Uczenie maszynowe

ZADANIE KLASYFIKACJI CZ. 3

Klasyfikatory regułowe

- Zbiór reguł IF-THEN

if(warunek) → *then {wniosek};*

w klasyfikacji *if(warunek dotyczący danych)* → *then {klasa}*

- Różne zasady tworzenia reguł – pokrywanie zbioru uczącego (wzajemnie wykluczające się, wyczerpujące…)
- Działanie na zmiennych dyskretnych (najczęściej)

Klasyfikatory regułowe - przykład

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
human	warm	yes	no	no	mammals
python	cold	no	no	no	reptiles
salmon	cold	no	no	yes	fishes
whale	warm	yes	no	yes	mammals
frog	cold	no	no	sometimes	amphibians
komodo	cold	no	no	no	reptiles
bat	warm	yes	yes	no	mammals
pigeon	warm	no	yes	no	birds
cat	warm	yes	no	no	mammals
leopard shark	cold	yes	no	yes	fishes
turtle	cold	no	no	sometimes	reptiles
penguin	warm	no	no	sometimes	birds
porcupine	warm	yes	no	no	mammals
eel	cold	no	no	yes	fishes
salamander	cold	no	no	sometimes	amphibians
gila monster	cold	no	no	no	reptiles
platypus	warm	no	no	no	mammals
owl	warm	no	yes	no	birds
dolphin	warm	yes	no	yes	mammals
eagle	warm	no	yes	no	birds

R1: (Give Birth = no) $\wedge$ (Can Fly = yes) $\rightarrow$ Birds

R2: (Give Birth = no) $\wedge$ (Live in Water = yes) $\rightarrow$ Fishes

R3: (Give Birth = yes) $\wedge$ (Blood Type = warm) $\rightarrow$ Mammals

R4: (Give Birth = no) $\wedge$ (Can Fly = no) $\rightarrow$ Reptiles

R5: (Live in Water = sometimes) $\rightarrow$ Amphibians

http://staffwww.itn.liu.se/~aidvi/courses/06/dm/lectures/lec4.pdf
http://www.cs.uml.edu/~cchen/544-F15/slides/ch5.pdf

Klasyfikatory regułowe - przykład

R1: (Give Birth = no) $\wedge$ (Can Fly = yes) $\rightarrow$ Birds

R2: (Give Birth = no) $\wedge$ (Live in Water = yes) $\rightarrow$ Fishes

R3: (Give Birth = yes) $\wedge$ (Blood Type = warm) $\rightarrow$ Mammals

R4: (Give Birth = no) $\wedge$ (Can Fly = no) $\rightarrow$ Reptiles

R5: (Live in Water = sometimes) $\rightarrow$ Amphibians

Lemur ?
Turtle?
Dogfish shark?

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
lemur	warm	yes	no	no	?
turtle	cold	no	no	sometimes	?
dogfish shark	cold	yes	no	yes	?

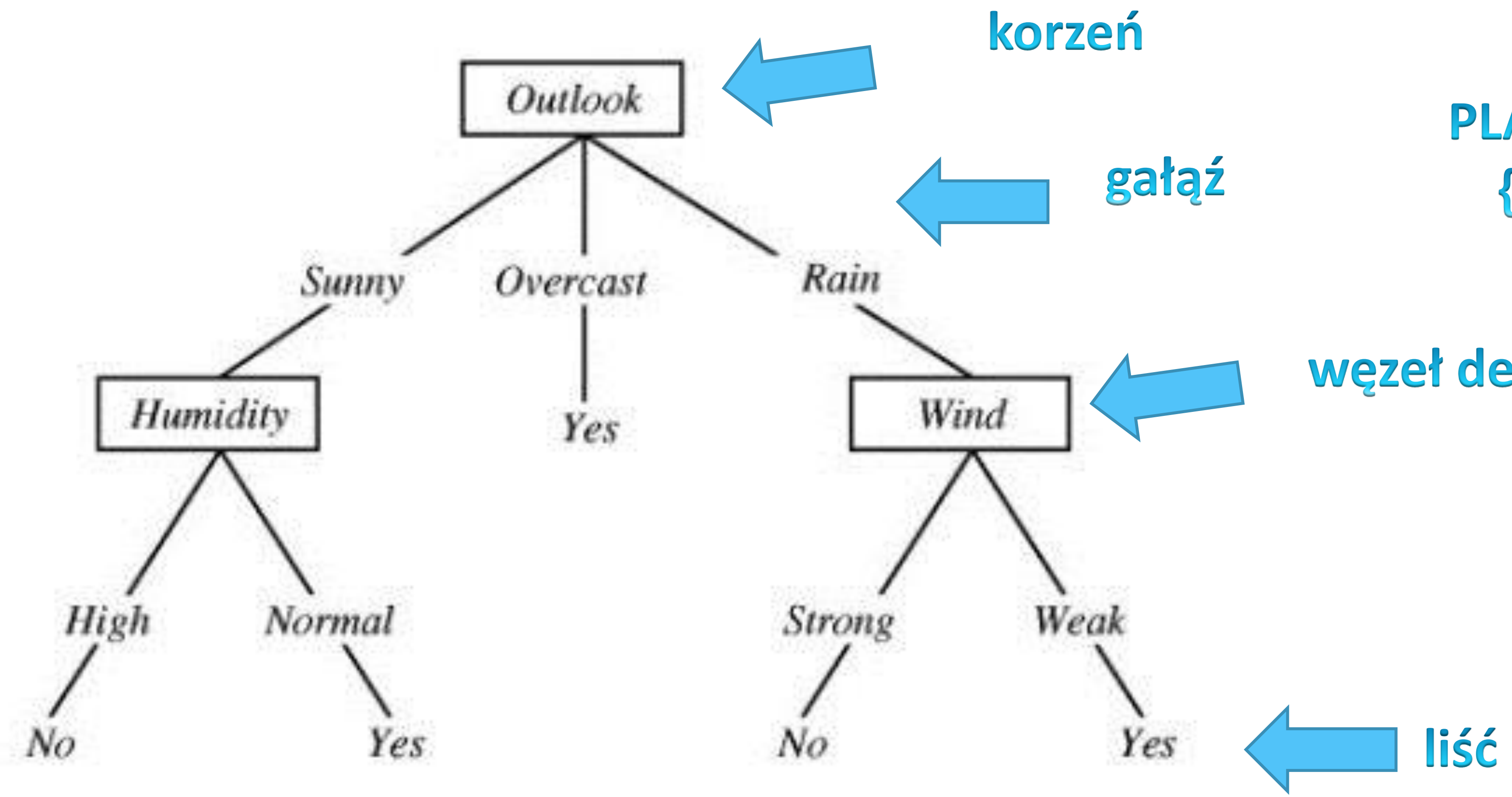
A **lemur** triggers rule R3 $\Rightarrow$ class = *mammal*

A **turtle** triggers both R4 and R5 $\rightarrow$ voting or ordering rules

A **dogfish shark** triggers none of the rules $\rightarrow$ default rule

Mammal
R4/R5 (głosowanie, priorytet reguł)
None (domyślna reguła)

Drzewo decyzyjne - budowa



PLAY TENNIS?
{YES, NO}

Rysunek: https://nullpointerexception1.wordpress.com/2017/12/16/a-tutorial-to-understand-decision-tree-id3-learning-algorithm/

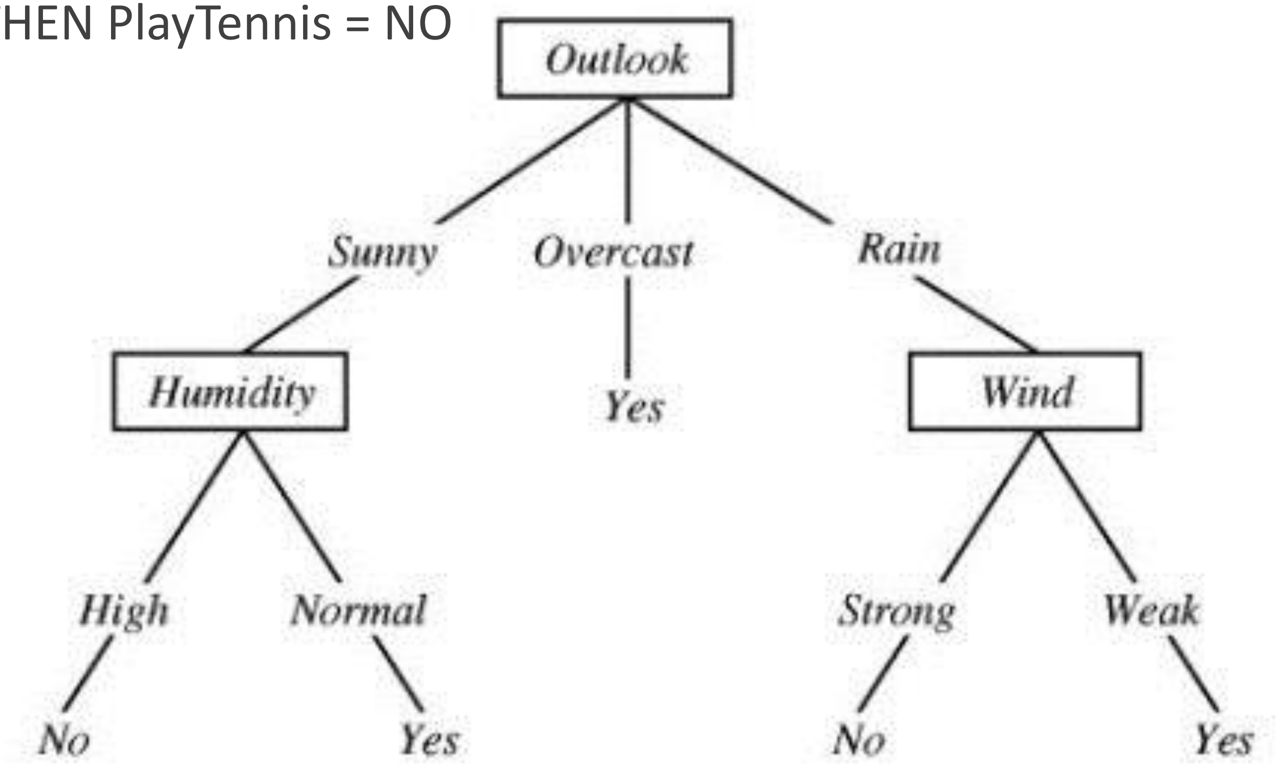
Reguły drzewa

Reguły dla lewej strony drzewa:

1. IF Outlook is Sunny AND Humidity is High THEN PlayTennis = NO

2. IF Outlook is Sunny AND
Humidity is Normal THEN PlayTennis = YES

....



Rysunek: https://nullpointerexception1.wordpress.com/2017/12/16/a-tutorial-to-understand-decision-tree-id3-learning-algorithm/

Algorytm C4.5

- Nie jest ograniczony do binarnych podziałów
- Dla zmiennych jakościowych algorytm tworzy osobne gałęzie dla każdej wartości atrybutu jakościowego („rozgałęzienie”)
- Metoda mierzenia jednorodności algorytmu opiera się na zysku informacyjnym (do wyboru optymalnego podziału)

Entropia $H(X) = -\sum_j p_j \, log_2(p_j)$

- Sortowanie danych w każdym węźle drzewa w celu znalezienia atrybutu najlepiej dzielącego zbiór. Algorytm C4.5 wybiera w każdym węźle drzewa atrybut na podstawie kryterium - znormalizowanego zysku informacyjnego (różnicy w entropii), który wynika z wyboru atrybutu dzielącego dane. Atrybut o najwyższym zysku informacyjnym jest wybierany do węzła.

Algorytm C4.5

Algorytm C4.5 wykorzystuje zysk informacyjny (ang. *information gain*) do wyboru optymalnego podziału S zbioru uczącego na T podzbiorów:

$$zysk(S) = H(T) - H_s(T), \tag{C.1}$$

gdzie $H(T)$ to entropia zbioru przed podziałem, a $H_S(T)$ opisuje ważoną sumę entropii dla pojedynczych podzbiorów:

$$H_S(T) = \sum_{i=1}^{k} P_i H_S(T_i), \tag{C.2}$$

gdzie P_i oznacza procent przypadków w i-tym podzbiorze. W każdym węźle algorytm wybiera podział optymalny - o największym zysku informacyjnym.

Algorytm C4.5

-

- Obliczenie zysku informacyjnego danego atrybutu, uwzględniając jego istniejące wartości.
- Możliwość wykorzystania danych ciągłych:

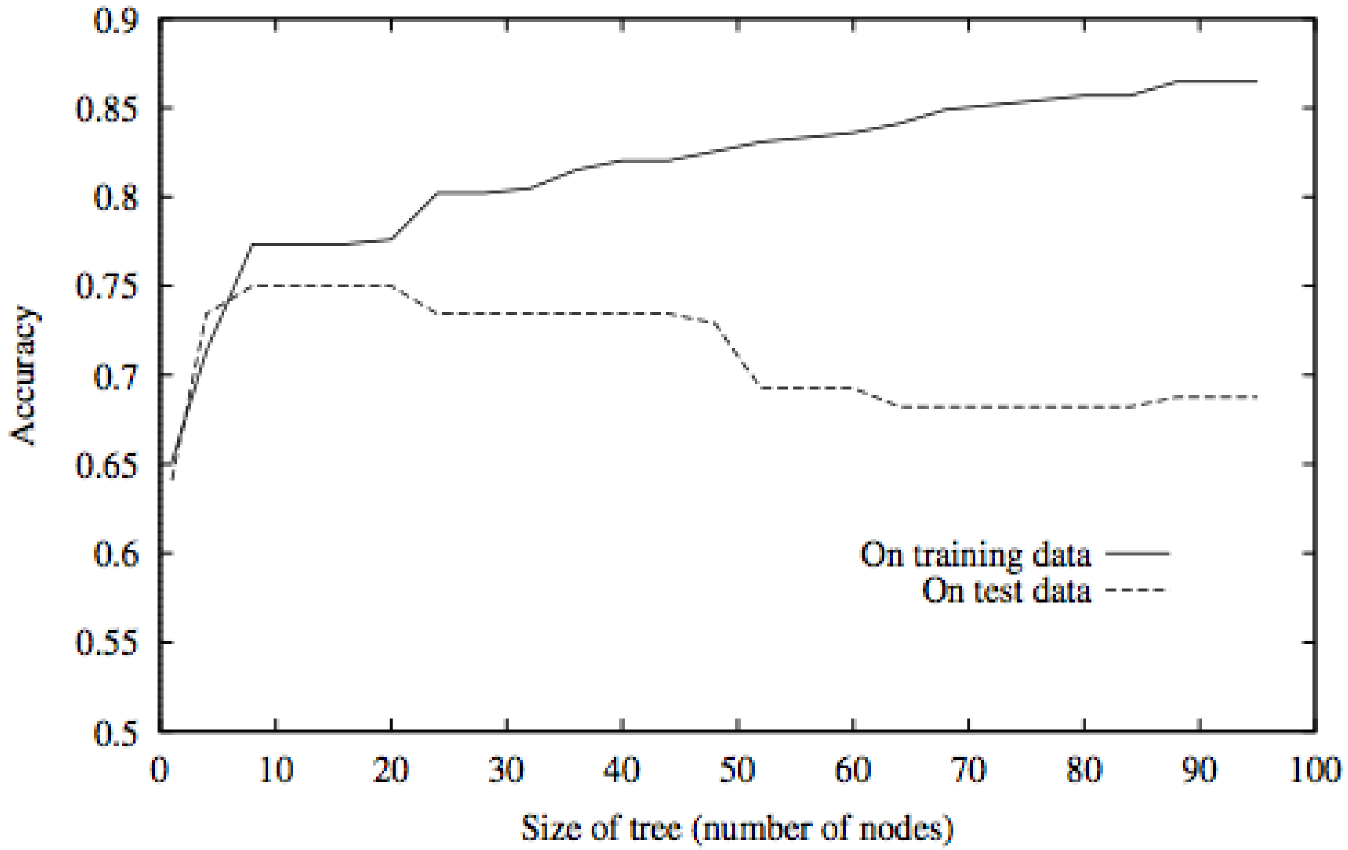
1. Sortowanie wartości atrybutu rosnąco
2. Dla każdej wartości wyznaczenie liczby elementów większych/mniejszych od tej wartości
3. Obliczenie dla każdego z podziałów zysku/zysku informacyjnego i wybranie tego o najwyższej wartości.

Algorytm C4.5 - *overfitting*

- Możliwość nadmiernego dopasowania do danych (tzw. przeuczenie),
- Zapobieganie: zmniejszenie rozmiaru drzewa - przycięcia (ang. *pruning*):

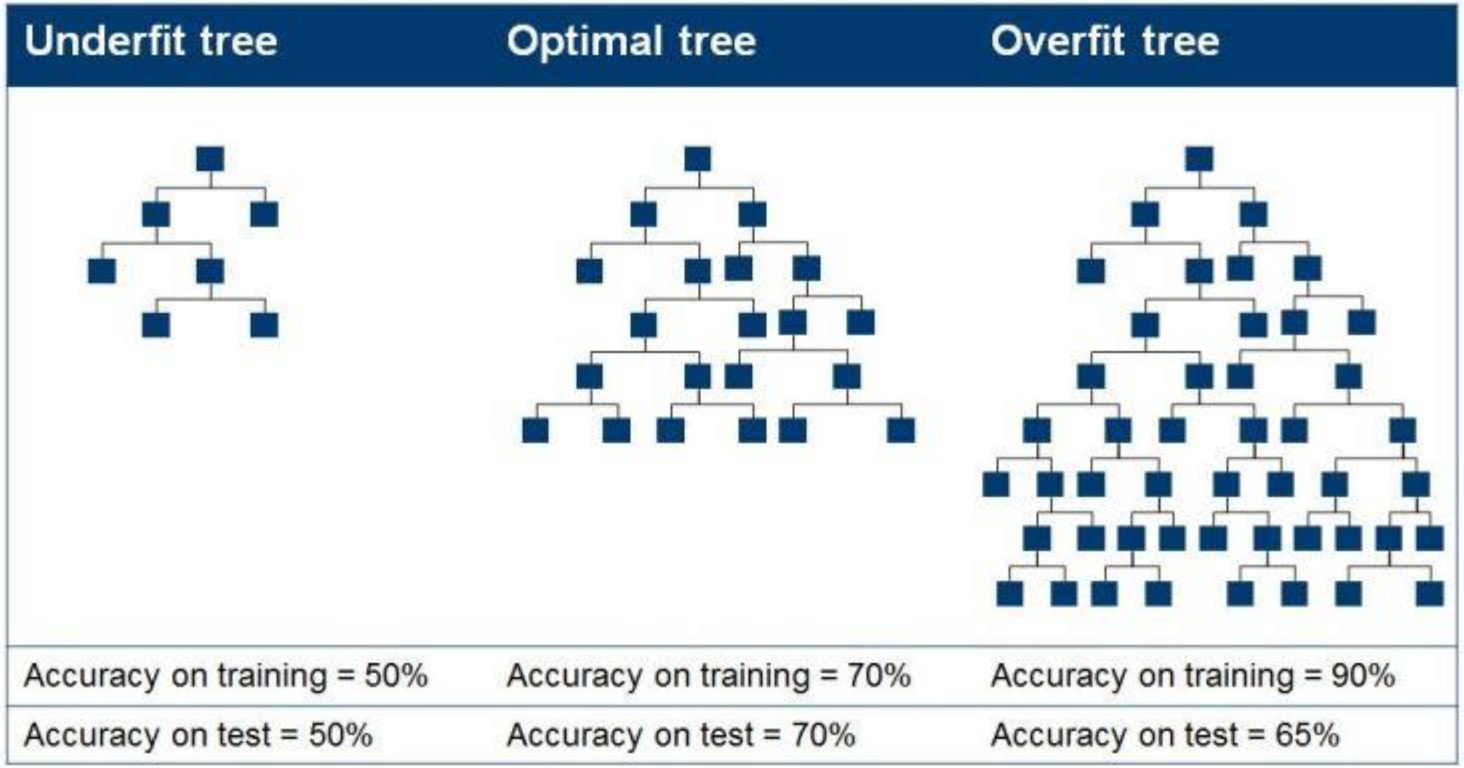
1. Zmniejszenie złożoności klasyfikatora
2. Oparcie algorytmu przycinania na pesymistycznym oszacowaniu liczby błędów E w zbiorze N, nienależących do klas najczęściej występujących; obliczenie górnej granicy prawdopodobieństwa (rozkład binominalny) dla *E* zdarzeń zaobserwowanych w *N* próbach z zadanym poziomem ufności (standardowo 0.25)
3. Rozpoczęcie przycinania od liści – obliczenie - dla każdego z mniejszych drzew szacowanego błędu, gdyby przekształcono je w liść (przycięcie w warunku niezwiększającego się błędu)

Przeuczenie (*overfitting*)



Rysunek: http://wiki.bethanycrane.com/overfitting-of-data

„Przeuczone” drzewo decyzyjne



Rysunek: https://www.analyticbridge.datasciencecentral.com/profiles/blogs/making-data-science-accessible-machine-learning-tree-methods

Zadanie – zbiór PlayTennis

Entropia NO i YES

$$H_{No} = -\frac{5}{14} log_2\left(\frac{5}{14}\right); H_{Yes} = -\frac{9}{14} log_2\left(\frac{9}{14}\right)$$

Całkowita entropia

$$H = \ H_{No} + H_{Yes}$$

Entropia dla Outlook = Sunny (2 YES, 3 NO)

$$H_{Sunny} = -\frac{2}{5} log_2\left(\frac{2}{5}\right) - -\frac{3}{5} log_2\left(\frac{3}{5}\right)$$

Entropia dla Outlook = Overcast (4 YES, 0 NO)

$$H_{Overcast} = -\frac{4}{4} log_2\left(\frac{4}{4}\right) - -\frac{0}{4} log_2\left(\frac{0}{4}\right)$$

Outlook	Temperature	Humidity	Windy	PlayTennis
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
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

Rysunek: https://nullpointerexception1.wordpress.com/2017/12/16/a-tutorial-to-understand-decision-tree-id3-learning-algorithm/

Zadanie – zbiór PlayTennis

Entropia dla Outlook = Rainy (2 YES, 3 NO)

$H_{Rainy} = H_{Sunny}$

Całkowita entropia gałęzi

$H_s = H_{Sunny} + H_{Overcast} + H_{Rainy}$

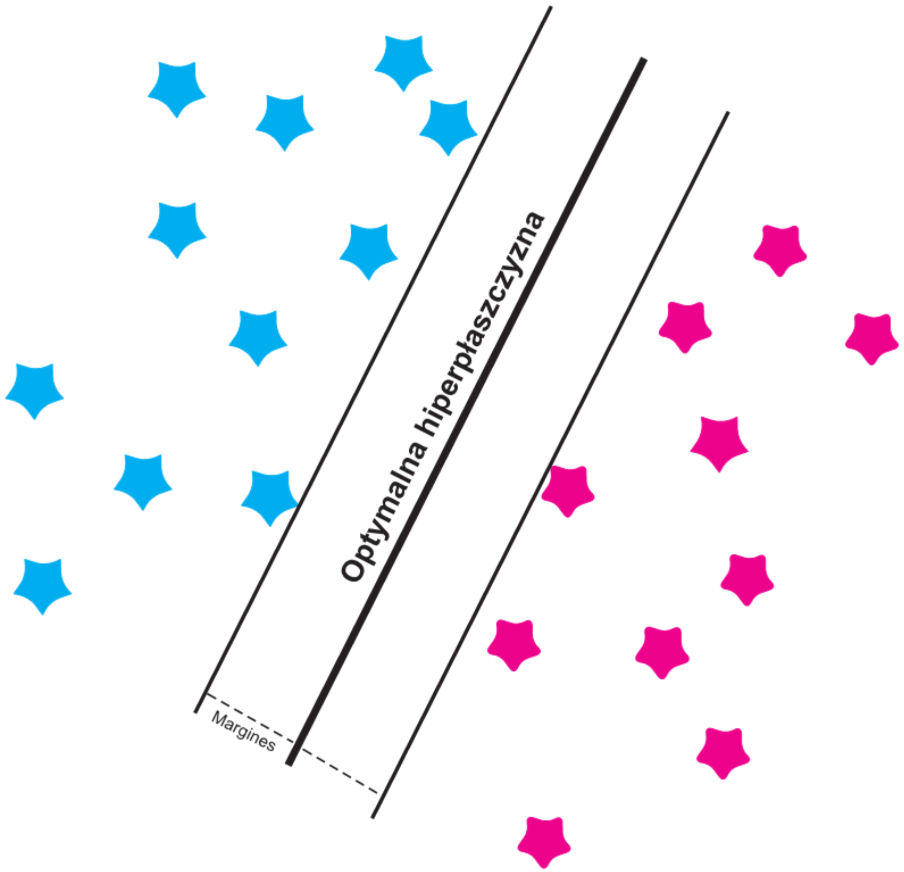
Redukcja niepewności – zysk informacyjny

$zysk = H - H_s$

Outlook	Temperature	Humidity	Windy	PlayTennis
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
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

Support Vector Machines - SVM

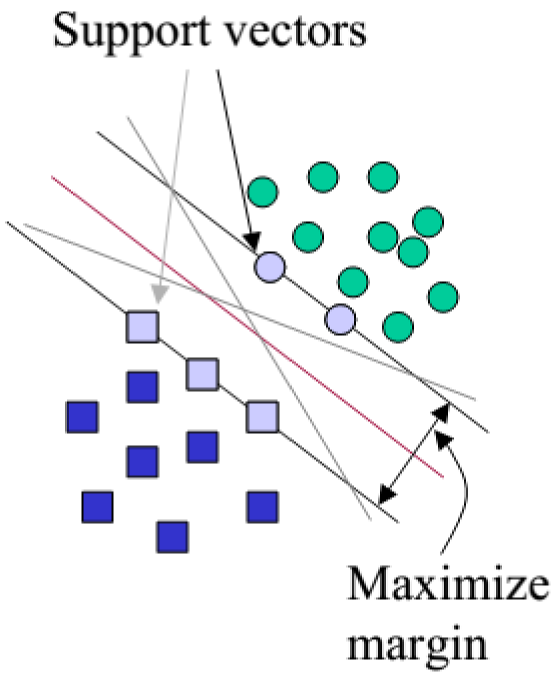
- Polskie tłumaczenie maszyna wektorów/podpierających
- Klasyfikator binarny oparty o optymalizację (odnalezienie hiperpłaszczyzny separującej liniowo)
- Zastosowanie jąder do transformowania nieliniowych problemów do liniowych → *kernel trick*
- Margines zaufania – odległość między optymalną hiperpłaszczyzną a najbliższym jej wektorem nośnym



Rysunek: http://fizyka.umk.pl/publications/kmk/99phd-nj.pdf

Wektory nośne

- Wektory nośne to punkty leżące najbliżej płaszczyzny decyzyjnej
- Najtrudniejsze do sklasyfikowania
- Maksymalizacja marginesu wokół hiperpłaszczyzny



Rysunek: http://lcsd2.svms.org/tutorials/Berwick2003.pdf

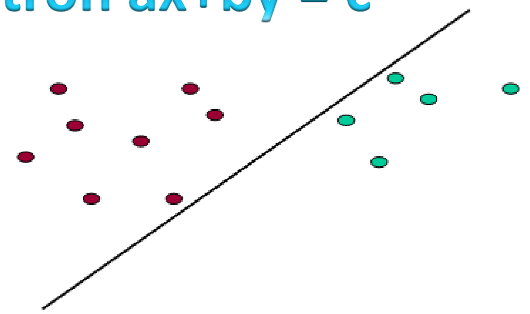
Separacja przez hiperpłaszczyzny

- Liniowa separowalność:

-2 wymiary → linia

- więcej wymiarów → hiperpłaszczyzna

Perceptron ax+by = c

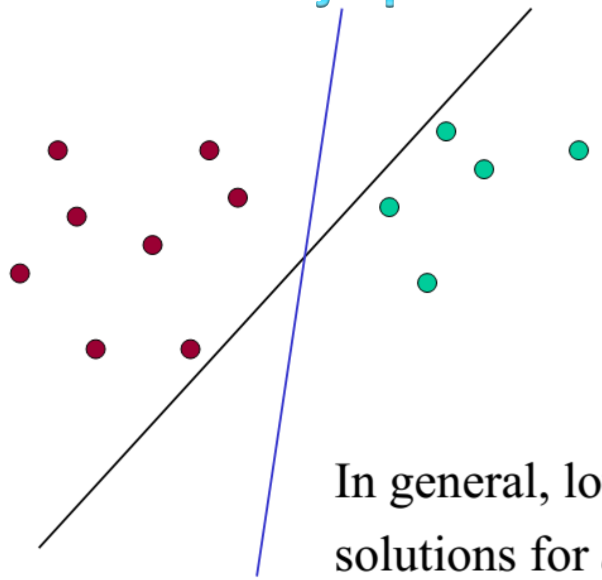


Find a,b,c, such that

$ax + by \geq c$ for red points

$ax + by \leq c$ for green points

Która najlepsza??



In general, lots of possible solutions for *a,b,c*.

Rysunek: http://lcsd2.svms.org/tutorials/Berwick2003.pdf

Liniowa separowalność

- Dwie klasy są liniowo separowalne, jeśli istnieje hiperpłaszczyzna H postaci g(**x**)

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

- przyjmująca wartości

$$\begin{cases} g(\mathbf{x}_i) > 0 & \mathbf{x}_i \in 1 \\ g(\mathbf{x}_i) < 0 & \mathbf{x}_i \in -1 \end{cases}$$

Wybór hiperpłaszczyzny

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

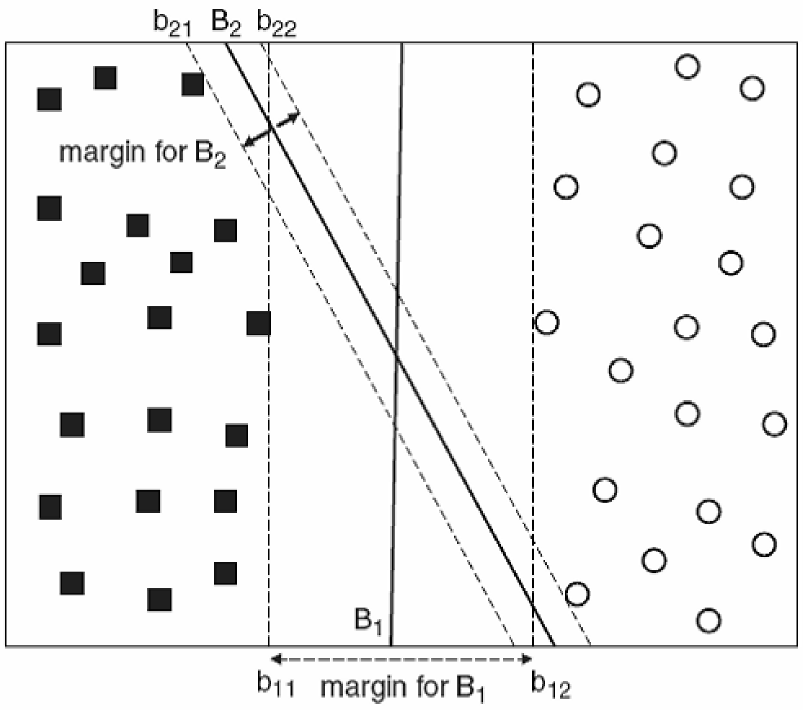
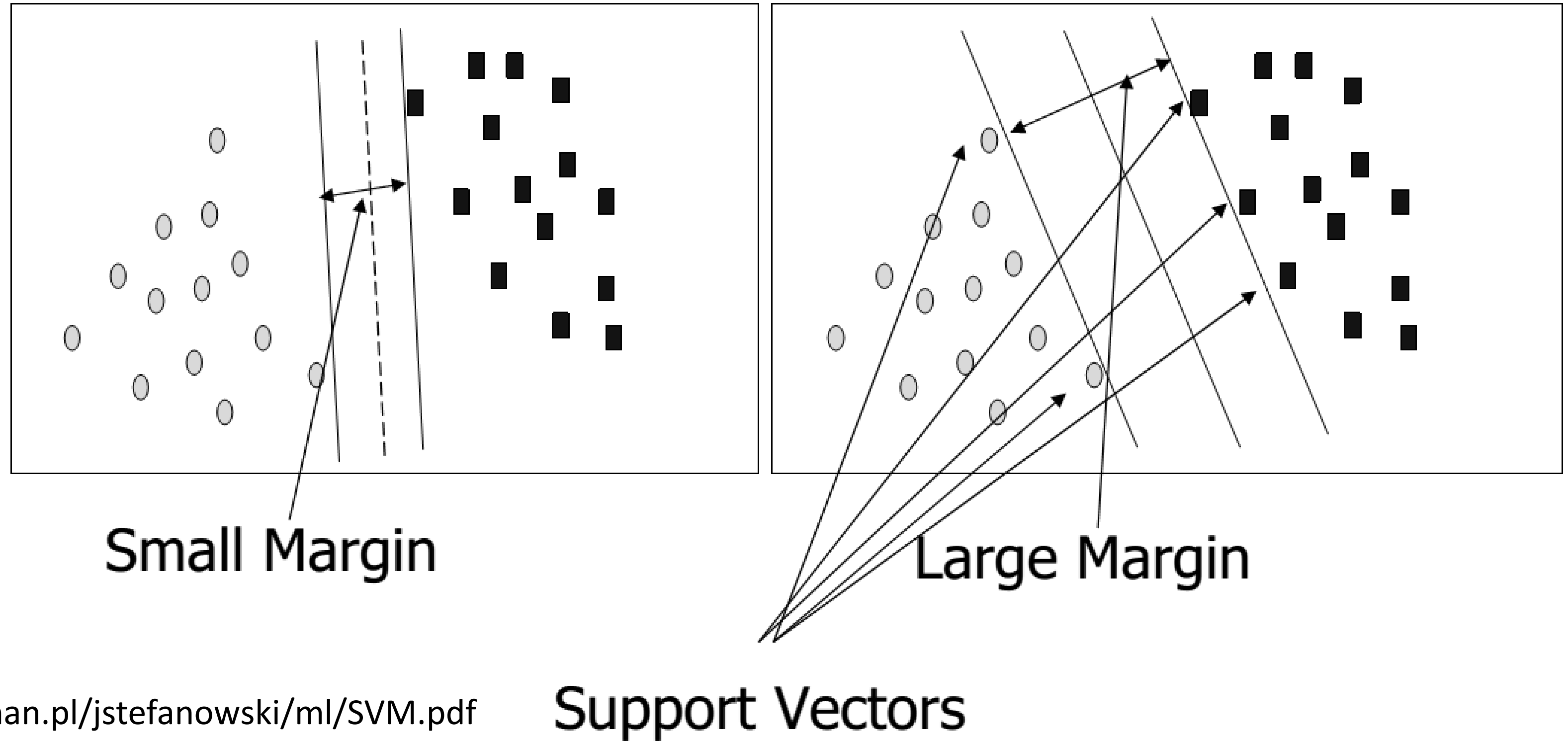


Figure 5.22. Margin of a decision boundary.

Margines – węższy czy szerszy?

- Szerszy margines $\rightarrow$ lepsze własności generalizacji, mniejsza podatność na ew. przeuczenie (overfitting)
- Wąski margines – mała zmiana granicy, radykalne zmiany klasyfikacji



http://www.cs.put.poznan.pl/jstefanowski/ml/SVM.pdf

Działanie SVM

- Vapnik – poszukuj „maximal margin classifier”

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

  gdzie **w** i **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 b_{i1} i b_{i2} były miejscem geometrycznym punktów **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}

http://www.cs.put.poznan.pl/jstefanowski/ml/SVM.pdf

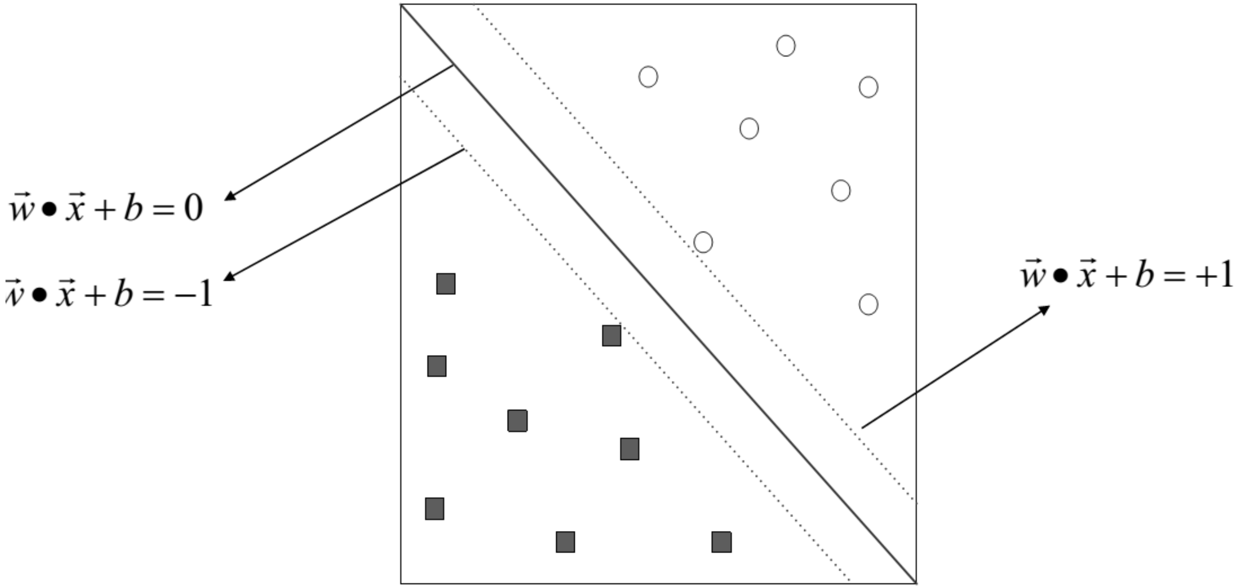
Poszukiwanie parametrów hiperpłaszczyzny

$$\mathbf{margin} = \frac{2}{\|\mathbf{w}\|}$$

$$\|\mathbf{w}\| \equiv \sqrt{w_1^2 + \ldots + w_p^2}$$

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

maximize **minimize** **minimize**



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

Sformułowanie mat. problemu:

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

- Przy warunkach ograniczających

$$y_i(w \cdot x_i + b) \geq 1 \quad i = 1,2,\ldots,N$$

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

http://www.cs.put.poznan.pl/jstefanowski/ml/SVM.pdf

LSVM

- Minimalizuj funkcję Lagrange'a

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

- parametry $\alpha \geq 0$ mnożniki Lagrange'a
- Powinno się różniczkować L po w i b – nadal trudności w rozwiązaniu

- Nadal zbyt wiele parametrów w,b,α do oszacowania
- Przechodzi się na postać dualną zadania optymalizacji
- Maksymalizuj L(α)

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

- Przy ograniczeniach

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

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

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

- Hiperpłaszczyzna decyzyjna

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

SVM - klasyfikacja

- Klasyfikacja – funkcja decyzyjna

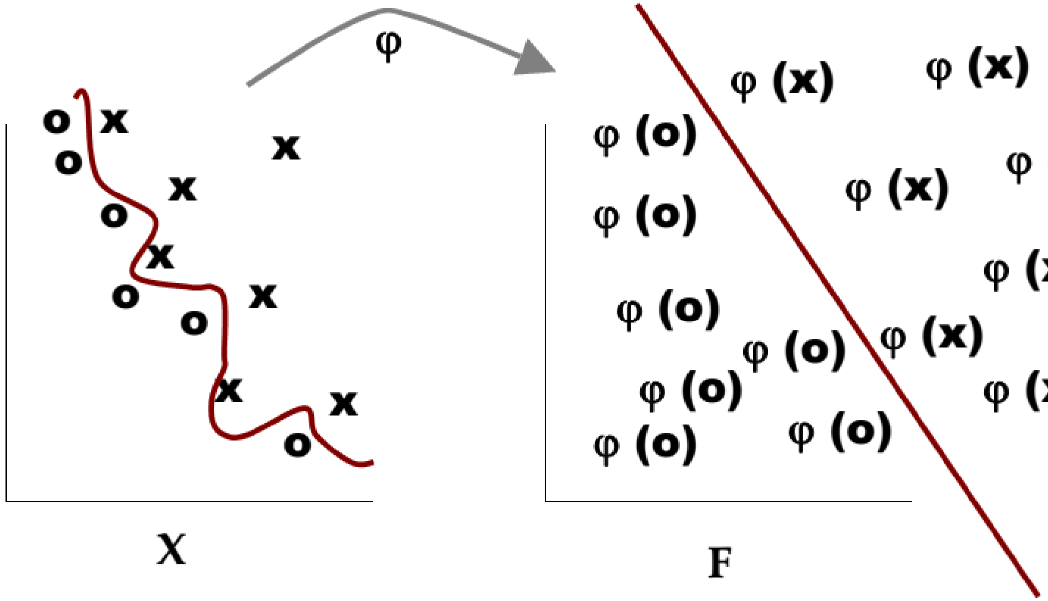
$$f(x) = sign(\sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b)$$

- O ostatecznej postaci hiperpłaszczyzny decydują wyłącznie wektory nośne (α_i >0)
- Im większa wartość α_i tym większy wpływ wektora na granicę decyzyjną
- Klasyfikacja zależy od iloczynu skalarnego nowego **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 treningowych (wektory nośne)

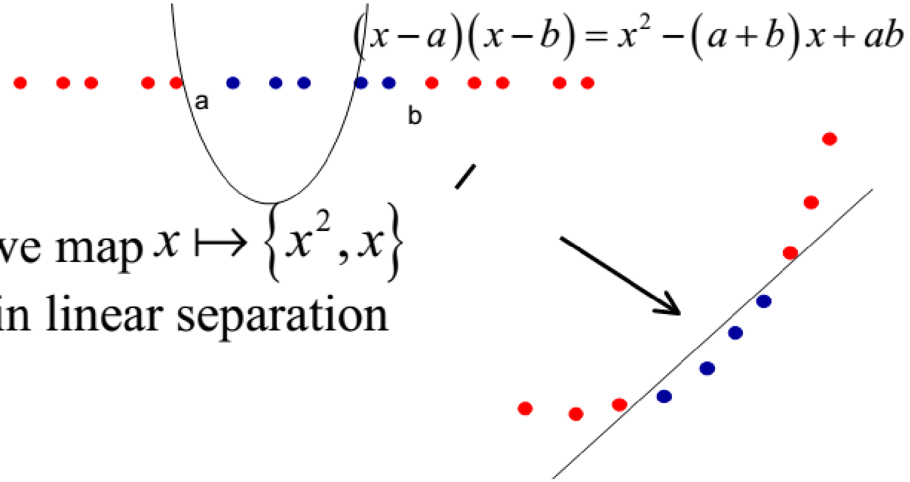
http://www.cs.put.poznan.pl/jstefanowski/ml/SVM.pdf

Problem separowalny nieliniowo

- Mapowanie danych do przestrzeni wielowymiarowych



– The following set can't be separated by a linear function, but can be separated by a quadratic one



– So if we map $x \mapsto \{x^2, x\}$ we gain linear separation

Rysunek: http://lcsd2.svms.org/tutorials/Berwick2003.pdf

Funkcje jądrowe (*kernel functions*)

Przykład prostego przekształcenia wielomianowego

The kernel trick:

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

$$=\ \Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)$$

Original optimization function:

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

Nie musimy znać funkcji Φ, wystarczy znać jądro (kernel)
i można pracować w nowej przestrzeni

Dopuszczalne typy jąder związane z SVM

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

Wykorzystanie funkcji jądrowych

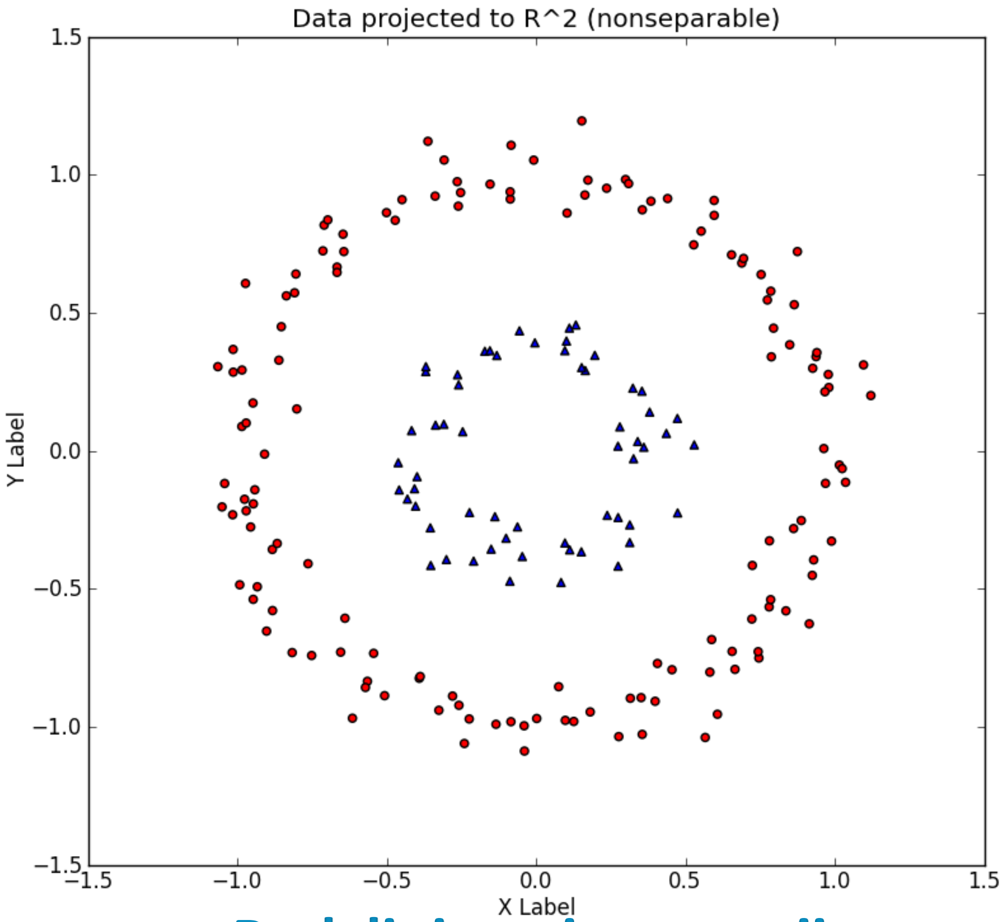
$$f(\mathbf{x})=\quad sign\left(\sum_{i=1}^{N}\alpha_i y_i\Phi(\mathbf{x}_i)\Phi(\mathbf{x})+b\right)$$

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

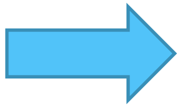
Model klasyfikacji binarnej rozszerza się na zagadnienie wieloklasowe K > 2

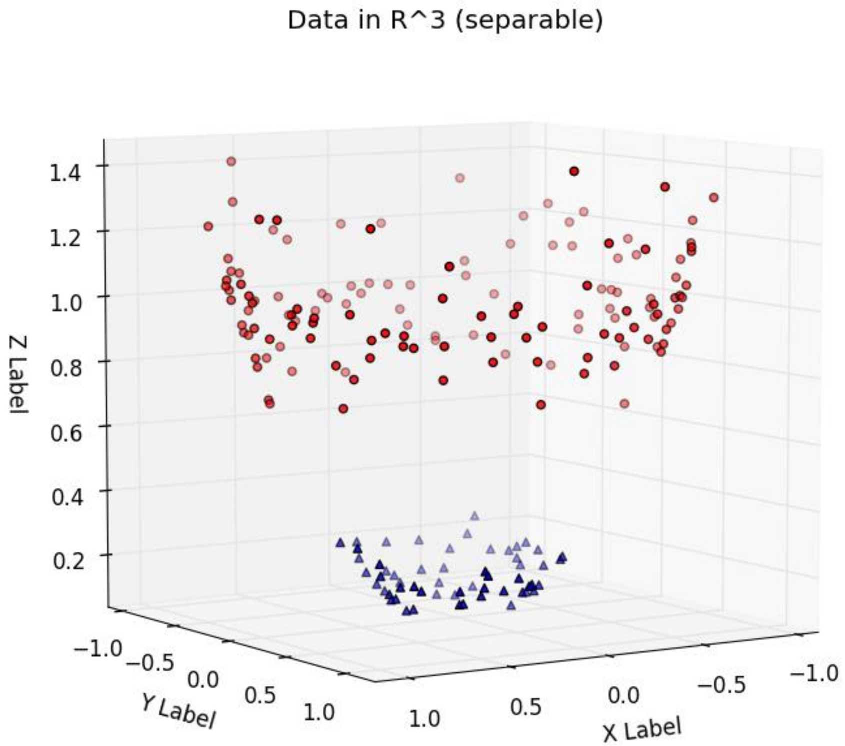
http://www.cs.put.poznan.pl/jstefanowski/ml/SVM.pdf

Kernel trick



Brak liniowej separacji





Liniowa separacja grup

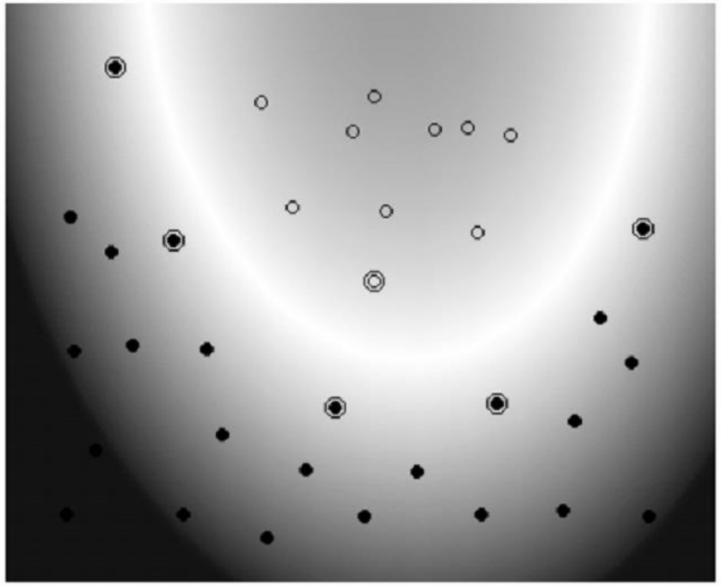
http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html

Kernel 1

Example: SVM with Polynomial of Degree 2

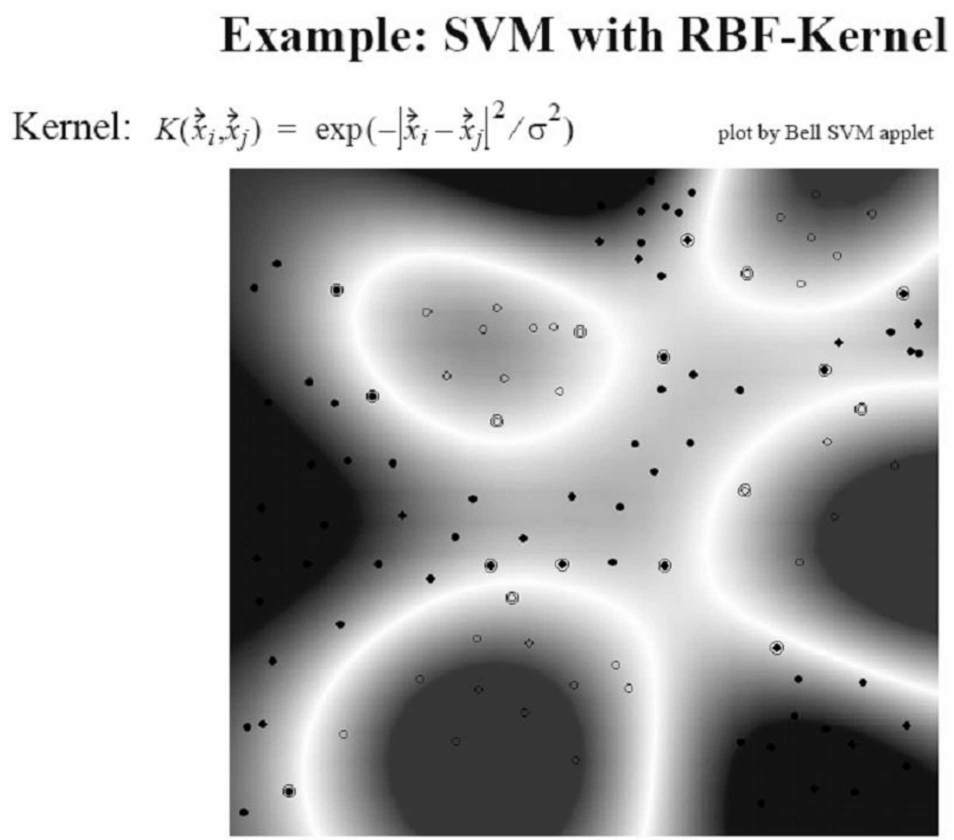
Kernel: $K(\vec{x}_i,\vec{x}_j) = [\vec{x}_i \cdot \vec{x}_j + 1]^2$

plot by Bell SVM applet



http://www.cs.put.poznan.pl/jstefanowski/ml/SVM.pdf

Kernel 2



http://www.cs.put.poznan.pl/jstefanowski/ml/SVM.pdf

SVM - zalety

Stopień skomplikowania/pojemność jest niezależna od liczby wymiarów.

Bardzo dobra podbudowa statystyczno-teoretyczna

Znajdowanie minimum glonalnego. Minimalizujemy funkcję kwadratową co gwarantuje zawsze znalezienie minimum. Algorytm jest wydajny i SVM generuje prawie optymalny klasyfikator. Nie jest tez czuły na przetrenowanie.

Dobre uogólnianie dzięki wielowymiarowej “feature space”.

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

SVM - wady

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

Rozwiązania też są skomplikowane (normalnie >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 używał 2 ukrytych węzłów, a SVM 285 wektorów.*

Trudno dodać własną wiedzę (prior knowledge)

http://www.cs.put.poznan.pl/jstefanowski/ml/SVM.pdf