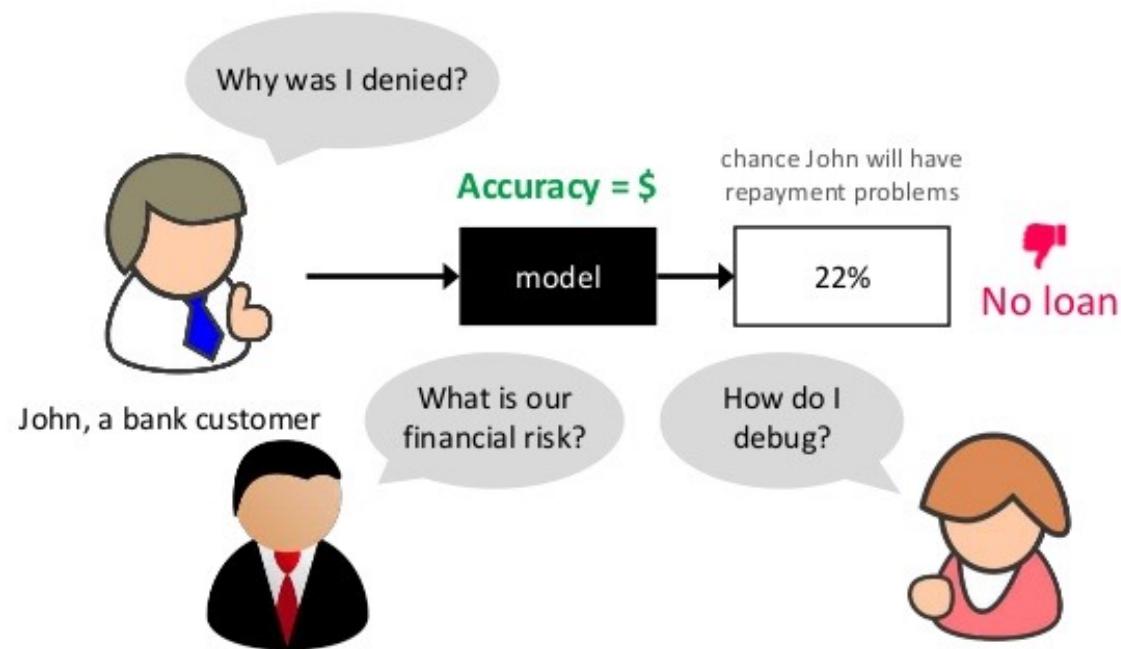
Ocena ryzyka kredytowego, dostęp do produktów finansowych, ubezpieczenia = często automatyzowanie procesu oceny wniosku klienta

Także decyzje administracyjne

The Big Read Artificial intelligence + Add to myFT

## Insurance: Robots learn the business of covering risk

Artificial intelligence could revolutionise the industry but may also allow clients to calculate if they need protection



Za wykład Scott Lundberg'a h2o world nyc 2019

# Prawo obywateli EU do wyjaśnień działania automatycznych systemów

(...) Algorytmy których decyzje są związane z predykcją użytkowników (ludzi) i mają znaczący wpływ na nich, powinny dostarczać wyjaśnień”

Dokumenty z 2016 nt. algorytmicznego podejmowania decyzji +2018 EU GDPR → a right to explanation

„a user can ask for an explanation of an algorithmic decision that was made about them”

2019 The high level expert group AI – the ethics guidelines for trustworthy AI



# Potrzeby wynikające z zastosowań ML

- Interpretowalność modeli uczenia maszynowego jest niezbędna dla **pozyskania zaufania ludzi** wobec takich systemów, zwłaszcza jeśli automatyczne podjęte decyzje są zaskakujące, nawet dla ekspertów [Stefanowski, Woźniak]
- Samek, Muller zauważają potrzeby wyjaśnialności w stosowaniu systemów predykcyjnych:
  - Weryfikowalność poprawności działania systemu predykcyjnego [przykład wykrycia niedoskonałości danych w problemie diagnozy zapalenia płuc].
  - Zrozumienie działania i wsparcie poprawy modelu
  - Uczenie się od systemów sztucznej inteligencji
  - Zgodność z prawodawstwem i postulatami regulacyjnymi

## Zdolność udzielania wyjaśnień o pracy systemu ML

System ML powinien móc udzielić wyjaśnień, dlaczego rekomenduje określoną decyzję.

- Rozróżnia się poziomy:
  - Funkcjonalnej interpretacji (jak i dlaczego model działa) i
  - Nisko-poziomowe algorytmiczne zrozumienie detali algorytmu, wpływu parametrów itd.
- Należy „powiązać atrybuty opisujące przykłady z wyjściem systemu w prosty, informatywny oraz posiadający znaczenie sposób”.
- Użytkownicy złożonych systemów muszą mieć możliwość odtworzenia całego procesu podejmowania decyzji

# Terminologia

- Rozróżnienie pomiędzy wyjaśnialnością (ang. Explainability) a interpretowalnością systemów ML – nie jest rygorystycznie precyzyjnie definiowane.
- F.Giannotti et al. artykuł:
  - **Explanation** – możliwości odpowiedzi na pytania “why”
  - **Interpretability** – opisanie wnętrza lub działania systemu w sposób potencjalnie zrozumiały dla człowieka (zależne od przygotowania wstępnego odbiorcy, jego wiedzy oraz kontekstu użycia)

## Elementy terminologiczne ad. Interpretowalność cd.

Pojęcie zrozumienia przez człowieka jest najważniejszych aspektów dobrej interpretowalności, lecz może zarówno dotyczyć zrozumienia procesu zbudowania modelu, jego ewentualnej reprezentacji, jak również zasad jego działania.

Wymienia się także inne dodatkowe pojęcia

- przejrzystość i czytelność [transparency],
- wierność odwzorowania zależności [fidelity],
- dopasowanie do zdolności poznawczych odbiorcy,
- zaufanie do modelu,

Pytanie o miary oceny spełnienia postulatów

# Wyjaśnienia dla kogo (Who)?

Nie ma uniwersalnych wyjaśnień!

Różni odbiorcy:

- Specjaliści ML
- Eksperci od zastosowania
- Użytkownicy (end user)
- Audytorzy/ regulatorzy

Odbiór wyjaśnienia w zależności od przygotowania odbiorcy i jego wiedzy

Patrz A.Weller Challenges for Transparency ICML 2017

A.Arrieta et al Explainable AI ... 2020 oraz F. Rossi, AI Ethics for Enterprise AI (2019)

# Kategoryzacja metod wyjaśnialności

- Explanation by design (wnętrze systemu)
- Black box explanation (post-hoc)
  - Model (próba wyjaśnienia podstaw całego systemu)
  - Outcome (przyczyny podjęcia specyficznej decyzji)
  - Inspection (na ogólnie wizualizacja graficzna fragmentu działania)
- Global vs. local explanations
- Model-specific vs. model agnostic

# Czytelne reprezentacje modeli ML

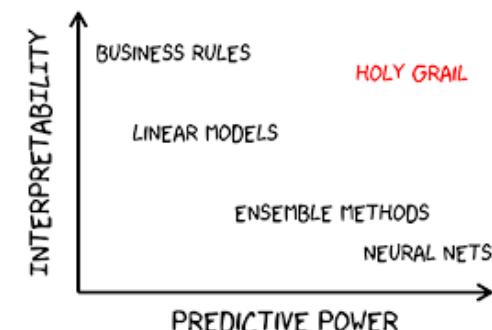
## Łatwiejsze do interpretacji

- Modele liniowe
- Reguły
- Drzewa
- K-NN
- Uogólnione model liniowe
- Modele Bayesowskie

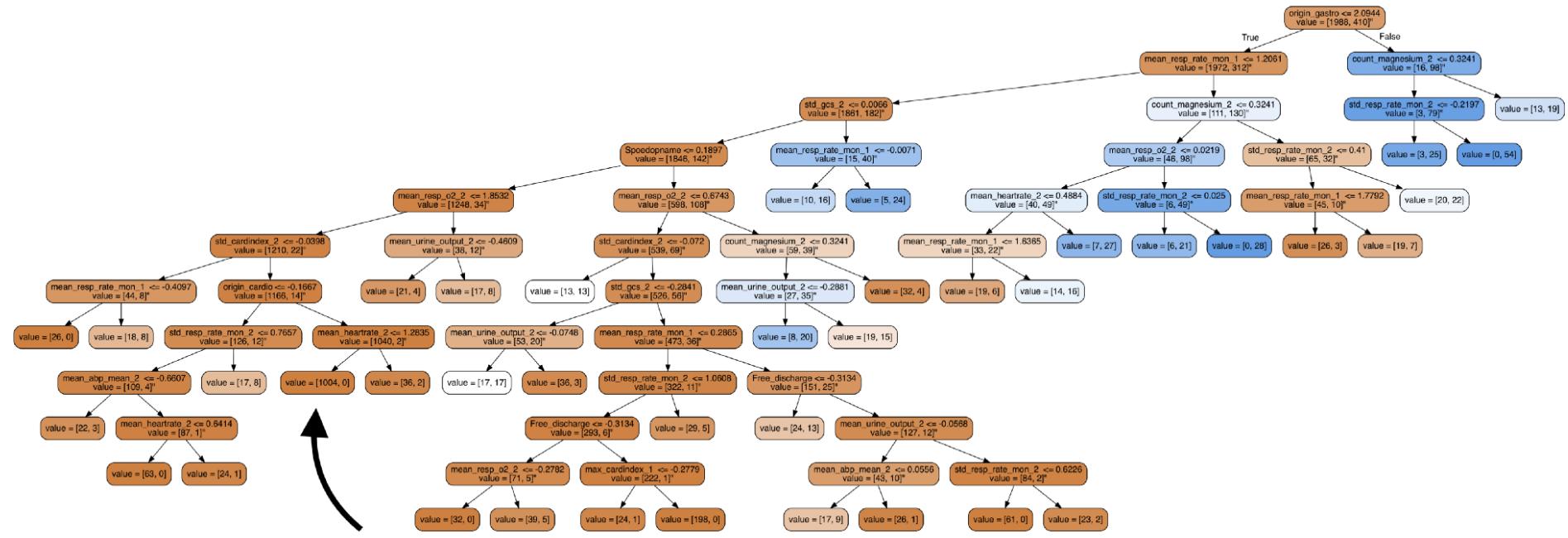
## Trudne do interpretacji

- SVM
- Zespoły modeli predykcyjnych (ensembles)
- ANN
- DNN
- Hybrydowe modele złożone

Przetarg między trafnością a złożonością / potencjalną czytelnością modelu



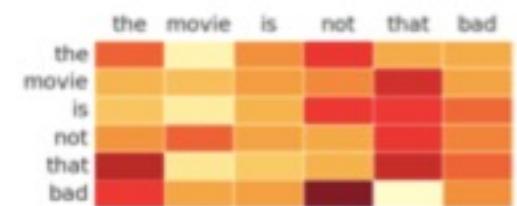
# Przykład oceny stanu chorych IOM –VUMc Amsterdam (Pacmed system)



do not come from gastroenterology  
had low respiratory rate on the first day of recovery  
have a stable score in the Glasgow Coma Scale (measuring alertness and being a proxy for neurological disorders)  
do not come from an emergency setting  
have respiratory measurements that are not worrisome at the moment  
come from cardiology  
have heart-rate measurements that are not worrisome at the moment

....

# Przykłady metod

TABULAR	IMAGE	TEXT								
<p><b>Rule-Based (RB)</b> A set of premises that the record must satisfy in order to meet the rule's consequence.</p> $r = \text{Education} \leq \text{College}$ $\rightarrow \leq 50k$	<p><b>Saliency Maps (SM)</b> A map which highlight the contribution of each pixel at the prediction.</p> 	<p><b>Sentence Highlighting (SH)</b> A map which highlight the contribution of each word at the prediction.</p> <p>the movie is not that bad</p>								
<p><b>Feature Importance (FI)</b> A vector containing a value for each feature. Each value indicates the importance of the feature for the classification.</p> <table border="1"> <tr> <td>capitalgain</td> <td>0.00</td> </tr> <tr> <td>education-num</td> <td>14.00</td> </tr> <tr> <td>relationship</td> <td>1.00</td> </tr> <tr> <td>hoursperweek</td> <td>3.00</td> </tr> </table>	capitalgain	0.00	education-num	14.00	relationship	1.00	hoursperweek	3.00	<p><b>Concept Attribution (CA)</b> Compute attribution to a target "concept" given by the user. For example, how sensitive is the output (a prediction of zebra) to a concept (the presence of stripes)?</p> 	<p><b>Attention Based (AB)</b> This type of explanation gives a matrix of scores which reveal how the word in the sentence are related to each other.</p> 
capitalgain	0.00									
education-num	14.00									
relationship	1.00									
hoursperweek	3.00									
<p><b>Prototypes (PR)</b> The user is provided with a series of examples that characterize a class of the black box</p> $p = \text{Age} \in [35, 60], \text{Education} \in [\text{College}, \text{Master}] \rightarrow \geq 50k$	$p =$  $\rightarrow$ <p>"cat"</p>	$p = \dots \text{not bad} \dots \rightarrow$ <p>"positive"</p>								

# Najpopularniejsze podejścia

- Feature summary statistics, evaluating their role, impact
- Tzw. PDP (partial dependency plots)
- Visualization (np. heat maps for ANN, ...)
- Model internal parameters (easier for typical interpretable models)
- Focus on some data elements (interpreting some data points / example based explanations) – tzw. Prototypy
- Kontrfakty
- Sensitivity / Perturbation analysis
- Use (or transform into) Intrinsically interpretable model: surrogate = zbliżone do post-hoc
- Specialized approaches to DeepANN and images (e.g. LRP Layer-Wise Relevance Propagation, decomposition)

Inne podziały:

Patrz literatura

# Ocena ważności atrybutów

- Przypomnij sobie np. feature importance w drzewach lub lasach (Breiman)

Najprostsza heurystyka analizy struktury drzewa – im wyżej i częściej występuje warunek z atrybutem

Oceń dla warunku redukcję miary (impurity) oraz wagę – liczbę przykładów w węźle

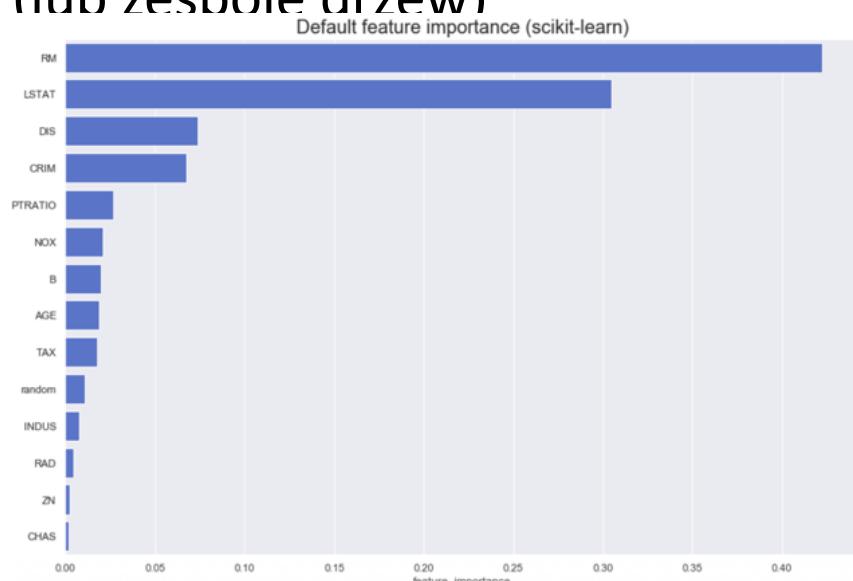
np. w węźle t dla atrybutu A spadek entropi  $\Delta(t)=0.1$  oraz  $N = 60$  przykładów, to ocena atr A =  $0.1 * 60 = 6$

Sumuj takie wystąpienia w całym drzewie (lub zespole drzew)

Pro: – szybkie obliczenia

Cons: - przybliżone obliczenia i może nadmiernie faworyzować wielo\_wartościowe oraz liczbowe atrybuty;

Brak interakcji pomiędzy cechami



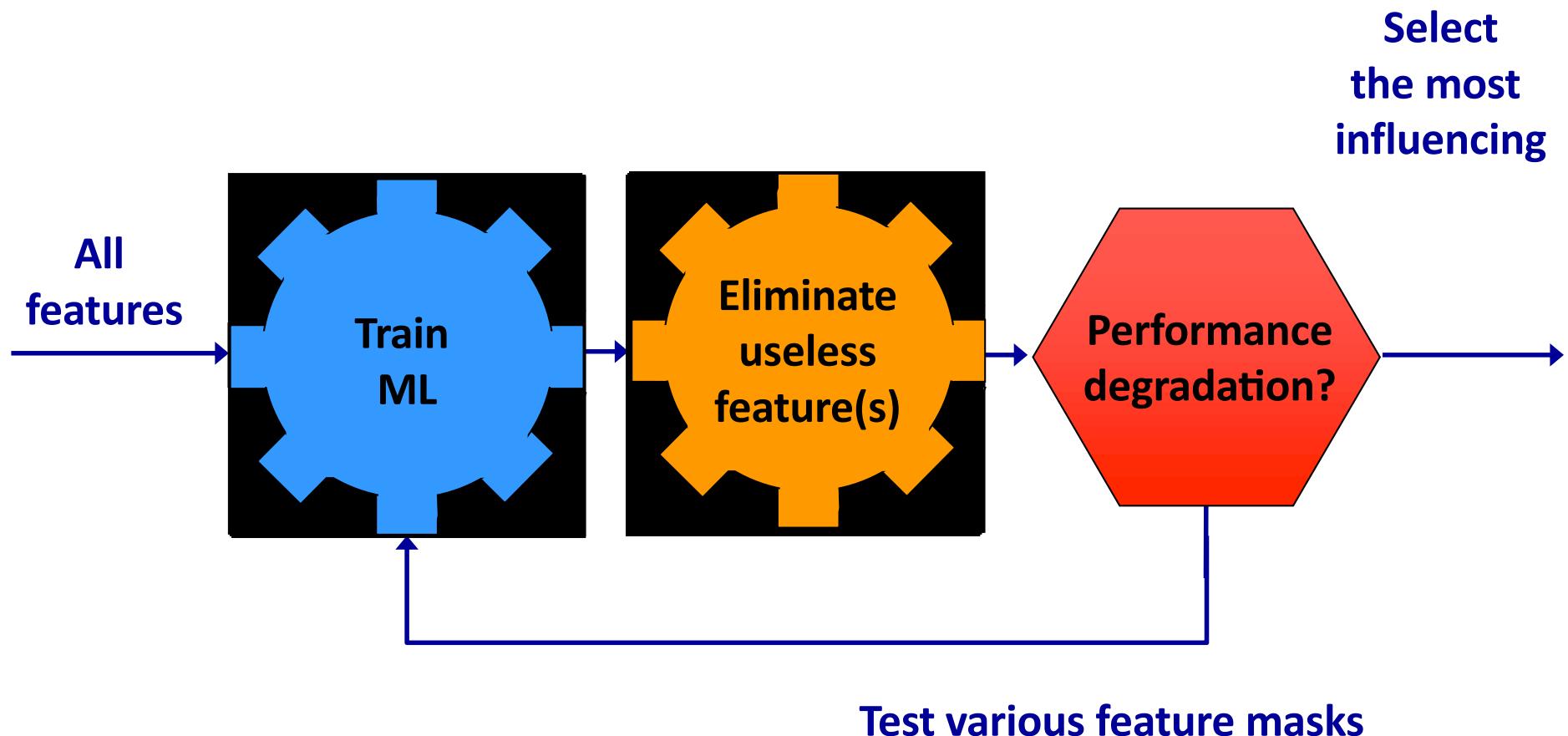
# Permutations – “noised-up” method

- L. Breiman wprowadził dla innego rankingu ważności cech poprzez analizę klasyfikowania w random forests / bagging
- Hipoteza – cecha jest tym ważniejsza, jeśli losowa zmiana jej wartości (permutacja) w zbiorze testowych mocno pogarsza trafność predykcji
- W bagging / RF – wykorzystuje się zbiór out of bag (OOB)dla każdego z drzew
  - najpierw ocenia się predykcję z oryginalnymi cechami O;
  - następnie dla każdej z cech tworzy zastępczy zbiór O':  
losowo permutuje się wartości cechy i ponownie testuje klasyfikator zachowuje się informacje o predykcjach w każdej z tych sytuacji
- Po zbudowaniu całego zespołu:
  - Dla każdego z przykładów uczących  $z_i$  (1...N) – określ predykcje klasyfikatora używając pamięci testowania OOB jeśli zawierają  $z_i$  – i oblicz dec. zespołu; następnie łączny błąd klasyfikowania e
  - Analogicznie dla każdej cechy  $x_j$  oblicz z pamięci OOB – O' błąd zespołu klasyfikatora
- Ważność cechy  $F_j = ej - e$  Im większa, tym ważniejsza  $x_j$

Pro: b. wiarygodna; stosowalna także dla innych klasyfikatorów niż drzewa

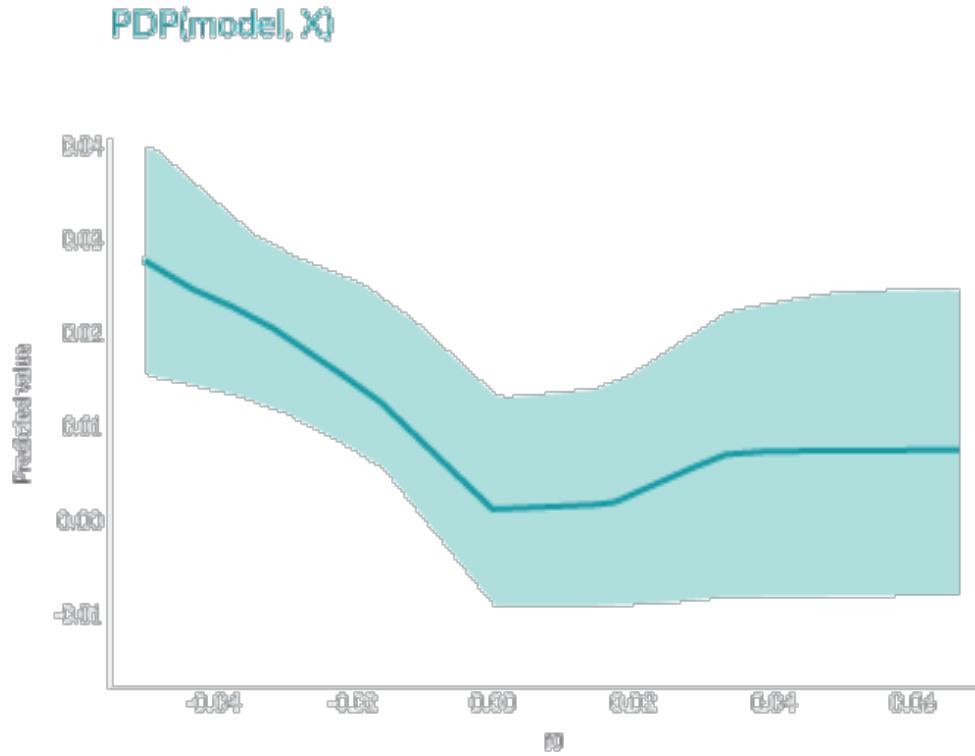
Cons: przeszacowuje skorelowane atrubuty, kosztown., nie w podst. bibliotekach

# Wrażliwość na wybrane cechy



Recursive Feature Elimination (RFE) SVM. [Guyon-Weston](#), - also for ranking them

# Partial Dependency Plots

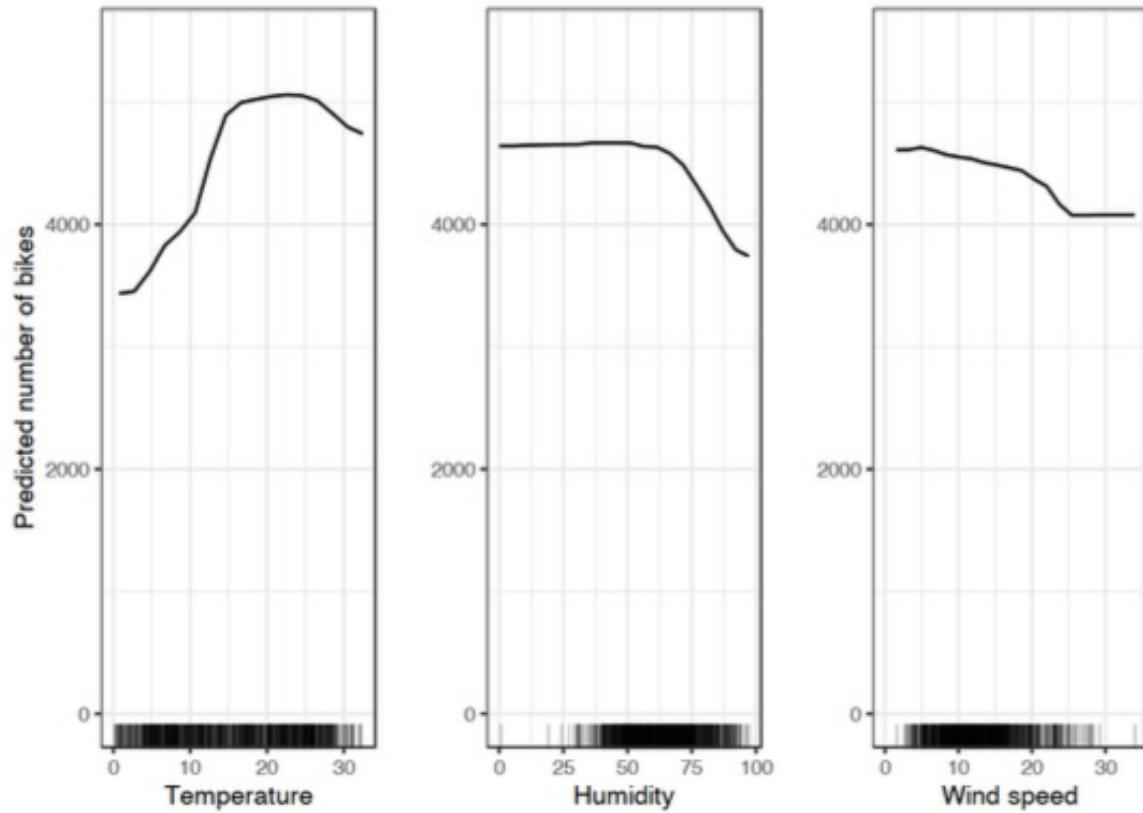


Wpływ zmian wartości jednej cechy na wyjście modelu / dla pojedynczej obserwacji “zamrażamy” inne cechy/ wynik zbiorczy wraz z zaznaczoną wartością średnią

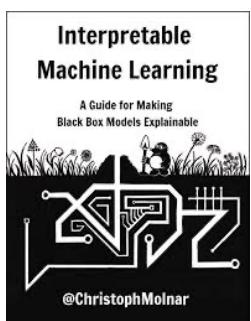
Prosta analiza wrażliwości – lecz bez dogłębniejszej analizy cech i ich interakcji

# Przykład użycia PDP

The task of prediction regression of bicycles in the city depending on external conditions[see Molnar]



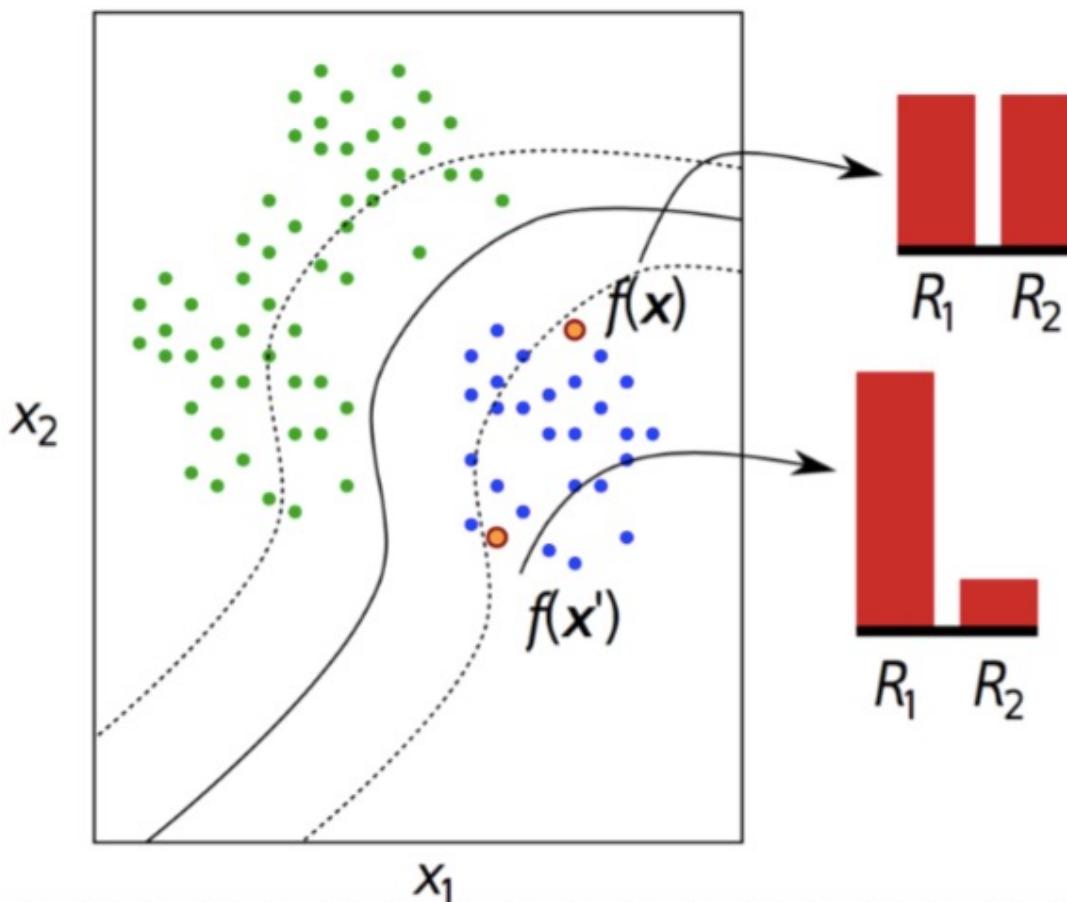
**FIGURE 7.1** PDPs for the bicycle count prediction model and temperature, humidity and wind speed. The largest differences can be seen in the temperature. The hotter, the more bikes are rented. This trend goes up to 20 degrees Celsius, then flattens and drops slightly at 30. Marks on the x-axis indicate the data distribution.



# Explaining Individual Decisions

---

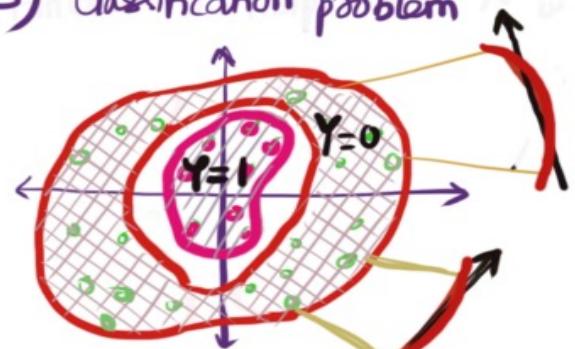
**Goal:** Determine the relevance of each input variable for a given decision  $f(x_1, x_2, \dots, x_d)$ , by assigning to these variables *relevance scores*  $R_1, R_2, \dots, R_d$ .



## Model Explainability (degree of explainability)

- 1) linearity
  - 2) monotonicity
- $f: X \rightarrow Y$  (mapping from  $x$  to  $y$  by  $f$ )

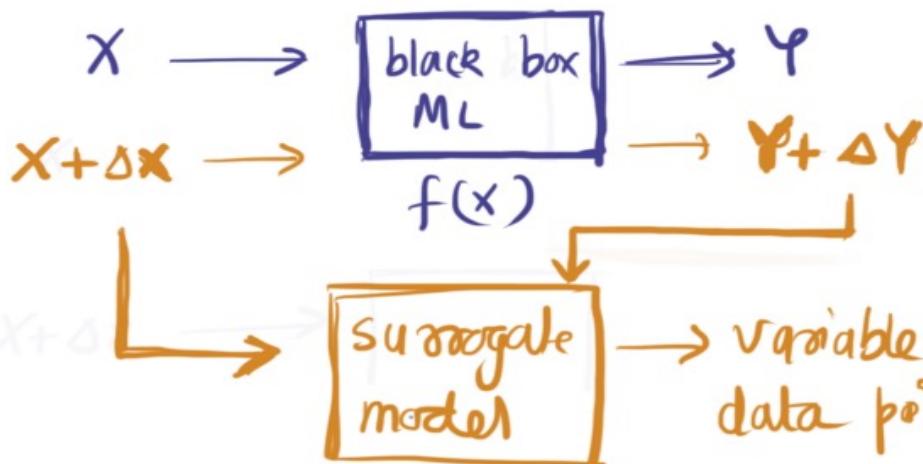
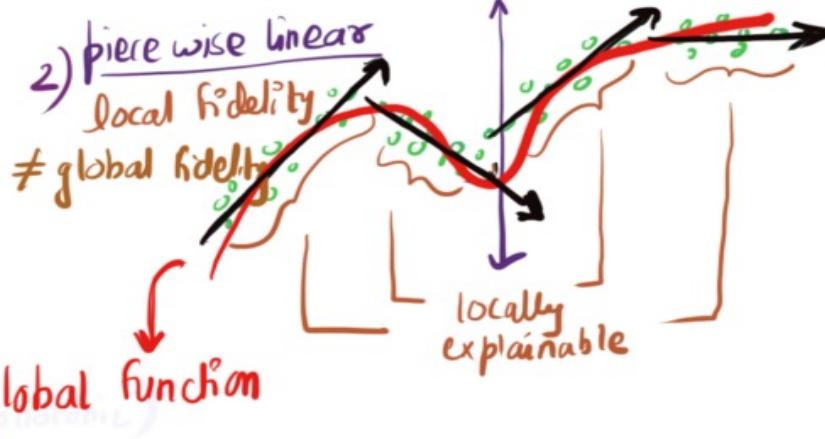
- 3) classification problem



globally : complex boundary  
locally : linear boundaries

$$f(x) = x\beta$$

1) linear regression  
global fidelity  
= local fidelity



$\Delta X$  : small change in  $X$   
 $\Delta Y$  : small change in  $Y$

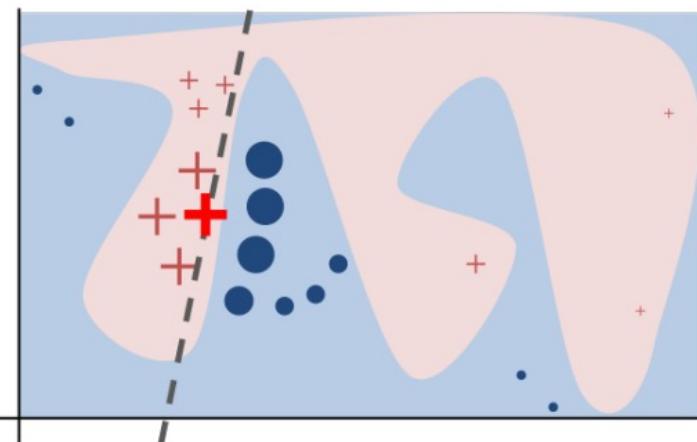
variable importance for individual  
data points

# LIME - Lokalne wyjaśnienie wpływu atrybutów na predykcje

- LIME – Local interpretable model agnostic explanations [Ribeiro et al KDD Conf. 2016]
- Globalne agnostyczne przybliżenie działania złożonego modelu  $f$  może być trudne
- Autorzy skupiają uwagę na przybliżeniu lokalnym

# LIME – ogólny schemat

- Podejście agnostyczne – lokalne – dla konkretnego przykładu  $x$ 
  - Skupienie zainteresowania na sąsiedztwie  $N(x)$  punktu  $x$



LIME (Ribeiro et. al.)

- Naucz model zastępczy  $g$  na podstawie danych z sąsiedztwa  $N(x)$  rozszerzone o losowe zaburzone punkty
- Liniowy model / regresji
- Interpretacja zmiennych w modelu linowym

# LIME – do czego dążymy?

Wyjaśnienie ważności cech / atrybutów i ich wartości dla predykcji modelu black box

- Lecz rozpatrywanych lokalnie

# LIME funkcja celu

Poszukuj modelu liniowego  $g$  o dużej lokalnie zgodności predykcji z “black box model”  $f$

$$\xi(x) = \operatorname{argmax}_{(in g \in G)} L(f, g, \pi_x) + \Omega(g)$$

gdzie  $\pi_x$  ważona odległość punktów z sąsiedztwa do  $x$

$\Omega$  czynnik regularyzacyjny – złożoność liniowego modelu  $f$  w odniesieniu do zmiennych

Autorzy LIME stosowali LASSO i regresje liniową

# Ogólny schemat algorytmu

To find an explanation for a single data point and a given classifier

- Sample the locality around the selected single data point uniformly and at random and generate a dataset of perturbed data points with its corresponding prediction from the model  $f$  we want to be explained
- Use the specified feature selection methodology to select the number of features that is required for explanation
- Calculate the sample weights using a kernel function and a distance function. (this captures how close or how far the sampled points are from the original point)
- Fit an interpretable linear model on the perturbed dataset using the sample weights to weigh the objective function.
- Provide local explanations using the newly trained interpretable model

Więcej w artykule autorów oraz na blogu

<https://www.kdnuggets.com/2019/12/interpretability-part-3-lime-shap.html>

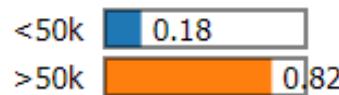
# Przykład ilustracyjny – dane o dochodach mieszkańców USA

```
marital_status           Married
education_num            Bachelors
hours_per_week           40
fnlwgt                   167065
sex                       Male
age                       50
random                   0.0412146
workclass_Private        1
occupation_Exec-managerial 1
race_White                1
Name: 23706, dtype: object
```

Label: >50K

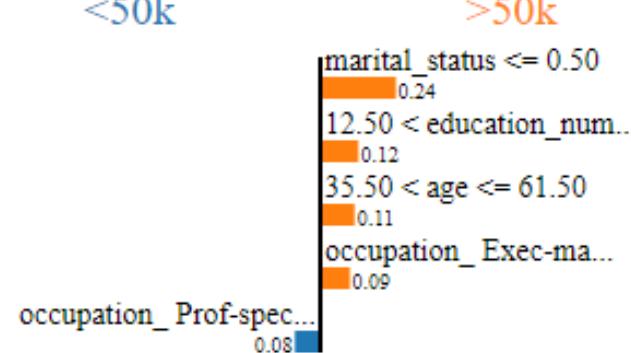
Prediction: >50K

Prediction probabilities



<50k

>50k

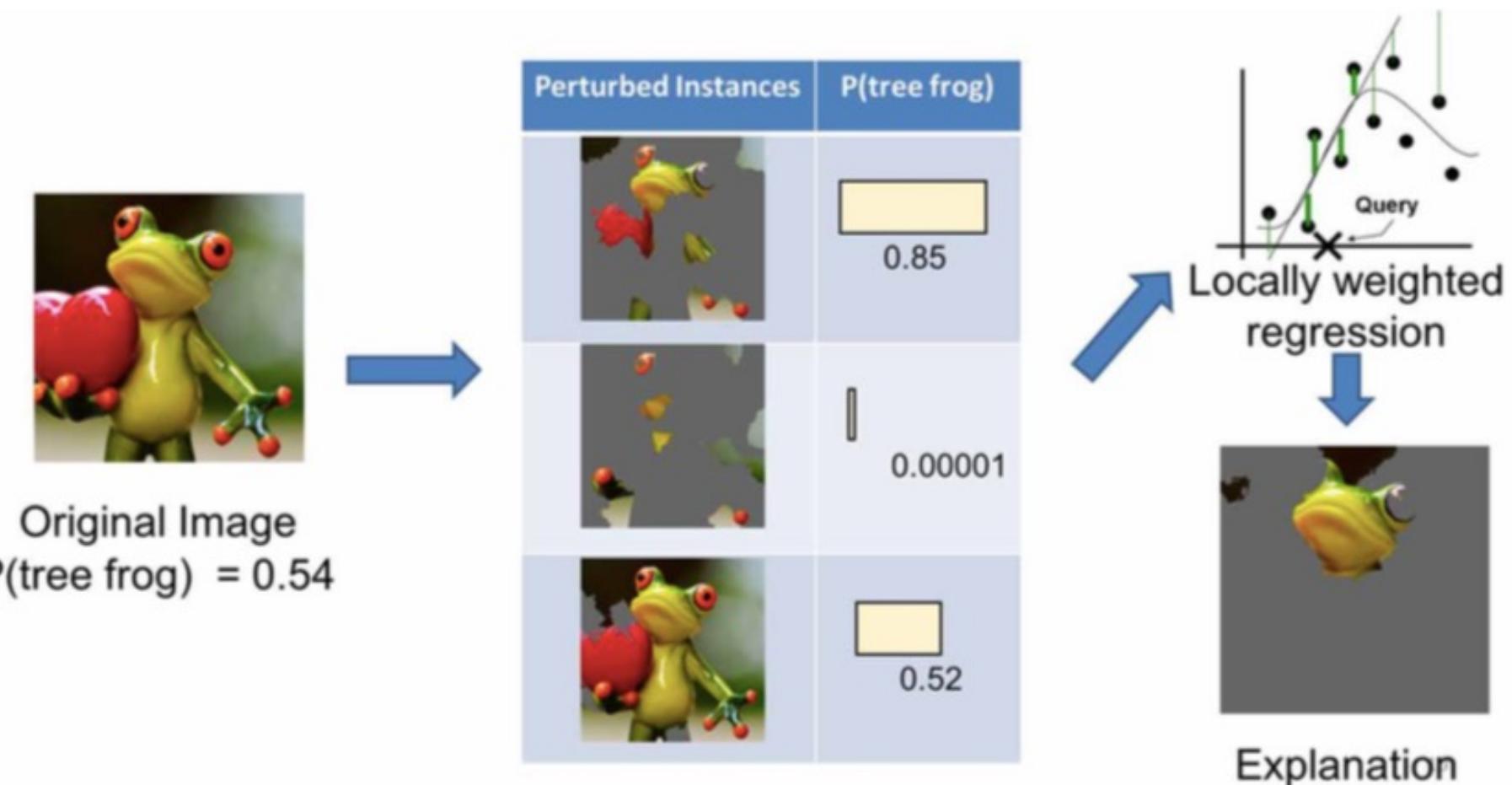


Feature Value

marital_status	0.00
education_num	13.00
age	50.00
occupation_Exec-managerial	1.00
occupation_Prof-specialty	0.00

Klasyfikacja : dwie kategorie dochodu; atrybuty opisujące pracę, wykształcenie, działalność, rasę, itd.

# LIME – image recognition



# 20 newsgroups text classification

Prediction probabilities



atheism

Posting  
0.15  
Host  
0.14  
NNTP  
0.11  
edu  
0.05  
base  
0.03  
There  
0.01

christian

## Text with highlighted words

From: johnchad@triton.unm.edu (jchadwic)

Subject: Another request for Darwin Fish

Organization: University of New Mexico, Albuquerque

Lines: 11

NNTP-Posting-Host: triton.unm.edu

Hello Gang,

There have been some notes recently asking where to obtain the DARWIN fish.

This is the same question I have and I have not seen an answer on the net. If anyone has a contact please post on the net or email me.

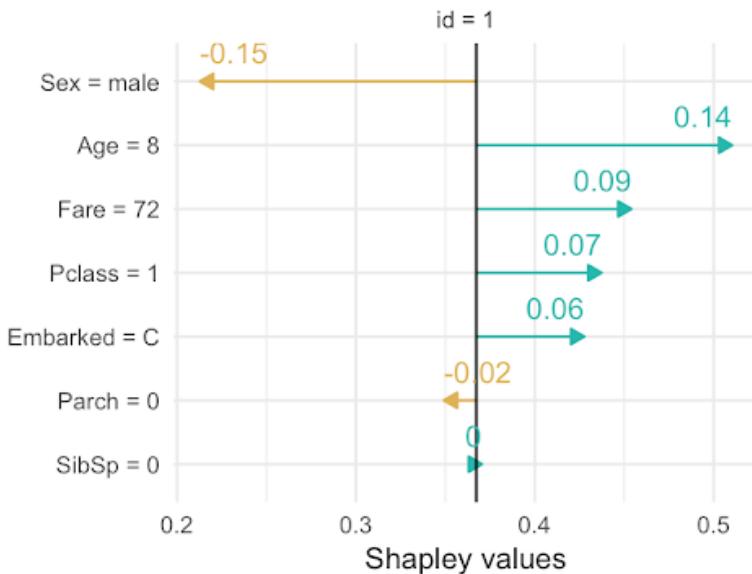
# SHAP

- Podejście agnostyczne do wyjaśniania modeli klasyfikacyjnych i regresyjnych
- Ocena globalna lub lokalna ważności / wkładu każdego atrybutu do predykcji
  - Wykorzystanie przybliżonych oszacowań wskaźników Shapley'a
  - Nacisk na ich efektywne obliczenie

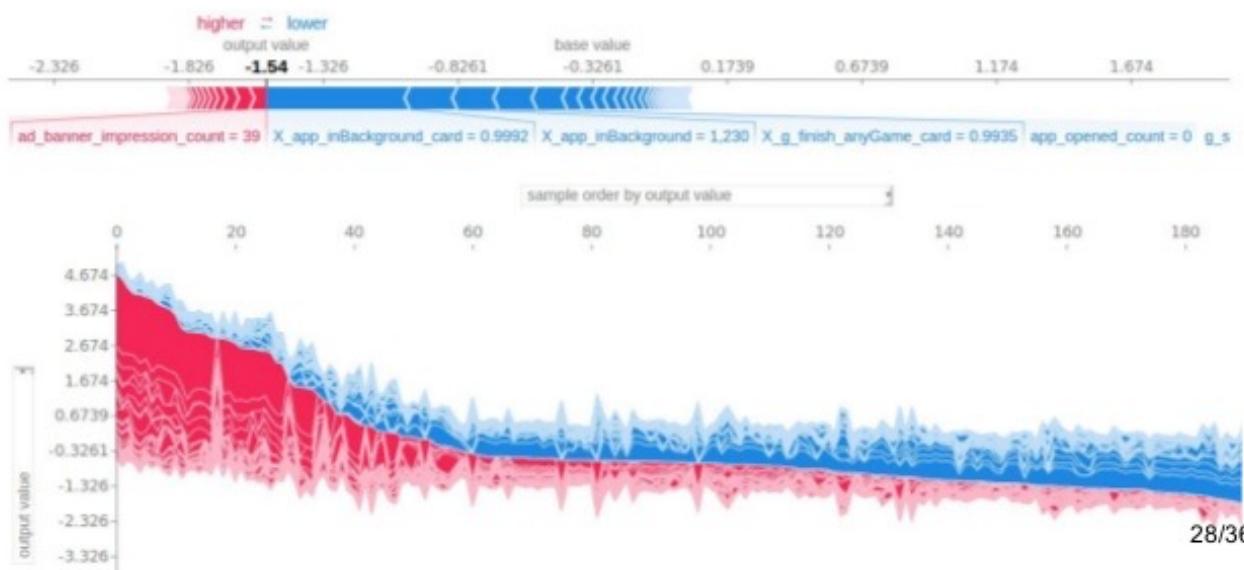
# Możliwe wyniki działania SHAP

Shapley values

Ocena wpływu atrybutów



SHAP: force plot



# Lloyd Stowell Shapley



- Amerykański matematyki i laureat Nagrody Nobla (1923-2016).
- Wkład naukowy w zastosowania ekonomiczne matematyki, w szczególności teorię gier
- Nowe propozycja w tzw. cooperative game 1953.
- Wskaźniki (Shapley values) – oszacowanie wkładu **każdego uczestnika do tzw. coalition game.**
  - Assume there are  $N$  players and  $S$  is a subset of the  $N$  players. Let  $v(S)$  be the total value of the  $S$  players. When player  $\{i\}$  join the  $S$  players, Player  $i$ 's marginal contribution is  $v(S \cup \{i\}) - v(S)$ .

# Podstawy tzw. funkcji na zbiorach

- $X = \{1, 2, \dots, n\}$  zbiór elementów (gracze, uczestnicy);  $P(X)$  – zbiór potęgowy  $X =$  zbiór wszystkich możliwych podzbiorów  $X$   
Podstawowa funkcja oceny - set function  $\mu : P(X) \rightarrow [0, 1]$
- Funkcja  $\mu$  - spełnia minimum założenia:
  - $\mu(\emptyset) = 0$  i  $\mu(X) = 1$
  - $A \subseteq B$  implikuje  $\mu(A) \leq \mu(B)$
  - „1” uznaje się za wartość max
- Praktyczna interpretacja  $\mu$  zależy od zastosowania
  - Zysk otrzymany przez graczy / agentów
  - Ważność kryteriów / atrybutów w MCDA lub analizy danych
- Transformacje funkcji  $\mu$ 
  - Wartości Shapley'a i Banzhaf'a odnoszą się do pojedynczych elementów  $i \in X$ , lecz także interakcji w parach, podzbiorów  $A \subseteq X$
  - Möbius representation  $m : P(X) \rightarrow R$

# Przykład ilustracyjny – Shapley value

- Shapley value – średni udział elementu w koalicji – zbiorze
- Niech  $X=\{1,2,3\}$  gdzie zysk z akcji udziału agenta w koalicjach  $\mu(\{1\})=5$ ,  $\mu(\{2\})=7$ ,  $\mu(\{3\})=4$ ,  $\mu(\{1,2\})=15$ ,  $\mu(\{1,3\})=12$ ,  $\mu(\{2,3\})=14$  and  $\mu(\{1,2,3\})=30$
- Jak podzielić zysk 30 jednostek agentów uwzględniając ich udział/wkład do różnych koalicji?
- Rozważając wspólny udział w  $A \subseteq X$ , podziel równo  $m(A)$  pomiędzy agentów  $m(A)/|A|$
- Każdy agent powinien otrzymać udział będący wartością Shapley'a (Shapley value)

$$\varphi_i(\mu) = \sum_{A \subseteq X: i \in A} \frac{m(A)}{|A|}$$

## Przykład ilustracyjny – Shapley value

- $X=\{1,2,3\}$  i zyski z ich udziału  $\mu(\{1\})=5$ ,  $\mu(\{2\})=7$ ,  $\mu(\{3\})=4$ ,  
 $\mu(\{1,2\})=15$ ,  $\mu(\{1,3\})=12$ ,  $\mu(\{2,3\})=14$  and  $\mu(\{1,2,3\})=30$

Shapley values dla każdego agenta

- $\phi_1(\mu)=m(\{1\})/1+m(\{1,2\})/2+m(\{1,3\})/2+m(\{1,2,3\})/3 =$   
 $5+3/2+3/2+5/3=9.67$
- $\phi_2(\mu)=m(\{2\})/1+m(\{1,2\})/2+m(\{2,3\})/2+m(\{1,2,3\})/3 =$   
 $7+3/2+3/2+5/3=11.67$
- $\phi_3(\mu)=m(\{3\})/1+m(\{1,3\})/2+m(\{2,3\})/2+m(\{1,2,3\})/3 =$   
 $5+3/2+3/2+5/3=9.67$

# Inne sformułowanie wzoru i rozszerzenia

Shapley value:

$$\Phi_i(\mu) = \sum_{A \subseteq X - \{i\}} \frac{(|X - A| - 1)! |A|!}{|X|!} \cdot [\mu(A \cup \{i\}) - \mu(A)]$$

Banzhaf value:

$$\Phi_{Bi}(\mu) = \frac{1}{2^{|X|-2}} \sum_{A \subseteq X - \{i\}} [\mu(A \cup \{i\}) - \mu(A)]$$

Interpreacja jako średni wkład elementu  $i$  we wszystkich możliwych koalicjach  $A$

Interaction indices  $(i,j) \rightarrow$  Morofushi i Soneda; Roubens

$$I_{MS}(i,j) = \sum_{A \subseteq X - \{i,j\}} \frac{(|X - A| - 2)! |A|!}{(|X| - 1)!} \cdot [\mu(A \cup \{i,j\}) - \mu(A \cup \{i\}) - \mu(A \cup \{j\}) + \mu(A)]$$

$$I_R(i,j) = \frac{1}{2^{n-2}} \sum_{A \subseteq X - \{i,j\}} [\mu(A \cup \{i,j\}) - \mu(A \cup \{i\}) - \mu(A \cup \{j\}) + \mu(A)]$$

# Inny przykład

Trzech znajomych chce partycypować w kosztach obiadu

$$v(c) = \begin{cases} 80, & \text{if } c = \{A\} \\ 56, & \text{if } c = \{B\} \\ 70, & \text{if } c = \{C\} \\ 80, & \text{if } c = \{A, B\} \\ 85, & \text{if } c = \{A, C\} \\ 72, & \text{if } c = \{B, C\} \\ 90, & \text{if } c = \{A, B, C\} \end{cases}$$

Wkład osoby A => 51.17

Zapis wszystkich obliczeń dostępny na KDDBlog

<https://www.kdnuggets.com/2019/12/interpretability-part-3-lime-shap.html>

# Wartości Shapley'a

- Ciekawa interpretacja – dostarcza więcej informacji o wpływie (zmiennej) niż prostsze metody statystyczne

Lecz,

- Obliczenia – wymaga rozważenia wszystkich permutacji elementów (w ML atrybutów).
- Koszty obliczeniowe + badanie wszystkich możliwych koalicji coraz trudniejsze dla rosnącej liczby atrybutów = zbyt kosztowne dla ML

2013 E. Štrumbelj I. Kononenko zaproponował przybliżone oszacowanie poprzez losowanie permutacji cech metodą Monte-Carlo

Nadal potrzebne efektywniejsze obliczenie przybliżenia wartości Shapley'a!

# Użycie Shapley value w metodzie SHAP

- SHAP – **SH**apley **A**dditive **eX**planations nowe podejście dla wyjaśnień modeli predykcyjnych
- Zaproponowane przez Lundberg i Lee (NIPS 2016) jako ogólne podejście do oceny klasyfikatorów oraz modeli regresji
- Popularne dzięki efektywnej obliczeniowej implementacji
- Spójrz na authors' repository  
<https://github.com/slundberg/shap>

Obliczają przybliżenie wartości Shapley lecz w innym środowisku modeli predykcyjnych

# Przeformułowanie w metodzie SHAP

Shapley regression value

$$\Phi_i = \sum_{A \subseteq F - \{i\}} \frac{(|F - A| - 1)! |A|!}{|F|!} \cdot [f(A \cup \{i\}) - f(A)]$$

Oszacowanie ważności atrybutu w modelu liniowych / nawet przy ich skorelowaniu

Predykcja -> model ze zbiorem cech A (bez elementu  $i$ ) oraz analogicznego modelu ze zbiorem rozszerzonym o  $\{i\}$

A- wszystkie możliwe podzbiory z F

Matematycznie ich wartości sumują się do 1

Ponadto – można oszacować wartości Shapleya dla predykcji przykładu  $x$  za pomocą addytywnego modelu liniowego

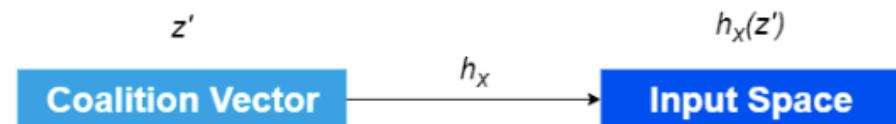
$$\hat{\phi}_j = \frac{1}{M} \sum_{m=1}^M \left( \hat{f}(x_{+j}^m) - \hat{f}(x_{-j}^m) \right)$$

# SHAP – specjalne przybliżenia

- Kernel (partly extend linear model inspired like LIME) – do not require the evaluation of all  $2^M$  sets
- Instead an additive attribute model – a weighted linear regression with simplified inputs  $z$  and estimation Shapley values by making calculation over a sample of instance predictions

$$g(z') = \phi_0 + \sum_{i=1}^M \phi_i z'_i,$$

where  $z' \in \{0, 1\}^M$ ,  $M$  is the number of simplified input features, and  $\phi_i \in \mathbb{R}$ .



The coalition vector will be all 1's

Instance to be explained  $x$

	age	marital_status	hours_worked
1	1		1

$x =$

	age	marital_status	hours_worked
34	married		40

Sampled Coalition Vector

	age	marital_status	hours_worked
1	0		0

$x =$

	age	marital_status	hours_worked
34	married		40
	single		50

Representation of how the coalition vector is converted to original input space.

# Szybsze obliczeniowo przybliżanie wartości Shapleya

- Podstawowe podejście - Kernel SHAP metoda przybliżenia wartości Shapleya wykorzystująca model addytywny liniowy.
- Możliwe jest szybsze przybliżanie wartości Shapleya, kiedy dodamy informację o rodzaju zastosowanego modelu uczenia maszynowego. Obecnie SHAP jest wyspecjalizowany dla:
  - Tree SHAP - XGBoost, LightGBM, CatBoost, scikit-learn, pyspark
  - Deep SHAP - TensorFlow, Keras, PyTorch
  - Linear SHAP

# SHAP oferuje różne wyjaśnienia

- **The global interpretability** - SHAP values wskazują wkład każdego atributu : pozytywny lub negatywny do wartości wyjściowej (ang. target variable)
- **The local interpretability** - dla każdego przykładu (i atributów z nim skojarzonych) tzw. local set of SHAP values oraz wpływu na konkretną wartość predykcji.
- Różne formy graficznej wizualizacji

# Ilustracja wykorzystania SHAP



# SHAP motywacyjne rozważanie

Rozważmy predykcje ceny mieszkań w pewnym mieście

- Wybieramy trzy najważniejsze atrybuty: flat size, year of building and a localization (region/district of the city)
- Dla oferty “40m<sup>2</sup>, building from 1920 and inside Old City this” model przewiduje cenę 450.000
- Wiedząc że **średnie ceny w tym rejonie miasta są approx. 400.000**, stawiamy pytania: jaką jest przyczyna wyższej ceny, jaki jest wkład wartość każdego atrubutu do ceny?
- SHAP values wskazały pozytywny wkład lokalizacji mieszkania (podniesienie ceny o około 70.000); negatywny związek z wiekiem (obniżenie o około 20.000), wielkość mieszkania nie ma znaczącego wpływu na cenę.

# Boston housing data

Dane o 506 nieruchomościach opisanych 13 atrybutami i jednym wyjściem (MEDV – the price of the house)

Atrybuty:

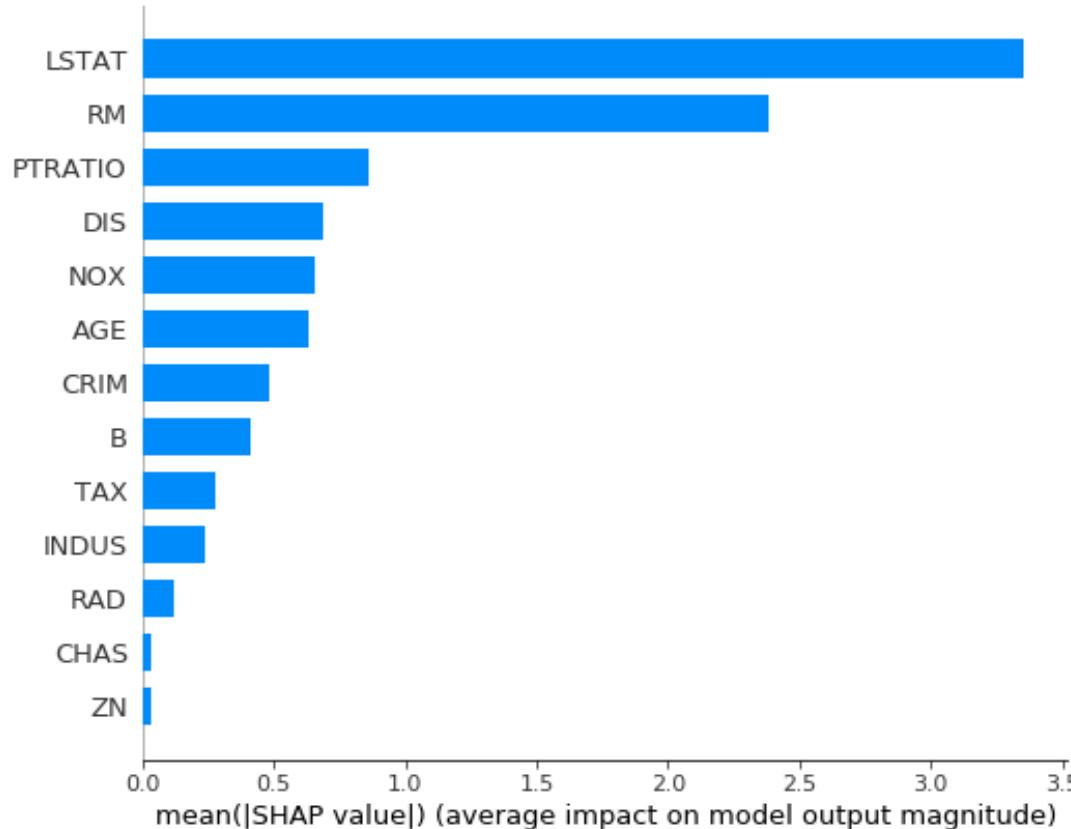
- CRIM – wskaźnik przestępcości na mieszkańców według miasta
- ZN – część działki pod zabudowę mieszkaniową pod działki o powierzchni ponad 25 000 **stóp kwadratowych**
- INDUS – odsetek niedetalicznych akrów biznesowych na miasto.
- CHAS – zmienna zmienna Charles River (1, jeśli trasa ogranicza rzekę; 0 w przeciwnym razie)
- NOX – stężenie tlenków azotu (części na 10 milionów)
- RM – średnia liczba pokoi na mieszkanie
- AGE – odsetek jednostek zajmowanych przez właścicieli wybudowanych przed 1940 r
- DIS – ważone odległości do pięciu centrów zatrudnienia w Bostonie
- RAD – indeks dostępności do radialnych autostrad
- TAX- pełna stawka podatku od nieruchomości od 10 000 USD
- PTRATIO – stosunek liczby uczniów do nauczycieli według miasta
- B –  $1000(Bk - 0,63)^2$ , gdzie Bk to odsetek czarnych według miasta
- LSTAT -% niższy status populacji
- MEDV – Medianą wartości domów zajmowanych przez właścicieli w tysiącach dolarów

Modele – wybrano XGBoost regressor vs. linear regression)

# Globalna interpretacja

Typowa wizualiacja rankingu atrybutów wg. Wartości Shapley'a / im wyżej, tym bardziej wpływowe

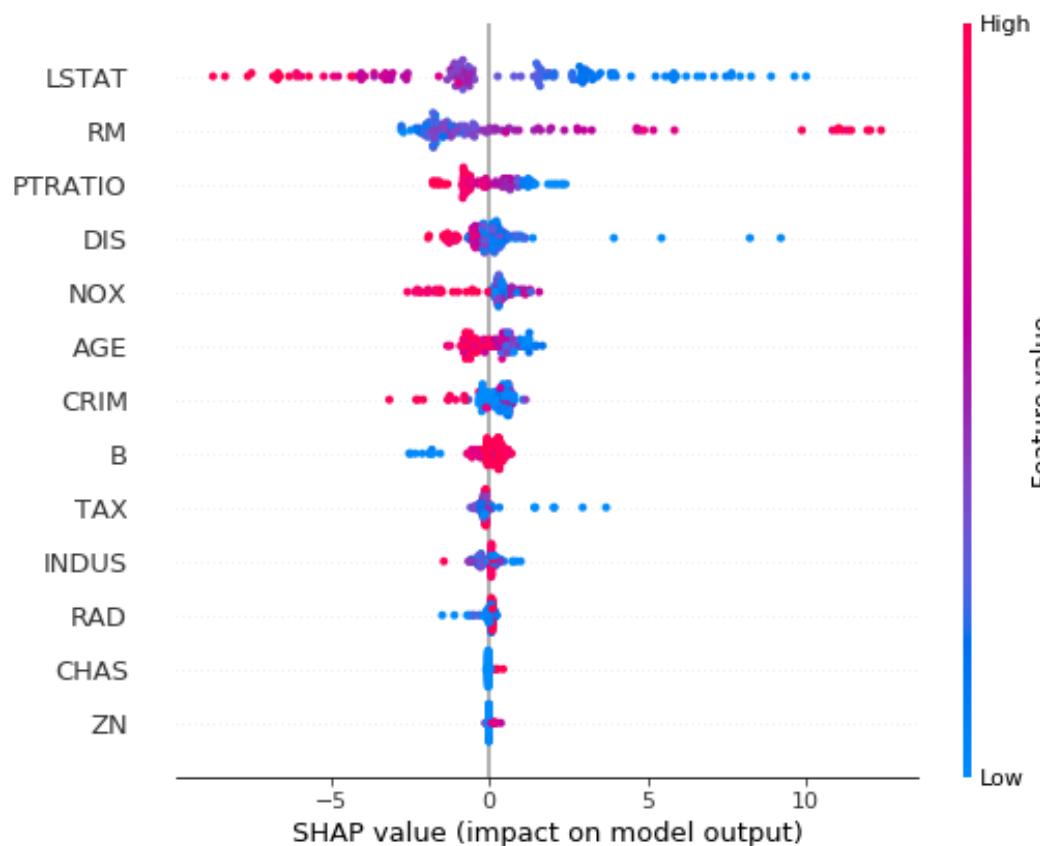
LSTAT, RM, ... najbardziej wpływowe, a CHAS i ZN najmniej



# Global kierunek wpływu

Można pokazać pozytywny lub negatywny wpływ każdego atrybutu na wartości zmiennej wyjściowej

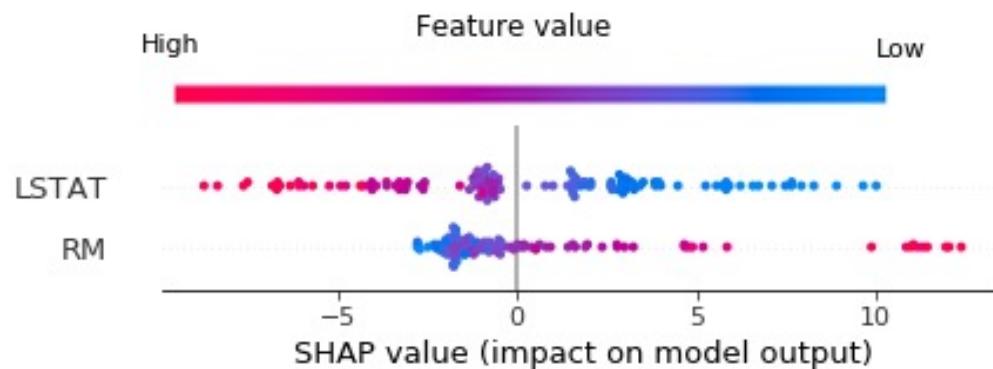
Czerwony kolor oznacza wpływ na podwyższenie wartości, a niebieski na zmniejszenie wartości wyjścia (tutaj ceny nieruchomości)



# Wykres kierunku wpływu

Wykres obejmuje przykłady uczące – każda kropka to pojedynczy przykład:

- Ranking ważności atrybutów wg. wyższych wartości Shapleya
- Kierunek wpływ: linia pozioma – czy oryginalna wartość atrybutu dla obiektu wpływa na wyższe (czerwone) czy zmniejszenie (niebieskie) wartości predykcji y

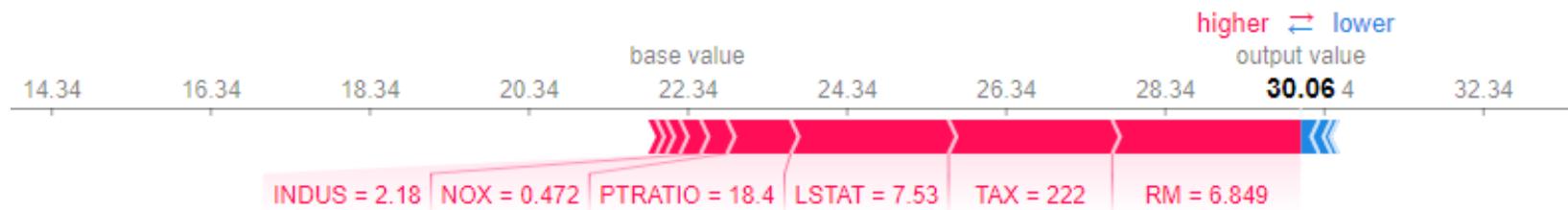


LSTAT : niższe wartości – większy wpływ na wyższą cenę

RM : inny rodzaj wpływu – wyższe wartości powiązane z wyższą ceną

# Lokalne wyjaśnienie dla pojedynczej predykcji – jak wartości konkretnego przykładu wpływają na wartość wyjścia

Dla tego przykładu:



Wyjście – predykcja modelu ( $y = 30,06$ ) – wyższa niż średnia wartość predykcji dla wszystkich przykładów (base value 22.34)

Kolory – czerwone – atrybuty, które swoimi wartościami wpływają na podwyższenie wartości  $y$  (ceny) i w jakim stopniu, niebieskie wkład do obniżenia  
Wskazane atrybuty częściowo zgodne z rankingiem ważności / największy wkład do wysokiej ceny mają RM, TAX później LSTAT i PTRATIO

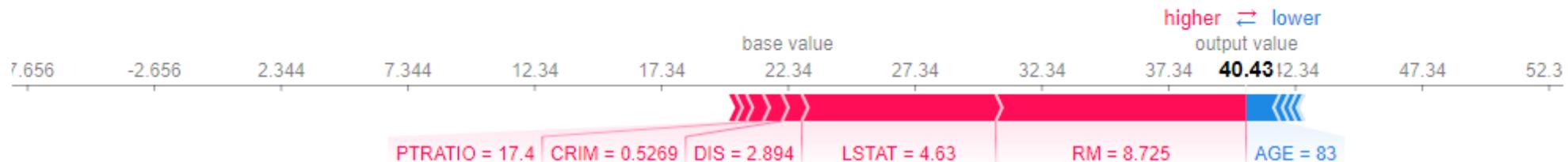
# Porównywanie kilku predykcji

Porównaj analizę predykcji dla ofert nr 17, 23 i 54

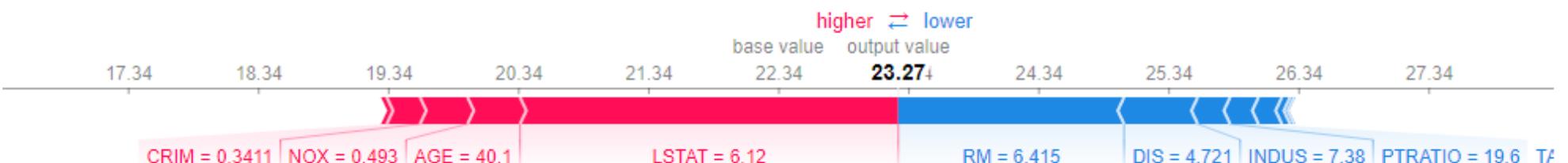
Zauważ inne wkłady atrybutów dla wyższych i mniejszy cen y

Możliwość dostosowania ofert dla klientów

Obserwacja nr: 17



Obserwacja nr: 23



Obserwacja nr: 54



# Niektóre biblioteki

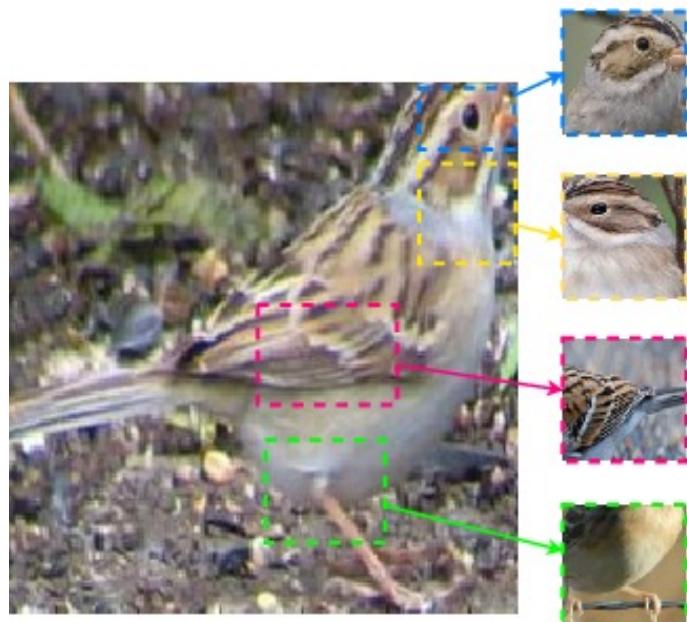
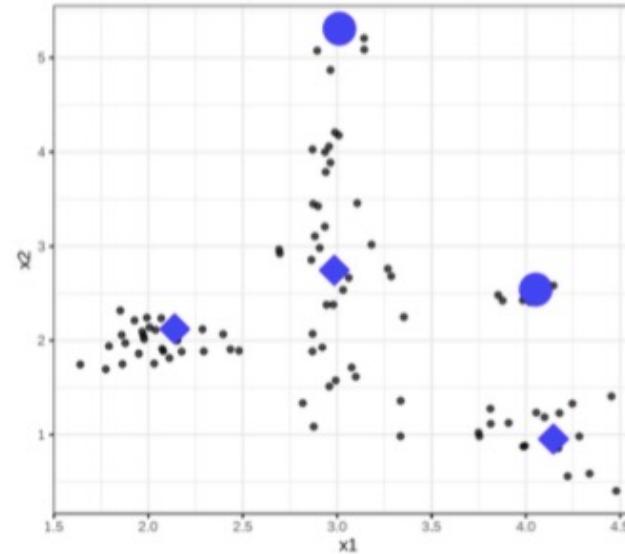
- DeepExplain: perturbation and gradient-based attribution methods for Deep Neural Networks interpretability.  
[github.com/marcoancona/DeepExplain](https://github.com/marcoancona/DeepExplain)
- iNNvestigate: A toolbox to iNNvestigate neural networks' predictions.  
[github.com/albermax/innvestigate](https://github.com/albermax/innvestigate)
- SHAP: SHapley Additive exPlanations. [github.com/slundberg/shap](https://github.com/slundberg/shap)
- ELI5: A library for debugging/inspecting machine learning classifiers and explaining their predictions. [github.com/TeamHG-Memex/eli5](https://github.com/TeamHG-Memex/eli5)
- Skater: Python Library for Model Interpretation/Explanations.  
[github.com/datascienceinc/Skater](https://github.com/datascienceinc/Skater)
- Yellowbrick: Visual analysis and diagnostic tools to facilitate machine learning model selection. [github.com/DistrictDataLabs/yellowbrick](https://github.com/DistrictDataLabs/yellowbrick)
- Lucid: A collection of infrastructure and tools for research in neural network interpretability. [github.com/tensorflow/lucid](https://github.com/tensorflow/lucid)

# Wyjaśnienia z prototypami

Prototyp – przykład, który jest reprezentatywny dla podzbioru danych

- Rzeczywisty przykład uczący
- Centroid skupiska
- Sztuczny przykład o specjalnych właściwościach

Klasyfikacja  $x$  – może być wyjaśniona poprzez podobieństwo do prototypów



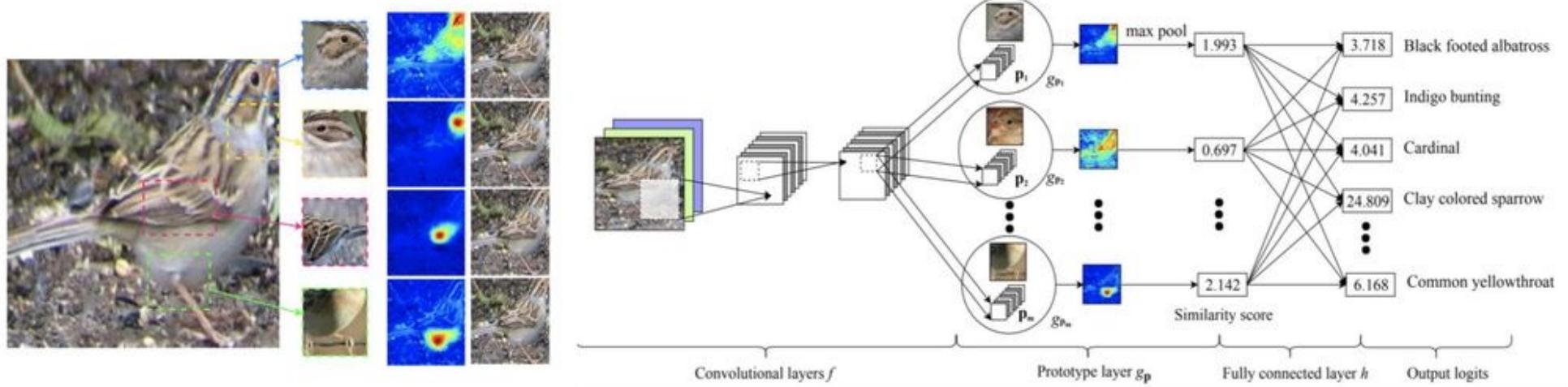
Obrazy – np. PROTOPNET [Chen et al 2019] This looks like this – specjalna odmiana CNN

Prototyp – na podstawie ukrytej reprezentacji w warstwach sieci – powiązany do części obrazu

Klasyfikacja = określa ich ważone podobieństwo w obrazie testowych do wyuczonych prototypów – proste do interpretacji

# ProtoPNet – porównywalna trafność do innych CNN

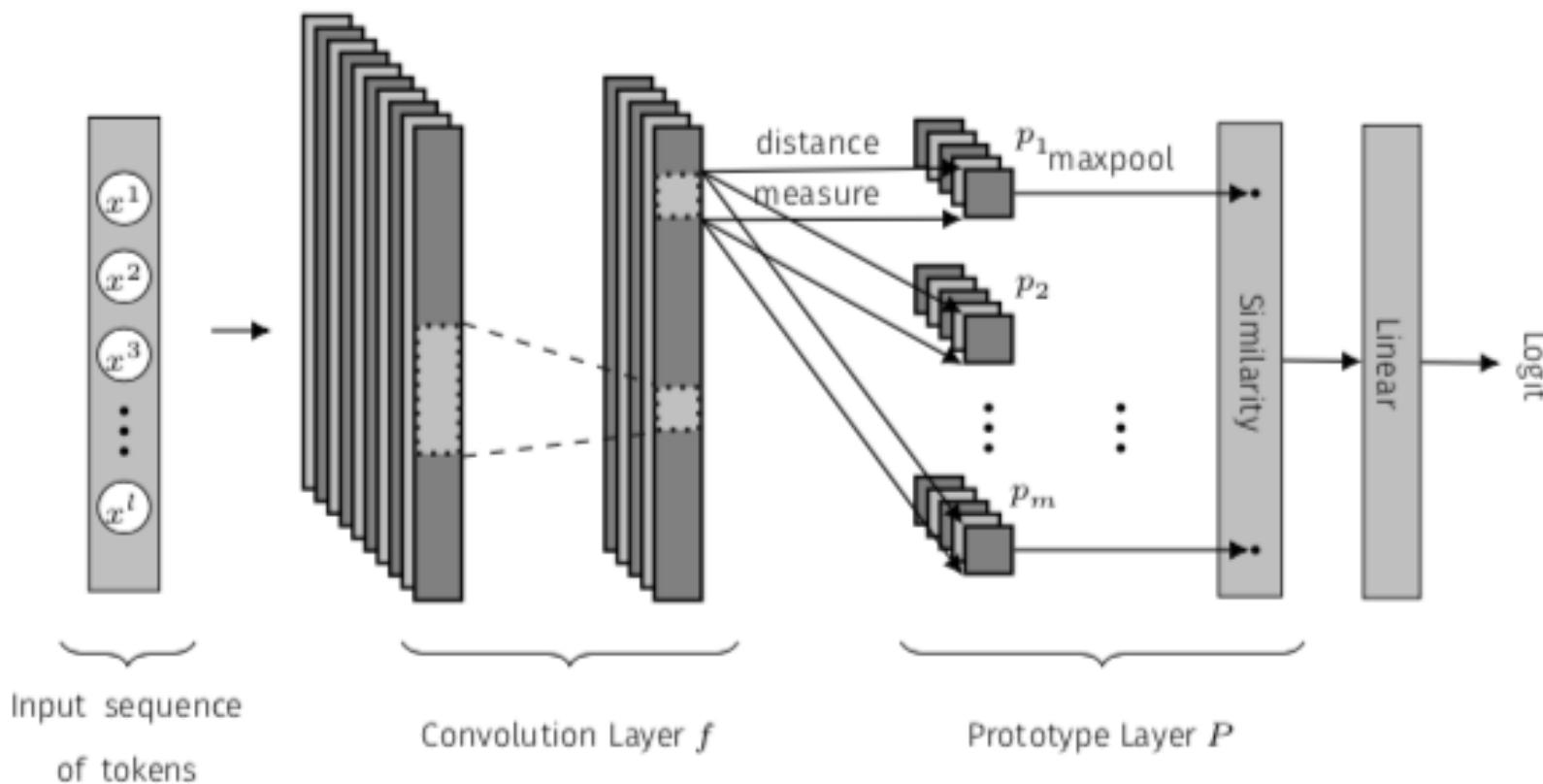
## Interpretable Deep Learning



This Looks Like That: Deep Learning for Interpretable Image Recognition. Chen et al 2018.

# K.Pluciński, M.Lango, J.Stefanowski: Prototypical Convolutional Neural Network; 2021

- Prototypy związane z najbardziej podobnymi frazami w dokumencie
- Architektura konwolucyjna z filtrami n-gram + “white” wyjście - liniowa ważona kombinacja podobieństwa przykładu testowego do wyuczonych prototypów – dynamiczny dobór ich liczby



# Przykład dialogu z użytkownikiem

Table 2: Explanation for a correct prediction.

**Input example:** about twenty minutes into this movie i was already bored quite simply these characters were fairly dull occasionally something enjoyable would happen but then things would slow down again fortunately my patience was eventually rewarded and the ending to this movie was n't bad at all however it was by no means good enough to justify sitting through the first ninety minutes so i would say that the movie was mediocre overall and considering all of the talent in the cast i 'd call this a disappointment.

Prediction: **Negative**, Gold standard: **Negative**

Evidence for negative sentiment:

Prototype	Most similar phrase	Similarity * Weight
unbelievable rambling nonsense that should a waste of film	was mediocre overall and considering characters were fairly dull occasionally	$2.17 * 0.96 = 2.08$
was the worst film i	the movie was mediocre overall	$2.08 * 0.86 = 1.79$
		$2.89 * 0.56 = 1.62$

Sum of evidence: **5.49**

Evidence for positive sentiment:

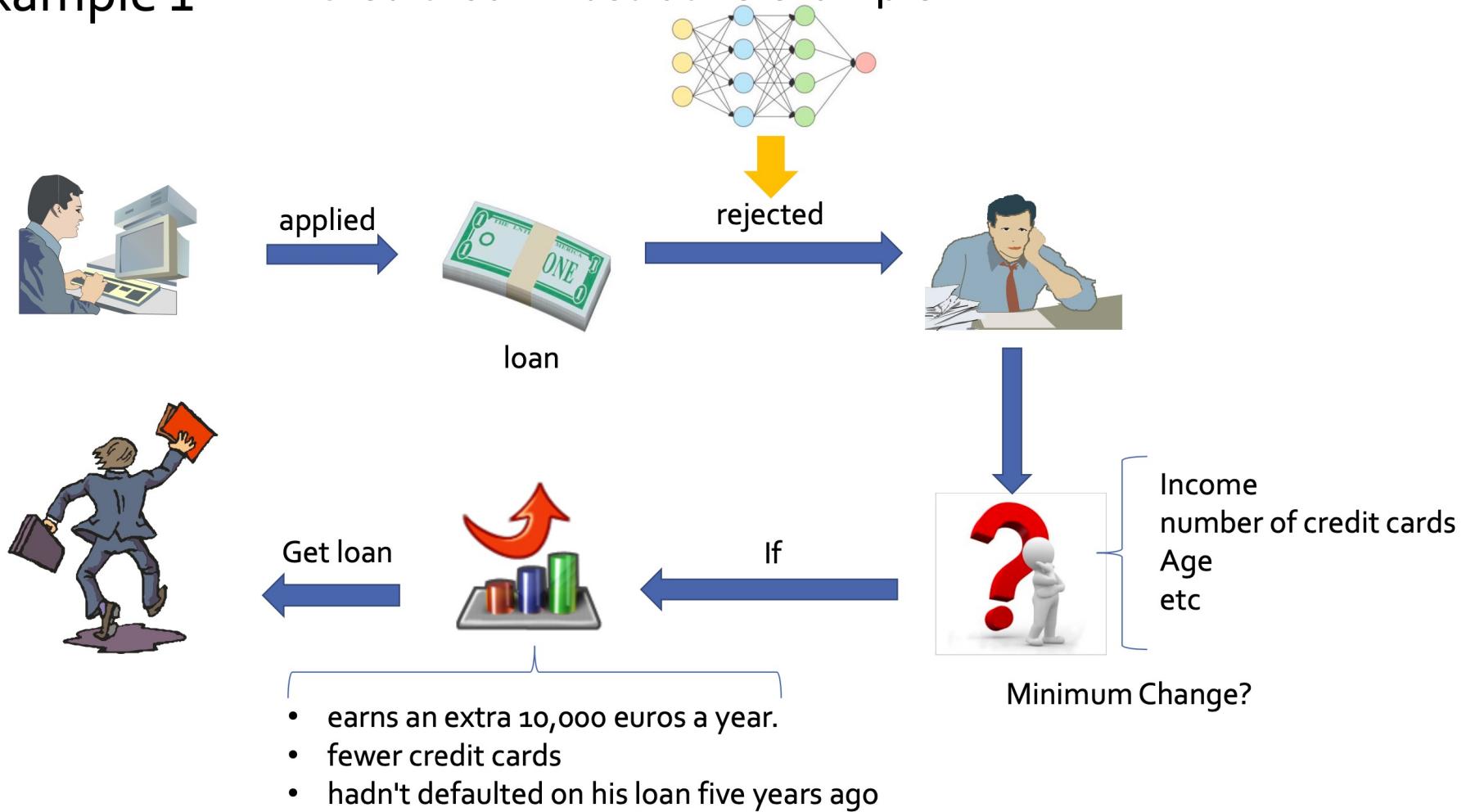
Prototype	Most similar phrase	Similarity * Weight
the best filmgoing experiences i is a wonderful film full lots of great comedy from	occasionally something enjoyable would happen the first ninety minutes so my patience was eventually rewarded	$0.60 * 0.97 = 0.58$ $0.61 * 0.78 = 0.48$ $0.82 * 0.45 = 0.36$

Sum of evidence: **2.01**

# Kontrfakty

## Example 1

### Credit loan illustrative example



Molnar, C. (2020). Interpretable machine learning. Lulu. com.

## Example 2 : apartment sale offers

# Wyjaśnienia z kontrfaktami

- **Counterfactual explanations** – opisują zależności przyczynowo skutkowe pomiędzy zmianą przesłanki, a zmianą decyzji,  
“if A had not occurred, then B would not have occurred” [Dandl et al. ]
- W ML dla przykładu  $(x,y)$  najmniejsze dopuszczalne zmiany  $x$ , które doprowadzą do (pożąданej zmiany predykcji  $b(x') \neq y$ )

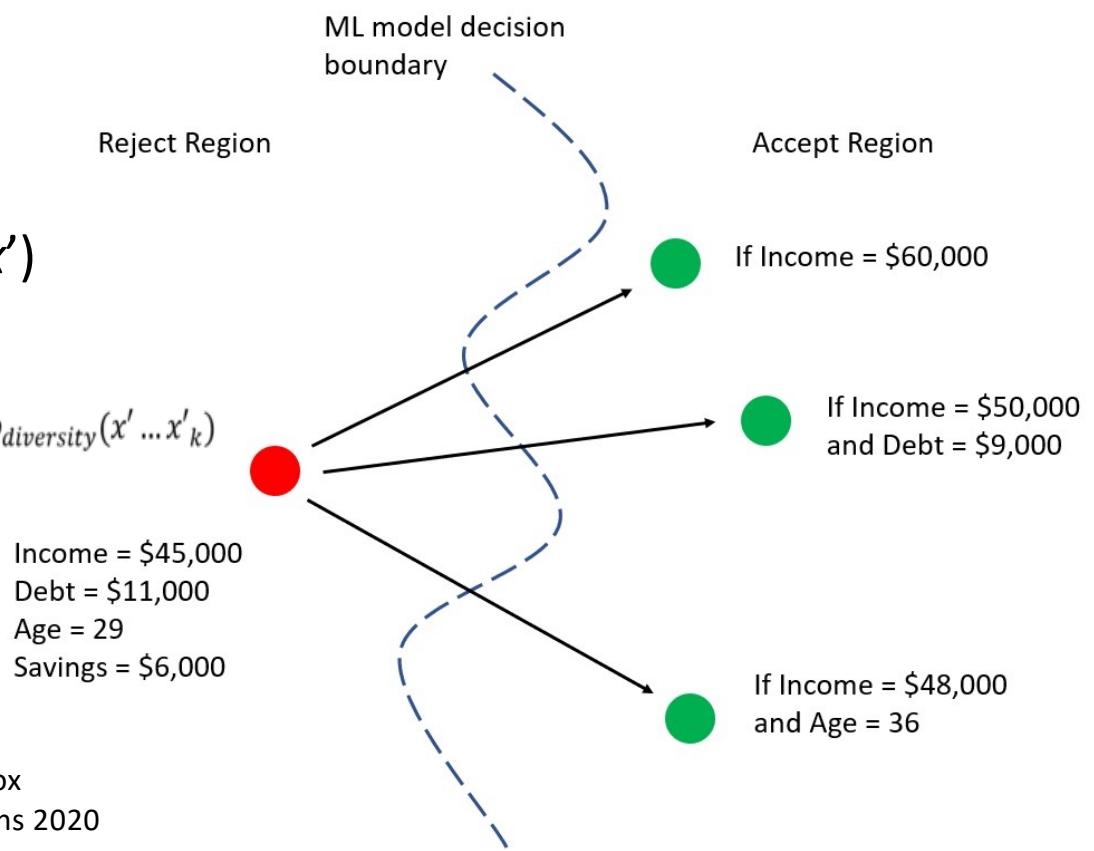
Przykład decyzji kredytowej

- Dopuszczalne atrybuty
- Optym. funkcji straty [Wachler]

$$C(x) = \arg \min L(b(x'), y) + \lambda d(x, x')$$

Bardziej złożone [DICE] – wiele C

$$C(x) = \arg \min_{x'_1 \dots x'_k} \frac{1}{k} \sum_{i=1}^k \mathcal{L}(f(x'_i), y) + \frac{\lambda_1}{k} d(x'_i, x) - \lambda_2 \cdot dpp_{diversity}(x' \dots x'_k)$$



Wachter, S., Mittelstadt, B., and Russell, C. (2017)

Counterfactual explanations without opening the black box

Dandl, S. et al. AI Multi-Objective Counterfactual Explanations 2020

# Uwagi podsumowujące

- XAI i interpretowalne ML – motywowane praktycznymi problemami i nowymi zastosowaniami
- Część rozwiązań wykorzystuje inspiracje z dawnych propozycji w ML i innych dziedzinach nauki
- Otwarte pytana badawcze : czy rozwijać metody post-hoc czy explainable by design?
- Zrozumienie wyniku wielu metod – wymaga jednak przygotowania i wiedzy dziedzinowej
- Miary i techniki oceny – ciągle poszukujemy

# Poczytaj więcej

Książki:

- Christoph Molnar, “Interpretable Machine Learning: A Guide for making black box models explainable”
- Przemysław Biecek, Explanatory Model Analysis (book under preparation)

Artykuły:

- Bibal A., Frenay B.: Interpretability of machine learning models and representations: an introduction. W: Proceedings of ESANN 2016
- Bratko I.: Machine learning: between accuracy and interpretability.
- Freitas A.: Comprehensible classification models: a position paper. ACM SIGKDD Exploration Newsletter, Vol. 15, Nr 1, 2014
- Guidotti R, Monreale A., Ruggieri S, Turini F., Giannotti F, Pedreschi D.: A Survey of Methods for Explaining Black Box Models, ACM Comput. Surv., 2018
- Ribeiro M. T., Singh S., Guestrin C.: Why should i trust you? Explaining the predictions of any classifier, W: Proc. of the 22nd ACM SIGKDD 2015
- Samek, W., Wiegand, T., Müller, K. R.: Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. arXiv preprint arXiv:1708.08296, 2017.
- Stefanowski J. Woźniak M.: Interpretacja modeli uczonych ze złożonych danych medycznych. W Informatyka w medycynie 2019.

# Pytanie i komentarze?

Dalszy kontakt:

[jerzy.stefanowski@cs.put.poznan.pl](mailto:jerzy.stefanowski@cs.put.poznan.pl)

<http://www.cs.put.poznan.pl/jstefanowski/>



Fundusze  
Europejskie  
Polska Cyfrowa



Rzeczpospolita  
Polska

Unia Europejska  
Europejski Fundusz  
Rozwoju Regionalnego

