



SVM

Wprowadzenie

- ▶ Support vector machines (maszyny wektorów wspierających, maszyny wektorów nośnych)
- ▶ SVM służy do:
 - ▶ w wersji podstawowej – klasyfikacji binarnej
 - ▶ w wersji z rozszerzeniami
 - ▶ regresji
 - ▶ wyboru najważniejszych cech
 - ▶ określania przypadków odstających
 - ▶ grupowania
- ▶ Zalety metody SVM
 - ▶ budowa prostych i skomplikowanych modeli
 - ▶ radzi sobie z małą liczbą przykładów uczących
 - ▶ radzi sobie z dużą liczbą atrybutów

Założenia

- ▶ Przykłady są opisane przez wektory
- ▶ Niech początek wektora będzie w początku układu współrzędnych
- ▶ Niech koniec wektora będzie w punkcie wyznaczonym przez wartości atrybutów dla reprezentowanego przez niego przykładu

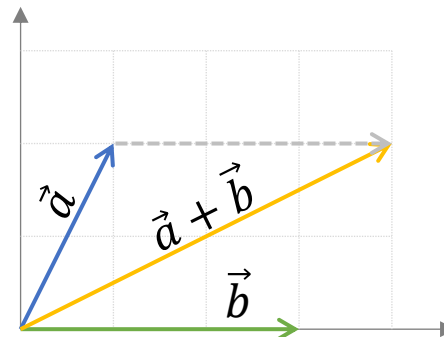
Rachunek wektorowy – przypomnienie (1)

► Dodawanie wektorów

$$\vec{a} = (a_1, a_2, \dots, a_n)$$

$$\vec{b} = (b_1, b_2, \dots, b_n)$$

$$\vec{a} + \vec{b} = (a_1 + b_1, a_2 + b_2, \dots, a_n + b_n)$$

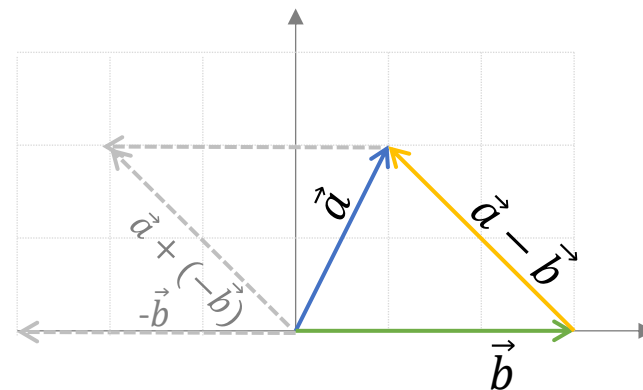


► Odejmowanie wektorów

$$\vec{a} = (a_1, a_2, \dots, a_n)$$

$$\vec{b} = (b_1, b_2, \dots, b_n)$$

$$\vec{a} - \vec{b} = (a_1 - b_1, a_2 - b_2, \dots, a_n - b_n)$$



Rachunek wektorowy - przypomnienie (2)

- Mnożenie wektora przez skalar

$$\vec{a} = (a_1, a_2, \dots, a_n)$$
$$c\vec{a} = (ca_1, ca_2, \dots, ca_n)$$

„rozciąga wektor” ale nie zmienia kierunku (zwrot zależy od znaku c)

- Długość euklidesowa

$$\|\vec{a}\|_2 = \|\vec{a}\| = \sqrt{a_1^2 + a_2^2 + \dots + a_n^2}$$

Rachunek wektorowy – przypomnienie (3)

- Iloczyn skalarny (ang. dot product)

$$\vec{a} = (a_1, a_2, \dots, a_n)$$

$$\vec{b} = (b_1, b_2, \dots, b_n)$$

$$\vec{a} \cdot \vec{b} = a_1 b_1 + a_2 b_2 + \dots + a_n b_n = \sum_{i=1}^n a_i b_i$$

$$\vec{a} \cdot \vec{b} = \|\vec{a}\| \|\vec{b}\| \cos \theta,$$

gdzie θ to kąt pomiędzy wektorami $\vec{a}$ i $\vec{b}$.

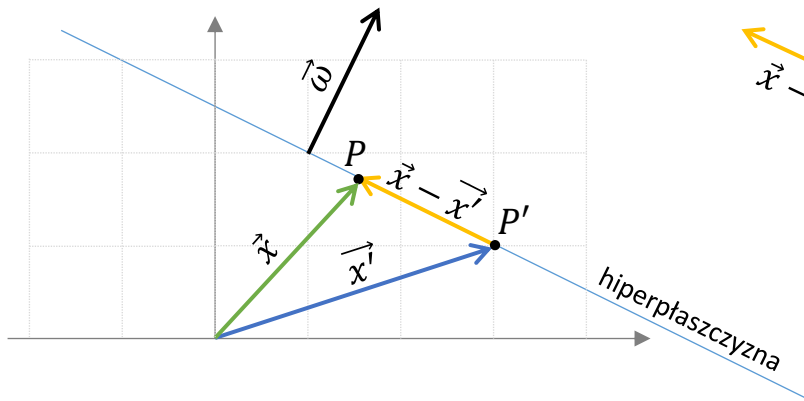
Wniosek 1: iloczyn skalarny wektorów prostopadłych jest równy 0

Wniosek 2: $\vec{a} \cdot \vec{a} = a_1 a_1 + a_2 a_2 + \dots + a_n a_n = \|\vec{a}\|^2$

Hiperpłaszczyzna

- ▶ Hiperpłaszczyzna w 2D to linia, a w 3D to płaszczyzna
- ▶ W ogólności do zdefiniowania hiperpłaszczyzny wystarczy wiedza o dowolnym jej punkcie P oraz o dowolnym wektorze do niej prostopadłym $\vec{\omega}$
- ▶ Jeżeli P' leży na hiperpłaszczyźnie, na której leży P wówczas zachodzi:

$$\begin{aligned}\vec{x} &= (1.5, 1.75) \\ \vec{x}' &= (3, 1) \\ \vec{x} - \vec{x}' &= (-1.5, 0.75)\end{aligned}$$



$$\vec{\omega} \cdot (\vec{x} - \vec{x}') = 0$$

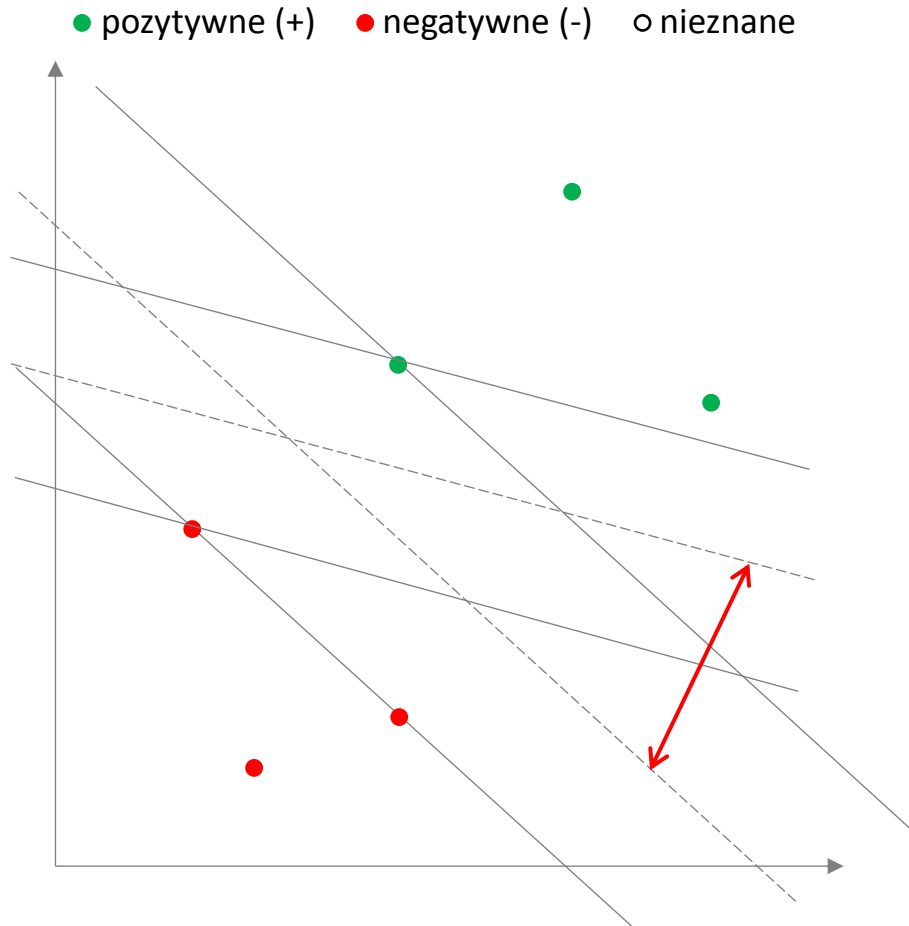
$$\vec{\omega} \cdot \vec{x} - \vec{\omega} \cdot \vec{x}' = 0$$

$$\vec{\omega} \cdot \vec{x} + b = 0$$

gdzie $b = -\vec{\omega} \cdot \vec{x}'$

Różne b wyznaczają „równoległe do siebie hiperpłaszczyzny”

SVM – ogólna idea

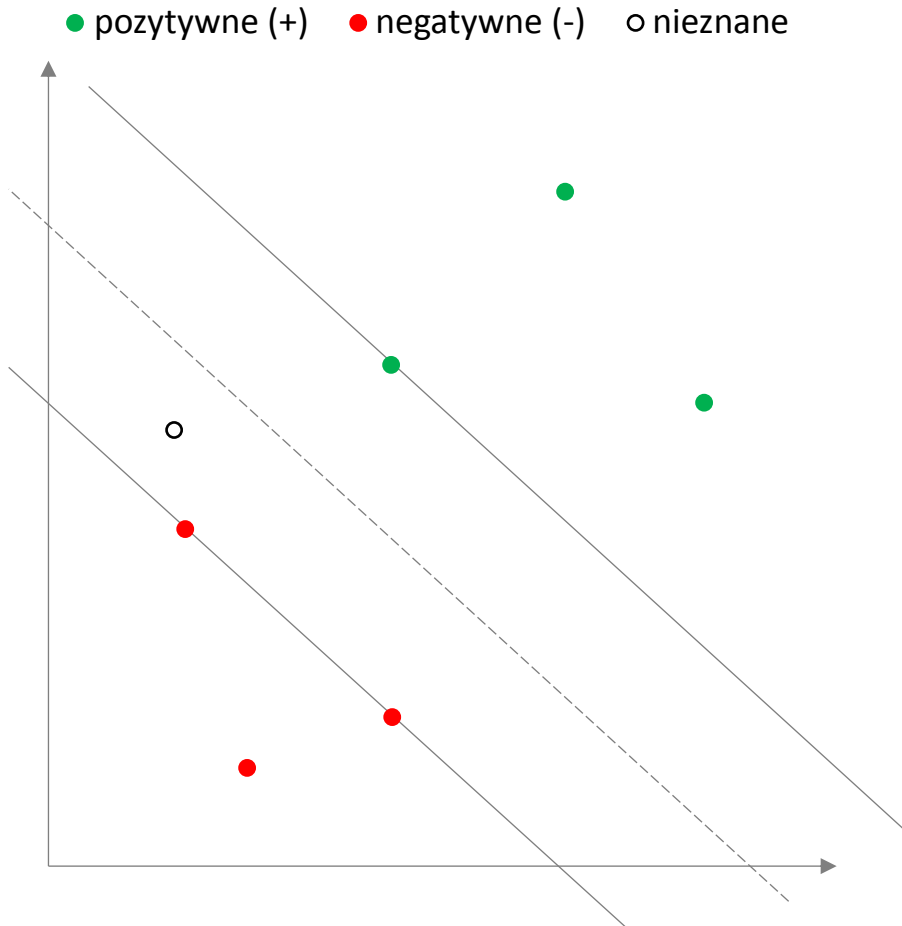


Idea: Znajdźmy linię, która najlepiej dzieli obserwacje negatywne i pozytywne

Metoda: znajdźmy najszerszą „ulicę”, którą można wstawić pomiędzy przypadki pozytywne i negatywne. Jej środkiem jest poszukiwana przez nas linia podziału.

Problem: jak znaleźć taką „ulicę”?

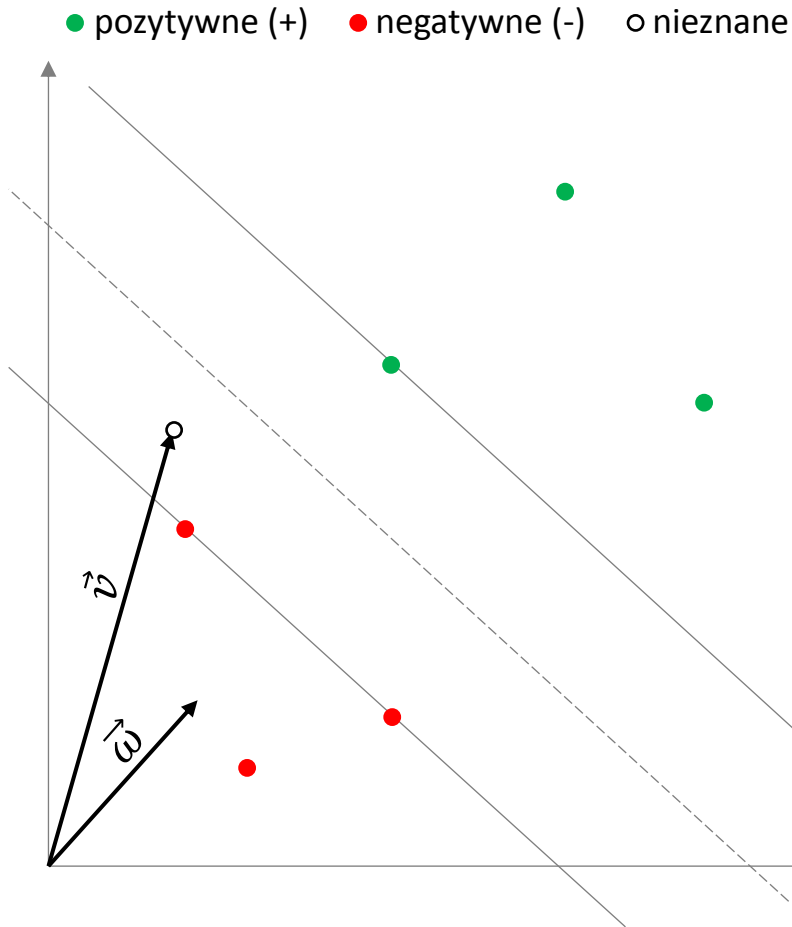
SVM – ogólna idea



Podstawowe pytanie:
po której stronie środka ulicy jest
nowa obserwacja?

Odpowiedź jest tożsama z
podjęciem decyzji o przydziale do
klasy pozytywnej lub negatywnej

SVM - wektory



$\vec{w}$ - dowolny wektor prostopadły do „ulicy” (nieważna długość)
 $\vec{v}$ - wektor do nowej obserwacji

Pytanie: po której stronie środka ulicy jest nowa obserwacja?



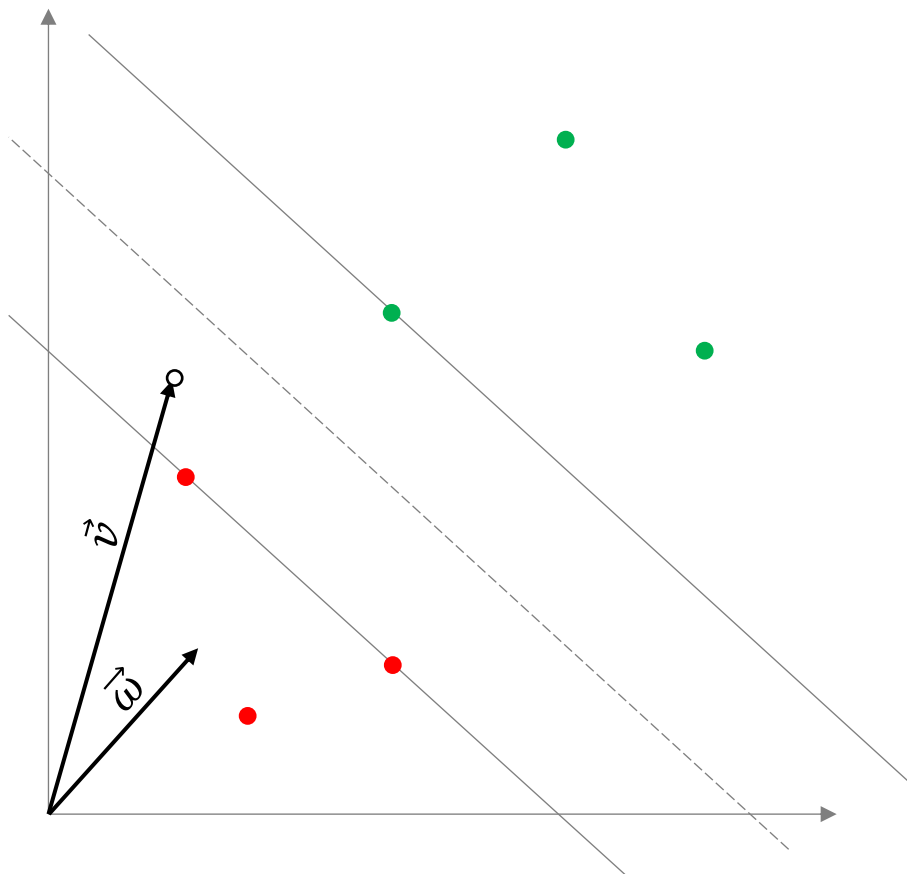
Pytanie: czy projekcja wektora $\vec{v}$ na wektor $\vec{w}$ skutkuje przekroczeniem środka „ulicy”?

Projekcja to iloczyn skalarny, zatem chcemy sprawdzić, czy $\vec{w} \cdot \vec{v}$ jest większe lub równe jakiejś stałej c

$$\vec{w} \cdot \vec{v} \geq c$$

SVM

● pozytywne (+) ● negatywne (-) ○ nieznane



Dla ułatwienia dalszych przekształceń
założmy, że $b = -c$, wówczas:

jeżeli $\vec{w} \cdot \vec{v} + b \geq 0$ to przykład (+),
jeżeli nie to przykład (-)

Problem: nie znamy zarówno $\vec{w}$
(których może istnieć wiele), jak
również b

Rozwiązanie: Trzeba wprowadzić
pewne założenia/ograniczenia

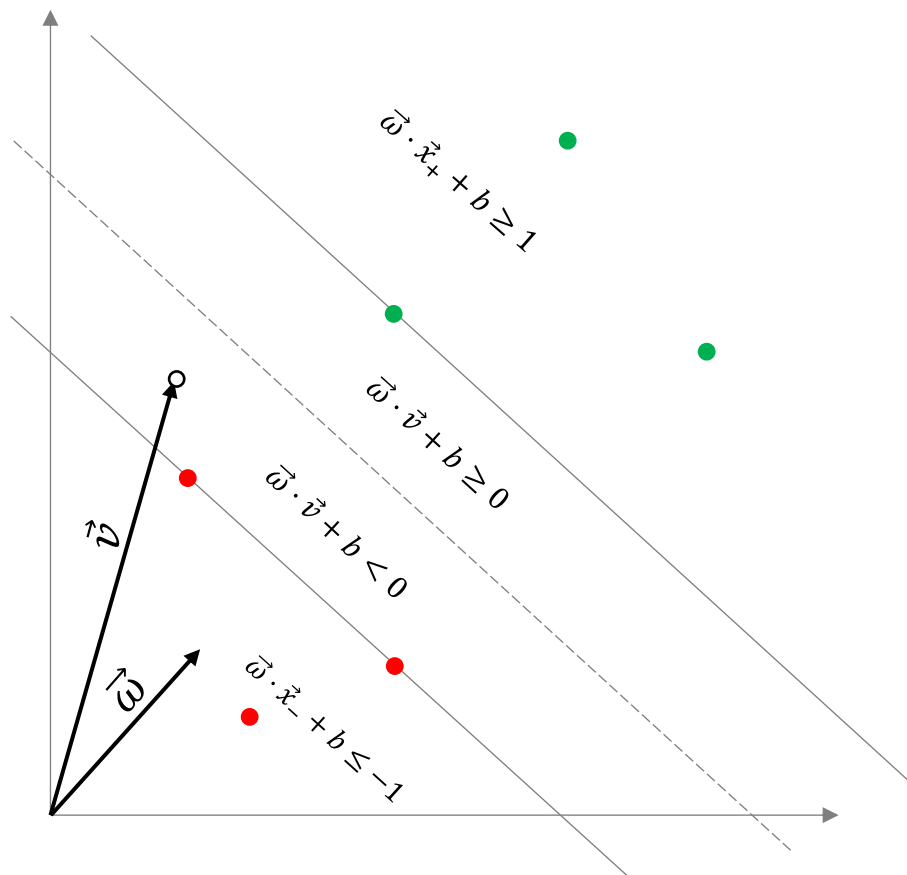
Niech

$\vec{x}_+$ - dowolny ZNANY przykład (+)

$\vec{x}_-$ - dowolny ZNANY przykład (-)

SVM -

● pozytywne (+) ● negatywne (-) ○ nieznane



Założmy, że

$$\vec{w} \cdot \vec{x}_+ + b \geq 1$$

$$\vec{w} \cdot \vec{x}_- + b \leq -1$$

i niech będzie dane y_i równe
+1 dla przykładów pozytywnych
-1 dla przykładów negatywnych

Pomnóżmy te nierówności przez y_i ,
wówczas dla przykładów

- pozytywnych:
 $y_i(\vec{w} \cdot \vec{x}_i + b) \geq y_i * 1$
- negatywnych
 $y_i(\vec{w} \cdot \vec{x}_i + b) \geq y_i * (-1)$

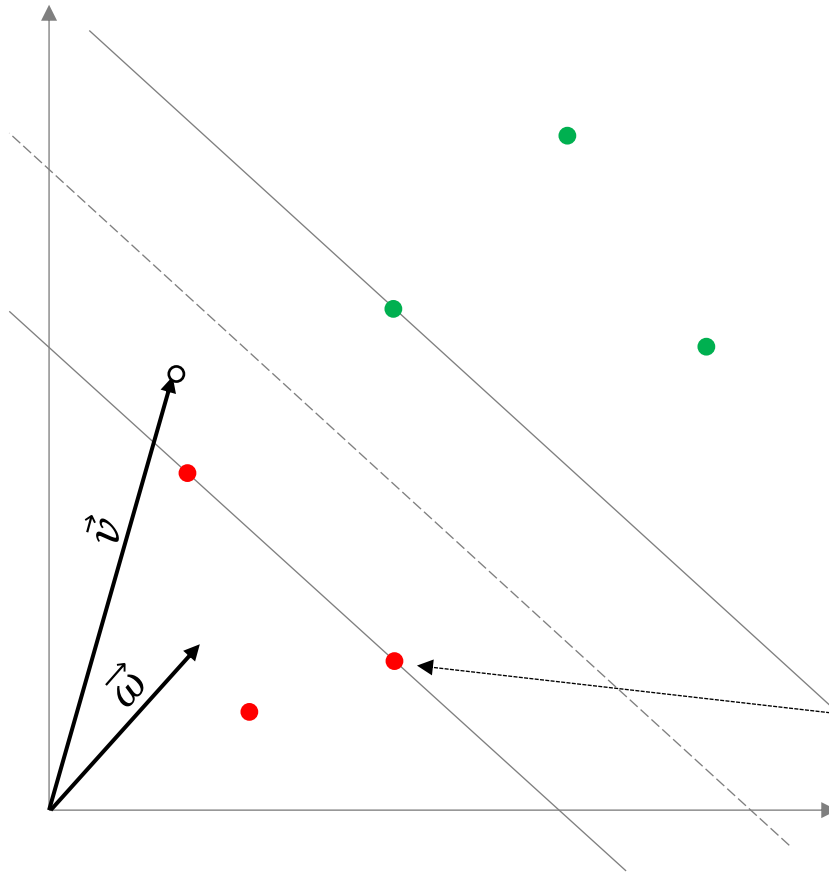
uważa na zmianę kierunku

Podstawiając 1 lub -1 po prawej
stronie, możemy zredukować te dwa
równania do jednego:

$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1$$

SVM – obserwacje na krawędzi

● pozytywne (+) ● negatywne (-) ○ nieznane



$$y_i(\vec{\omega} \cdot \vec{x}_i + b) \geq 1$$



$$y_i(\vec{\omega} \cdot \vec{x}_i + b) - 1 \geq 0$$

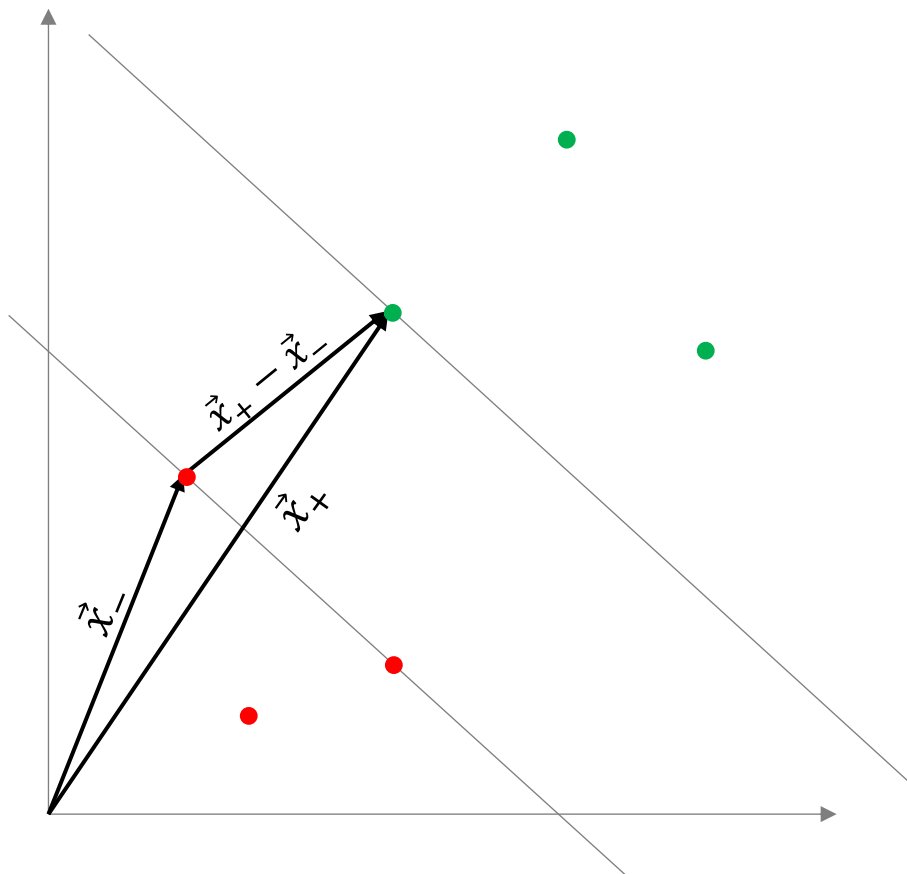
Pytanie: zawsze jest nieujemne, ale kiedy równe 0 ?

Odpowiedz: dla obserwacji znajdujących się dokładnie na krawędzi ulicy

$$y_i(\vec{\omega} \cdot \vec{x}_i + b) - 1 = 0$$

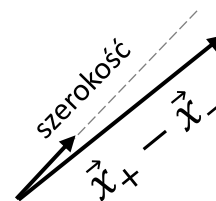
SVM – szerokość „ulicy”

● pozytywne (+) ● negatywne (-) ○ nieznane



Weźmy dwa dowolne wektory dla przykładów pozytywnego i negatywnego leżących na krawędzi. Na podstawie tych wektorów można wyprowadzić ogólny wzór na szerokość „ulicy”

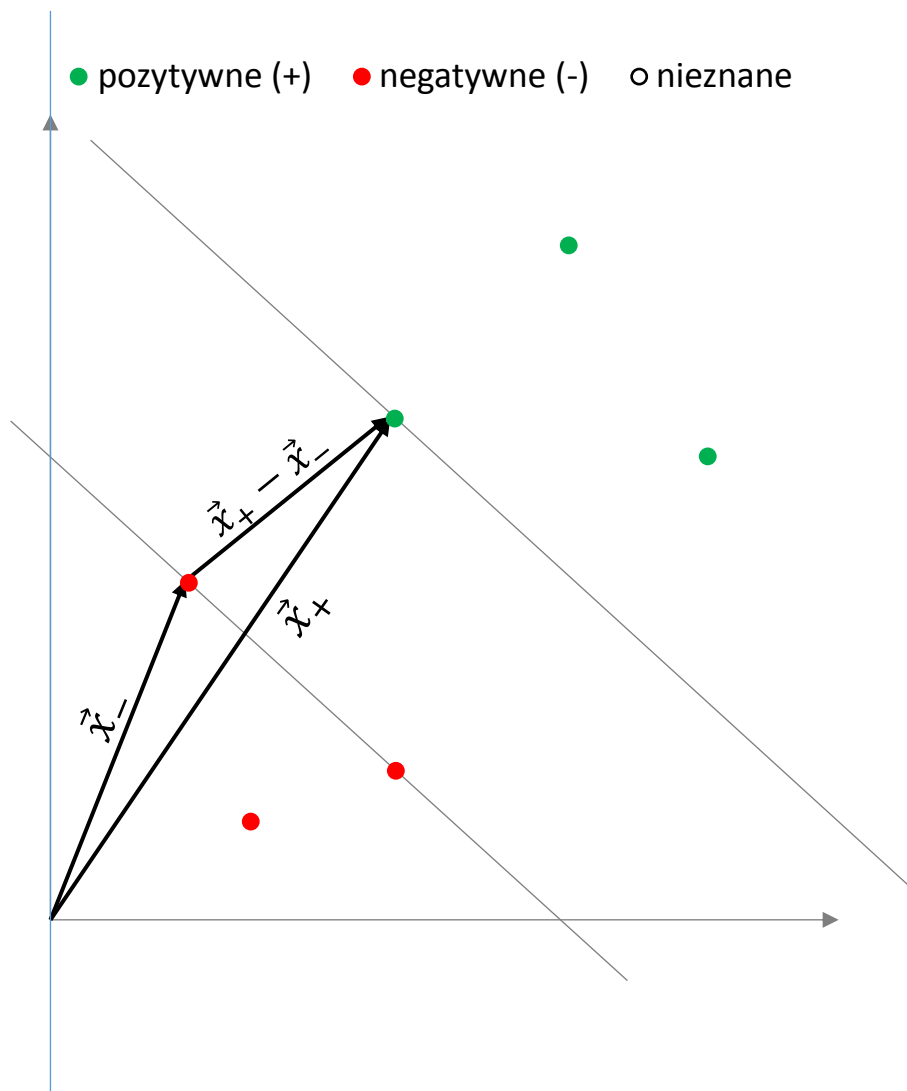
- odejmujemy wektory
- wektor wynikowy rzutujemy na jednostkowy wektor prostopadły do „ulicy”



$$\text{szerokość} = (\vec{x}_+ - \vec{x}_-) \cdot \frac{\vec{\omega}}{\|\vec{\omega}\|^2}$$

„jednostkowanie”

SVM – szerokość „ulicy”



$$\text{szerokość} = (\vec{x}_+ - \vec{x}_-) \cdot \frac{\vec{\omega}}{\|\vec{\omega}\|}$$

Problem: nie znamy $\vec{x}_+$ ani $\vec{x}_-$,
ale przypomnijmy, że

$$y_i(\vec{\omega} \cdot \vec{x}_i + b) - 1 = 0$$

Dla $\vec{x}_+$ wartość $y_i = 1$, czyli

$$1(\vec{\omega} \cdot \vec{x}_+ + b) - 1 = 0$$

$$\vec{\omega} \cdot \vec{x}_+ = 1 - b$$

a dla $\vec{x}_-$ wartość $y_i = -1$, czyli

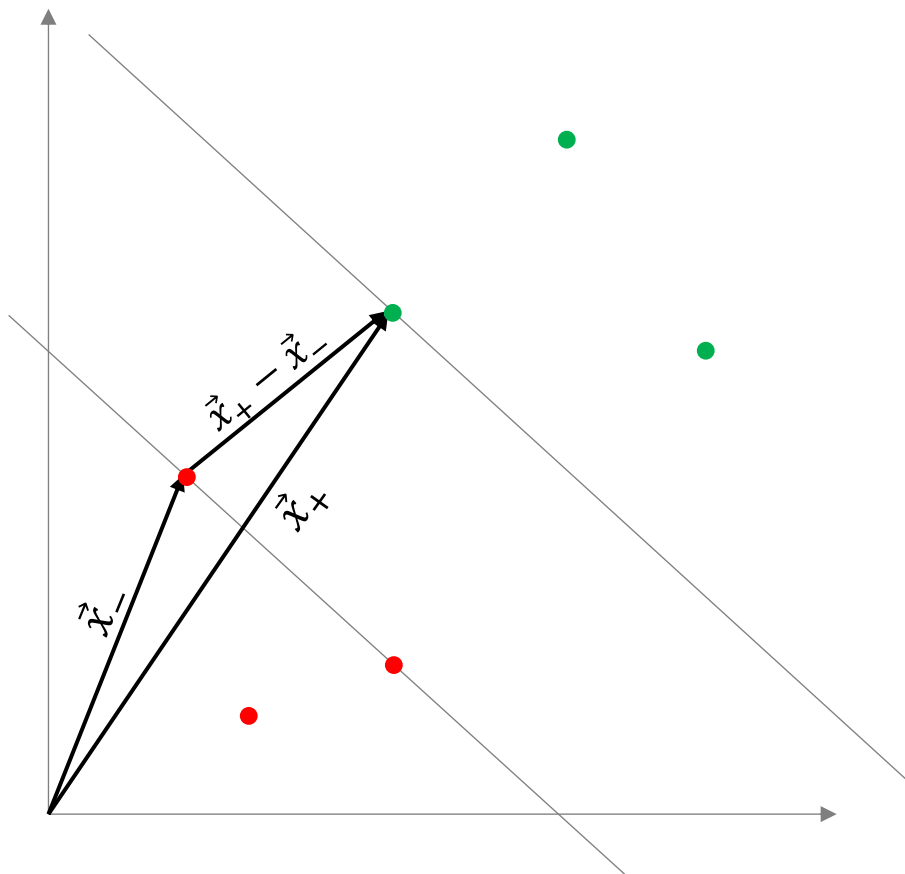
$$-1(\vec{\omega} \cdot \vec{x}_- + b) - 1 = 0$$

$$-(\vec{\omega} \cdot \vec{x}_-) = 1 + b$$

$$(\vec{x}_+ - \vec{x}_-) \cdot \frac{\vec{\omega}}{\|\vec{\omega}\|}$$

SVM – szerokość „ulicy”

● pozytywne (+) ● negatywne (-) ○ nieznane



Podstawiając otrzymujemy:

$$\frac{1 - b + (1 + b)}{\|\vec{\omega}\|} = \frac{2}{\|\vec{\omega}\|}$$

$$\text{szerokość} = \frac{2}{\|\vec{\omega}\|}$$

Maksymalizowanie szerokości oznacza zatem minimalizowanie $\|\vec{\omega}\|$

Dla ułatwienia kolejnych przekształceń (liczenia pochodnych) można założyć zatem, że minimalizujemy wyrażenie:

$$\frac{1}{2} \|\vec{\omega}\|^2$$

SVM – sformułowanie prymalne

- ▶ Zadanie maksymalizacji „szerokości” drogi można wyrazić w postaci problemu programowania kwadratowego, w którym jest $d + 1$ zmiennych $(\omega_1, \dots, \omega_d, b)$ oraz i ograniczeń

primal formulation

- ▶ Dla przykładów $x_1, \dots, x_n \in \mathbb{R}^d$ z etykietami $y_1, \dots, y_n \in \{-1, 1\}$

$$\min_{\vec{\omega}, b} \frac{1}{2} \|\vec{\omega}\|^2$$

przy ograniczeniach $y_i(\vec{\omega} \cdot \vec{x}_i + b) \geq 1, i = 1, \dots, n$

- ▶ Klasyfikacja nowego elementu $\vec{v}$ następuje poprzez sprawdzenie, czy

$$\vec{\omega} \cdot \vec{v} + b > 0$$

- ▶ Problem w postaci pierwotnej można rozwiązać, ale lepsze jest przejście do sformułowania dualnego

SVM – mnożniki Lagrange'a

- ▶ Przypomnienie: jak znaleźć ekstremum? Policzyć pochodną i sprawdzić dla jakiego parametru przyjmuje wartość 0
- ▶ Mnożniki Lagrange'a to metoda obliczania tzw. ekstremum warunkowych funkcji
- ▶ Mając funkcję f i zbiór A opisany za pomocą pewnego równania można znaleźć ekstrema funkcji f dla zbioru A .
- ▶ Przykład: niech będzie dana funkcja $f(x, y) = x^2 - y^2$ i warunek $x^2 + y^2 - 1 = 0$. Znajdź ekstrema warunkowe funkcji $f(x, y)$
- ▶ Metoda rozwiązania polega na
 - ▶ zbudowaniu funkcji Lagrange'a
 - ▶ $L(x, y) = x^2 - y^2 + \lambda(x^2 + y^2 - 1)$
 - ▶ wyliczeniu układu równań
 - ▶ $\frac{\partial L}{\partial x} = 0$
 - ▶ $\frac{\partial L}{\partial y} = 0$

SVM – mnożniki Lagrange'a

► Wersja dla SVM

Mnożniki Lagrange'a

$$L(\vec{\omega}, b, \alpha) = \frac{1}{2} \|\vec{\omega}\|^2 - \sum_i \alpha_i \underbrace{[y_i(\vec{\omega} \cdot \vec{x}_i + b) - 1]}_{\text{ograniczenia}}$$

(„ograniczone” do wartości 0)

To chcemy maksymalizować

$$\sum_i \alpha_i \geq 0$$

Uwaga: $\alpha_i > 0$ oznacza, że aby ograniczenie było równe 0, $y_i(\vec{\omega} \cdot \vec{x}_i + b)$ musi być równe 1. To oznacza natomiast, że przykład skojarzony z wektorem $\vec{x}_i$ jest wektorem wspierającym.

Mnożniki Lagrange'a w SVM są zatem równe 0 dla wektorów, które nie leżą na krawężnikach drogi

SVM – transformacja do sformułowania dualnego

Idea: rozwiązując problem dualny (czyli wyznaczając α) otrzymujemy rozwiązanie problemu pierwotnego

$$\max_{\alpha} \min_{\vec{\omega}, b} L(\vec{\omega}, b, \alpha) = \frac{1}{2} \|\vec{\omega}\|^2 - \sum_i \alpha_i [y_i (\vec{\omega} \cdot \vec{x}_i + b) - 1]$$
$$\alpha_i \geq 0$$

$$\frac{\partial L}{\partial \vec{\omega}} = \vec{\omega} - \sum_i \alpha_i y_i \vec{x}_i \quad \Rightarrow \quad \frac{\partial L}{\partial \vec{\omega}} = 0 \quad \Rightarrow \quad \vec{\omega} = \sum_i \alpha_i y_i \vec{x}_i$$

$$\frac{\partial L}{\partial b} = - \sum_i \alpha_i y_i \quad \Rightarrow \quad \frac{\partial L}{\partial b} = 0 \quad \Rightarrow \quad \sum_i \alpha_i y_i = 0$$

SVM – pochodne

...dokonajmy prostych przekształceń i zobaczmy jakie są zależności

$$L(\vec{\omega}, b, \alpha) = \frac{1}{2} \|\vec{\omega}\|^2 - \sum_i \alpha_i [y_i (\vec{\omega} \cdot \vec{x}_i + b) - 1]$$

$\vec{\omega} = \sum_i \alpha_i y_i \vec{x}_i$

$$L(b, \alpha) = \frac{1}{2} \left(\sum_i \alpha_i y_i \vec{x}_i \right) \left(\sum_j \alpha_j y_j \vec{x}_j \right) - \left(\sum_i \alpha_i y_i \vec{x}_i \right) \cdot \left(\sum_j \alpha_j y_j \vec{x}_j \right) - \sum_i \alpha_i y_i b + \sum_i \alpha_i$$

Uwaga na indeksy

$$L(b, \alpha) = -\frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j + \underbrace{\sum_i \alpha_i y_i b}_{=0} + \sum_i \alpha_i$$

=0 ponieważ $\sum_i \alpha_i y_i = 0$

$$L(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j$$

SVM – sformułowanie dualne

dual formulation

- Dla przykładów $x_1, \dots, x_n \in \mathbb{R}^d$ z etykietami $y_1, \dots, y_n \in \{-1, 1\}$

$$\max_{\alpha} \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j$$

przy ograniczeniach $\sum_i \alpha_i y_i = 0 \wedge \alpha_i \geq 0$

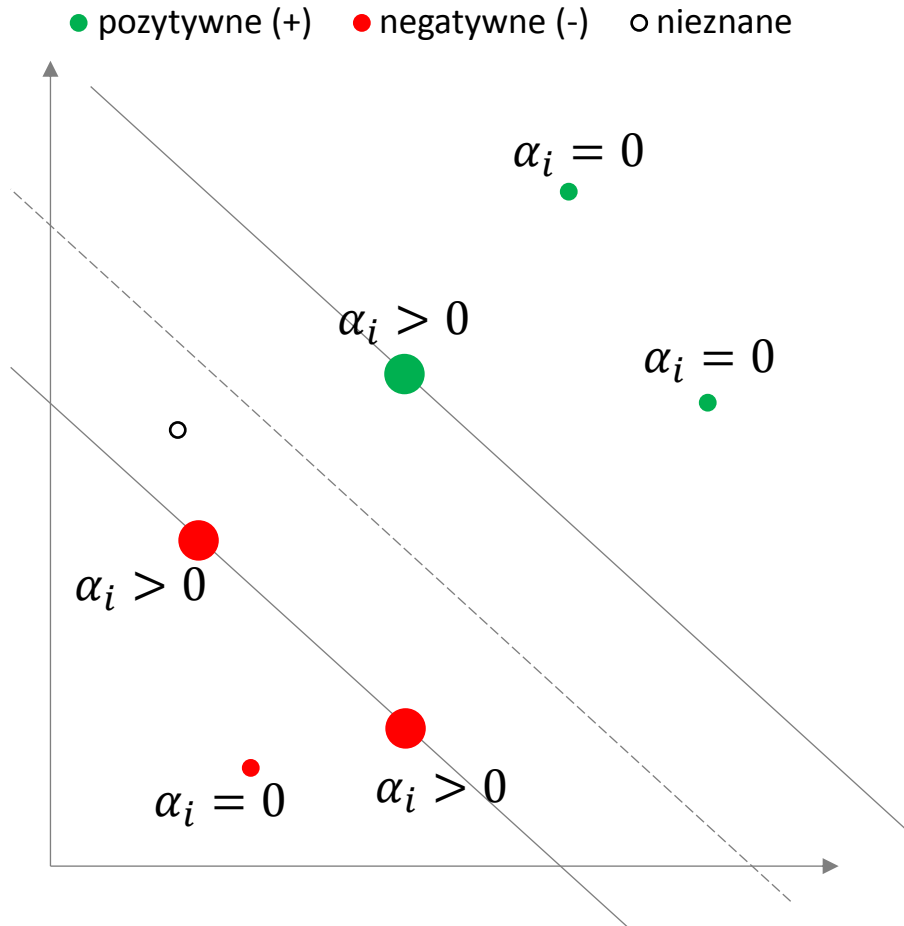
- Jest to również problem programowania kwadratowego z n zmiennymi $(\alpha_1, \dots, \alpha_n)$ i $2n$ ograniczeniami

SVM – obserwacje

$$L(\alpha) = \sum_i \alpha_i + \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j$$

- ▶ Czyli zależy tylko od iloczynu skalarnego przykładów
- ▶ Można to rozwiązać metodami iteracyjnymi (nie ma lokalnych ekstremów więc w nich nie „utkniemy”)
- ▶ Zauważmy, że reguła dla sformułowania pierwotnego:
jeżeli $\vec{\omega} \cdot \vec{v} + b \geq 0$ to przykład pozytywny
może zostać zastąpiona (przez podstawienie $\vec{\omega} = \sum_i \alpha_i y_i \vec{x}_i$)
dla sformułowania dualnego na:
jeżeli $\sum_i \alpha_i y_i \vec{x}_i \cdot \vec{v} + b \geq 0$ to przykład pozytywny
- ▶ Zatem decyzja też zależy od iloczynu skalarnego

SVM – sformułowanie dualne



Rozwiązując problem dualny poznajemy wartości α

Na ich podstawie wyliczamy

$\vec{\omega}$, ze wzoru $\vec{\omega} = \sum_i \alpha_i y_i \vec{x}_i$

b , ze wzoru $b = y_k - \vec{\omega} \cdot \vec{x}_k$ (dla dowolnego k , dla którego $\alpha_k > 0$)

Inaczej mówiąc używamy wektorów wspierających dla wyznaczenia $\vec{\omega}$ oraz b

SVM – kernel trick

- ▶ wiele zbiorów danych może nie być liniowo separowalna
- ▶ Rozwiązanie: transformacja danych wejściowych, tak by były separowalne liniowo (najczęściej transformacja do przestrzeni o większej liczbie wymiarów)
- ▶ Każdą obserwację $\vec{x}_i$ należałoby zatem poddać transformacji ϕ co zapisujemy jako $\phi(\vec{x}_i)$
- ▶ Zauważmy, że maksymalizacja szerokości drogi zależy tylko od iloczynu skalarnego, w docelowej przestrzeni maksymalizować trzeba $\phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$
- ▶ Klasyfikacja wymaga sprawdzenia nierówności:
 - ▶ dla postaci prymalnej $\vec{w} \cdot \phi(\vec{v}) + b > 0$
 - ▶ dla postaci dualnej $\sum_i \alpha_i y_i \phi(\vec{x}_i) \cdot \phi(\vec{v}) + b \geq 0$
- ▶ Najważniejsza obserwacja: w wersji dualnej **nie musimy znać transformacji**, musimy tylko znać funkcję (kernel function), która dostarcza wynik iloczynu skalarnego w pewnej założonej przestrzeni

$$K(\vec{x}_i, \vec{x}_j) = \phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$$

SVM – rodzaje kerneli

- ▶ Liniowy

$$K(\vec{x}_i, \vec{x}_j) = \vec{x}_i \cdot \vec{x}_j$$

- ▶ Gausowski

$$K(\vec{x}_i, \vec{x}_j) = e^{-\gamma \|\vec{x}_i - \vec{x}_j\|^2}$$

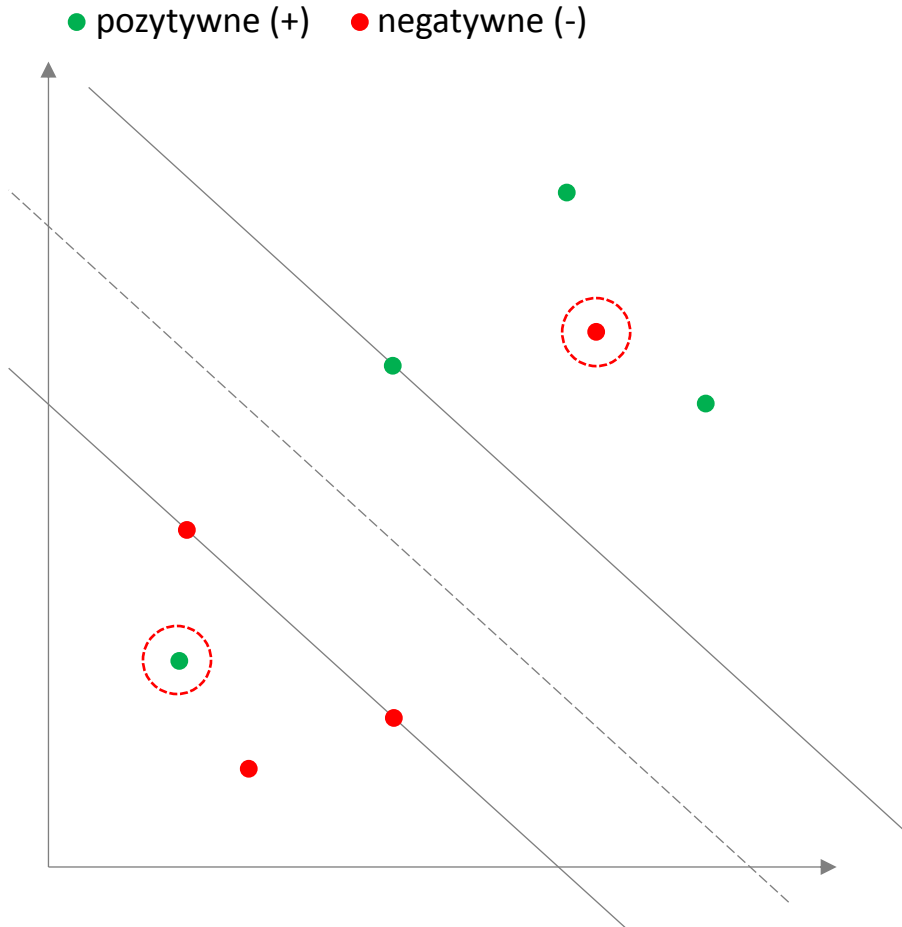
- ▶ Wykładniczy

$$K(\vec{x}_i, \vec{x}_j) = e^{-\gamma \|\vec{x}_i - \vec{x}_j\|}$$

- ▶ Wielomianowy

$$K(\vec{x}_i, \vec{x}_j) = (p + \vec{x}_i \cdot \vec{x}_j)^q$$

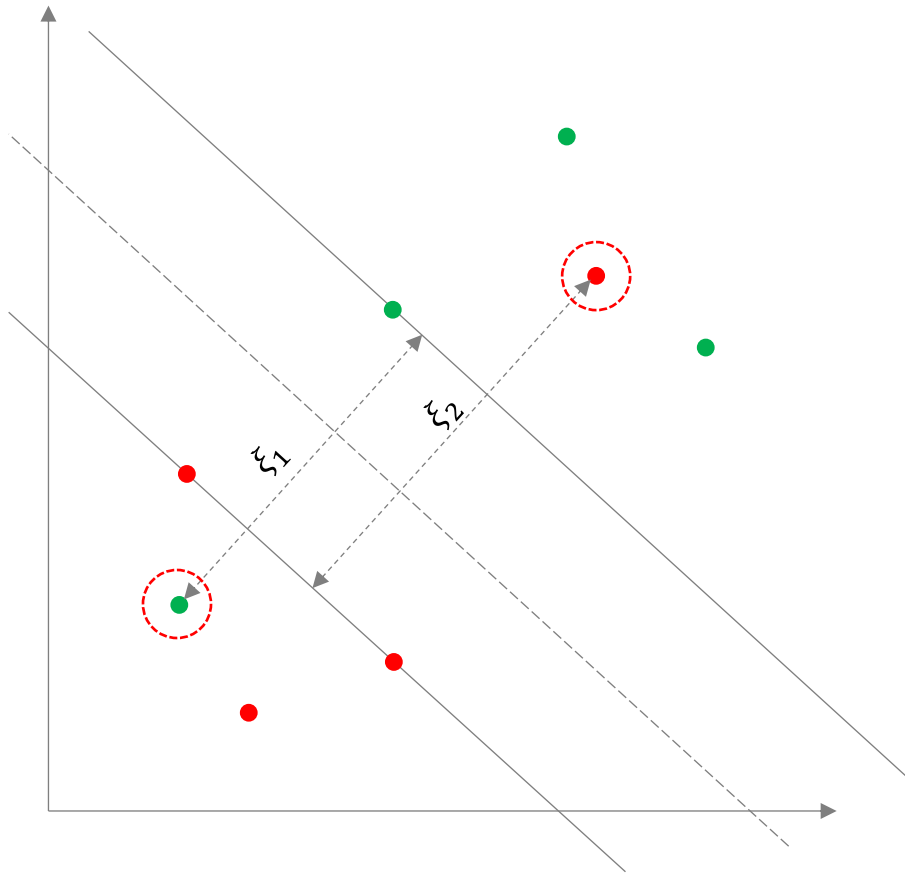
SVM – soft margin (1)



- oprócz transformacji do większej liczby wymiarów można stosować przekształcenia przykładów wejściowych (ale ryzyko overfitting)
- jeżeli nadal nie można uzyskać liniowej separowalności możemy pozwolić na pewien błąd
- Trywialne podejście: policzmy ile razy się mylimy, tj. „jesteśmy po złej stronie drogi”
- Inne podejście: policzmy jak bardzo się mylimy

SVM – soft margin (2)

● pozytywne (+) ● negatywne (-)



Trywialne podejście:

$$\min_{\vec{w}, b} \|\vec{w}\|^2 + C (\text{liczba pomyłek})$$

C to wsp. „tradeoff” pomiędzy szerokością „drogi” a błędnym rozróżnieniem klas

Problem: nie rozróżniamy ,delikatnej’ i ,mocnej’ pomyłki (ocena binarna), a dodatkowo nie jest to problem programowania kwadratowego

Lepsze rozwiązanie:

$$\min_{\vec{w}, b} \|\vec{w}\|^2 + C \sum_i \xi_i$$

liniowa zależność
od „wielkości”
pomyłki

przy ograniczeniach

$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 - \xi_i, \quad i = 1, \dots, n$$

gdzie ξ_i to odległość i -tego przykładu od hiperpłaszczyzny

SVM – soft margin (3)

► Primal formulation

$$\min_{\vec{\omega}, b} \|\vec{\omega}\|^2 + C \sum_i \xi_i$$

przy ograniczeniach $y_i(\vec{\omega} \cdot \vec{x}_i + b) \geq 1 - \xi_i, \quad i = 1, \dots, n$

► Dual formulation

$$\max_{\alpha} \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j$$

przy ograniczeniach $\sum_i \alpha_i y_i = 0 \wedge 0 \leq \alpha_i \leq C$

► Parametr C

- Duża wartość zbliża SVM do wersji „hard-margin” ($C = \infty$)
- Mała wartość dopuszcza większy błąd
- Nie ma dobrej, uniwersalnej wartości

SVM – jeszcze o błędach (1)

- ▶ Na przykładzie sformułowania prymalnego

$$\min_{\vec{\omega}, b} \|\vec{\omega}\|^2 + C \sum_i \xi_i$$

przy ograniczeniach $y_i(\vec{\omega} \cdot \vec{x}_i + b) \geq 1 - \xi_i, \quad i = 1, \dots, n$

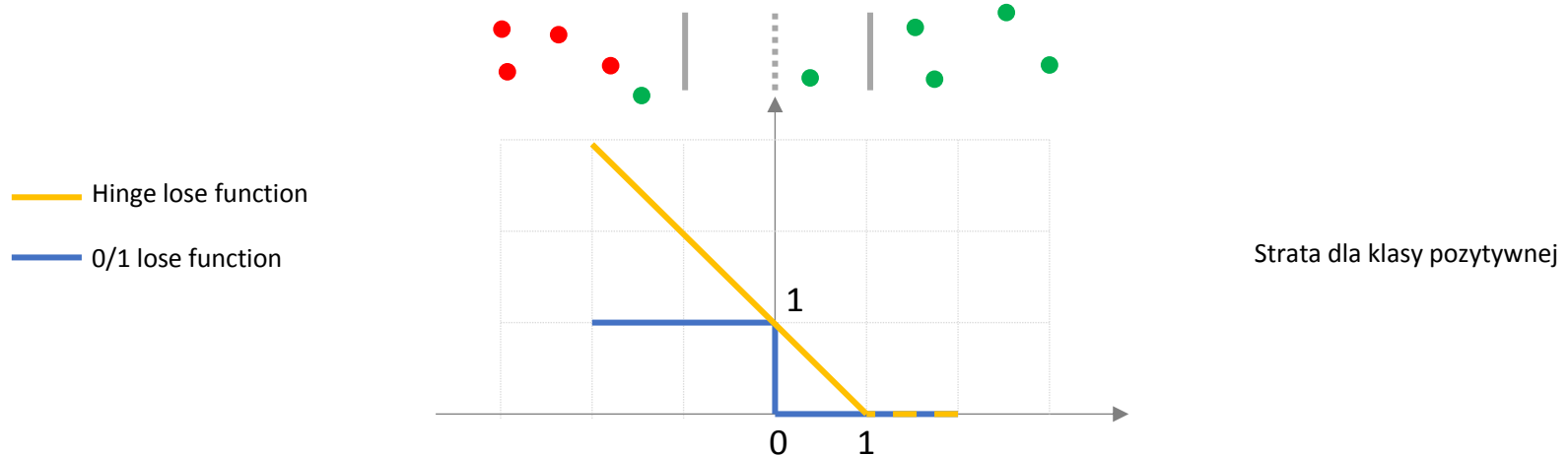
- ▶ zauważmy, że ograniczenie $y_i(\vec{\omega} \cdot \vec{x}_i + b) \geq 1 - \xi_i$ można zapisać w postaci $y_i f(\vec{x}_i) \geq 1 - \xi_i$, a po przekształceniu otrzymujemy $\xi_i \geq 1 - y_i f(\vec{x}_i)$
- ▶ Ponieważ $\xi_i \geq 0$, cały problem można zapisać w postaci

$$\min_{\vec{\omega}, b} \|\vec{\omega}\|^2 + C \sum_i \underbrace{\max(0, 1 - y_i f(\vec{x}_i))}_{\text{Funkcja straty}}$$

bez ograniczeń

SVM – jeszcze o błędach (2)

- ▶ $\max(0, 1 - y_i f(\vec{x}_i))$ - jak to interpretować?
- ▶ Gdy punkt jest
 - ▶ Po właściwej stronie hiperpłaszczyzny i poza „drogą”, wówczas $y_i f(\vec{x}_i) > 1$. Nie zwiększa to straty ponieważ max da wynik równy 0.
 - ▶ Po właściwej stronie hiperpłaszczyzny i na krawędzi „drogi”, wówczas $y_i f(\vec{x}_i) = 1$. Nie zwiększa to straty ponieważ max da wynik równy 0.
 - ▶ Po niewłaściwej stronie hiperpłaszczyzny lub na „drodze”, wówczas $y_i f(\vec{x}_i) < 1$. Zawsze zwiększa to straty ponieważ max da wynik równy większy od 1 (im większa odległość, tym większa wartość straty).



SVM

- ▶ Jak wybrać model?
 - ▶ Można zastosować podejście „data-driven”, w którym używany jest dodatkowy etap (nested cross-validation) dla różnych wartości C oraz transformacji. Innym podejściem może być wykorzystanie np. MDL.
- ▶ SVM a wiele klas decyzyjnych
 - ▶ one vs rest, one vs all – dla każdej z M klas tworzony jest klasyfikator, w którym za klasę pozytywną uznaje się m -tą klasę, a pozostałe przypadki traktowane są jako przykłady klasy negatywnej. Przy klasyfikacji wygrywa klasa, dla której wartość funkcji jest największa
 - ▶ OVO - one vs one – dla M klas tworzonych jest $M(M - 1)/2$ klasyfikatorów (każda klasa z każdą). Decyzja jest podejmowana przez głosowanie większościowe.
 - ▶ ECOC – error correcting output code – (następny slajd)
 - ▶ DAGSVM – tworzy klasyfikatory jak w OVO, ale inaczej wygląda klasyfikacja. Tworzone jest drzewo DAG (direct acyclic graph), które przypomina binarne drzewo decyzyjne i jest używane dla szybszego eliminowania klas
 - ▶ Weston-Watkins oraz Crammer-Singer (M-SVM) zmiana definicji problemu, jeden klasyfikator

SVM - ECOC

Class	Code word						
	f_1	f_2	f_3	f_4	f_5	f_6	f_7
C_0	0	0	0	0	0	0	0
C_1	1	1	1	1	0	0	0
C_2	1	1	0	0	1	1	0
C_3	1	0	1	0	1	0	1

- ▶ coding matrix $M_{Q \times S}$.
 - ▶ wiersze to klasy, kolumny to klasyfikatory binarne
 - ▶ Wartości 1/0 oznacza, że dla q -tej klasy i s -tego klasyfikatora przykład jest pozytywny/negatywny
- ▶ Podczas trenowania tworzone są klasyfikatory binarne
- ▶ Nowy przykład dostaje wektor wynikowy, który porównywany jest z wektorami dla klas (np. odległość Hamminga)

One-class SVM

- ▶ To nie jest problem klasyfikacyjny – nie ma żadnych, z góry znanych klas
- ▶ Idea (intuicja):
 - ▶ znajdź region, który zawiera większość przykładów
 - ▶ zbuduj model, który dla przykładów z tego regionu da wynik pozytywny, a dla pozostałych przykładów da wynik negatywny
 - ▶ Przykłady negatywne są obserwacjami dotąd niespotykanymi lub przypadkami osobliwymi
- ▶ Zaleta: nie musimy mieć w przykładach uczących pełnych danych o negatywnych przykładach (a często ich nie znamy, np. różne możliwe wady produktów mogą nie być łatwe w przewidzeniu)
- ▶ W praktyce dodajemy dodatkowy parametr określający górną granicę na względną liczbę nowych obserwacji/przypadków osobliwych