Algorytm SVM – cz. I

Maciej Kusy

Katedra Podstaw Elektroniki

Spis treści

Algorytm SVM

Algorytm wektorów wspierających (ang. support vector machines, SVM) – koncept w statystycznej teorii uczenia zaproponowany przez **Vladimira Vapnika** w 1995 r.

Stosowany jest do:

- klasyfikacji binarnej (Vapnik, 1995),
- analizy regresji (Drucker et al., 1997).

Wykorzystuje regułę minimalizacji ryzyka strukturalnego (ang. structural risk minimisation, SRM), która polega na minimalizacji górnej granicy wymiaru Vapnika-Chervonenkisa, czyli błędu generalizacji.

Podejście SRM jest lepsze od reguły minimalizacji ryzyka empirycznego (ang. empirical risk minimisation, ERM), gdyż ERM bazuje na minimalizacji błędu, który wyznaczany jest na danych uczących.

Wymiar Vapnika-Chervonenkisa

Wymiar Vapnika-Chervonenkisa (VC) jest miarą mocy zbioru funkcji, które można otrzymać za pomocą algorytmu statystycznej klasyfikacji binarnej.

Jest on zdefiniowany jako liczność największego zbioru punktów, który może zostać *rozbity* (ang. shatter) przez dany algorytm.

Pierwotnie został zdefiniowany przez Vladimira Vapnika i Alexeya Chervonenkisa w (Vapnik i Chervonenkis, 1971).

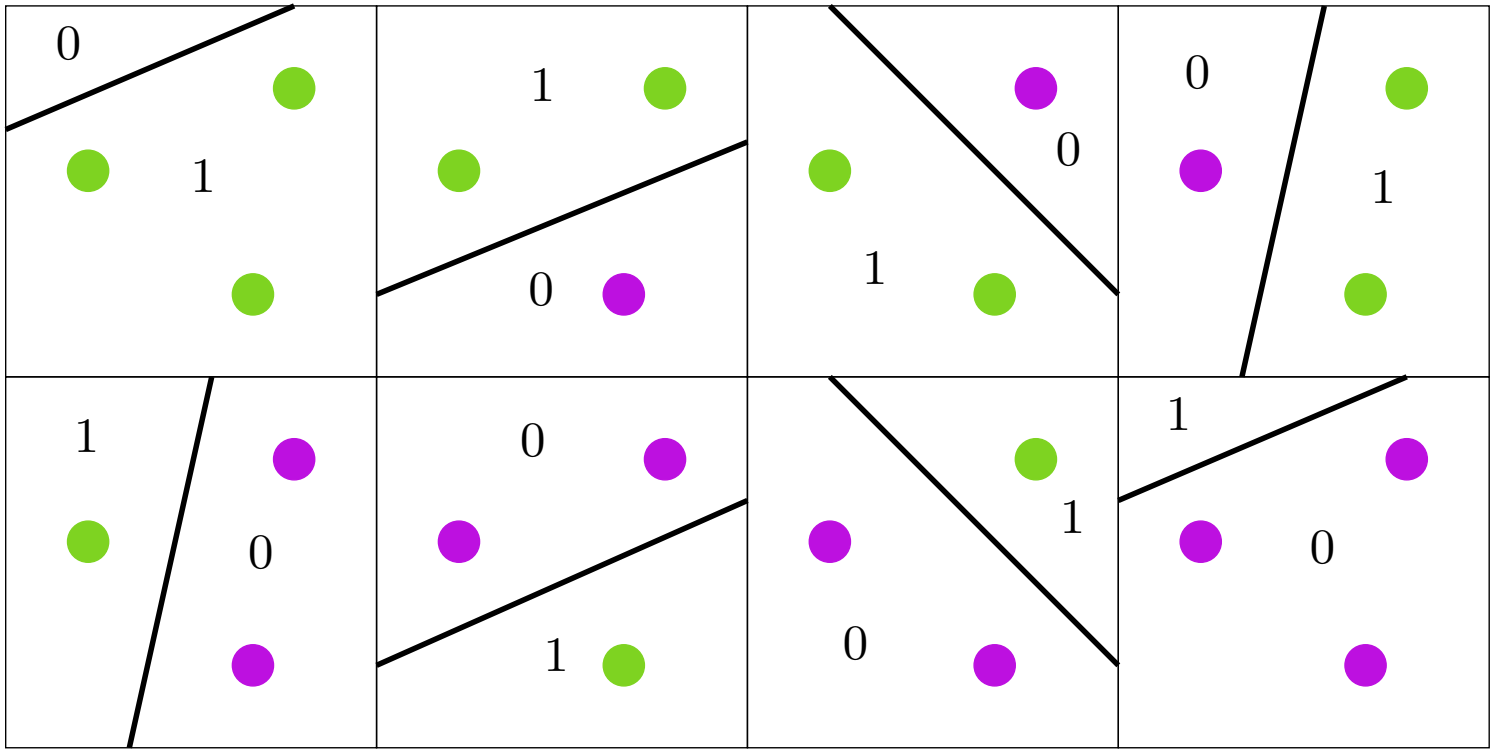
Wymiar VC modelu klasyfikacyjnego:

Binarny model klasyfikacji $M(p)$ o parametrach p rozbija zbiór punktów $\mathbf{x}_1, \ldots, \mathbf{x}_L$, jeśli dla wszystkich etykiet $y_1, \ldots, y_L$ przypisanych do tych punktów istnieje p takie, że model M nie popełnia błędów podczas klasyfikacji tego zbioru danych.

Na przykład dla perceptronu każde **3** niewspółliniowe punkty mogą być rozbite za pomocą prostej.

Wymiar VC – przykład

Łatwo jest znaleźć zbiór **3** punktów, które można poprawnie sklasyfikować za pomocą *liniowej hiperpłaszczyzny, (LH)* bez względu na to, do jakiej należą klasy.



Dla wszystkich $2^3 = 8$ możliwych przypisań istnieje LH, która je realizuje.

Nie jest możliwe znalezienie zbioru 4 etykiet, aby za pomocą LH poprawnie zrealizować wszystkie $2^4 = 16$ możliwe przypisań $\Rightarrow VC = 3$.

Problem liniowo separowalny – założenia

Dane są wektory wejściowe w formie par:

$$\langle \mathbf{X}, \mathbf{y} \rangle = \{ \langle \mathbf{x}_1, y_1 \rangle, \ldots, \langle \mathbf{x}_L, y_L \rangle \}, \tag{1}$$

gdzie:

- $\mathbf{x}_l = [x_{l1}, \ldots, x_{lI}] \in \mathbb{R}^I$ – l-ty wektor zbioru wejściowego,
- $y_l \in \{-1, +1\}$ – etykieta klasy przypisana do $\mathbf{x}_l$.

Dane wejściowe można odseparować za pomocą liniowej hiperpłaszczyzny:

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \tag{2}$$

gdzie $\mathbf{w}$ to współczynnik kierunkowy hiperpłaszczyzny, natomiast b to przesunięcie.

Liniowa separowalność

Hiperpłaszczyzna separująca w postaci kanonicznej musi spełniać następujące warunki:

$$y_l \left[(\mathbf{w} \cdot \mathbf{x}_l) + b \right] \geqslant 1, \quad l = 1, \ldots, L \tag{3}$$

co jest równoważne z:

$$\begin{cases} \mathbf{w} \cdot \mathbf{x}_l + b \geqslant +1 & \text{dla} \quad y_l = +1 \\ \mathbf{w} \cdot \mathbf{x}_l + b \leqslant -1 & \text{dla} \quad y_l = -1 \end{cases} \tag{4}$$

Wszystkie punkty spełniające równość w (4) leżą najbliżej hiperpłaszczyzny $(\mathbf{w}, b)$.

Wyznaczanie odległości pomiędzy najbliższymi punktami

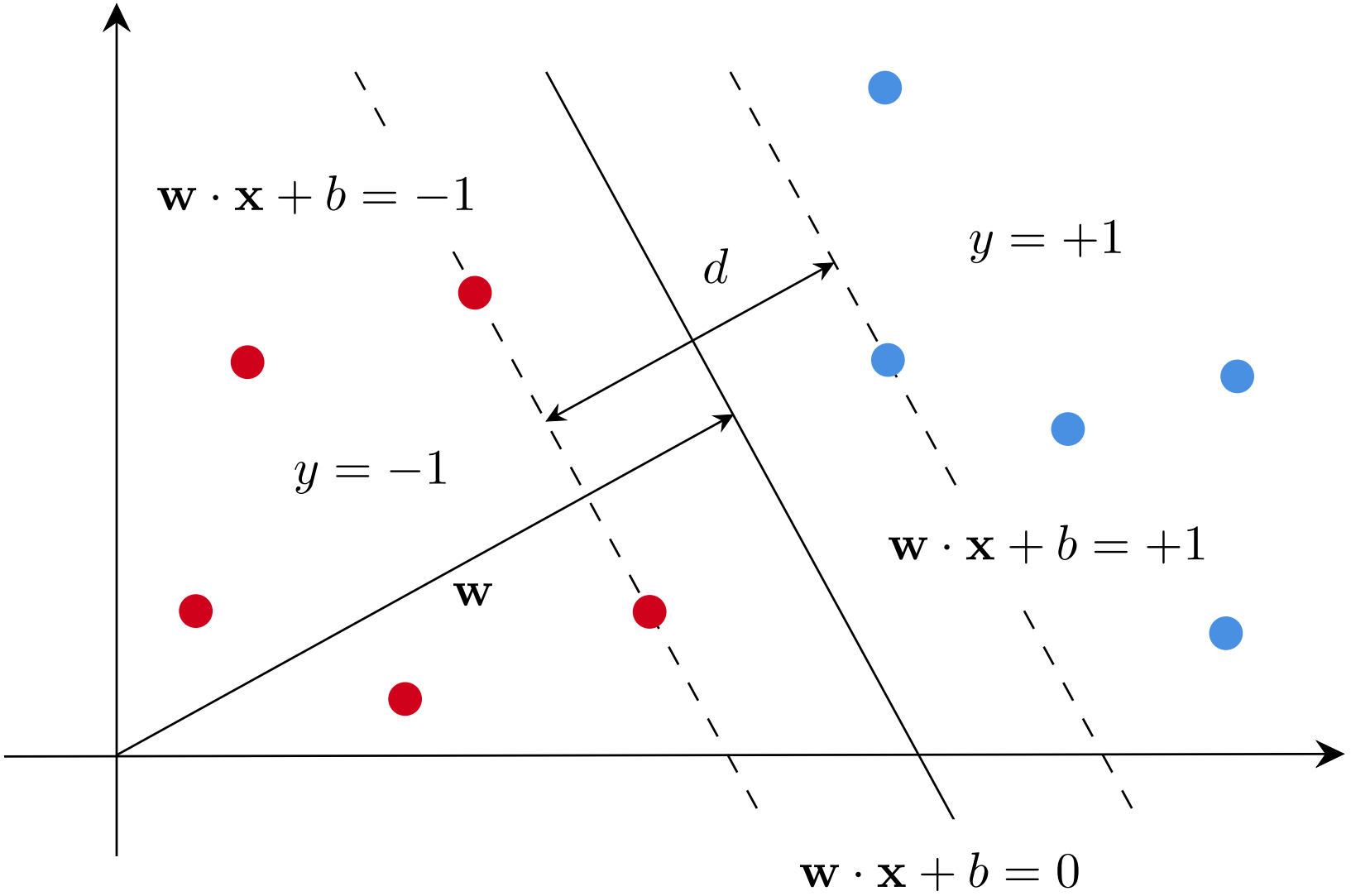
Wiadomo, że:

$$d\left(\mathbf{w}, b; \mathbf{x}\right) = \frac{|\mathbf{w} \cdot \mathbf{x} + b|}{\|\mathbf{w}\|}, \tag{5}$$

stąd odległość pomiędzy punktami z klasy o etykiecie -1 oraz $+1$ leżącymi najbliżej hiperpłaszczyzny $(\mathbf{w}, b)$ można wyznaczyć jako:

$$\begin{aligned}
d &= \min_{\{\mathbf{x}_l : y_l = +1\}} d_l\left(\mathbf{w}, b; \mathbf{x}_l\right) + \min_{\{\mathbf{x}_k : y_k = -1\}} d_k\left(\mathbf{w}, b; \mathbf{x}_k\right) \\
&= \min_{\{\mathbf{x}_l : y_l = +1\}} \frac{|\mathbf{w} \cdot \mathbf{x}_l + b|}{\|\mathbf{w}\|} + \min_{\{\mathbf{x}_k : y_k = -1\}} \frac{|\mathbf{w} \cdot \mathbf{x}_k + b|}{\|\mathbf{w}\|} \\
&= \frac{1}{\|\mathbf{w}\|} \left(\min_{\{\mathbf{x}_l : y_l = +1\}} |\underbrace{\mathbf{w} \cdot \mathbf{x}_l + b}_{1}| + \min_{\{\mathbf{x}_k : y_k = -1\}} |\underbrace{\mathbf{w} \cdot \mathbf{x}_k + b}_{-1}| \right) \\
&= \frac{2}{\|\mathbf{w}\|}
\end{aligned}$$

Graficzna interpretacja odległości d



Optymalna hiperpłaszczyzna separująca

Optymalna hiperpłaszczyzna separująca (ang. optimal separating hyperplane, OSH) – funkcja, która oddziela dane dwóch klas w sposób optymalny.

OSH wyznaczana jest poprzez maksymalizację odległości d, co jest równoważne z rozwiązaniem następującego problemu:

$$\begin{cases} \min \Phi(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{jeżeli:} \\ y_l\left[(\mathbf{w}\cdot\mathbf{x}_l)+b\right] \geqslant 1 \end{cases} \tag{6}$$

dla $l = 1, \ldots, L$.

Równanie (6) stanowi klasyczny problem optymalizacyjny – **problem podstawowy** (ang. primal problem).

Rozwiązanie problemu optymalizacyjnego

Problem podstawowy (6) rozwiązuje się tworząc **Lagrangian**.

Wprowadzenie mnożników Lagrange'a α_l do dla każdego rekordu wejściowego:

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w} \cdot \mathbf{w} - \sum_{l=1}^{L} \alpha_l \left\{ y_l \left[(\mathbf{w} \cdot \mathbf{x}_l) + b \right] - 1 \right\} \tag{7}$$

pod warunkiem $\alpha_l \geqslant 0$, $\forall_{\alpha_l}$.

Minimalizacja Lagrangianu (7) po $\mathbf{w}, b$ oraz maksymalizacja po $\boldsymbol{\alpha}$ wyznacza tzw. punkt siodłowy.

Problem dualny:

$$\max_{\boldsymbol{\alpha}} W(\boldsymbol{\alpha}) = \max_{\boldsymbol{\alpha}} \left\{ \min_{\mathbf{w},b} L(\mathbf{w}, b, \boldsymbol{\alpha}) \right\} \tag{8}$$

Minimalizacja L po $\mathbf{w}$

$$\begin{aligned}
\frac{\partial L}{\partial \mathbf{w}} &= \frac{\partial}{\partial \mathbf{w}}\left(\frac{1}{2}\mathbf{w}\cdot\mathbf{w} - \sum_{l=1}^{L}\alpha_l\left\{y_l\left[(\mathbf{w}\cdot\mathbf{x}_l)+b\right]-1\right\}\right)\\
&= \mathbf{w} - \frac{\partial}{\partial \mathbf{w}}\left(\sum_{l=1}^{L}\left[\alpha_l y_l\left(\mathbf{w}\cdot\mathbf{x}_l\right)+\alpha_l y_l b-\alpha_l\right]\right)\\
&= \mathbf{w} - \sum_{l=1}^{L}\left(\frac{\partial}{\partial \mathbf{w}}\alpha_l y_l\left(\mathbf{w}\cdot\mathbf{x}_l\right)+\frac{\partial}{\partial \mathbf{w}}\alpha_l y_l b-\frac{\partial}{\partial \mathbf{w}}\alpha_l\right)\\
&= \mathbf{w} - \sum_{l=1}^{L}\alpha_l y_l \mathbf{x}_l.
\end{aligned}$$

W związku z tym, że $\frac{\partial L}{\partial \mathbf{w}} = 0$:

$$\mathbf{w} = \sum_{l=1}^{L}\alpha_l y_l \mathbf{x}_l. \tag{9}$$

Minimalizacja L po b

$$\begin{aligned}
\frac{\partial L}{\partial b} &= \frac{\partial}{\partial b}\left(\frac{1}{2}\mathbf{w}\cdot\mathbf{w} - \sum_{l=1}^{L}\alpha_l\left\{y_l\left[(\mathbf{w}\cdot\mathbf{x}_l)+b\right]-1\right\}\right) \\
&= \frac{\partial}{\partial b}\left(\frac{1}{2}\mathbf{w}^2\right) - \frac{\partial}{\partial b}\left(\sum_{l=1}^{L}\left[\alpha_l y_l\left(\mathbf{w}\cdot\mathbf{x}_l\right)+\alpha_l y_l b-\alpha_l\right]\right) \\
&= 0 - \sum_{l=1}^{L}\left(\frac{\partial}{\partial b}\alpha_l y_l\left(\mathbf{w}\cdot\mathbf{x}_l\right)+\frac{\partial}{\partial b}\alpha_l y_l b-\frac{\partial}{\partial b}\alpha_l\right) \\
&= -\sum_{l=1}^{L}\alpha_l y_l.
\end{aligned}$$

W związku z tym, że $\frac{\partial L}{\partial b}=0$:

$$\sum_{l=1}^{L}\alpha_l y_l = 0. \tag{10}$$

Warunki Karusha-Kuhna-Tuckera (KKT)

Dla problemu liniowo separowalnego:

$$\begin{cases} \dfrac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \displaystyle\sum_{l=1}^{L} \alpha_l y_l \mathbf{x}_l = \mathbf{0} \\ \dfrac{\partial L}{\partial b} = \displaystyle\sum_{l=1}^{L} \alpha_l y_l = 0 \\ y_l \left[(\mathbf{w} \cdot \mathbf{x}_l) + b \right] - 1 \geqslant 0 \\ \alpha_l \geqslant 0 \\ \alpha_l \left\{ y_l \left[(\mathbf{w} \cdot \mathbf{x}_l) + b \right] - 1 \right\} = 0 \end{cases} \tag{11}$$

dla $l = 1, \ldots, L$.

Rozwinięcie problemu dualnego

Uwzględniając warunki KKT w (11) oraz $L(\mathbf{w}, b, \boldsymbol{\alpha})$, problem dualny (8) można przekształcić do następującego funkcjonału:

$$
\begin{aligned}
W(\boldsymbol{\alpha}) &= \frac{1}{2}(\mathbf{w}\cdot\mathbf{w}) - \sum_{l=1}^{L}\alpha_l\left\{y_l\left[(\mathbf{w}\cdot\mathbf{x}_l)+b\right]-1\right\} \\
&= \frac{1}{2}(\mathbf{w}\cdot\mathbf{w}) - \mathbf{w}\cdot\sum_{l=1}^{L}\alpha_l y_l\mathbf{x}_l - \sum_{l=1}^{L}\alpha_l y_l b - \sum_{l=1}^{L}\alpha_l(-1) \\
&= \frac{1}{2}(\mathbf{w}\cdot\mathbf{w}) - (\mathbf{w}\cdot\mathbf{w}) - 0 + \sum_{l=1}^{L}\alpha_l = \sum_{l=1}^{L}\alpha_l - \frac{1}{2}(\mathbf{w}\cdot\mathbf{w}) \\
&= \sum_{l=1}^{L}\alpha_l - \frac{1}{2}\left(\sum_{l=1}^{L}\alpha_l y_l\mathbf{x}_l\cdot\sum_{l=1}^{L}\alpha_l y_l\mathbf{x}_l\right) \\
&= \sum_{l=1}^{L}\alpha_l - \frac{1}{2}\sum_{l=1}^{L}\sum_{k=1}^{K}\alpha_l\alpha_k y_l y_k(\mathbf{x}_l\cdot\mathbf{x}_k)
\end{aligned}
$$

Ostateczna postać problemu dualnego

Dla problemu liniowo separowalnego, znalezienie OSH sprowadza się do rozwiązania następującego problemu dualnego:

$$\begin{cases} \max\limits_{\boldsymbol{\alpha}} W(\boldsymbol{\alpha}) = \max\limits_{\boldsymbol{\alpha}} \left\{ \sum\limits_{l=1}^{L} \alpha_l - \frac{1}{2} \sum\limits_{l=1}^{L} \sum\limits_{k=1}^{K} \alpha_l \alpha_k y_l y_k \left(\mathbf{x}_l \cdot \mathbf{x}_k \right) \right\} \\ \text{jeżeli dla } l = 1, \ldots, L : \\ \alpha_l \geqslant 0 \\ \sum\limits_{l=1}^{L} \alpha_l y_l = 0. \end{cases} \tag{12}$$

Powyższa postać dualna to klasyczny problem programowania kwadratowego (ang. quadratic programming, QP).

Optymalny klasyfikator

Rozwiązaniem problemu (12) jest wektor mnożników Lagrange'a $\boldsymbol{\alpha}^\star$. Optymalne współczynniki OSH przyjmują wówczas postać:

$$\mathbf{w}^\star = \sum_{l=1}^{L} \alpha_l^\star y_l \mathbf{x}_l, \tag{13}$$

$$b^\star = -\frac{1}{2}\mathbf{w}^\star \cdot (\mathbf{x}_r + \mathbf{x}_s), \tag{14}$$

gdzie $\mathbf{x}_r$ i $\mathbf{x}_s$ są dowolnymi wektorami wspierającymi z każdej klasy, dla których spełnione są warunki: $\{\alpha_r^\star, \alpha_s^\star\} > 0$, $y_r = +1$, $y_s = -1$.

Optymalny twardy klasyfikator wyznacza się według wzoru:

$$y = \operatorname{sign}(\mathbf{w}^\star \cdot \mathbf{x} + b^\star). \tag{15}$$

Wektory wspierające

Uwzględniając warunek KKT: $\alpha_l^\star \left\{ y_l \left[(\mathbf{w}^\star \cdot \mathbf{x}_l) + b^\star \right] - 1 \right\} = 0$, można zaobserwować, że aby został on spełniony:

$$y_l \left(\mathbf{w}^\star \cdot \mathbf{x}_l + b^\star \right) = 1. \tag{16}$$

Zatem, tylko te wektory $\mathbf{x}_l$, dla których spełniona jest równość (16), mają niezerowe współczynniki Lagrange'a – są to tzw. **wektory wspierające** (ang. support vectors, SV).

Jeśli dane są liniowo separowalne, wszystkie SV będą leżały na odległości d, przez co ich liczba może być bardzo mała.

Pozostałe dane wejściowe mogą zostać usunięte. Ponownie wyznaczona OSH będzie miała identyczny kształt.

Przykład separowalności liniowej

Dane wejściowe:

Zbiór wektorów reprezentujących kwiaty Irysa, $\mathbf{x}_l \in \mathbb{R}^4$, $l = 1, \ldots, 150$:

- trzy klasy:
  - *setosa*: $l_1 = 50$;
  - *versicolor*: $l_2 = 50$;
  - *virginica*: $l_3 = 50$;
- cztery atrybuty wejściowe (w cm):
  - długość płatka,
  - szerokość płatka,
  - długość kielicha,
  - szerokość kielicha.

Dla wizualizacji przykładu, dane klasy *virginica* oraz atrybuty długość i szerokość płatka zostały usunięte.

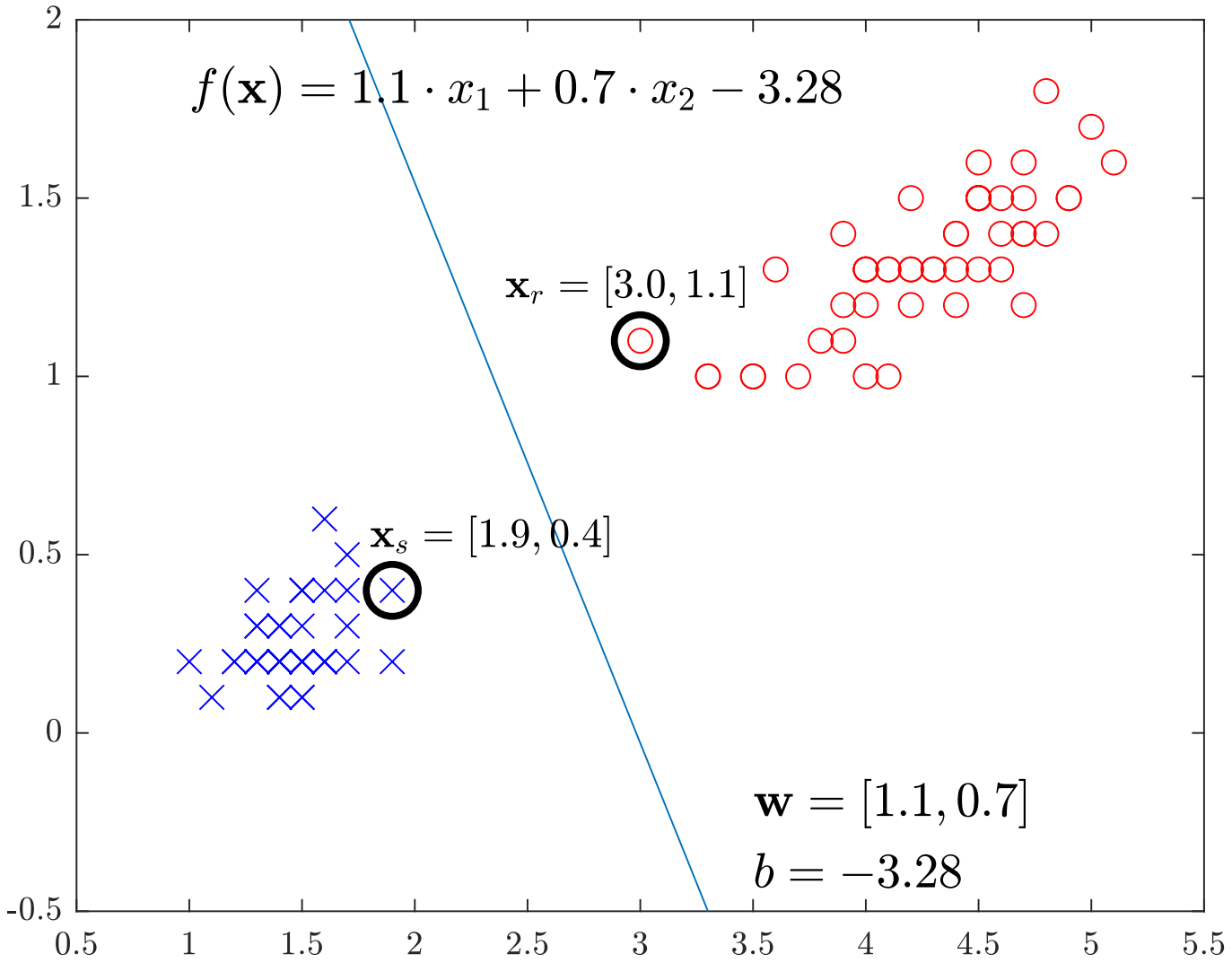
Przykład separowalności liniowej – kod w Matlabie

Zbiór danych kwiatów Irysa: $L = 150$ rekordów, $\mathbf{x}_l \in \mathbb{R}^4$, $\mathbf{t}_l \in \{1, 2, 3\}$.

Do klasyfikacji zostanie wykorzystanych tylko 100 pierwszych rekordów (reprezentują dwie klasy) oraz cechy 3 (długość kielicha) i 4 (szerokość kielicha) do wizualizacji w przestrzeni $\mathbb{R}^2$.

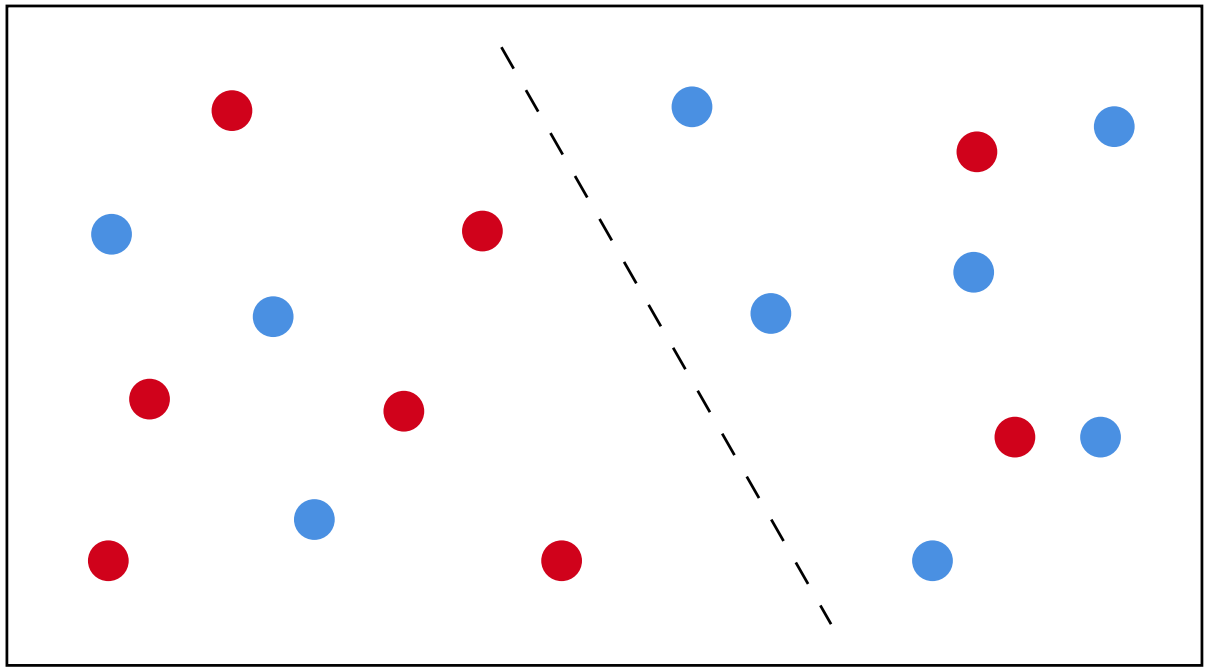
```
load('Iris');
% Wyłuskanie rekordów wejściowych i klas:
X = Iris(1:100,3:4); Y = Iris(1:100,5);
% Zastosowanie liniowego SVM:
SVMModel = fitcsvm(X, Y);
% Współczynnik kierunkowy i przesunięcie:
w = SVMModel.Beta;
b = SVMModel.Bias;
% Rysowanie OSH w zadanym przedziale:
x = 0.8:.05:5.3;
f = (w(1)*x + b)/-w(2);
plot(x,f);
```

Przykład separowalności liniowej – rezultat



Problem nieliniowo separowalny

Dane są wektory wejściowe w formie L par: $\langle \mathbf{x}_1, y_1 \rangle, \ldots, \langle \mathbf{x}_L, y_L \rangle$.



Danych wejściowych nie można odseparować za pomocą liniowej hiperpłaszczyzny $\mathbf{w} \cdot \mathbf{x} + b = 0$.

Konieczne jest użycie złożonej (nieliniowej) funkcji aby wyznaczyć odległość/granicę pomiędzy danymi dwóch klas.

Rozwiązanie problemu nieliniowego

Aby uogólnić OSH do problemu nieliniowego, konieczne jest wprowadzenie nieujemnych zmiennych ξ_l oraz zmodyfikowanie warunku liniowej separowalności (3) do postaci:

$$y_l \left[(\mathbf{w} \cdot \mathbf{x}_l) + b \right] \geqslant 1 - \xi_l, \quad l = 1, \ldots, L \tag{17}$$

przy czym $\xi_l \geqslant 0$.

Uogólniona OSH wyznaczana jest na podstawie wektora $\mathbf{w}$ i minimalizuje ona funkcjonał (**problem podstawowy**):

$$\Phi(\mathbf{w}, \boldsymbol{\xi}) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{l=1}^{L} \xi_l, \tag{18}$$

gdzie C jest stałą regularyzującą przy spełnionym warunku (17).

Rozwiązanie problemu optymalizacyjnego

Rozbudowany Lagrangian:

$$\begin{aligned} L(\mathbf{w}, b, \boldsymbol{\alpha}, \boldsymbol{\xi}, \boldsymbol{\beta}) &= \frac{1}{2}(\mathbf{w}\cdot\mathbf{w}) + C\sum_{l=1}^{L}\xi_l \\ &\quad - \sum_{l=1}^{L}\alpha_l\left\{y_l\left[(\mathbf{w}\cdot\mathbf{x}_l)+b\right]-1+\xi_l\right\} - \sum_{l=1}^{L}\beta_l\xi_l. \end{aligned} \tag{19}$$

gdzie β_l są mnożnikami Lagrange'a wymuszającymi dodatniość ξ_l.

Minimalizacja Lagrangianu (19) po $\mathbf{w}$, b i $\boldsymbol{\xi}$ oraz maksymalizacja po $\boldsymbol{\alpha}$ oraz $\boldsymbol{\beta}$ wyznacza punkt siodłowy.

Problem dualny:

$$\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} W(\boldsymbol{\alpha},\boldsymbol{\beta}) = \max_{\boldsymbol{\alpha},\boldsymbol{\beta}}\left\{\min_{\mathbf{w},b,\boldsymbol{\xi}} L(\mathbf{w}, b, \boldsymbol{\alpha}, \boldsymbol{\xi}, \boldsymbol{\beta})\right\} \tag{20}$$

Warunki KKT dla problemu nieliniowego

Dla każdego $l = 1, \ldots, L$:

$$\begin{cases} \dfrac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \displaystyle\sum_{l=1}^{L} \alpha_l y_l \mathbf{x}_l = \mathbf{0} \\ \dfrac{\partial L}{\partial b} = \displaystyle\sum_{l=1}^{L} \alpha_l y_l = 0 \\ \dfrac{\partial L}{\partial \boldsymbol{\xi}} = C - \alpha_l - \beta_l = 0 \\ y_l \left[(\mathbf{w} \cdot \mathbf{x}_l) + b \right] - 1 + \xi_l \geqslant 0 \\ \alpha_l \left\{ y_l \left[(\mathbf{w} \cdot \mathbf{x}_l) + b \right] - 1 + \xi_l \right\} = 0 \\ \beta_l \xi_l = 0 \\ \alpha_l \geqslant 0, \xi_l \geqslant 0, \beta_l \geqslant 0 \end{cases} \tag{21}$$

Ostateczna postać problemu dualnego

Uwzględniając warunki KKT w (21) oraz $L(\mathbf{w}, b, \boldsymbol{\alpha}, \boldsymbol{\xi}, \boldsymbol{\beta})$, problem dualny (20) można przekształcić do następującego funkcjonału:

$$\begin{cases} \max\limits_{\boldsymbol{\alpha}} W(\boldsymbol{\alpha}) = \max\limits_{\boldsymbol{\alpha}} \left\{ \sum\limits_{l=1}^{L} \alpha_l - \frac{1}{2} \sum\limits_{l=1}^{L} \sum\limits_{k=1}^{K} \alpha_l \alpha_k y_l y_k \left(\mathbf{x}_l \cdot \mathbf{x}_k \right) \right\} \\ \text{jeżeli dla } l = 1, \ldots, L: \\ 0 \leqslant \alpha_l \leqslant C \\ \sum\limits_{l=1}^{L} \alpha_l y_l = 0. \end{cases} \tag{22}$$

Różnica pomiędzy problemami dualnymi (12) i (22) sprowadza się do górnego ograniczenia mnożników α_l współczynnikiem C. C musi być odpowiednio dobrane, gdyż wpływa on na złożoność OSH.

Istotne uwagi

1. Jak można zaobserwować $\max_{\boldsymbol{\alpha}} W(\boldsymbol{\alpha})$ zależy od iloczynu skalarnego dwóch wektorów, tak jak w przypadku problemu liniowo separowalnego.

2. Taki iloczyn może "wygenerować" wyłącznie liniową OSH.

3. Co należy zrobić, aby maksymalizacja (22) rozwiązywała zadanie klasyfikacji danych nieliniowo separowalnych?

4. Czy można uogólnić problem dualny (22) do przypadku, gdy liniowa OSH nie może być zastosowana do odseparowania danych dwóch klas?

Uogólnienie do wielowymiarowej przestrzeni cech

W przypadku, gdy liniowa OSH jest nieodpowiednia do odseparowania danych, konieczne jest użycie nieliniowego odwzorowania $\mathbf{\Psi} : \mathbb{R}^I \to \mathbf{F}$, tj. z przestrzeni wejściowej do przestrzeni cech o większym wymiarze.

W oryginalnej I-wymiarowej przestrzeni wejściowej nie jest możliwe znalezienie OSH, natomiast w przestrzeni cech jest to możliwe.

Nowa OSH nosi nazwę uogólnionej optymalnej hiperpłaszczyzny separującej (ang. generalized separating hyperplane, GSH).

Funkcjonał (22) przyjmuje wówczas postać:

$$\max_{\boldsymbol{\alpha}} \left\{ \sum_{l=1}^{L} \alpha_l - \frac{1}{2} \sum_{l=1}^{L} \sum_{k=1}^{K} \alpha_l \alpha_k y_l y_k \mathbf{\Psi}\left(\mathbf{x}_l\right) \cdot \mathbf{\Psi}\left(\mathbf{x}_k\right) \right\}. \tag{23}$$

Przykład odwzorowania $\mathbf{\Psi}$

Dane są dwa wektory wejściowe w $\mathbb{R}^2$: $\mathbf{u} = [u_1, u_2]$ i $\mathbf{v} = [v_1, v_2]$.
Należy wyznaczyć $\mathbf{\Psi} : \mathbb{R}^2 \to \mathbb{R}^6$ takie, że: $\Psi(\mathbf{u}) \cdot \Psi(\mathbf{v}) = ((\mathbf{u} \cdot \mathbf{v}) + 1)^2$.

Rozwiązanie:

$$\begin{aligned}
\Psi(\mathbf{u}) \cdot \Psi(\mathbf{v}) &= ((\mathbf{u} \cdot \mathbf{v}) + 1)^2 \\
&= ([u_1, u_2] \cdot [v_1, v_2] + 1)^2 = (u_1 v_1 + u_2 v_2 + 1)^2 \\
&= u_1^2 v_1^2 + 2 u_1 v_1 u_2 v_2 + u_2^2 v_2^2 + 2 \cdot (u_1 v_1 + u_2 v_2) + 1 \\
&= u_1^2 v_1^2 + u_2^2 v_2^2 + 2 u_1 v_1 u_2 v_2 + 2 u_1 v_1 + 2 u_2 v_2 + 1.
\end{aligned}$$

Stąd wynika, że:

$$\Psi(\mathbf{u}) = \left[u_1^2, u_2^2, \sqrt{2} u_1 u_2, \sqrt{2} u_1, \sqrt{2} u_2, 1\right],$$

$$\Psi(\mathbf{v}) = \left[v_1^2, v_2^2, \sqrt{2} v_1 v_2, \sqrt{2} v_1, \sqrt{2} v_2, 1\right],$$

Przez co, równanie GSH przyjmuje postać:

$$w_1 u_1^2 + w_2 u_2^2 + w_3 u_1 u_2 + w_4 u_1 + w_5 u_2 + b = 0. \tag{24}$$

Praktyczna realizacja odwzorowania do przestrzeni cech

Odwzorowanie $\mathbf{\Psi} : \mathbb{R}^I \to \mathbf{F}$ może zostać zrealizowane za pomocą funkcji jądra. Ma ona umożliwić przeprowadzenie operacji w przestrzeni wejściowej a nie w przestrzeni o wyższym wymiarze.

Przez to iloczyn skalarny $\Psi(\mathbf{u}) \cdot \Psi(\mathbf{v})$ nie musi być jawnie wyznaczany.

Można przyjąć, że iloczyn skalarny w przestrzeni cech posiada równoważną funkcję jądra w przestrzeni wejściowej:

$$\mathcal{K}(\mathbf{u}, \mathbf{v}) = \Psi(\mathbf{u}) \cdot \Psi(\mathbf{v}) \tag{25}$$

jeżeli funkcja $\mathcal{K}$ jest dodatnio określona i spełnia warunki Mercera:

$$\begin{gathered} \mathcal{K}(\mathbf{u}, \mathbf{v}) = \sum_{m=1}^{\infty} \alpha_m \Psi(\mathbf{u}) \Psi(\mathbf{v}) \\ \int \int \mathcal{K}(\mathbf{u}, \mathbf{v}) g(\mathbf{u}) g(\mathbf{v}) \, d\mathbf{u} d\mathbf{v} > 0, \quad \int g^2(\mathbf{u}) \, d\mathbf{u} < \infty. \end{gathered} \tag{26}$$

Przykłady funkcji jądra

Najbardziej popularne funkcje jądra spełniające warunki Mercera:

- kernel wielomianowy:

$$\mathcal{K}(\mathbf{u},\mathbf{v}) = [(\mathbf{u}\cdot\mathbf{v})+1]^d, \quad d = 2,3\ldots,D, \tag{27}$$

- kernel o rozkładzie Gaussa:

$$\mathcal{K}(\mathbf{u},\mathbf{v}) = \exp\left(-\frac{(\mathbf{u}-\mathbf{v})^2}{2\sigma^2}\right), \tag{28}$$

- kernel wykładniczy:

$$\mathcal{K}(\mathbf{u},\mathbf{v}) = \exp\left(-\frac{\|\mathbf{u}-\mathbf{v}\|}{2\sigma^2}\right), \tag{29}$$

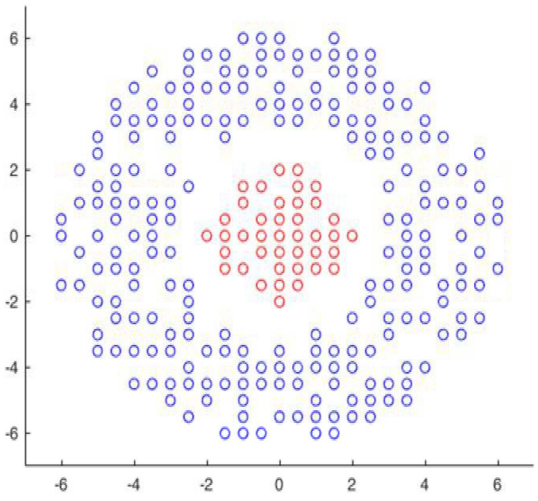
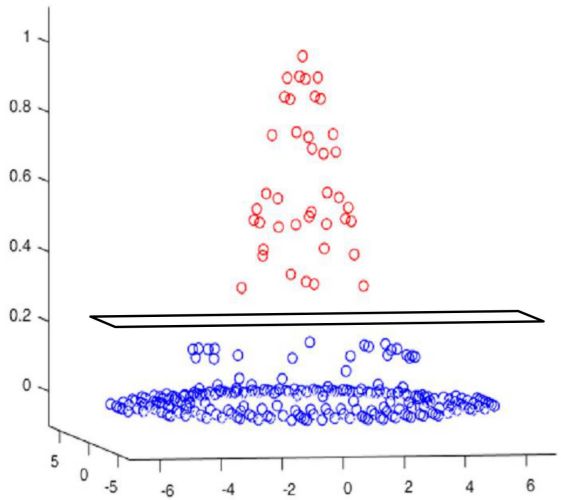
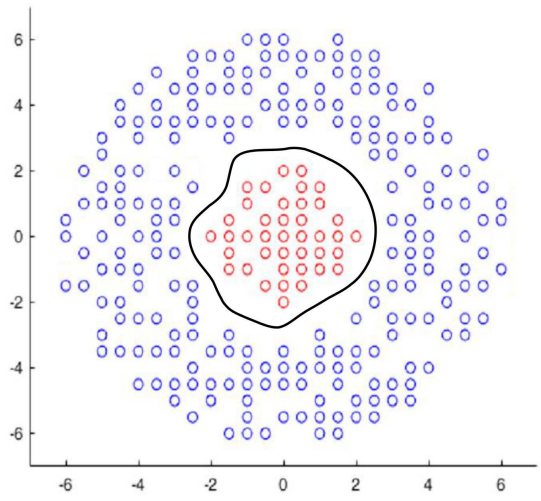
- kernel perceptronowy:

$$\mathcal{K}(\mathbf{u},\mathbf{v}) = \tanh\left[a(\mathbf{u}\cdot\mathbf{v})+b\right]. \tag{30}$$

Odwzorowanie $\mathbf{\Psi} : \mathbb{R}^I \to \mathbf{F}$

Liniowa hiperpłaszczyzna powstaje w przestrzeni $\mathbf{F}$.

To daje nieliniową funkcję separującą w przestrzeni $\mathbb{R}^I$.

Dzięki funkcji jądra $\mathcal{K}$ nie jest konieczne jawne odwzorowanie $\mathbf{\Psi}$.

Uogólniony problem dualny

Realizując nieliniowe odwzorowanie do przestrzeni cech za pomocą funkcji jądra, które spełniają warunki Mercera, funkcjonał (23) przyjmuje następującą postać:

$$\begin{cases} \max\limits_{\boldsymbol{\alpha}} W(\boldsymbol{\alpha}) = \max\limits_{\boldsymbol{\alpha}} \left\{ \sum\limits_{l=1}^{L} \alpha_l - \frac{1}{2} \sum\limits_{l=1}^{L} \sum\limits_{k=1}^{K} \alpha_l \alpha_k y_l y_k \mathcal{K}\left(\mathbf{x}_l, \mathbf{x}_k\right) \right\} \\ \text{pod warunkiem, że:} \\ 0 \leqslant \alpha_l \leqslant C, \quad l = 1, \ldots, L \\ \sum\limits_{l=1}^{L} \alpha_l y_l = 0. \end{cases} \tag{31}$$

Dobór funkcji $\mathcal{K}$ oraz wartości parametru C ma istotny wpływ na rozwiązanie problemu (31).

Uogólniony optymalny klasyfikator

Rozwiązaniem problemu (31) jest wektor mnożników Lagrange'a $\boldsymbol{\alpha}^\star$.

Uogólniony optymalny miękki klasyfikator (ang. soft margin classifier) wyznacza się według wzoru:

$$y = \operatorname{sign}(f(\mathbf{x})) = \operatorname{sign}\left(\sum_{v \in SV} \alpha_v^\star y_v \mathcal{K}\left(\mathbf{x}_v, \mathbf{x}\right) + b^\star\right), \tag{32}$$

gdzie:

$$b^\star = -\frac{1}{2} \sum_{v \in SV} \alpha_v^\star y_v \left[\mathcal{K}\left(\mathbf{x}_v, \mathbf{x}_r\right) + \mathcal{K}\left(\mathbf{x}_v, \mathbf{x}_s\right)\right], \tag{33}$$

gdzie SV oznacza zbiór indeksów wektorów wspierających.

Takie podejście w literaturze nosi nazwę C–SVM.

Uogólniony optymalny klasyfikator – przykład

Zbiór danych raka piersi: $L = 569$ rekordów, $\mathbf{x}_l \in \mathbb{R}^{30}$, $\mathbf{t}_l \in \{1, 2\}$.

```
load('DBC');
% Wyłuskanie rekordów wejściowych i klas:
X = DBC(:,1:end-1); Y = DBC(:,end);
% Parametry SVM:
kernel = 'gaussian'; % 'linear', 'rbf', 'polynomial'
kern_par = 10;       % parametr funkcji jądra
solver = 'SMO';      % 'ISDA', 'L1QP'
C = 1000;            % C constant (box constraint)
SVMModel = fitcsvm(X,Y, 'KernelFunction', kernel, ...
                        'KernelScale', kern_par, ...
                        'Solver', solver, 'BoxConstraint', C);
CVSVMModel = crossval(SVMModel);  % Walidacja krzyżowa
Acc = 1 - kfoldLoss(CVSVMModel);
% Wektory wspierające:
sv = X(SVMModel.IsSupportVector,:);
% Predykcja dla zbioru testującego dla struktury CVSVMModel.  Struktura
ta przechowuje 10 klasyfikatorów SVM; do predykcji wybrano nr 2:
[labels,~] = predict(CVSVMModel.Trained{2,1},X_Test(:,1:end-1));
```

Algorytm SVM w klasyfikacji na wiele klas

Każda z funkcji decyzyjnych (15) oraz (32) tworzy klasyfikator binarny.

W związku z tym algorytmu SVM nie można bezpośrednio zastosować do problemów separacji wieloklasowej.

Algorytm można jednak rozszerzyć tak, aby "radził" sobie z problemami wieloklasowymi. W tym celu można skorzystać z dwóch znanych podejść:

- jeden przeciwko reszcie (ang. one against all),
- jeden przeciwko jednemu (ang. one against one).

Istnieje również inna możliwość – konstrukcja funkcji decyzyjnej wykorzystując wszystkie klasy naraz (Weston i Watkins, 1998).

Algorytm SVM w klasyfikacji na wiele klas . . .

One against all

To podejście składa się z dwóch etapów: najpierw z całego zbioru danych wybierana jest jedna klasa a pozostałe traktowane są jako klasa "przeciwna"; następnie dla powstałego problemu binarnego tworzony jest optymalny klasyfikator. Procedura powtarzana jest dla każdej klasy z osobna. Na etapie predykcji, wykorzystuje się wszystkie z K funkcji decyzyjnych f_k do sklasyfikowania rekordu testującego. Rozpoznana klasa to ta, dla której zachodzi $y = \underset{k\in\{1,\ldots,K\}}{\operatorname{argmax}} f_k(\mathbf{x})$.

One against one

W tej metodzie tworzy się optymalny klasyfikator dla każdej pary klas spośród wszystkich dostępnych. Powstaje zatem $K(K-1)/2$ funkcji decyzyjnych. Na etapie predykcji stosuje się strategię głosowania, co powoduje, że rekord testujący przydzielany jest do klasy, "która otrzymała największą liczbę głosów".

Przykład SVM dla wielu klas – kod w Matlabie

Zbiór danych kwiatów Irysa: $L = 150$ rekordów, $\mathbf{x}_l \in \mathbb{R}^4$, $\mathbf{t}_l \in \{1, 2, 3\}$.

```
load('Iris');
X = Iris(:,1:4); Y = Iris(:,5); L = length(Y);
kernel = 'gaussian'; kern_par = 10; solver = 'SMO'; C = 1000;
SVMs = cell(3,1);
Acc = zeros(3,1);
for i = 1 :  3
  T = zeros(L,1);
  ind = find(Y == i);
  T(ind) = 1;
  SVMs{i} = fitcsvm(X,T,'Standardize' , true, ...
                        'KernelFunction', kernel, ...
                        'KernelScale', kern_par, ...
                        'Solver', solver, 'BoxConstraint', C);
  CVSVM = crossval(SVMs{i});
  Acc(i) = 1 - kfoldLoss(CVSVM);
end
```

Implementacje algorytmu SVM

Najbardziej znane propozycje:

- Chunking (Vapnik, 2006);
- Algorytm redukcji (Osuna et al., 1997);
- SVM^{light} (Joachims, 1998)
- Sequential minimal optimization (Platt, 1998).

Można też wykorzystać funkcję `quadprog` oprogramowania Matlab.

Narzędzia programistyczne posiadające wbudowaną funkcję do klasyfikacji danych i analizy regresji za pomocą algorytmu SVM:

- Matlab,
- Statistica,
- IBM SPSS Modeler,
- DTREG,
- LIBSVM – A Library for Support Vector Machines.

Literatura

Vapnik V.N. (1995) The Nature of Statistical Learning Theory. Springer-Verlag, New York

Vapnik, V.N., Chervonenkis, A.Y. (1971) On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities. Theory of Probability & Its Applications 16(2), s. 264

Drucker H., Burges C., Kaufman L., Smola A.J., Vapnik V (1997) Support Vector Regression Machines. Advances in Neural Information Processing Systems 9, NIPS 1997, s. 155–161, MIT Press

Vapnik V. (2006) Estimation of dependences based on empirical data. Springer-Verlag Berlin, Heidelberg

Osuna E., Freund R., Girosi F. (1997) Improved Training Algorithm for Support Vector Machines, Proceedings of Neural Networks for Signal Processing Workshop

Joachims T. (1998) Making Large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning. Red. B. Schölkopf, C.J.C. Burges i A.J. Smola, The MIT Press, Cambridge

Platt J.C. (1998) A Fast Algorithm for Training Support Vector Machines. Technical Report MSR-TR-98-14

Weston J., Watkins C. (1998) Multi–class Support Vector Machines. Technical Report CSD-TR-98-04