SVM: Maszyny Wektorów Podpierających

Nguyen Hung Son

Outline

Liniony klasyfikator

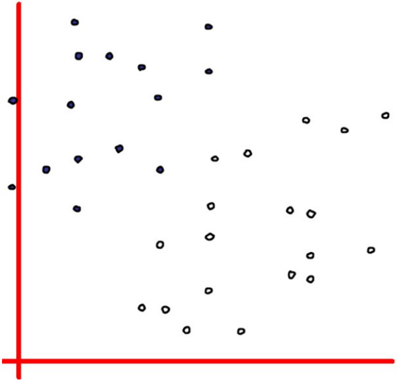
Dana jest próbka treningowa

$$D = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\} \text{ gdzie } y_i \in \{-1, 1\}, \mathbf{x}_i \in \Re^p$$

Liniony klasyfikator

Dana jest próbka treningowa

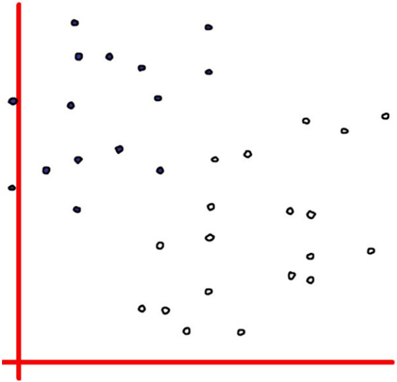
$$D = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\} \text{ gdzie } y_i \in \{-1, 1\}, \mathbf{x}_i \in \Re^p$$



Liniony klasyfikator

Dana jest próbka treningowa

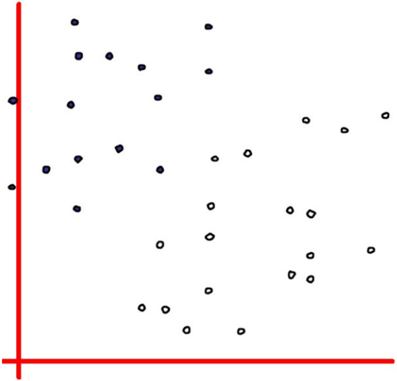
$$D = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\} \text{ gdzie } y_i \in \{-1, 1\}, \mathbf{x}_i \in \Re^p$$



Liniony klasyfikator

Dana jest próbka treningowa

$$D = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\} \text{ gdzie } y_i \in \{-1, 1\}, \mathbf{x}_i \in \Re^p$$



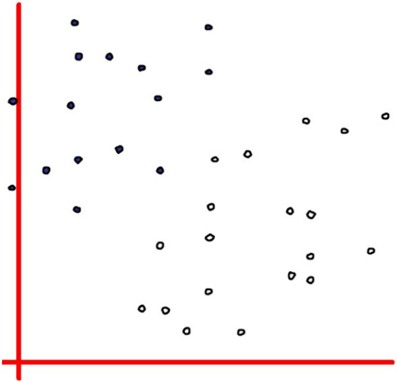
- Klasyfikatorem liniowym nazywamy funkcję

$$f_{\mathbf{w},b}(\mathbf{x}) = \mathrm{sgn}(\mathbf{w} \cdot \mathbf{x} + b)$$

Liniony klasyfikator

Dana jest próbka treningowa

$$D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\} \text{ gdzie } y_i \in \{-1, 1\}, \mathbf{x}_i \in \Re^p$$



- Klasyfikatorem liniowym nazywamy funkcję

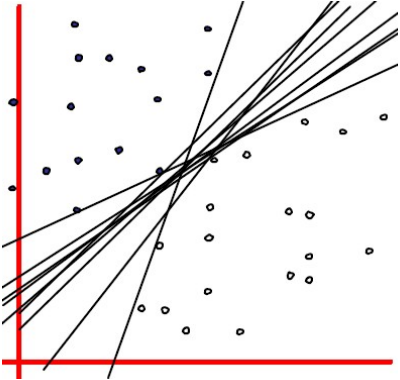
$$f_{\mathbf{w},b}(\mathbf{x}) = \mathrm{sgn}(\mathbf{w} \cdot \mathbf{x} + b)$$

- Próbka D jest liniowo seperowana jeśli istnieje klasyfikator liniowy $f_{\mathbf{w},b}(\ )$ taki, że

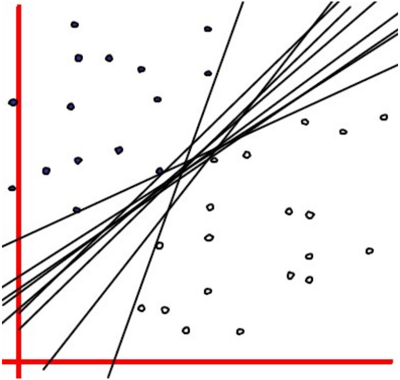
$$\mathbf{w} \cdot \mathbf{x}_i + b \geq +1, \text{ gdy } y_i = +1$$
$$\mathbf{w} \cdot \mathbf{x}_i + b < -1, \text{ gdy } y_i = -1$$

Liniowy klasyfikator

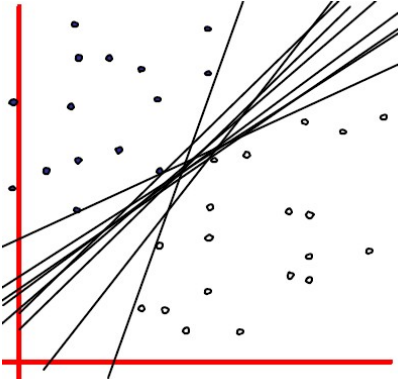
Liniowy klasyfikator



Liniowy klasyfikator



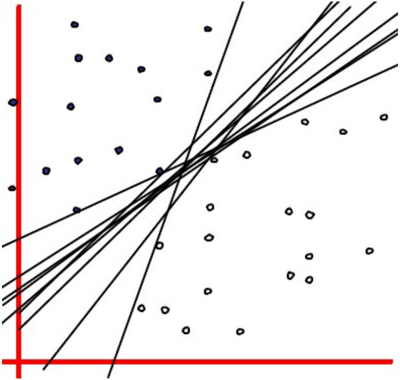
Liniowy klasyfikator



- Uproszczony warunek:

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

dla każdego i.

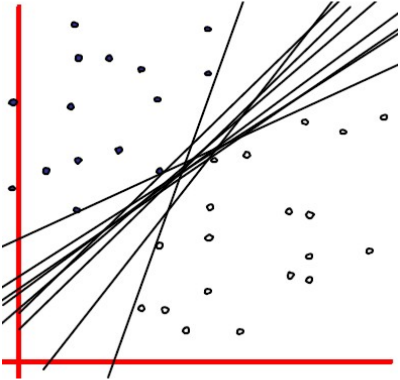


- Uproszczony warunek:

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

  dla każdego i.
- Każda z tych linii dobrze seperuje punkty;

Liniowy klasyfikator



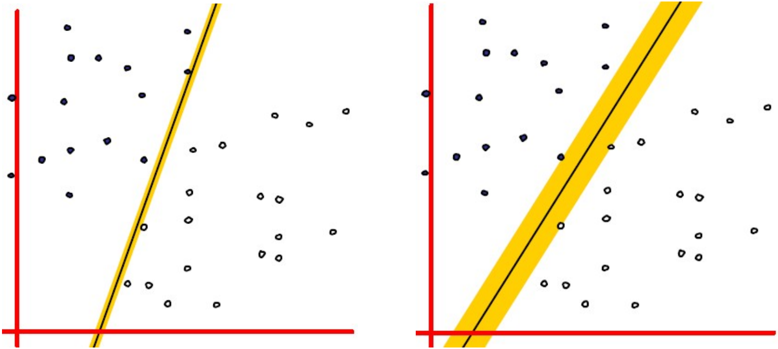
- Uproszczony warunek:

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

  dla każdego i.
- Każda z tych linii dobrze seperuje punkty;
- Która z nich jest najlepsza?

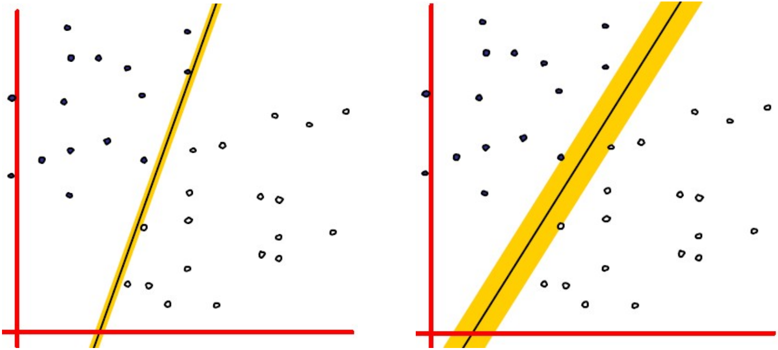
Margines klasyfikatora liniowego

- Margines = szerokość bezpiecznego obszaru po obu stronach hiperpłaszczyzny;



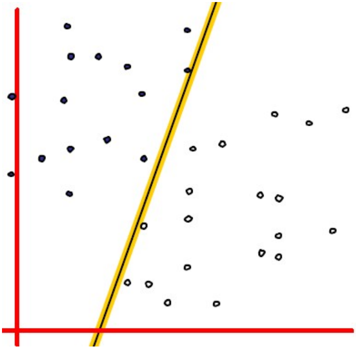
Margines klasyfikatora liniowego

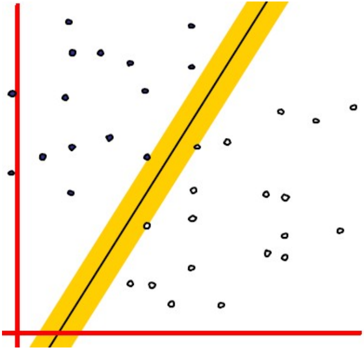
- Margines = szerokość bezpiecznego obszaru po obu stronach hiperpłaszczyzny;
- Margines jest wyznaczony przez 2 hiperpłaszczyzny.



Margines klasyfikatora liniowego

- Margines = szerokość bezpiecznego obszaru po obu stronach hiperpłaszczyzny;
- Margines jest wyznaczony przez 2 hiperpłaszczyzny.
- Chcemy znaleźć klasyfikator z maksymalnym marginesem





Maksymalizujemy $d(H_1, H_2)$ lub minimalizujemy $\frac{||\mathbf{w}||}{2}$.

Wektory podpierające

Maksymalizujemy $d(H_1, H_2)$ lub minimalizujemy $\frac{||\mathbf{w}||}{2}$.

Wektory podpierające

- Niech H_1, H_2 będą brzegami marginesu;

Maksymalizujemy $d(H_1, H_2)$ lub minimalizujemy $\frac{||\mathbf{w}||}{2}$.

Wektory podpierające

- Niech H_1, H_2 będą brzegami marginesu;
- Powinniśmy oprzeć brzegi o punkty z próbki treningowej

Maksymalizujemy $d(H_1, H_2)$ lub minimalizujemy $\dfrac{||\mathbf{w}||}{2}$.

Wektory podpierające

- Niech H_1, H_2 będą brzegami marginesu;
- Powinniśmy oprzeć brzegi o punkty z próbki treningowej
- Te punkty nazywamy *wektorami podpierającymi*

Maksymalizujemy $d(H_1, H_2)$ lub minimalizujemy $\frac{||\mathbf{w}||}{2}$.

Wektory podpierające

- Niech H_1, H_2 będą brzegami marginesu;
- Powinińmy oprzeć brzegi o punkty z próbki treningowej
- Te punkty nazywamy *wektorami podpierającymi*
- Równanie H_1 i H_2:

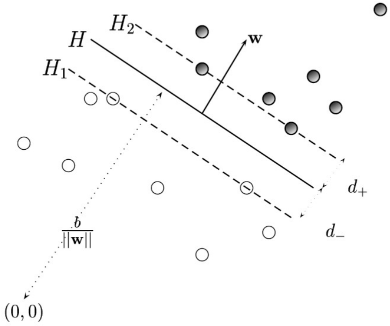
$$H_1 : \mathbf{w} \cdot \mathbf{x} + b = 1$$
$$H_2 : \mathbf{w} \cdot \mathbf{x} + b = -1$$

Maksymalizujemy $d(H_1, H_2)$ lub minimalizujemy $\frac{||\mathbf{w}||}{2}$.

Wektory podpierające

- Niech H_1, H_2 będą brzegami marginesu;
- Powinniśmy oprzeć brzegi o punkty z próbki treningowej
- Te punkty nazywamy *wektorami podpierającymi*
- Równanie H_1 i H_2:

$$H_1 : \mathbf{w} \cdot \mathbf{x} + b = 1$$
$$H_2 : \mathbf{w} \cdot \mathbf{x} + b = -1$$



Odległość między H_1 a H_2:

$$d(H_1, H_2) = \frac{2}{||\mathbf{w}||}$$

Maksymalizujemy $d(H_1, H_2)$ lub minimalizujemy $\frac{||\mathbf{w}||}{2}$.

Problem optymalizacyjny

Zadanie LSVM

Znaleźć $\mathbf{w}$ i b które minimalizują

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

przy ograniczeniach

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0; i = 1, ..., n$$

Q: Jak rozwiązać tego typu zagadnienia?

A: Metodą gradientu?, symulowanego wyżrzania?, odwracanie macierzy? EM? Newton? PROGRAMOWANIE KWADRATOWE?

Pogramowanie kwadratowe QP

- Znaleźć

$$\arg\max_{\mathbf{u}} c + \mathbf{d}^T\mathbf{u} + \frac{\mathbf{u}^T R \mathbf{u}}{2}$$

To zagadnienie jest dobrze zbadane i istnieją bardzo efektywne algorytmy rozwiązujące ten problem

Pogramowanie kwadratowe QP

- Znaleźć

$$\arg\max_{\mathbf{u}} c + \mathbf{d}^T\mathbf{u} + \frac{\mathbf{u}^T R\mathbf{u}}{2}$$

- przy założeniach:

$$A_1\mathbf{u} \leq \mathbf{b}_1$$

To zagadnienie jest dobrze zbadane i istnieją bardzo efektywne algorytmy rozwiązujące ten problem

Pogramowanie kwadratowe QP

- Znaleźć

$$\arg\max_{\mathbf{u}} c + \mathbf{d}^T\mathbf{u} + \frac{\mathbf{u}^T R \mathbf{u}}{2}$$

- przy założeniach:

$$A_1\mathbf{u} \leq \mathbf{b}_1$$

- oraz

$$A_2\mathbf{u} = \mathbf{b}_2$$

To zagadnienie jest dobrze zbadane i istnieją bardzo efektywne algorytmy rozwiązujące ten problem

Metoda Lagrange'a

- Rozwiązywanie zagadnienia LSVM $\Leftrightarrow$ znalezienie *punktu siodłowego* wielomianu Lagrange'a:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}(\mathbf{w} \cdot \mathbf{w}) - \sum_{i=1}^{n} \alpha_i \left\{[(\mathbf{x}_i \cdot \mathbf{w}) - b]y_i - 1\right\}$$

gdzie $\alpha = (\alpha_1, ..., \alpha_n)$ jest wektorem nieujemnych współczynników Lagrange'a

Metoda Lagrange'a

- Rozwiązywanie zagadnienia LSVM $\Leftrightarrow$ znalezienie *punktu siodłowego* wielomianu Lagrange'a:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}(\mathbf{w} \cdot \mathbf{w}) - \sum_{i=1}^{n} \alpha_i \left\{[(\mathbf{x}_i \cdot \mathbf{w}) - b]y_i - 1\right\}$$

  gdzie $\alpha = (\alpha_1, ..., \alpha_n)$ jest wektorem nieujemnych współczynników Lagrange'a

- Punkt siodłowy = maksimum funkcji względem $\alpha_i \geq 0$ i minimum funkcji względem $\mathbf{w}$ i b.

Metoda Lagrange'a (c.d.)

Twierdzenie Karusha-Kuhna-Tuckera: warunkiem koniecznym istnienia punktu siodlowego jest

- zerowanie się gradientu wzg. $\mathbf{w}$, czyli

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

Metoda Lagrange'a (c.d.)

Twierdzenie Karusha-Kuhna-Tuckera: warunkiem koniecznym istnienia punktu siodlowego jest

- zerowanie się gradientu wzg. $\mathbf{w}$, czyli

$$\mathbf{w} = \sum_{i=1}^{n} y_i \alpha_i \mathbf{x}_i \qquad (1)$$

- zerowanie się pochodnej wzg. b, czyli

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \qquad (2)$$

Metoda Lagrange'a (c.d.)

Twierdzenie Karusha-Kuhna-Tuckera: warunkiem koniecznym istnienia punktu siodlowego jest

- zerowanie się gradientu wzg. $\mathbf{w}$, czyli

$$\mathbf{w} = \sum_{i=1}^{n} y_i \alpha_i \mathbf{x}_i \qquad (1)$$

- zerowanie się pochodnej wzg. b, czyli

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \qquad (2)$$

- spełnienie warunku:

$$\alpha_i \{[(\mathbf{x}_i \cdot \mathbf{w}_0) - b_0]y_i - 1\} = 0, \text{ for } i = 1, ..., n \qquad (3)$$

Po uwzględnieniu tych warunków (na istnienie punktu siodłowego) mamy nowy problem optymalizacyjny:

- Znaleźć wektor α będący maksimum funkcji

$$W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{ij=1}^{n} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \tag{4}$$

Po uwzględnieniu tych warunków (na istnienie punktu siodłowego) mamy nowy problem optymalizacyjny:

- Znaleźć wektor α będący maksimum funkcji

$$W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{ij=1}^{n} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \tag{4}$$

- przy ograniczeniach

$$\alpha_i \geq 0, i = 1, .., n \quad oraz \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{5}$$

Metoda Lagrange'a (c.d.)

Niech wektor $(\alpha_1^0, ..., \alpha_n^0)$ będzie rozwiązaniem maksymalizującym (4) przy ograniczeniach (5).

Metoda Lagrange´a (c.d.)

Niech wektor $(\alpha_1^0, ..., \alpha_n^0)$ będzie rozwiązaniem maksymalizującym (4) przy ograniczeniach (5).

- Z (3) wynika, że jeśli $\mathbf{x}_i$ nie jest wektorem podpierającym to $\alpha_i^0 = 0$;

Metoda Lagrange'a (c.d.)

Niech wektor $(\alpha_1^0, ..., \alpha_n^0)$ będzie rozwiązaniem maksymalizującym (4) przy ograniczeniach (5).

- Z (3) wynika, że jeśli $\mathbf{x}_i$ nie jest wektorem podpierającym to $\alpha_i^0 = 0$;
- Równania (4) i (5) odbywają się faktycznie tylko po wektorach podpierających.

Metoda Lagrange'a (c.d.)

Niech wektor $(\alpha_1^0, ..., \alpha_n^0)$ będzie rozwiązaniem maksymalizującym (4) przy ograniczeniach (5).

- Z (3) wynika, że jeśli $\mathbf{x}_i$ nie jest wektorem podpierającym to $\alpha_i^0 = 0$;
- Równania (4) i (5) odbywają się faktycznie tylko po wektorach podpierających.
- Hyperpłaszczyzna rozdzielająca ma postać $\mathbf{w}_0\mathbf{x} - b_0 = 0$ gdzie

$$\mathbf{w}_0 = \sum_{\text{wektory podp.}} y_i\alpha_i^0\mathbf{x}_i \quad b_0 = \frac{1}{2}[(\mathbf{w}_0\cdot\mathbf{x}^{(1)})+(\mathbf{w}_0\cdot\mathbf{x}^{(-1)})]$$

tu $\mathbf{x}^{(1)}$ i $\mathbf{x}^{(-1)}$ są dowolnymi wektorami podpierającymi z każdej z klas.

Ostateczna funkcja decyzyjna:

$$f(\mathbf{x}) = \mathrm{sgn}\Big(\sum_{\text{wektory podp.}} y_i \alpha_i^0 (\mathbf{x}_i \cdot \mathbf{x}) + b_0\Big) \tag{6}$$

Outline

1. Wprowadzenie
2. Brak liniowej separowalności danych
   - Nieznaczna nieseparowalność
   - Zmiana przetrzeń atrybutów
3. Implementacja

Przypadek nieseparowalności

- Założenie o separowalności klas jest nienaturalna;

Przypadek nieseparowalności

- Założenie o separowalności klas jest nienaturalna;
- Modyfikacja:

$$\mathbf{w} \cdot \mathbf{x}_i + b \geq 1 - \xi_i, \text{ gdy } y_i = +1$$
$$\mathbf{w} \cdot \mathbf{x}_i + b \leq -1 + \xi_i, \text{ gdy } y_i = -1$$

gdzie stałe ξ_i spełniają warunki:

$$\xi_i \geq 0, i = 1, ..., n.$$

Przypadek nieseparowalności

- Założenie o separowalności klas jest nienaturalna;
- Modyfikacja:

$$\mathbf{w} \cdot \mathbf{x}_i + b \geq 1 - \xi_i, \text{ gdy } y_i = +1$$
$$\mathbf{w} \cdot \mathbf{x}_i + b \leq -1 + \xi_i, \text{ gdy } y_i = -1$$

gdzie stałe ξ_i spełniają warunki:

$$\xi_i \geq 0, i = 1, ..., n.$$

- Zmodyfikowana funkcja do minimalizacji

$$\frac{||\mathbf{w}||^2}{2} + C \sum_{i=1}^{n} \xi_i$$

C jest stałą "karą" za niespełnienie idealnych warunków.

Przypadek nieseparowalności

Ogólne zadanie SVM

Znaleźć $\mathbf{w}$ i b które minimalizują

$$\frac{||\mathbf{w}||^2}{2} + C\sum_{i=1}^{n} \xi_i$$

przy ograniczeniach

$$\mathbf{w} \cdot \mathbf{x}_i + b \geq 1 - \xi_i,\ \text{gdy}\ y_i = +1$$
$$\mathbf{w} \cdot \mathbf{x}_i + b \leq -1 + \xi_i,\ \text{gdy}\ y_i = -1$$
$$\xi_i \geq 0, i = 1, ..., n.$$

Przypadek nieseparowalności

Można pokazać, że zarówno i w tym przypadku, możemy sprowadzić do problemu

- Znaleźć wektor $(\alpha_1^0, ..., \alpha_n^0)$ będący maksimum funkcji

$$W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{ij=1}^{n} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \quad (7)$$

A następnie stosować funkcję decyzyjną:

Przypadek nieseparowalności

Można pokazać, że zarówno i w tym przypadku, możemy sprowadzić do problemu

- Znaleźć wektor $(\alpha_1^0, ..., \alpha_n^0)$ będący maksimum funkcji

$$W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{ij=1}^{n} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \tag{7}$$

- przy ograniczeniach

$$C \geq \alpha_i \geq 0, i = 1, .., n \quad oraz \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{8}$$

A następnie stosować funkcję decyzyjną:

Przypadek nieseparowalności

Można pokazać, że zarówno i w tym przypadku, możemy sprowadzić do problemu

- Znaleźć wektor $(\alpha_1^0, ..., \alpha_n^0)$ będący maksimum funkcji

$$W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{ij=1}^{n} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \tag{7}$$

- przy ograniczeniach

$$C \geq \alpha_i \geq 0, i = 1, .., n \quad oraz \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{8}$$

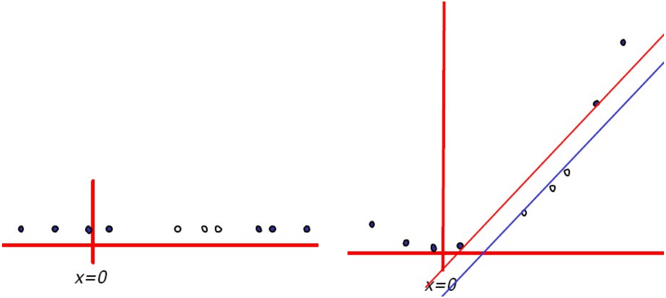
A następnie stosować funkcję decyzyjną:

$$f(\mathbf{x}) = \mathrm{sgn}\left(\sum_{\text{wektory podp.}} y_i \alpha_i^0 (\mathbf{x}_i \cdot \mathbf{x}) + b_0 \right) \tag{9}$$

Zanurzenie obserwacji w bogatszej przetrzeni

- Przekształcimy dane w bogatszej przetrzeni cech:

$$\phi : \Re^p \rightarrow \mathcal{F} \subseteq \Re^N, \quad N \gg p$$

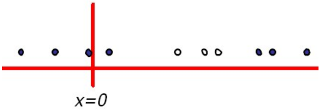


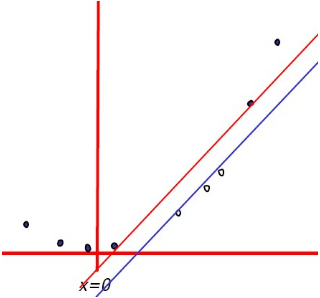
Zanurzenie obserwacji w bogatszej przetrzeni

- Przekształcimy dane w bogatszej przetrzeni cech:

$$\phi : \Re^p \rightarrow \mathcal{F} \subseteq \Re^N, \quad N \gg p$$

- wystarczy zamienić iloczyny skalarne $(\mathbf{x_i} \cdot \mathbf{x_j})$ we wszystkich wzorach na $(\phi(\mathbf{x_i}) \cdot \phi(\mathbf{x_j}))$





- PROBLEM OBLICZENIOWY: czas wykonania iloczynu skalarnego $(\phi(\mathbf{x_i}) \cdot \phi(\mathbf{x_j}))$ wynosi $O(N^2)$

- PROBLEM OBLICZENIOWY: czas wykonania iloczynu skalarnego $(\phi(\mathbf{x_i}) \cdot \phi(\mathbf{x_j}))$ wynosi $O(N^2)$
- TRIK: Kernel functions

$$K((\mathbf{x_i} \cdot \mathbf{x_j})) = (\phi(\mathbf{x_i}) \cdot \phi(\mathbf{x_j}))$$

- PROBLEM OBLICZENIOWY: czas wykonania iloczynu skalarnego $(\phi(\mathbf{x_i}) \cdot \phi(\mathbf{x_j}))$ wynosi $O(N^2)$
- TRIK: Kernel functions

$$K((\mathbf{x_i} \cdot \mathbf{x_j})) = (\phi(\mathbf{x_i}) \cdot \phi(\mathbf{x_j}))$$

- Zmieniona funkcja:

$$\begin{aligned} W(\alpha) &= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{ij=1}^{n} \alpha_i \alpha_j y_i y_j (\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)) \\ &= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{ij=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i \cdot \mathbf{x}_j) \end{aligned}$$

$$\begin{pmatrix} 1 \\ \sqrt{2}a_1 \\ \sqrt{2}a_2 \\ \vdots \\ \sqrt{2}a_m \\ a_1^2 \\ a_2^2 \\ \vdots \\ a_m^2 \\ \sqrt{2}a_1a_2 \\ \sqrt{2}a_1a_3 \\ \vdots \\ \sqrt{2}a_1a_m \\ \sqrt{2}a_2a_3 \\ \vdots \\ \sqrt{2}a_1a_m \\ \vdots \\ \sqrt{2}a_{m-1}a_m \end{pmatrix} \bullet \begin{pmatrix} 1 \\ \sqrt{2}b_1 \\ \sqrt{2}b_2 \\ \vdots \\ \sqrt{2}b_m \\ b_1^2 \\ b_2^2 \\ \vdots \\ b_m^2 \\ \sqrt{2}b_1b_2 \\ \sqrt{2}b_1b_3 \\ \vdots \\ \sqrt{2}b_1b_m \\ \sqrt{2}b_2b_3 \\ \vdots \\ \sqrt{2}b_1b_m \\ \vdots \\ \sqrt{2}b_{m-1}b_m \end{pmatrix} = 1 + \sum_{i=1}^{m} 2a_ib_i + \sum_{i=1}^{m} a_i^2b_i^2 + \sum_{i=1}^{m}\sum_{j=i+1}^{m} 2a_ia_jb_ib_j$$

$$(\mathbf{a}.\mathbf{b}+1)^2$$

$$= (\mathbf{a}.\mathbf{b})^2 + 2\mathbf{a}.\mathbf{b} + 1$$

$$= \left(\sum_{i=1}^{m} a_ib_i\right)^2 + 2\sum_{i=1}^{m} a_ib_i + 1$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{m} a_ib_ia_jb_j + 2\sum_{i=1}^{m} a_ib_i + 1$$

$$= \sum_{i=1}^{m} (a_ib_i)^2 + 2\sum_{i=1}^{m}\sum_{j=i+1}^{m} a_ib_ia_jb_j + 2\sum_{i=1}^{m} a_ib_i + 1$$

Outline

Uczenie SVM

Problem

Problem

Wejście: $D = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\}$; K: funkcja jądrowa.

Uczenie SVM

Problem

Wejście: $D = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\}$; K: funkcja jądrowa.

Wyjście: wektor $(\alpha_1^0, ..., \alpha_n^0)$ będący maksimum funkcji

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

przy ograniczeniach

$$C \geq \alpha_i \geq 0, i = 1, .., n \quad oraz \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

Przykład metody gradientu dla QP

$$\frac{\partial W(\alpha)}{\partial \alpha_i} = 1 - y_i \sum_{j=1}^{l} \alpha_j y_j K(x_i, x_j)$$

$$\alpha_i \leftarrow \alpha_i + \eta \frac{\partial W(\alpha)}{\partial \alpha_i}$$ η **= learning rate**

$parameter\ \omega \in (0,2)$

$$\alpha_i \leftarrow \alpha_i + \frac{\omega}{K(x_i, x_i)} \left(1 - y_i \sum_{j=1}^{l} \alpha_j y_j K(x_i, x_j)\right)$$

Given Training set S and learning rate η

$\alpha \leftarrow 0$

Repeat

for all train set $i = 1\ to\ l$

update α_i

if

$\alpha < 0,\ then\quad \alpha \leftarrow 0$

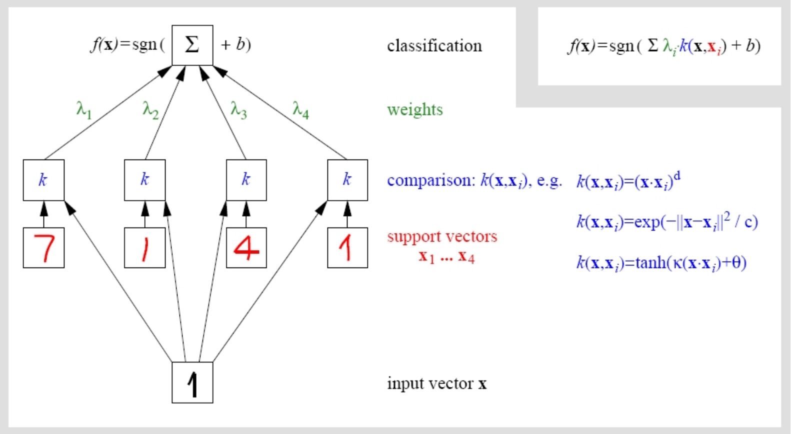
$\alpha > C,\ then\quad \alpha \leftarrow C$

End for

Until stop criterion satisfied

return α

Działanie SVM



$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{\text{wektory podp.}} y_i \alpha_i^0 K(\mathbf{x}_i \cdot \mathbf{x}) + b_0 \right)$$