Wstęp do uczenia maszynowego

Maszyny wektorów wspierających

Ewa Szczurek + BW (modyfikacje)

bartek@mimuw.edu.pl
Instytut Informatyki
Uniwersytet Warszawski

UNIWERSYTET
WARSZAWSKI

Maszyny wektorów wspierających:

(1) Klasyfikator o maksymalnym marginesie;

(2) Klasyfikator wektorów wspierających;

(3) Maszyny wektorów wspierających

Hiperpłaszczyzna (1)

- W przestrzeni p wymiarowej, **hiperpłaszczyzna** to $p-1$ wymiarowa podprzestrzeń afiniczna przestrzeni $\mathbb{R}^p$.
- Słowo '*afiniczna*' oznacza, że hiperpłaszczyzna nie musi przechodzić przez początek układu (czyli wektor zerowy nie musi należeć do hiperpłaszczyzny). Jest to zwykła podprzestrzeń $p-1$ wymiarowa przesunięta o pewien wektor.
- W dwóch wymiarach hiperpłaszczyzna jest prostą opisaną równaniem

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0$$

- W ogólności, w p wymiarach, hiperpłaszczyzna jest opisana równaniem

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p = 0.$$

Hiperpłaszczyzna (2)

Wektory nie leżące na hiperpłaszczyźnie podzielone są na **dwie** klasy:

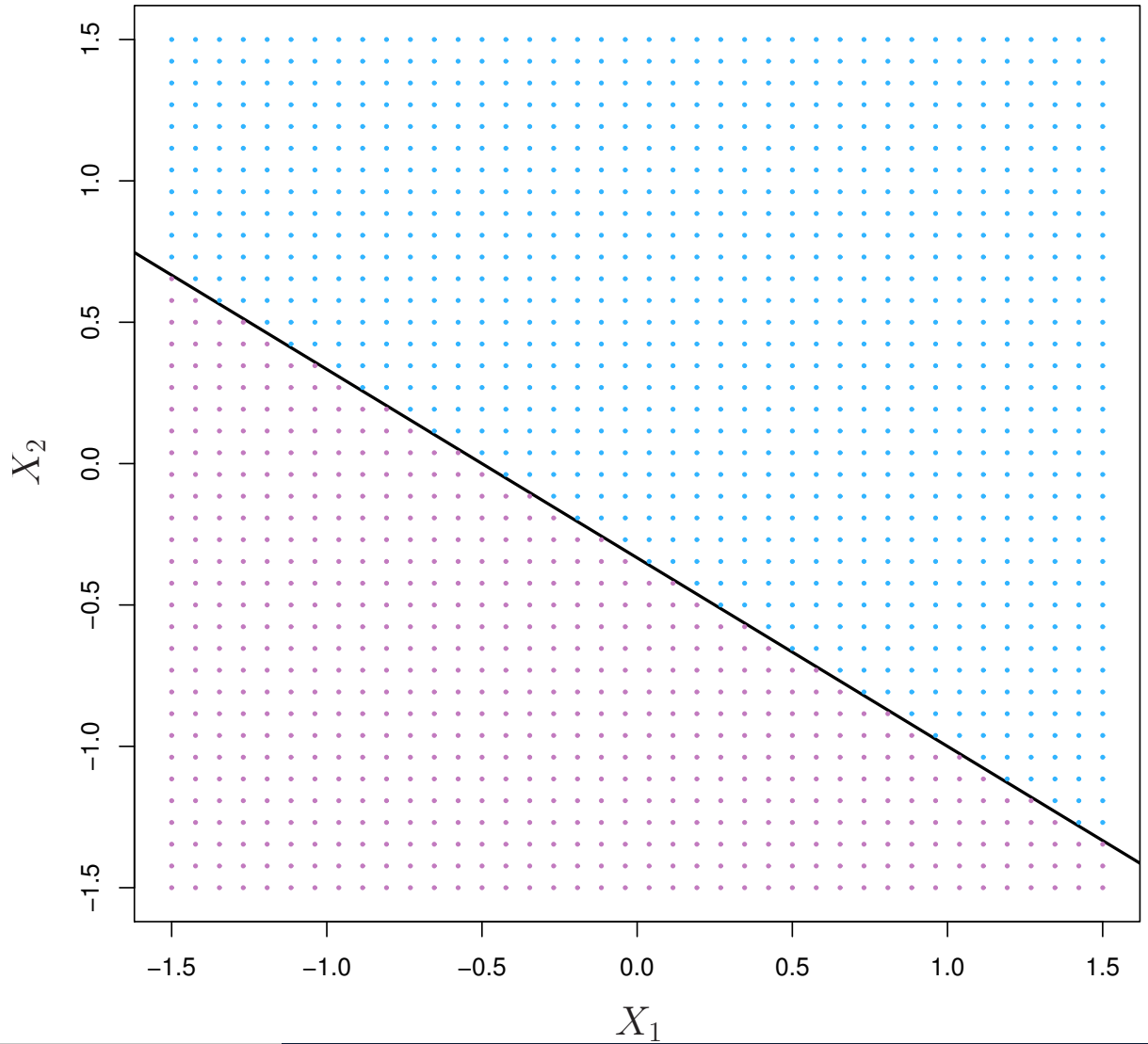
- Te które leżą po jedej stronie hiperpłaszczyzny spełniają warunek

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p > 0.$$

- Te które leżą po drugiej stronie spełniają

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p < 0.$$

Podział płaszczyzny prostą $1 + 2X_1 + 3X_2 = 0$ na dwie klasy



Klasyfikacja poprzez rozdzielenie hiperpłaszczyzną

- Przyjmijmy, że mamy $n \times p$ macierz danych X zawierającą n *treningowych obserwacji* p-wymiarowych (wiersze macierzy X):

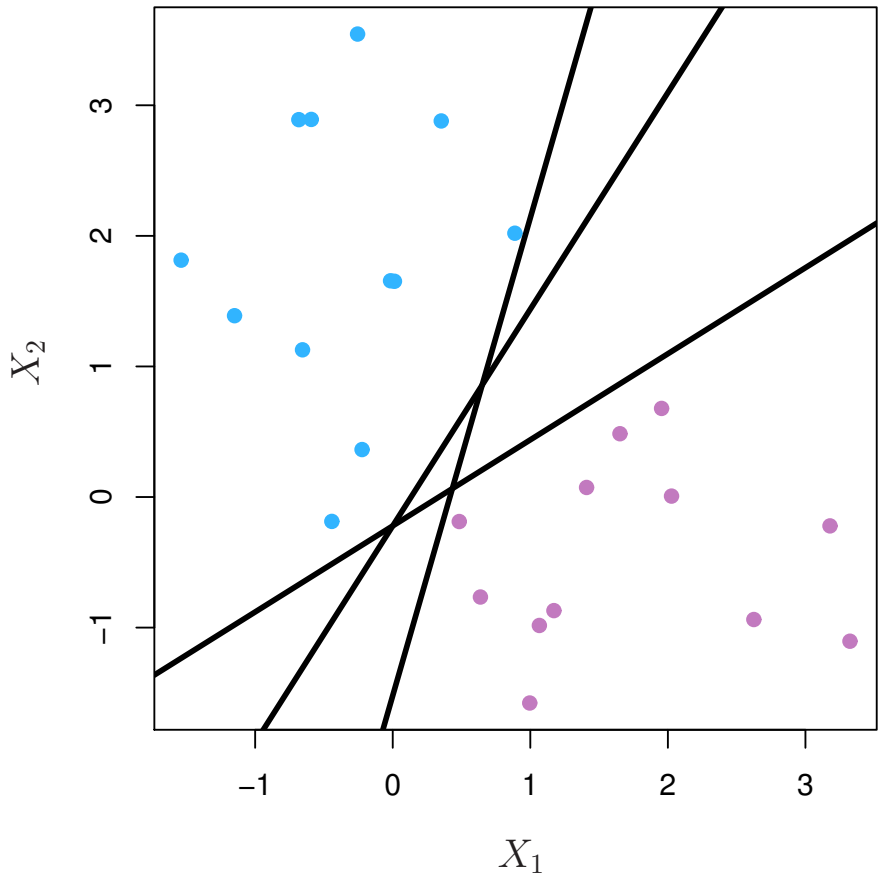
$$x_1 = \begin{bmatrix} x_{11} \\ \vdots \\ x_{1p} \end{bmatrix}, \ldots, x_n = \begin{bmatrix} x_{n1} \\ \vdots \\ x_{np} \end{bmatrix},$$

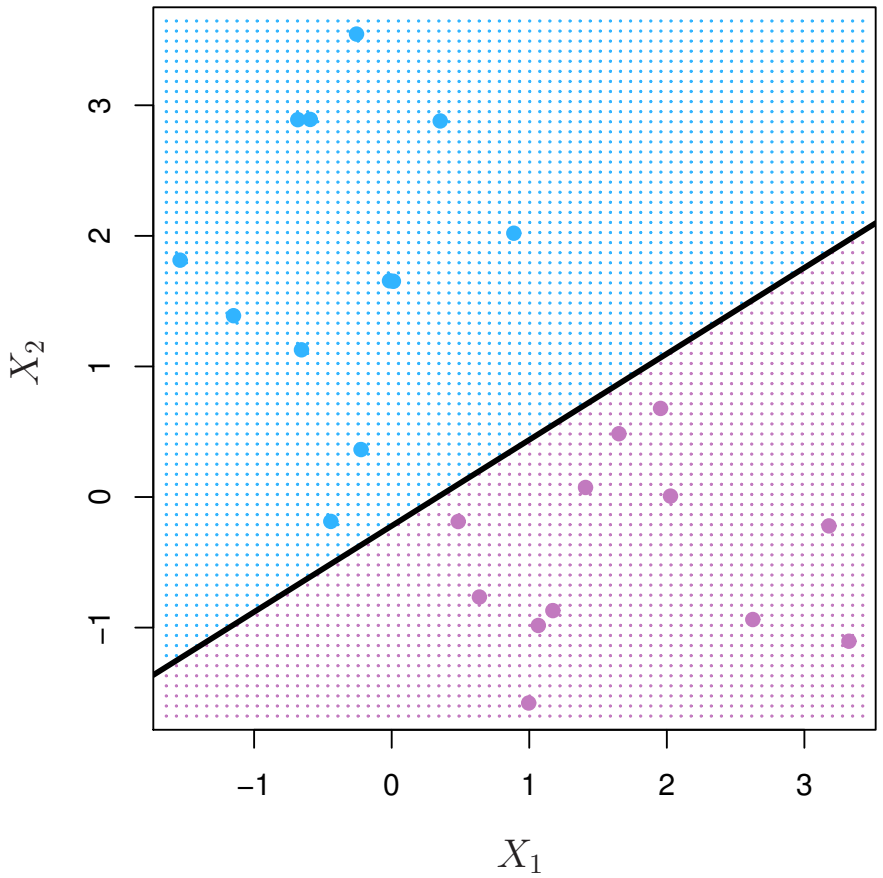
  które są poklasyfikowane na dwie klasy: $y_1, \ldots, y_n \in \{-1, 1\}$.
- *Testowa obserwacja* to wektor cech $x^* = (x_1^* \ \ldots \ x_p^*)^T$.
- Hiperpłaszczyzna rozdzielająca (o ile istnieje) opisuje się warunkiem

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) > 0$$

  dla wszystkich $i = 1, \ldots, n$.

Jeśli jest jedna hiperpłaszczyzna rozdzielająca to jest ich nieskończenie wiele

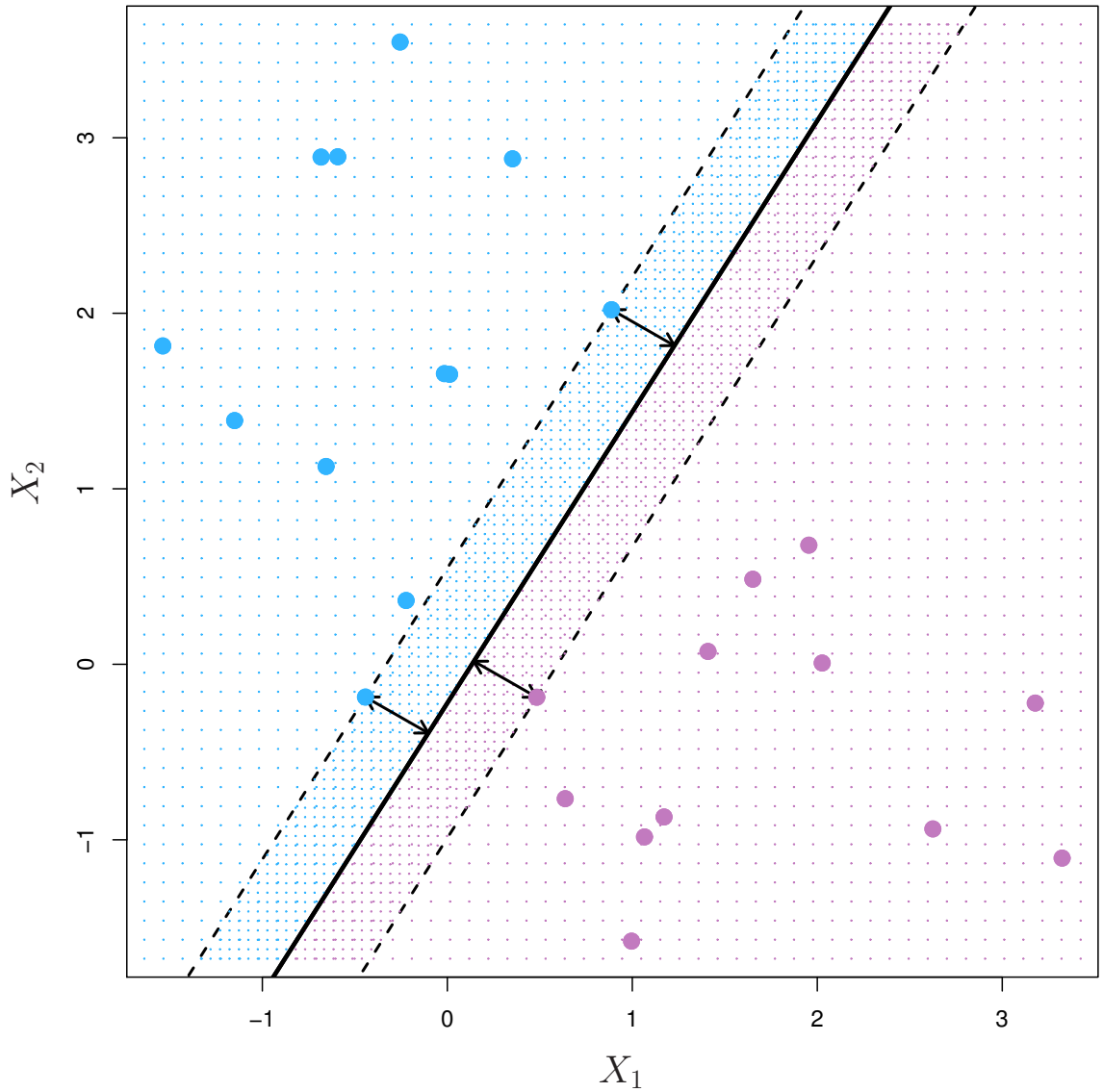




Klasyfikujemy obserwację x^* do klasy $+1$, gdy

$f(x^*) = \beta_0 + \beta_1 x_1^* + \ldots + \beta_p x_p^* > 0$, oraz do klasy -1 gdy znak jest ujemny.

Klasyfikator z maksymalnym marginesem



Klasyfikator z maksymalnym marginesem jako zadanie optymalizacyjne

Dane jest n treningowych obserwacji $x_1, \ldots x_n \in \mathbb{R}^p$ oraz związanych z nimi etykiet klas $y_1, \ldots, y_n \in \{-1, 1\}$.

Maksymalizuj M (dobierając parametry $\beta_0, \beta_1, \ldots \beta_p$ spełniające warunek $\sum_{j=1}^{p} \beta_j^2 = 1$) w wyrażeniu

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M, \qquad \text{dla } i = 1, \ldots n.$$

Dygresja: łatwe wyznaczanie odległości punktu od hiperpłaszczyzny (1)

Niech $\beta = [\beta_1, \ldots, \beta_p]^T \in \mathbb{R}^p$ będzie wektorem. Warunek $\sum_{j=1}^{p} \beta_j^2 = 1$ oznacza, że długość $\|\beta\| = 1$ (długość $\|\cdot\|$ to druga norma $\|\cdot\|_2$).
Hiperpłaszczyzna wyznaczona przez parametry $\beta_0, \beta_1, \ldots, \beta_p$ to zbiór

$$V^* = \{x \in \mathbb{R}^p \mid \beta_0 + \beta^T x = 0\}.$$

Hiperpłaszczyznę tę możemy przesunąć o wektor b tak aby otrzymać podprzestrzeń liniową

$$V = \{x \in \mathbb{R}^p \mid \beta^T x = 0\}$$

Wektor b musi spełniać warunek $\beta^T b = \beta_0$. Wówczas $V = b + V^*$ (bo jeśli $x \in V^*$ to $\beta^T(x + b) = \beta^T x + \beta^T b = -\beta_0 + \beta_0 = 0$, zatem $x + b \in V$. Również na odwrót).

Dygresja: łatwe wyznaczanie odległości punktu od hiperpłaszczyzny (2)

Zatem podprzestrzeń V zawiera wszystkie wektory prostopadłe do β. Aby wyznaczyć odległość dowolnego punktu $a \in \mathbb{R}^p$ od V^*, przesuwamy wszystko o wektor b i zadanie sprowadza się do obliczenia odległości $a+b$ od V. Wystarczy w tym celu zrzutować wektor $a+b$ na prosta L_β wyznaczoną przez wektor β i obliczyć długość tego rzutu.
Rzut $a+b$ na prostą L_β wyraża się wzorem

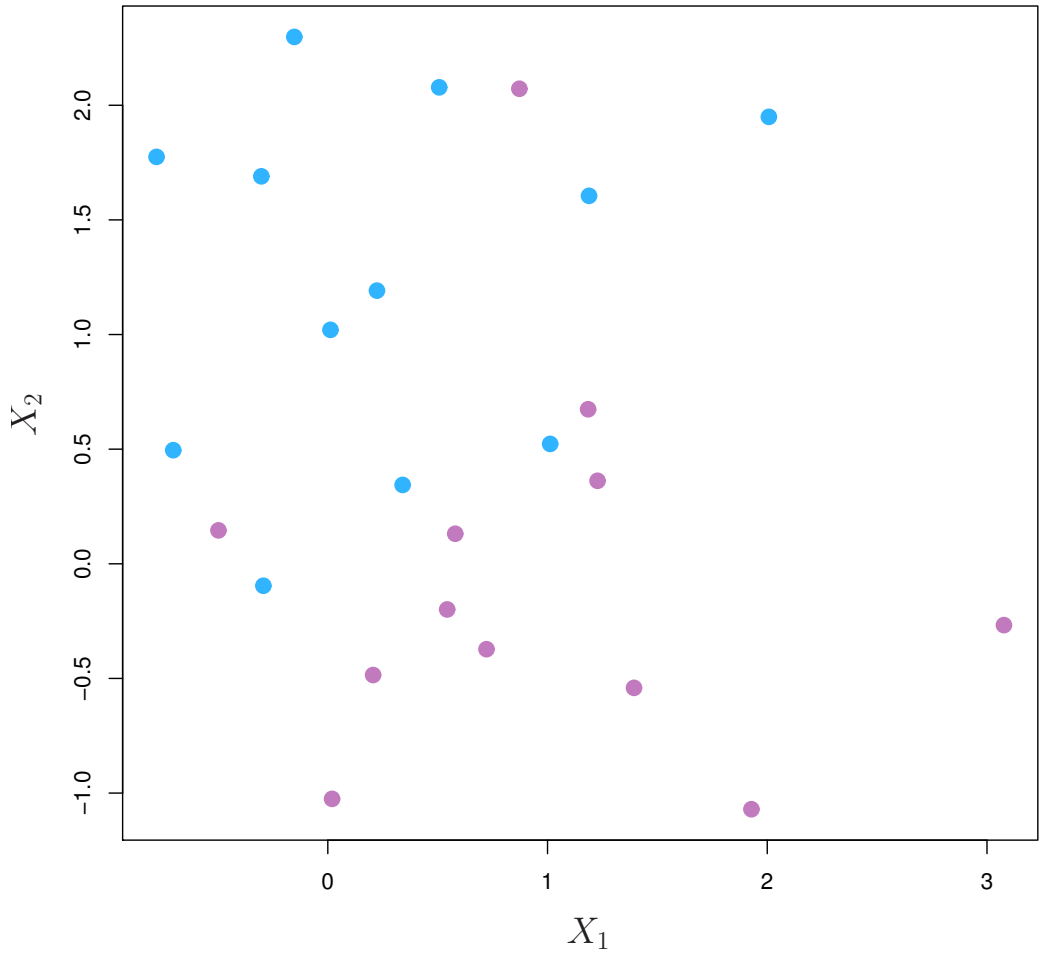
$$\frac{\beta^T(a+b)}{\|\beta\|^2}\beta = (\beta^T a + \beta^T b)\beta = (\beta^T a + \beta_0)\beta.$$

Ponieważ $\|\beta\| = 1$ to długość tego rzutu wynosi

$$||\beta^T a + \beta_0||.$$

Zatem warunek $y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M$ oznacza, że odległość x_i od hiperpłaszczyzny musi być nie mniejsza od M.

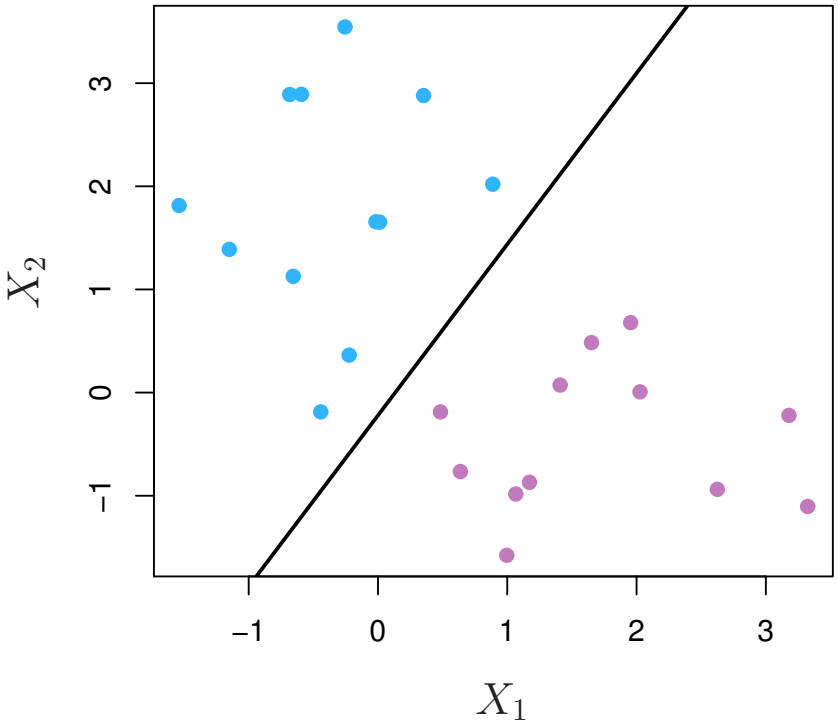
Co zrobic gdy nie istnieje hiperpłaszczyzna rozdzielająca klasy?



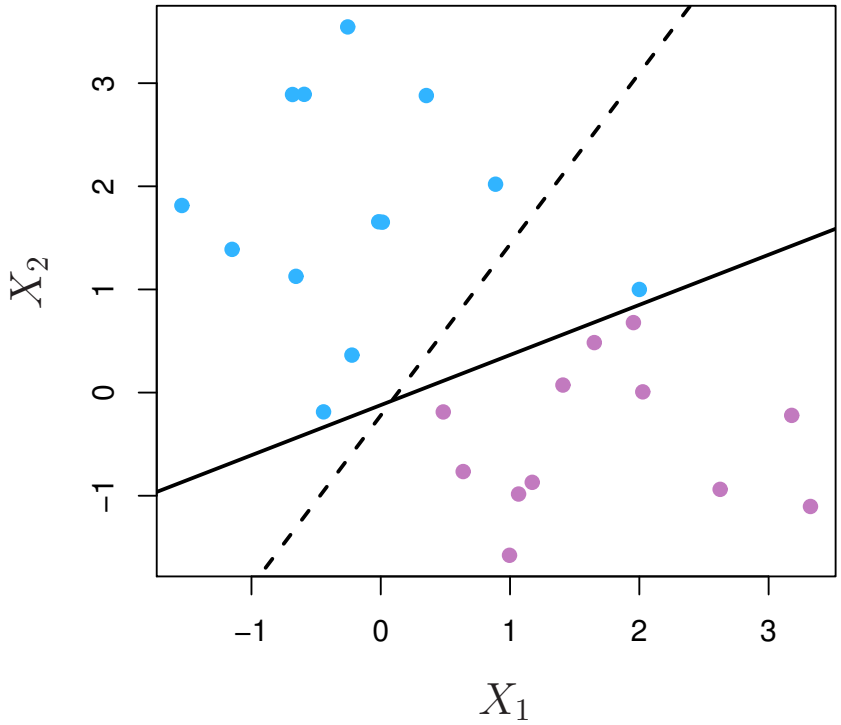
Wprowadzamy **klasyfikator wektorów wspierających**.

Klasyfikator wektorów wspierających

Poprzednia metoda jest bardzo wrażliwa na pojedyncze obserwacje



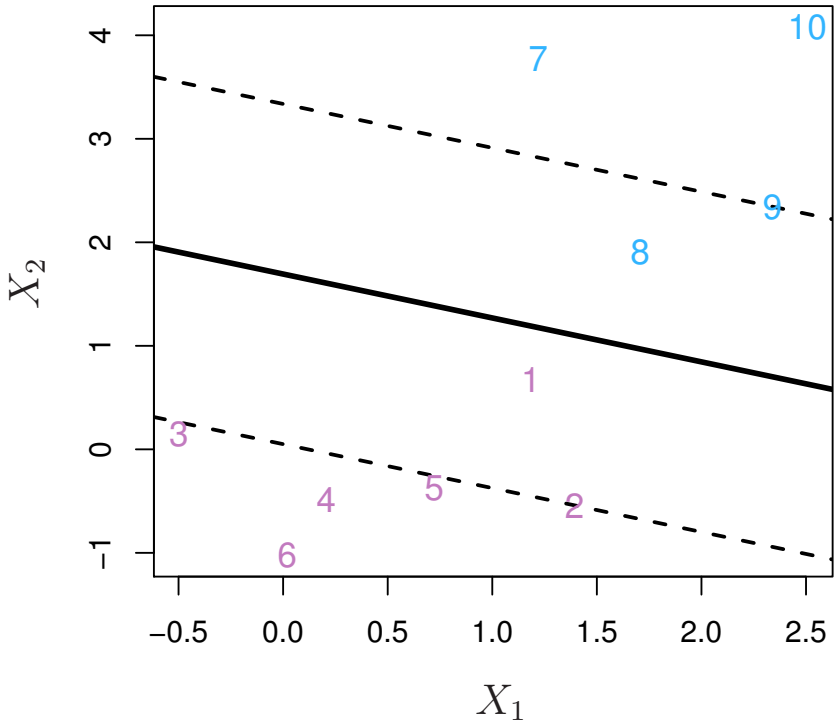
prosta rozdzielająca punkty o maksymalnym marginesie.

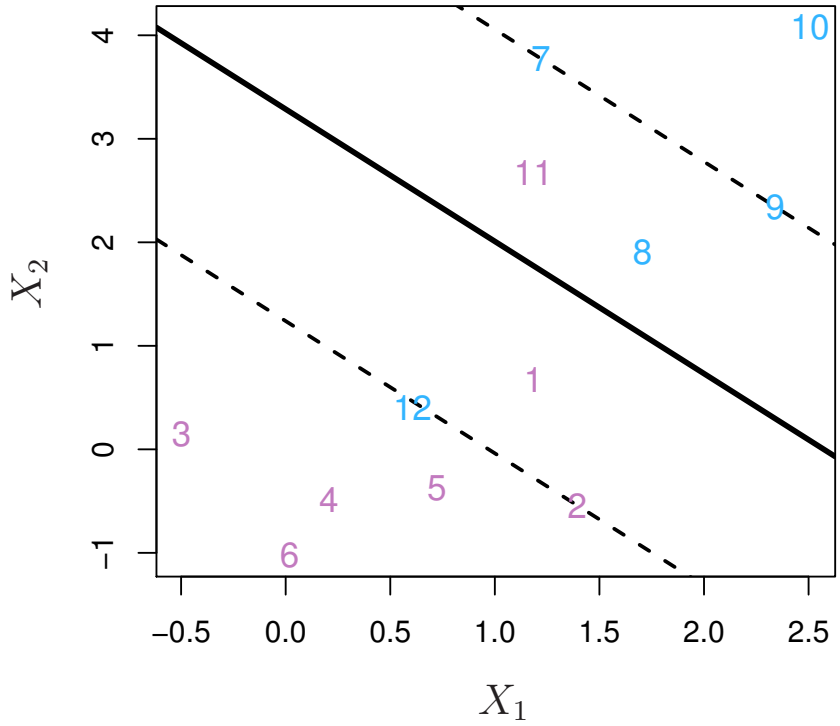


prosta rozdzielająca punkty o maksymalnym marginesie, po dodaniu jednego punktu.

Klasyfikator wektorów wspierających – 'miękki' margines

Ciągła linia: hiperpłaszczyzna rozdzielająca; przerywana: marginesy.





wszystkie obserwacje są po poprawnej stronie hiperpłaszczyzny, ale nie wszystkie są po poprawnej stronie marginesu (8 i 1).

dodane obserwacje 11 i 12 są po niepoprawnej stronie hiperpłaszczyzny.

Klasyfikator wektorów wspierających – zadanie optymalizacyjne

Maksymalizuj wartość M (szerokość marginesu) przez dobór parametrów: $\beta_0, \beta_1, \ldots, \beta_p, \epsilon_1, \ldots, \epsilon_n$ spełniających warunki

$$\sum_{j=1}^{p} \beta_j^2 = 1; \quad \epsilon_i \geq 0 \ \ (\text{dla } i = 1, \ldots, n); \quad \sum_{i=1}^{n} \epsilon_i \leq C$$

($C \geq 0$ jest ustalonym parametrem metody) w wyrażeniu

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M(1 - \epsilon_i), \quad i = 1, \ldots, n$$

Klasyfikator wektorów wspierających – własności

- Po wybraniu wartości parametrów β_j, ϵ_i obserwację x^* klasyfikujemy na podstawie tego po której stronie hiperpłaszczyzny x^* się znajduje.
- Jeśli $\epsilon_i = 0$, to i-ta obserwacja znajduje się po poprawnej stronie marginesu. Jeśli $\epsilon_i > 0$, to i-ta obserwacja znajduje się po niepoprawnej stronie marginesu. Jeśli $\epsilon_i > 1$ to i-ta obserwacja znajduje się po niepoprawnej stronie hiperpłaszczyzny.
- W szczególności, jeśli $C = 0$, to powyższy problem sprowadza się do znajdowania klasyfikatora o maksymalnym marginesie. Jeśli $C > 0$ to nie więcej niż C obserwacji może znajdować się po niepoprawnej stronie hiperpłaszczyzny. Im większe C tym bardziej tolerancyjne jest podejście do zaburzania marginesu i dlatego szerokość marginesu będzie mogła być większa.

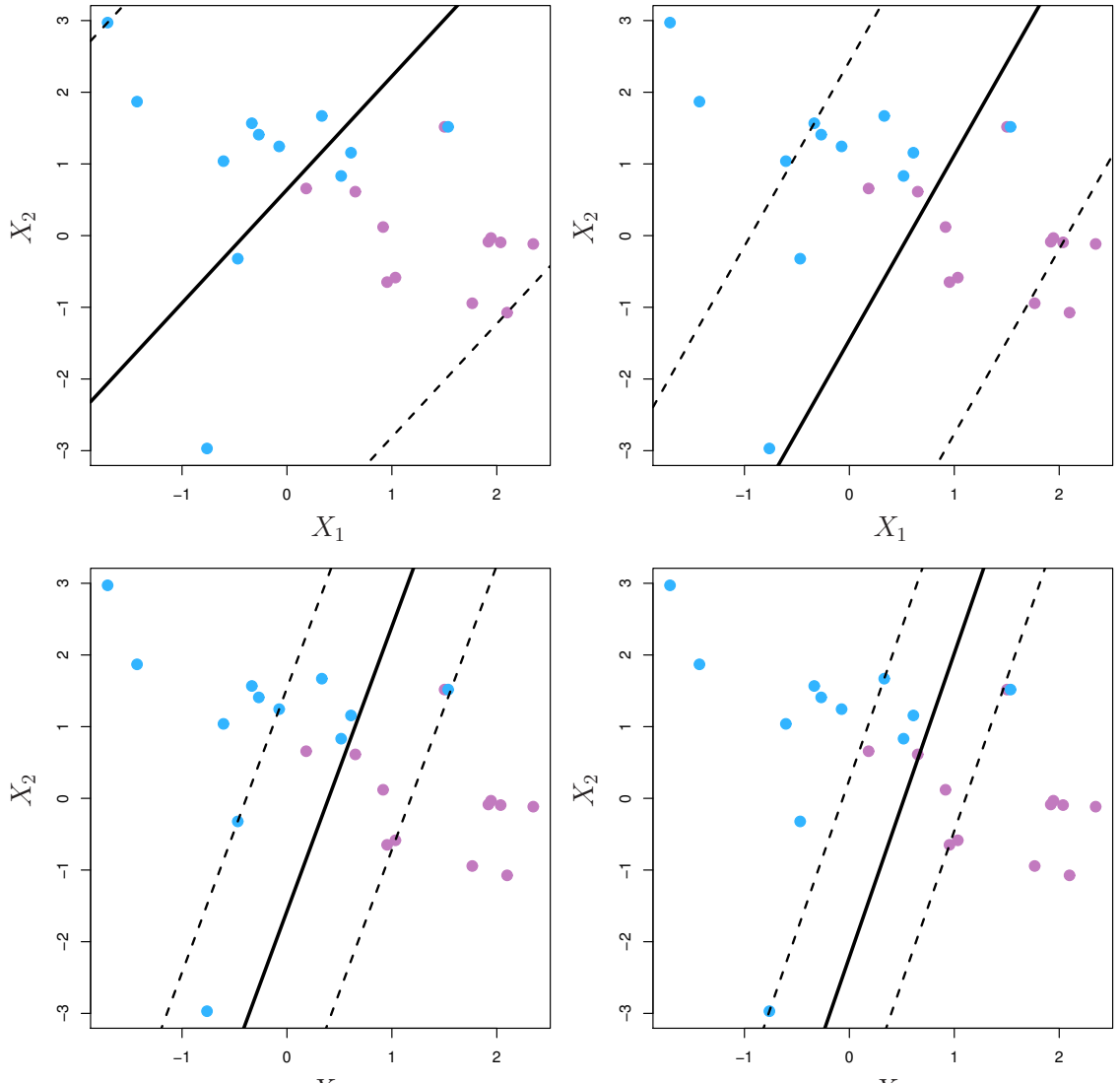
Klasyfikator wektorów wspierających – własności

- Obserwacje, które leżą bezpośrednio na marginesie, lub znajdują się po niepoprawnej stronie marginesu nazywane są **wektorami wspierającymi**.
- Obserwacje nie będące wektorami wspierającymi nie mają żadnego wpływu na wybór optymalnej hiperpłaszczyzny, a zatem na wybór klasyfikatora.

Rola "budżetu na błędy" C

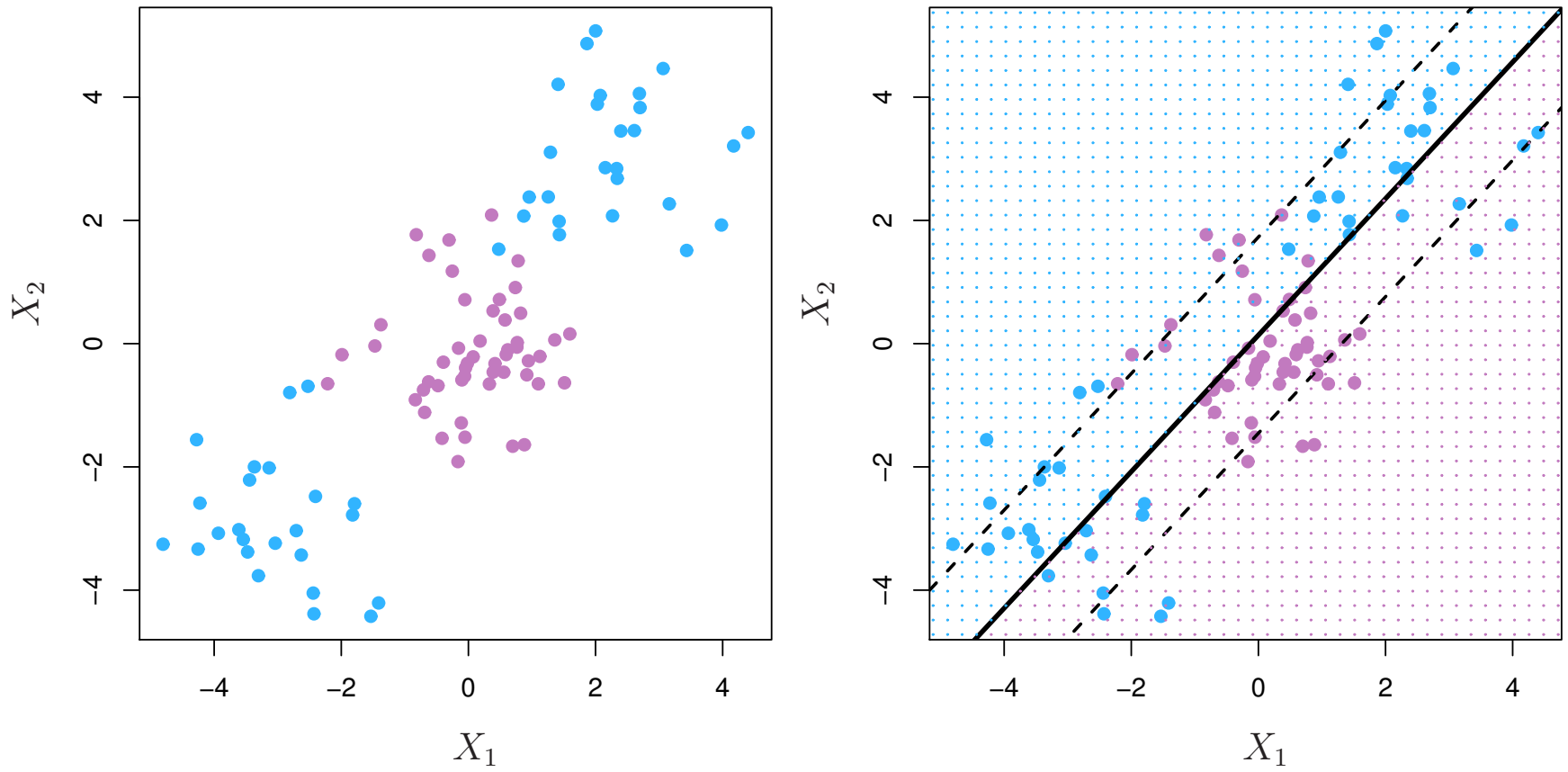
- Zwiększenie wartości C prowadzi zwykle do zwiększenia liczby wektorów wspierających. A zatem w tym przypadku jest więcej obserwacji wykorzystanych do wyznaczenia hiperpłaszczyzny.
- C ustala obciążenie i wariancję klasyfikatora.
  - Dla C małego: wąski margines, zaburzony przez mało obserwacji, a zatem klasyfikator jest dobrze dopasowany do danych treningowych i ma małe obciążenie, a potencjalnie dużą wariancję.
  - Dla C dużego: szeroki margines, zaburzony przez więcej obserwacji, mniej dopasowany, większe obciążenie a mniejsza wariancja.
- Optymalną wartość C znajduje sie poprzez walidację krzyżową.
- Reguła klasyfikacyjna zależy tu jedynie od potencjalnie niedużego zbioru obserwacji (od wektorów wspierających). Wynika stąd mała wrażliwość metody na pojedyncze obserwacje (podobnie do logistycznej regresji).

Klasyfikator wektorów wspierających dla różnych wartości C



Maszyny wektorów wspierających

Klasyfikator wektorów wspierających źle się spisuje dla danych z nieliniową granicą decyzyjną



Klasyfikacja dla problemów z nieliniową granicą decyzyjną

- Zamiast estymować klasyfikator wektorów wspierających w oparciu o obserwacje $X_1, \ldots, X_p$ możemy estymować ten klasyfikator używając $2p$ cech: $X_1, X_1^2, X_2, X_2^2, \ldots X_p, X_p^2$.
- Prowadzi to do następującego problemu optymalizacyjnego:
  - Używając parametrów $\beta_0, \beta_{11}, \beta_{12}, \ldots, \beta_{p1}, \beta_{p2}, \epsilon_1, \ldots, \epsilon_n$ maksymalizuj wartość M przy spełnieniu warunków:
  - $y_i(\beta_0 + \sum_{j=1}^{p} \beta_{j1} x_{ij} + \sum_{j=1}^{p} \beta_{j2} x_{ij}^2) \geq M(1 - \epsilon_i)$ dla $i = 1, \ldots, n$.
  - $\sum_{i=1}^{n} \epsilon_i \leq C$, $\epsilon_i \geq 0$ dla $i = 1, \ldots, n$.
  - $\sum_{j=1}^{p} \sum_{k=1}^{2} \beta_{jk}^2 = 1$.
- Prowadzi to do znaczącego wzrostu liczby parametrów.
- Maszyny wektorów wspierających pozwalają istotnie zwiększyć przestrzeń cech zachowując możliwość wykonywania efektywnych obliczeń.

Inne spojrzenie na klasyfikator wektorów wspierających (1)

- Wprowadzimy oznaczenie na **iloczyn sklarny wektorów**: $\langle a, b \rangle = a^T b = b^T a$.
- Można pokazać, że klasyfikator wektorów wspierających ma postać

$$f(x) = \beta_0 + \sum_{i=1}^{n} \alpha_i \langle x, x_i \rangle,$$

  gdzie parametry α_i są estymowane dla każdej treningowej obserwacji x_i ($x_i = [x_{i1}, \ldots, x_{ip}]^T$ jest i-tym wierszem macierzy X).
- Można też pokazać, że dla estymacji parametrów $\alpha_1, \ldots, \alpha_n$ wystarczy znać $\binom{n}{2} = n(n-1)/2$ iloczynów skalarnych $\langle x_i, x_{i'} \rangle$.

Inne spojrzenie na klasyfikator wektorów wspierających (2)

- Współczynniki α_i we wzorze na funkcję klasyfikującą są zerami dla wektorów, które nie są wspierające.
- Prowadzi to do znacznego zmniejszenia liczby składników w sumie. Jeśli S jest zbiorem indeksów wektorów wspierających, to

$$f(x) = \beta_0 + \sum_{i \in S} \alpha_i \langle x, x_i \rangle.$$

Jądra dla maszyn wektorów wspierających

- Podejście maszyn wektorów wspierających oparte jest na pomyśle zastąpienia iloczynu skalarnego w funkcji klasyfikującej tzw. **funkcją jądra** $K(x_i, x_{i'})$.
- Jądro $K(x_i, x_{i'}) = \langle x_i, x_{i'} \rangle$ jest tzw. **jądrem liniowym**. Jądro to mierzy podobieństwo pomiędzy obserwacjami (w zasadzie korelacja Pearsona). Ale jest wiele innych możliwości wyboru jądra.
- **Jądro wielomianowe** (stopnia d):

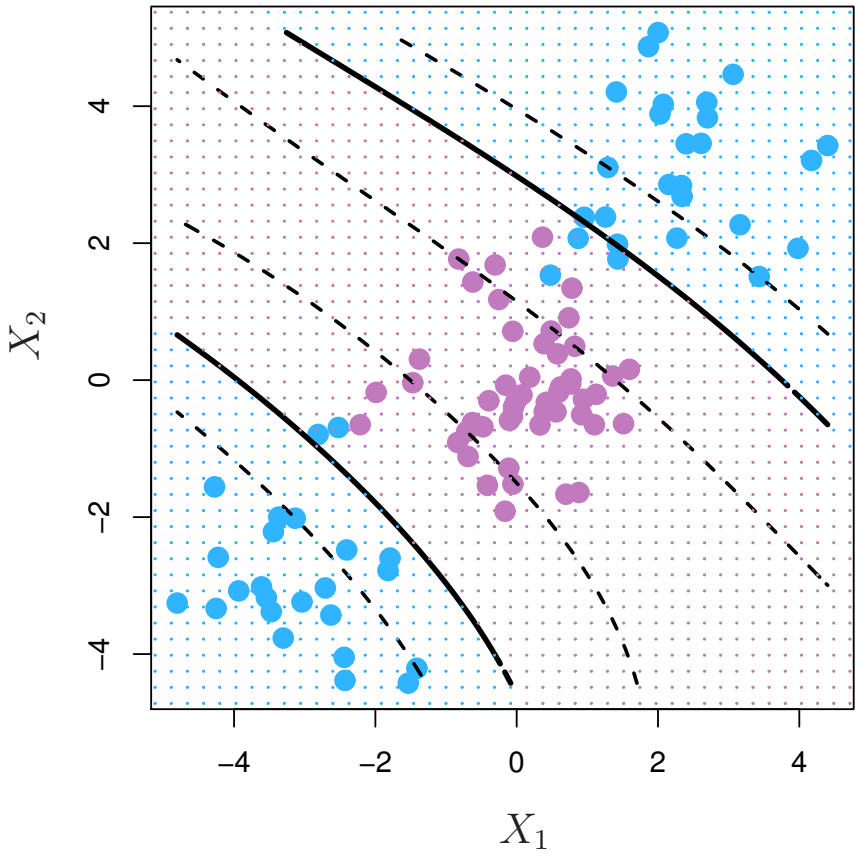
$$K(x_i, x_{i'}) = (1 + \langle x_i, x_{i'} \rangle)^d.$$

- **Jądro radialne**

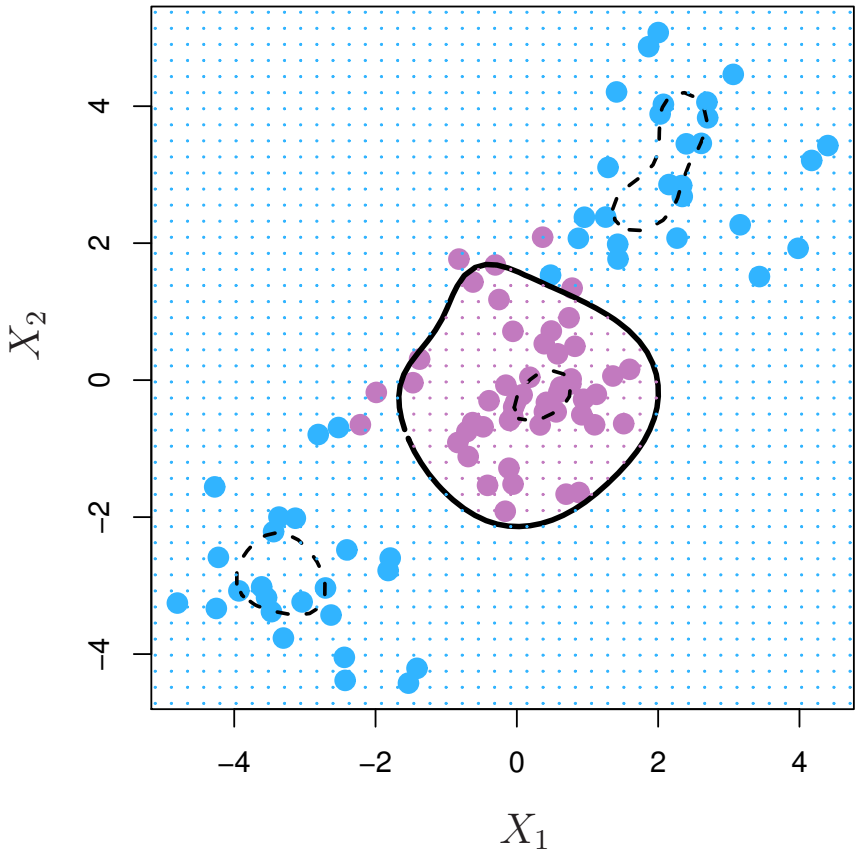
$$K(x_i, x_{i'}) = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2) = \exp(-\gamma \|x_i - x_{i'}\|^2),$$

gdzie γ jest dodatnią stałą.

Wybór jądra jest sprawą kluczową dla jakości klasyfikacji poprzez SVM



jądro sześcienne



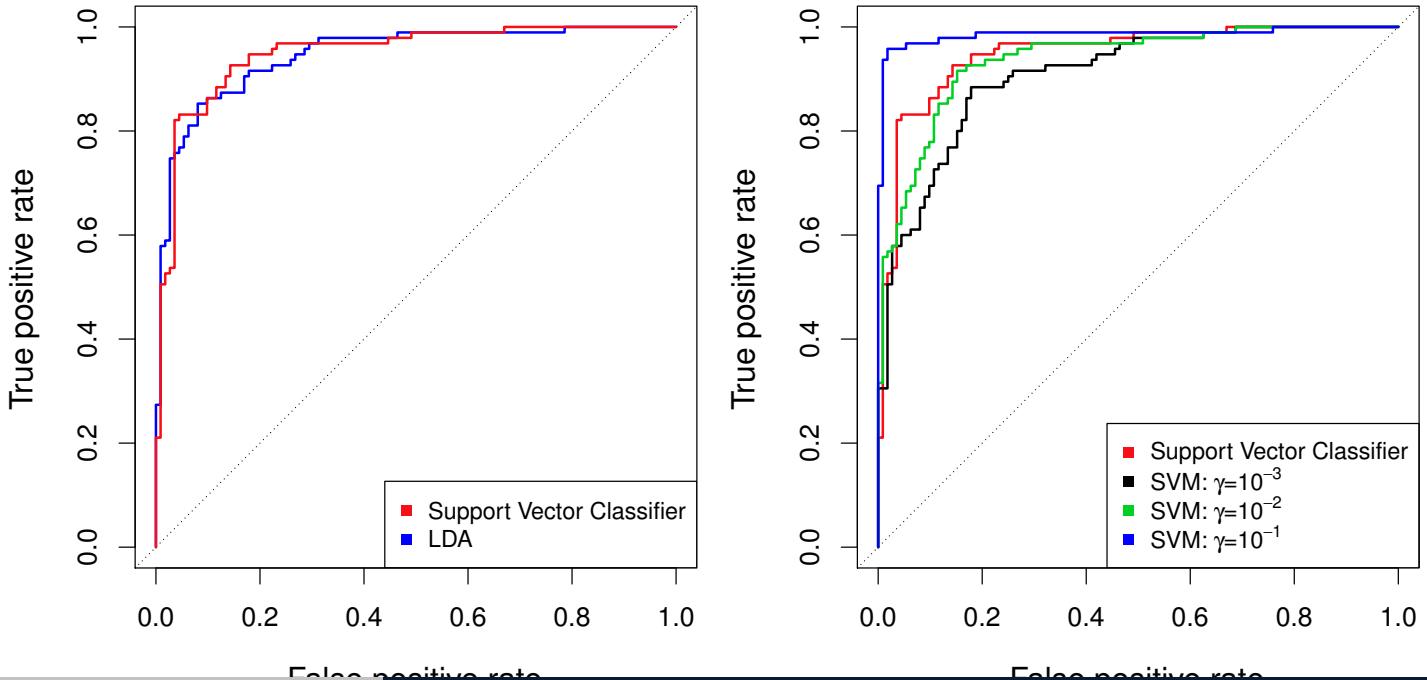
jądro radialne.

Zaleta używania jąder zamiast poszerzania przestrzeni cech

- **Korzyść jest obliczeniowa:** wystarczy obliczyć wartości jądra $K(x_i, x_{i'})$ dla wszystkich $\binom{n}{2}$ różnych par wektorów obserwacji i, i'.
- Poszerzona przestrzeń cech w metodzie SVM jest obecna tylko implicite i w ogólności jest ogromna.
- Przykładowo, dla jądra radialnego ta przestrzeń poszerzona jest nieskończenie wymiarowa, więc i tak nie można tam przeprowadzić obliczeń.

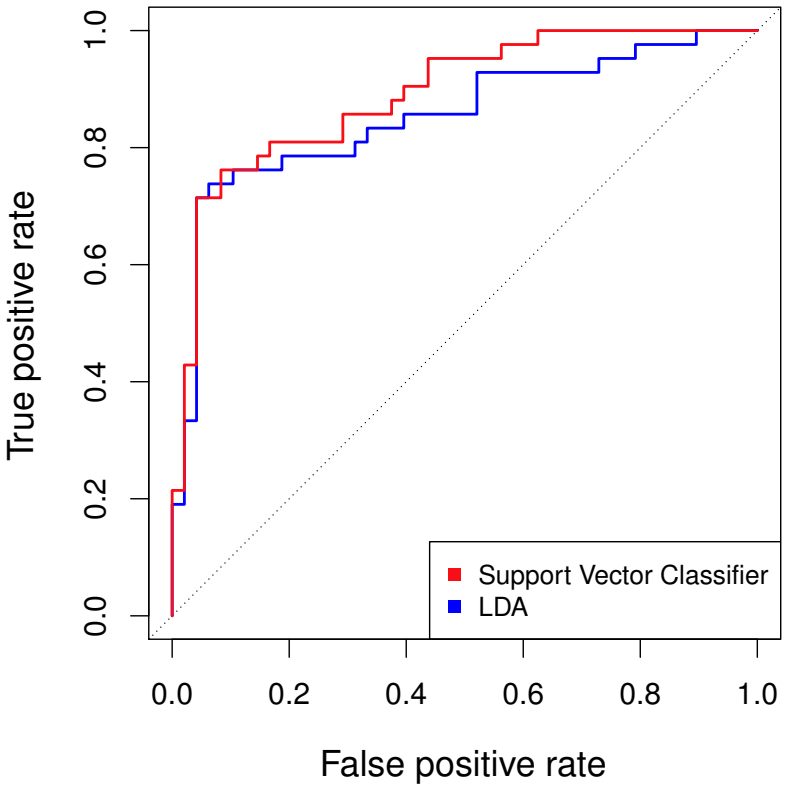
Zastosowanie SVM do danych choroby serca (dopasowanie do danych treningowych)

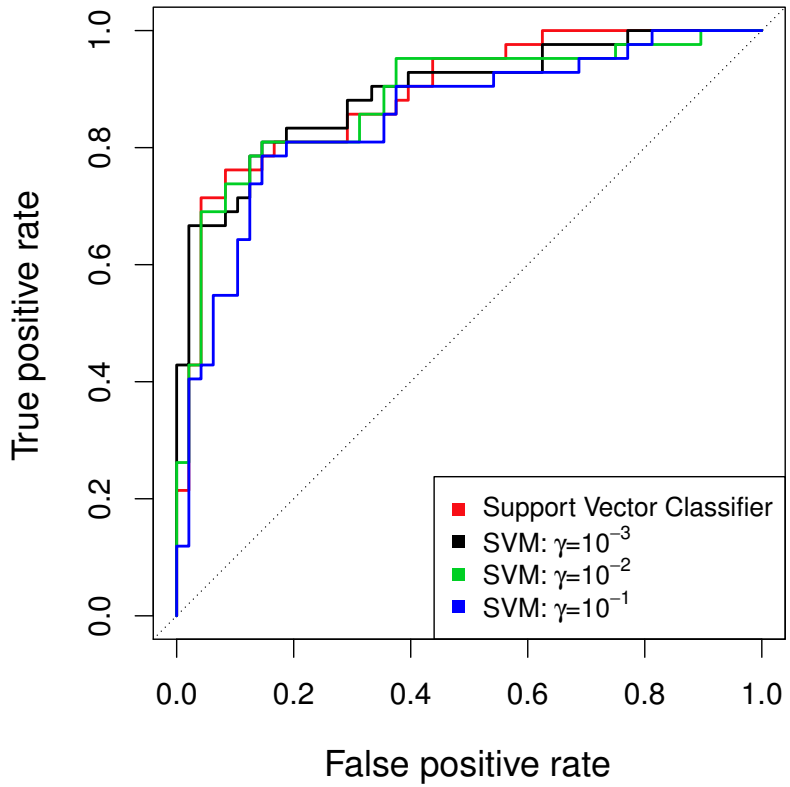
Lewy panel: porównanie klasyfikatora wektorów wspierających z LDA (krzywe ROC). **Prawy panel:** porównanie klasyfikatora wektorów wspierających z SVM opartym na jądrze radialnym dla różnych wartości parametru γ (dla większych wartości γ dopasowanie jest coraz badziej nieliniowe).



Zastosowanie SVM do danych choroby serca (dopasowanie do danych testowych)

Teraz SVM z jądrem radialnym i $\gamma = 10^{-1}$ wypada najgorzej (bardziej elastyczna metoda lepiej dopasowuje się do danych treningowych, co nie musi prowadzić do poprawy jakości dla danych testowych).





SVM dla więcej niż dwóch klas

Mamy $K > 2$ klas.

- (**Klasyfikacja jeden-na-jednego**) Dla każdej z $\binom{K}{2}$ par różnych klas budujemy klasyfikator SVM i dla danych testowych zliczamy ile razy każda z klas została wskazana przez klasyfikator. Zwracamy tę klasę, która najczęściej była wskazana.
- (**Klasyfikacja jeden-na-wszystkich**) Dla każdej z K klas budujemy klasyfikator SVM rozdzielający tę klasę od $K-1$ pozostałych. Niech $\beta_{0k}, \beta_{1k}, \ldots, \beta_{pk}$ będą parametrami funkcji klasyfikującej znalezionymi dla klasy $k \leq K$ (zakodowanej jako +1) w stosunku do pozostałych klas. Dla testowej obserwacji x^* obliczamy wartości $\beta_{0k} + \beta_{1k}x_1^* + \ldots + \beta_{pk}x_p^*$ dla wszystkich $1 \leq k \leq K$. Zwracamy klasę, dla której ta wartość jest największa.

Związek klasyfikatora wektorów wspierających z metodą logistycznej regresji

- Klasyfikator wektorów wspierających można przedstawić jako poszukiwanie funkcji klasyfikującej $f(X) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$ takiej, która minimalizuje wartość wyrażenia

$$\sum_{i=1}^{n} \max[0, 1 - y_i f(x_i)] + \lambda \sum_{j=1}^{p} \beta_j^2,$$

  gdzie $\lambda \geq 0$ jest parametrem związanym ze stałą C w oryginalnym problemie optymalizacyjnym (mała wartość λ odpowiada małej wartości C i na odwrót).
- Zauważmy, że rola $\lambda \sum_{j=1}^{p} \beta_j^2$ jest tu bardzo podobna do roli wyrażenia kary w regresji grzbietowej i kontroluje relację pomiędzy poziomem wariancji a poziomem obciążenia.

Wspólne przedstawienie problemów w postaci 'strata + kara' (Loss + Penalty)

- Postać strata + kara dla zadania optymalizacyjnego polega na minimalizacji (ze względu na parametry $\beta_0, \beta_1, \ldots, \beta_p$) wyrażenia

$$L(X, y, \beta) + \lambda P(\beta),$$

gdzie $L(X, y, \beta)$ to **funkcja straty**, a $P(\beta)$ to **funkcja kary**.

- Przykładowo, dla regresji grzbietowej i metody lasso funkcja straty ma postać

$$L(X, y, \beta) = \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2,$$

a funkcja kary dla regresji grzbietowej to $\sum_{j=1}^{p} \beta_j^2$, a dla lasso to $\sum_{j=1}^{p} |\beta_j|$.

Funkcje straty dla logistycznej regresji oraz dla klasyfikatora wektorów wspierających

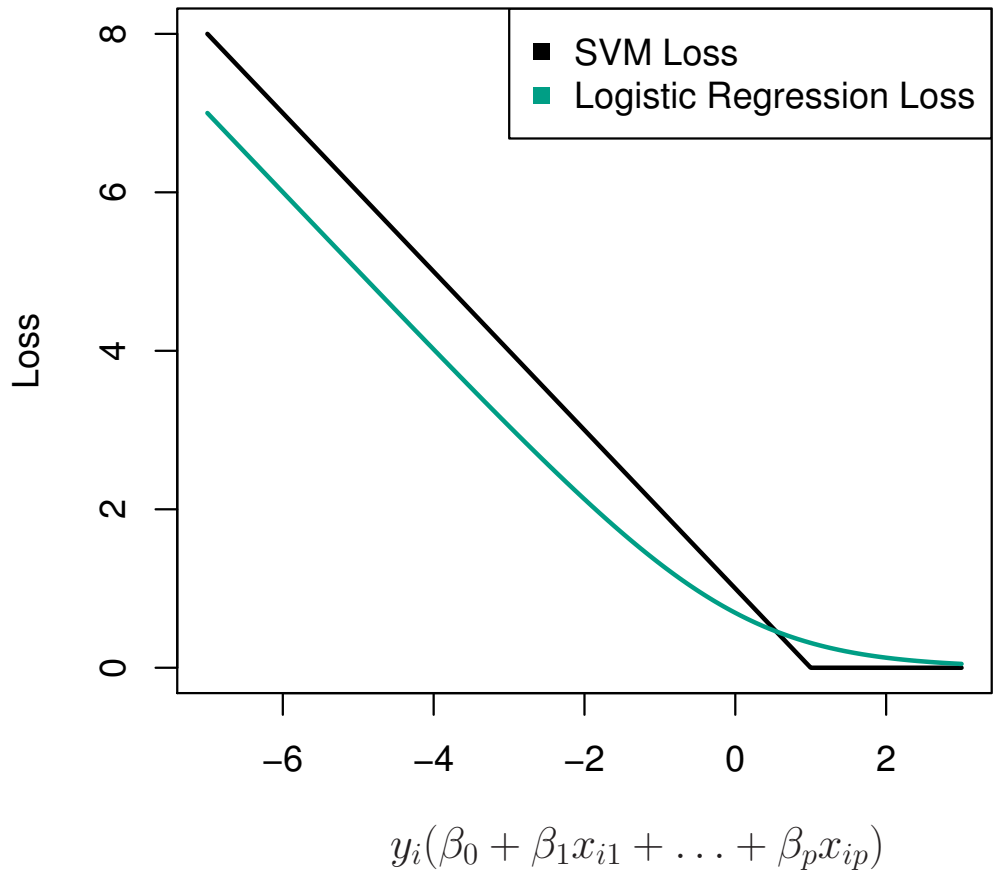
- Dla logistycznej regresji

$$L(X, y, \beta) = \sum_{i=1}^{n} \log(1 + \exp(-y_i(\beta_0 + \beta_1 x_{i1} + \ldots \beta_p x_{ip})))$$

- Dla klasyfikatora wektorów wpierających

$$L(X, y, \beta) = \sum_{i=1}^{n} \max[0, 1 - y_i(\beta_0 + \beta_1 x_{i1} + \ldots \beta_p x_{ip})].$$

Obie funkcje mają podobne przebiegi



Co prowadzi do zbliżonych wyników klasyfikacji.

Cytowania

- Gareth James, Daniela Witten, Trevor Hastie and Rob Tibshirani, *An Introduction to Statistical Learning*