Klasyfikatory SVM

Przemysław Klęsk

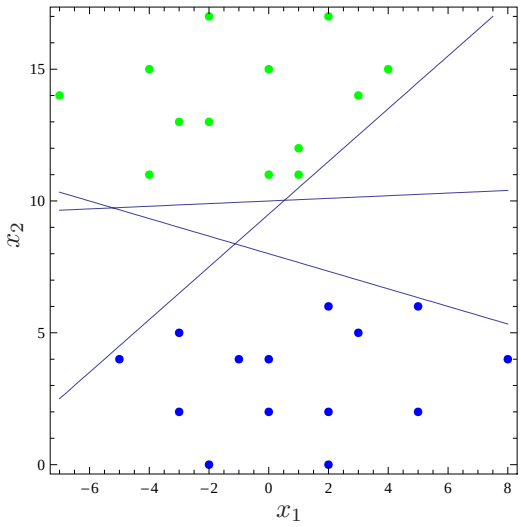
Spis treści

1 Wiadomości ogólne

Klasyfikatory SVM (ang. *Support Vector Machines*[1]) zostały opracowane przez Vapnika (1963, 1992, 1995). SVM'y służą do *klasyfikacji binarnej*. Oznacza to, że mamy dokładnie *dwie* klasy obiektów (np.: chorzy, zdrowi) i szukamy funkcji — klasyfikatora — która będzie przyporządkowywać nowo przychodzące obiekty do odpowiednich klas.

[1] Tłumaczenia dobrze oddające sens: *maszyny punktów podparcia* lub *maszyny wektorów podparcia*.

Omówienie metody SVM dobrze jest rozpocząć od analogii do nieco podobnej metody — *algorytmu perceptronu* Rosenblatt'a (1962). Algorytm ten poszukuje prostej lub w ogólności płaszczyzny separacji sposobem *on-line*, polegającym na krokowym aktualizowaniu dotychczasowej płaszczyzny na podstawie błędnie sklasyfikowanych dotychczas punktów. Jeżeli zbiór danych jest liniowo separowalny, to algorytm Rosenblatt'a na pewno się zatrzyma i znajdzie jedną z możliwych płaszczyzn separacji. Niestety nie mamy wpływu na to, jakie będzie końcowe ułożenie tej płaszczyzny. Zależy ono bowiem od losowego porządku przeglądania zbioru danych. Wiadomo tylko, że ta płaszczyzna trafi „gdzieś pomiędzy" klasy, patrz rys. 1. Jak widać z rysunku końcowe położenia płaszczyzny klasyfikacji mogą być istotnie różne.



Rysunek 1: Przykłady możliwych prostych separacji dla danych dwuwymiarowych.

Widzimy też, że niektóre z nich mogą przebiegać bardzo blisko punktów danych. Można zastanawiać się, które z położeń płaszczyzny są lepsze, a które gorsze.

Intuicyjnie, za lepsze moglibyśmy uznać te płaszczyzny, które przebiegają możliwie daleko od obu klas obiektów. Wynika to z domysłu, że być może prawdziwa granica (której nie znamy) tkwiąca w badanym zjawisku, a wg której następuje rozróżnienie obiektów na klasę pierwszą lub drugą, przebiega właśnie gdzieś „po środku" i z dala od skupień. Po drugie, przyjmując taką granicę, oczekujemy, że będziemy rzadziej popełniać błędy klasyfikacji dla nowo przychodzących obiektów. Szczególnie jeśli obiekty te byłyby nietypowe i leżałyby właśnie gdzieś blisko „środka". Innymi słowy, oczekujemy, że klasyfikator będzie dzięki temu potrafił dobrze *uogólniać*.

O metodzie SVM można ogólnie powiedzieć, że:

- w przypadku *liniowo separowalnym*, tj. wtedy, gdy istnieje przynajmniej jedna płaszczyzna separacji oddzielająca klasy, metoda ta gwarantuje znalezienie takiej płaszczyzny, która ma maksymalny tzw. ***margines separacji***;
- w przypadku *nieseparowalnym liniowo*, metoda SVM pozwala na znalezienie płaszczyzny, która

klasyfikuje obiekty na tyle poprawnie, na ile jest to możliwe i jednocześnie przebiega możliwie daleko od typowych skupień dla każdej z klas (będziemy tu mówili o największym marginesie w sensie pewnej zadanej heurystyki);

- w przypadku *nieseparowalnym liniowio*, stostując tzw. *podniesienie wymiarowości*, można za pomocą metody SVM znaleźć krzywoliniową granicę klasyfikacji o dużym marginesie separacji.

2 Margines separacji

Formalizując zadanie, dany jest zbiór znanych przykładów, wyrażony jako zbiór par:

$$\big\{(\mathbf{x}_i, y_i)\big\}_{i=1,2,\ldots,I} ,$$

gdzie $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{in}) \in \mathbb{R}^n$ są punktami w przestrzeni n-wymiarowej, a $y_i \in \{-1, 1\}$ są numerami (etykietami) przyporządkowanej do $\mathbf{x}_i$ klasy[2]. Rozmiar zbioru danych jest równy I. Oznaczenie klas właśnie jako $\{-1, 1\}$, a nie np. jako $\{1, 2\}$ lub $\{0, 1\}$, jest wygodne z pewnych powodów rachunkowych w metodzie SVM, o których dalej.

2.1 Wzór na odległość punktu od płaszczyzny

Do zdefiniowania pojęcia *marginesu separacji* potrzebny jest wzór na odległość punktu od płaszczyzny. Niech płaszczyzna będzie określona przez swój wektor normalny (prostopadły do niej) $\mathbf{w} = (w_1, w_2, \ldots, w_n)$ oraz przez wyraz wolny w_0. Równanie płaszczyzny ma postać:

$$w_0 + \underbrace{w_1 x_1 + w_2 x_2 + \cdots + w_n x_n}_{\langle \mathbf{w}, \mathbf{x} \rangle} = 0. \tag{1}$$

Notacja pary wektorów $\langle \mathbf{w}, \mathbf{x} \rangle$ oznacza po prostu ich iloczyn skalarny i będzie wygodnym i skrótowym sposobem do zapisywania potrzebnych nam równań.

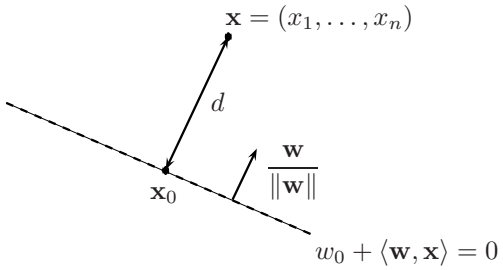
Wzór na odległość d punktu $\mathbf{x}$ od płaszczyzny określonej przez $\mathbf{w}$ i w_0 jest następujący:

$$d(\mathbf{x}, \mathbf{w}, w_0) = \frac{|w_0 + \langle \mathbf{w}, \mathbf{x} \rangle|}{\|\mathbf{w}\|}. \tag{2}$$

Zapis $\|\mathbf{w}\|$ oznacza *normę* lub inaczej długość wektora $\mathbf{w}$. Czyli $\|\mathbf{w}\| = \sqrt{{w_1}^2 + {w_2}^2 + \cdots + {w_n}^2}$.

Dowód: Dla ustalonej płaszczyzny, każdy punkt $\mathbf{x}$ możemy przedstawić jako $\mathbf{x} = \mathbf{x}_0 \pm d \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|}$, gdzie $\mathbf{x}_0$ jest jest rzutem $\mathbf{x}$ na rozpatrywaną płaszczyznę, a $d \geqslant 0$ odległością od płaszczyzny, patrz rys. 2. Innymi słowy, startując z punktu $\mathbf{x}_0$ i idąc wzdłuż jednostkowego wektora normalnego $\pm\frac{\mathbf{w}}{\|\mathbf{w}\|}$ przez odległość d

[2]Możemy myśleć w ten sposób, że dany punkt $\mathbf{x}_i$ zawiera pewne informacje (czynniki, cechy, właściwości i ich kombinacje), z których wynika jego przynależność do klasy y_i. Na przykład na podstawie temperatury ciała, ciśnienia krwi, poziomu leukocytów możemy próbować ocenić, czy człowiek jest zdrowy czy chory.



Rysunek 2: Przedstawienie punktu $\mathbf{x}$ jako jego rzut na płaszczyznę plus/minus d jednostek w kierunku wektora normalnego.

jednostek (odpowiednio na plus lub minus) trafiamy w punkt $\mathbf{x}$. Wstawiamy powyższe przedstawienie $\mathbf{x}$ do wzoru na odległość:

$$\begin{aligned}
\frac{|w_0 + \langle \mathbf{w}, \mathbf{x} \rangle|}{\|\mathbf{w}\|} &= \frac{|w_0 + \langle \mathbf{w}, \mathbf{x}_0 \pm d\frac{\mathbf{w}}{\|\mathbf{w}\|} \rangle|}{\|\mathbf{w}\|} \\
&= \frac{|w_0 + \langle \mathbf{w}, \mathbf{x}_0 \rangle \pm \langle \mathbf{w}, d\frac{\mathbf{w}}{\|\mathbf{w}\|} \rangle|}{\|\mathbf{w}\|} \\
&= \frac{|0 \pm \frac{d}{\|\mathbf{w}\|} \langle \mathbf{w}, \mathbf{w} \rangle|}{\|\mathbf{w}\|} \\
&= \frac{|\frac{d}{\|\mathbf{w}\|} \|\mathbf{w}\|^2|}{\|\mathbf{w}\|} = d. \qquad \blacksquare
\end{aligned}$$

Przejście z linii pierwszej do drugiej wynika z rozdzielności dodawania wektorów względem iloczynu skalarnego. Przejście z linii drugiej do trzeciej wynika z faktu, że punkt $\mathbf{x}_0$ spełnia równanie płaszczyzny, a zatem $w_0 + \langle \mathbf{w}, \mathbf{x}_0 \rangle = 0$.

Marginesem separacji dla zbioru danych $\{(\mathbf{x}_i, y_i)\}_{i=1,\ldots,I}$ i dla ustalonej płaszczyzny określonej przez $\mathbf{w}$ i w_0, nazywamy liczbę

$$\tau(\mathbf{w}, w_0) = \min_{i=1,\ldots,I} \frac{y_i(w_0 + \langle \mathbf{w}, \mathbf{x} \rangle)}{\|\mathbf{w}\|}. \tag{3}$$

A zatem τ jest to odległość od płaszczyzny do najbliższego do niej punktu danych. Należy zwrócić uwagę, że we wzorze (3) wykorzystany jest wzór na odległość (2), przy czym wartość bezwzględną, która zniknęła, rekompensuje domnożenie przez y_i, czyli odpowiednio przez ± 1. Jeżeli wszystkie punkty są dobrze sklasyfikowane, to dla punktów leżących po „dodatniej" stronie płaszczyzny mamy $y_i = 1$, a dla punktów leżących po „ujemnej" stronie płaszczyzny $y_i = -1$. Wówczas też $\tau \geqslant 0$ i jest ono równe odległości od płaszczyzny do najbliższego z punktów. Jeżeli zaś niektóre punkty są źle klasyfikowane przez daną płaszczyznę, to mają one niejako odległość ujemną od niej. Wówczas $\tau < 0$ i jest ono równe odległości do najdalszego bezwzględnie błędnie klasyfikowanego punktu, tyle że odległość ta jest ze znakiem minus.

3 Przypadek liniowej separowalności danych — znajdowanie płaszczyzny o największym marginesie separacji

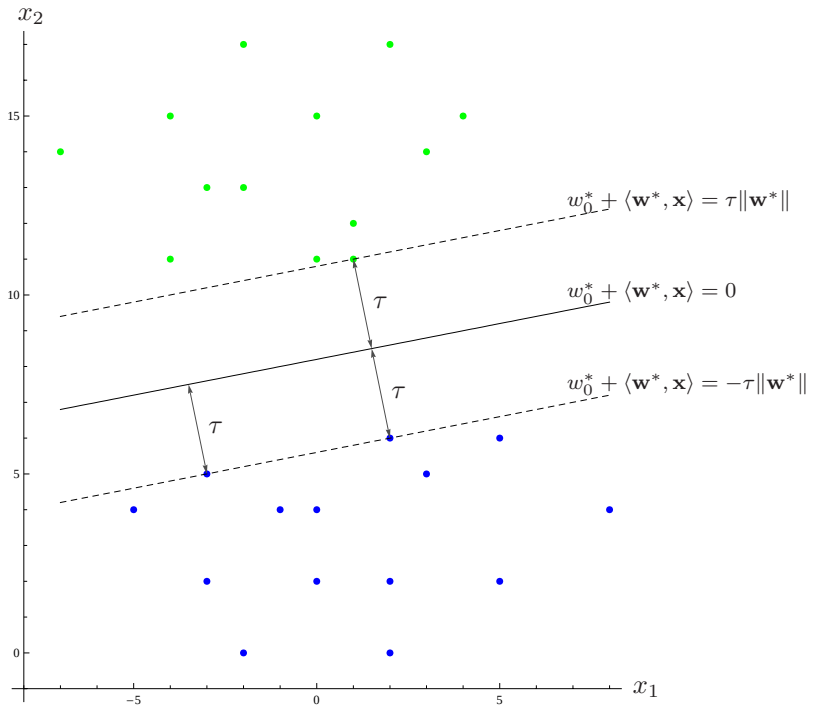
3.1 Oryginalne postawienie problemu

W metodzie SVM celem jest znalezienie *optymalnej płaszczyzny*, takiej która po pierwsze poprawnie klasyfikuje dane (o ile to możliwe), i po drugie, dla której margines separacji τ jest największy. A więc mamy zadanie optymalizacji z ograniczeniami — chcemy maksymalizować

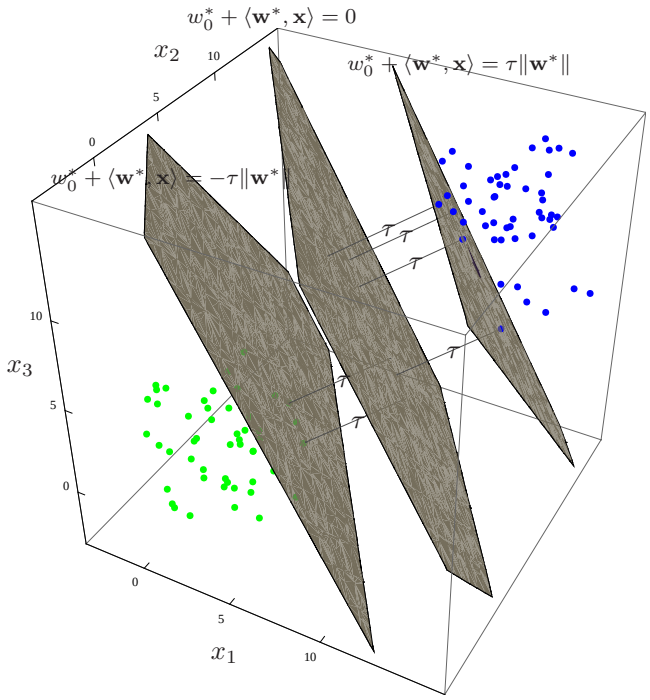
$$\tau(\mathbf{w}, w_0)$$
$$\text{przy ograniczeniach: } \forall i \quad y_i(w_0 + \langle \mathbf{w}, \mathbf{x}_i \rangle) \geqslant \tau(\mathbf{w}, w_0)\|\mathbf{w}\|. \tag{4}$$

Ograniczeń jest tyle, ile punktów danych, czyli I. Mówią one po prostu to, że każdy punkt ma leżeć po odpowiedniej stronie płaszczyzny i to w odległości co najmniej τ.

Nazwijmy optymalne rozwiązanie przez $\mathbf{w}^*$, w_0^*. Od tej pory będziemy również pisać krócej τ zamiast $\tau(\mathbf{w}, w_0)$. Patrz rysunki 3 i 4.



Rysunek 3: Przykład optymalnej prostej o maksymalnym marginesie separacji.



Rysunek 4: Klasyfikacja w przestrzeni $\mathbb{R}^3$. Przykład optymalnej płaszczyzny o maksymalnym marginesie separacji.

3.2 Unormowane postawienie problemu

Można zauważyć, że nawet jeśli znajdziemy taką optymalną płaszczyznę, to tak naprawdę mamy nieskończenie wiele rozwiązań, które reprezentują tę samą płaszczyznę, a różnią się jedynie skalowaniem $\mathbf{w}^*$ i w_0^*. Wynika to z faktu, że mnożenie równania płaszczyzny (1) stronami przez dowolną liczbę, oczywiście nie zmienia położenia tej płaszczyzny.

Powiedzmy, że chcielibyśmy coś na to zaradzić i dla ustalonego marginesu móc jednoznacznie określić tylko jedną płaszczyznę. Jednym ze sposobów, aby to zrobić, mogłoby być założenie, że szukamy $\mathbf{w}$, tylko wśród wektorów jednostkowych, tj. $\|\mathbf{w}\| = 1$. Jest to jedna możliwość, ale okazuje się, że istnieje jeszcze wygodniejsza. Powiążmy τ i $\mathbf{w}$ następującym więzem:

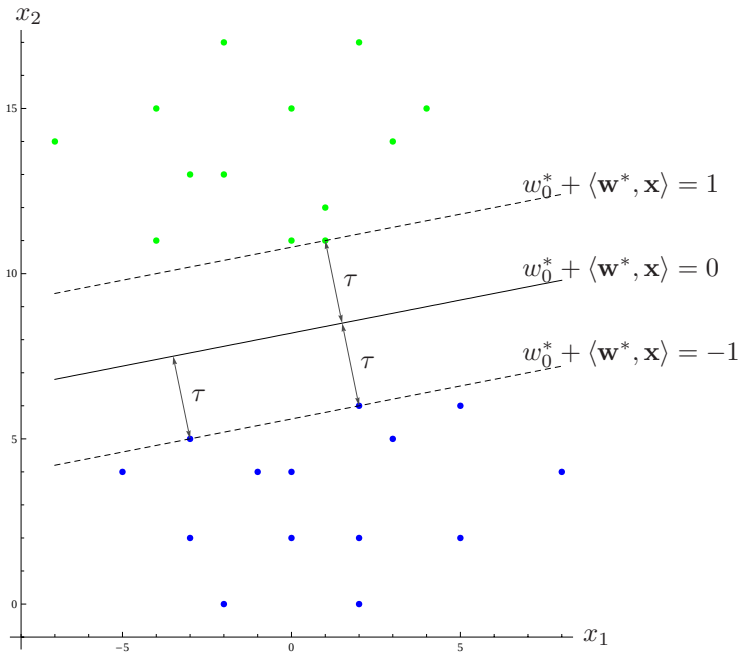
$$\|\mathbf{w}\| \cdot \tau = 1. \tag{5}$$

Innymi słowy szukamy wektora $\mathbf{w}$ tylko wśród takich, których długość pomnożona przez margines jest równa jeden. Przypomijmy, że do tej pory ustalenie płaszczyzny tj. $\mathbf{w}$, w_0, jednoznacznie określało margines τ, ale w drugą stronę: ustalenie marginesu nie określało jednoznacznie płaszczyzny (a nieskończenie wiele płaszczyzn). Od tej pory ten problem znika. Po nałożeniu więzu (5), zauważamy, że nasze ograniczenia przybierają postać

$$\forall i \quad y_i(w_0 + \langle \mathbf{w}, \mathbf{x}_i \rangle) \geqslant 1. \tag{6}$$

Wartość wyrażenia $w_0 + \langle \mathbf{w}, \mathbf{x} \rangle$ dla punktów $\mathbf{x}$ leżących w pasie marginesowym została unormowana do

przedziału $[-1, 1]$, ale w sensie metrycznym mamy nadal odległość 2τ pomiędzy zewnętrznymi granicami, patrz rys. 5.



Rysunek 5: Unormowanie pasa marginesowego więzem $\|\mathbf{w}\| \cdot \tau = 1$, patrz równania prostych granicznych.

Zauważamy również, że po nałożeniu więzu (5), mamy $\tau = \frac{1}{\|\mathbf{w}\|}$. Czyli im mniejsza norma $\mathbf{w}$, tym większy margines τ. A więc na przykład, jeżeli tylko udałoby nam się znaleźć pewien zbiór płaszczyzn spełniających unormowane ograniczenia (6), to spośród nich należy wybrać tę płaszczyznę, której wektor $\mathbf{w}$ ma najmniejszą normę, ponieważ ta płaszczyzna ma największy margines τ. Jest to wygodny zabieg. Zmienia on nam zadanie optymalizacji na równoważne, określone następująco. Minimalizuj

$$Q(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2,$$

$$\text{przy ograniczeniach: } \forall i \quad y_i(w_0 + \langle \mathbf{w}, \mathbf{x}_i \rangle) \geqslant 1. \tag{7}$$

Patrzymy na $\|\mathbf{w}\|^2$ zamiast na $\|\mathbf{w}\|$ dla wygody obliczeniowej przy późniejszym wyliczaniu pochodnej, co wolno na zrobić, ponieważ obie wielkości są tak samo monotoniczne i osiągają minimum w tym samym miejscu. Mnożnik $\frac{1}{2}$ jest dopisany, tak aby przy wyliczaniu pochodnej kasował się z dwójką z wykładnika do jedności.

Tak sformułowane zadanie można już teraz rozwiązywać. Można to zrobić na przykład za pomocą procedury `quadprog` w MATLABie, jako że w wyrażenie $\frac{1}{2}\|\mathbf{w}\|^2 = \frac{1}{2}({w_1}^2 + {w_2}^2 + \cdots + {w_n}^2)$ jest formą kwadratową[3], a więc potrzebujemy optymalizatora, który rozwiązuje zadanie programowania kwadratowego przy ograniczeniach. Proponujemy w tym miejscu przeliczyć sobie na próbę jakiś prosty przykład

[3]Np. dla dwóch zmiennych x, y pełna forma kwadratowa to: $A + Bx + Cy + Dx^2 + Ey^2 + Fxy$. U nas zmiennymi są $w_1, \ldots, w_n$.

za pomocą `quadprog` i sprawdzić chociażby graficznie, czy daje to poprawną prostą/płaszczyznę o maksymalnym marginesie. Zbiór danych separowalny liniowo można odpowiednio wylosować.

3.3 Postawienie problemu w terminach mnożników Lagrange'a — punkty podparcia

Do tej pory nie stało się jednak w żaden sposób jasne pojęcie *punktów podparcia*, które pojawia się w nazwie metody. Zajmijmy się tym teraz. Do tego celu wygodne jest jeszcze inne przeformułowanie problemu optymalizacji, także równoważne. Otóż, za pomocą techniki mnożników Lagrange'a możemy wpleść ograniczenia (w naszym zadaniu nierównościowe) do optymalizowanego wyrażenia:

$$\begin{aligned} Q(\mathbf{w}, w_0, \alpha_1, \alpha_2, \ldots, \alpha_I) &= \frac{1}{2}\|\mathbf{w}\|^2 - \alpha_1\big(y_1(w_0 + \langle \mathbf{w}, \mathbf{x}_1\rangle) - 1\big) \\ &\quad - \alpha_2\big(y_2(w_0 + \langle \mathbf{w}, \mathbf{x}_2\rangle) - 1\big) \\ &\quad \vdots \\ &\quad - \alpha_I\big(y_I(w_0 + \langle \mathbf{w}, \mathbf{x}_I\rangle) - 1\big) \\ &= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{I} \alpha_i\big(y_i(w_0 + \langle \mathbf{w}, \mathbf{x}_i\rangle) - 1\big) \\ \text{przy ograniczeniach: } \forall i \quad \alpha_i &\geqslant 0. \end{aligned} \tag{8}$$

Jeżeli i-ta nierówność z ograniczeń, patrz (7), jest spełniona, to wyrażenie $y_i(w_0 + \langle \mathbf{w}, \mathbf{x}_i\rangle) - 1$ jest dodatnie. Wyjaśnia to znaki minus stojące przed mnożnikami Lagrange'a α_i — żądamy minimalizacji ze względu na $\|\mathbf{w}\|$, a więc przy ustalonych $\alpha_i \geqslant 0$ spełnianie ograniczeń ma pomniejszać optymalizowane wyrażenie.

Należy jednocześnie zauważyć, że ustawienie mnożników $\alpha_i = \infty$ powodowałoby, że całe wyrażenie Q staje się $-\infty$. Co prowadziłoby w trywialny sposób do minimalizacji. Widać stąd, że minimalizacja Q musi się odbywać tylko ze względu na $\|\mathbf{w}\|$, natomiast ze względu na α_i należy maksymalizować Q. I tak rozwiązaniem będzie pewien punkt siodłowy, co można zilustrować symbolicznie tak jak na rys. 6.

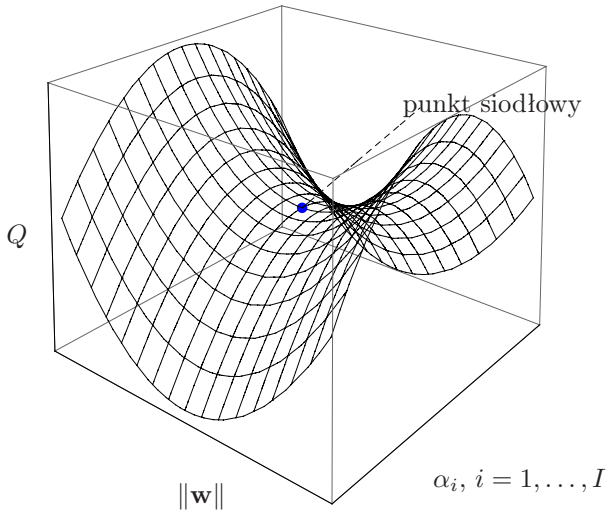
Warunek konieczny istnienia punktu siodłowego jest taki sam jak warunek konieczny istnienia ekstremum, tj. pochodne cząstkowe Q ze względu na nasze parametry mają być zerami:

$$\frac{\partial Q}{\partial \mathbf{w}} = \mathbf{0}, \tag{9}$$

$$\frac{\partial Q}{\partial w_0} = 0, \tag{10}$$

$$\forall i \quad \frac{\partial Q}{\partial \alpha_i} = 0 \qquad \text{(spełnienie ograniczeń na brzegach)}. \tag{11}$$

Należy zaznaczyć, że pierwsze z powyższych równań jest wektorowe, to znaczy mamy tak naprawdę n równań, tyle ile wynosi wymiarowość wektora $\mathbf{w} = (w_1, \ldots, w_n)$, po jednym równaniu dla każdej



Rysunek 6: Symboliczna ilustracja punktu siodłowego. W zadaniu SVM żądamy minimalizacji Q ze względu na $\|\mathbf{w}\|$ oraz maksymalizacji Q ze względu na wszystkie α_i.

współrzędnej. I tak z równania (9) otrzymujemy

$$\mathbf{w} - \sum_i \alpha_i y_i \mathbf{x}_i = 0, \qquad \text{więc}$$
$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i. \tag{12}$$

Z równania (10) otrzymujemy

$$-\sum_{i=1}^{I} \alpha_i y_i = 0. \tag{13}$$

Obie powyższe informacje wykorzystamy, aby wyrazić zadanie optymalizacyjne dane w formie (8) w terminach samych α_i, bez użycia $\mathbf{w}$ i w_0.

$$\begin{aligned}
&\frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{I} \alpha_i \big(y_i(w_0 + \langle \mathbf{w}, \mathbf{x}_i\rangle) - 1\big) \\
&= \frac{1}{2}\langle \underbrace{\mathbf{w}}_{\sum_i \alpha_i y_i \mathbf{x}_i}, \underbrace{\mathbf{w}}_{\sum_i \alpha_i y_i \mathbf{x}_i} \rangle - w_0 \underbrace{\sum_{i=1}^{I} \alpha_i y_i}_{0} - \sum_{i=1}^{I} \alpha_i y_i \langle \underbrace{\mathbf{w}}_{\sum_i \alpha_i y_i \mathbf{x}_i}, \mathbf{x}_i\rangle + \sum_{i=1}^{I} \alpha_i \\
&= \frac{1}{2}\sum_{i=1}^{I}\sum_{j=1}^{I} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j\rangle - \sum_{i=1}^{I}\sum_{j=1}^{I} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j\rangle + \sum_{i=1}^{I} \alpha_i.
\end{aligned}$$

A zatem ostatecznie mamy zadanie — maksymalizuj:

$$Q(\alpha_1, \ldots, \alpha_I) = -\frac{1}{2}\sum_{i=1}^{I}\sum_{j=1}^{I}\alpha_i\alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j\rangle + \sum_{i=1}^{I}\alpha_i$$

przy ograniczeniach: $\forall i \quad \alpha_i \geqslant 0$,

$$\sum_{i=1}^{I}\alpha_i y_i = 0. \tag{14}$$

Ostatnie ograniczenie jest potrzebne, ponieważ używaliśmy tej własności do wyzerowania pewnych fragmentów z wyrażenia na Q, a więc nie możemy teraz tego faktu pominąć. Należy zwrócić uwagę, że w związku z tym, że wszystko mamy teraz wyrażone w terminach samych α_i, to mamy już tylko zadanie maksymalizacji.

Znalezione optymalne rozwiązanie $\alpha_1^*, \alpha_2^*, \ldots, \alpha_I^*$ (podobnie jak wcześniej, można je znaleźć za pomocą `quadprog` w MATLABie), należy przełożyć teraz na $\mathbf{w}$ i w_0, tak abyśmy otrzymali pożądaną płaszczyznę. Jak to zrobić? Po pierwsze, przyda się w tym miejscu parę uwag o własnościach rozwiązania $\alpha_1^*, \alpha_2^*, \ldots, \alpha_I^*$. Otóż:

- w typowym praktycznym przypadku większość α_i^* wyjdzie równa zeru,
- punkty danych $\mathbf{x}_i$, dla których odpowiadające im $\alpha_i^* > 0$, będziemy nazywać ***punktami podparcia*** (ang. *support vectors*),
- zgodnie ze wzorem (12) optymalny wektor $\mathbf{w}^*$ można znaleźć jako

  $$\mathbf{w}^* = \sum_i \alpha_i^* y_i \mathbf{x}_i,$$

  a zatem stwierdzamy, że wektor $\mathbf{w}^*$ powstaje jako ***kombinacja liniowa punktów podparcia***, tj. tych $\mathbf{x}_i$, dla których $\alpha_i^* > 0$, bo pozostałe składniki w sumie będą zerami; można to łatwo zrozumieć geometrycznie (patrz rysunki 3, 4) — tylko wybrane punkty danych wpływają na położenie optymalnej płaszczyzny klasyfikacji, punkty „na tyłach" klas nie mają na to wpływu, a dostawienie lub odjęcie jednego z takich tylnych punktów nic by nie zmieniało. Na rysunkach, punktami podparcia są te punkty, przy których zaznaczono odcinkami margines τ. Właśnie te punkty wyznaczają płaszczyzny brzegowe $w_0^* + \langle \mathbf{w}^*, \mathbf{x}\rangle = \pm 1$ oraz główną płaszczyznę separacji $w_0^* + \langle \mathbf{w}^*, \mathbf{x}\rangle = 0$.
- dodatkowo można powiedzieć, że im więcej wyjdzie punktów podparcia w stosunku do rozmiaru całego zbioru, tym dany problem czy dane zjawisko jest trudniejsze do klasyfikacji, ponieważ dużo punktów przebywa „na styku" klas,
- im mniej punktów podparcia, tym możemy liczyć na lepszą zdolność do *uogólniania* naszego klasyfikatora — błędy klasyfikacji dla nowo przychodzących nieznanych punktów będą się zdarzały rzadziej niż wtedy, gdybyśmy mieli stosunkowo dużo punktów podparcia.

Pozostaje jeszcze kwestia, jak wyznaczyć wyraz wolny w_0^*. Otóż, można go wyznaczyć z dowolnego ograniczenia $y_i(w_0^* + \langle \mathbf{w}^*, \mathbf{x}_i \rangle) - 1 = 0$, przy czym $\mathbf{x}_i$ musi być jednym z punktów podparcia. I wówczas mamy:

$$\begin{aligned} y_i w_0^* &= 1 - y_i \langle \mathbf{w}^*, \mathbf{x}_i \rangle \qquad \text{(mnożymy stronami przez } y_i = \pm 1\text{)}, \\ w_0^* &= y_i - \langle \mathbf{w}^*, \mathbf{x}_i \rangle. \end{aligned} \tag{15}$$

3.4 Podsumowanie (przypadek liniowej separowalności danych)

W przypadku liniowej separowalności danych znalezienie płaszczyzny o największym marginesie separacji można zrealizować rozwiązując zadanie optymalizacji (programowania kwadratowego) postawione w postaci (7) lub w postaci (14). Obie postaci winny dać takie samo rozwiązanie. Jedyna różnica jest taka, że w pierwszym przypadku dostajemy jako rozwiązanie od razu $\mathbf{w}^*$ i w_0^*, natomiast w drugim przypadku dostajemy najpierw optymalne wartości mnożników Lagrange'a $\alpha_1 *, \ldots, \alpha_I^*$, a dopiero później przekładamy je na $\mathbf{w}^*$ i w_0^*. Pewną zaletą z drugiej wersji jest to, że możemy jawnie dowiedzieć się ile i które punkty są *punktami podparcia*. Będą to te $\mathbf{x}_i$, dla których $\alpha_i^* > 0$.

W MATLABie obie wersje zadania optymalizacji mogą zostać zadane do procedury `quadprog`, odpowiednio podając parametry formy kwadratowej, tj. macierz H i wektor f, oraz ograniczenia wyrażone przez macierze A i Aeq oraz wektory b i beq (patrz dokumentacja MATLABa). Należy pamiętać, że quadprog realizuje minimalizację, a więc jeżeli chcielibyśmy podać wersję drugą maksymalizującą Q względem α_i, to należy podać wyrażenie Q poprzedzić minusem, czyli podać $-H$ i $-f$.

Należy dodatkowo wspomnieć, że można podać do `quadprog` zadanie optymalizacji w wersji pierwszej i mimo to poprosić MATLABa o wartości mnożników Lagrange'a jako jeden z wyników (o nazwie `lambda`). Wynika to z faktu, że MATLAB w swoich wewnętrznych obliczeniach przełoży sobie ograniczenia właśnie na postać z mnożnikami.

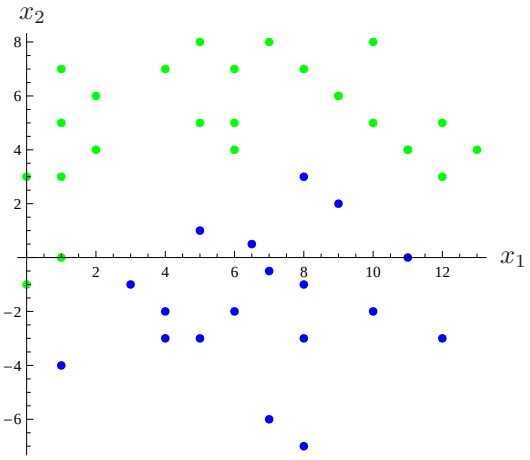
4 Przypadek danych nieseparowalnych liniowo

Przykładem zbioru danych nieseparowalnych liniowo jest zbiór przedstawiony na rys. 7. W ogólności oznacza to, że nie istnieje płaszczyzna, która bezbłędnie oddziela dane — tj. tak ażeby po „dodatniej" stronie płaszczyzny były tylko punkty z etykietą klasy $y_i = 1$, a po „ujemnej" tylko punkty z $y_i = -1$.

W takim przypadku można nadal poszukiwać optymalnej w pewnym sensie płaszczyzny, zgadzając się na to, aby niektóre punkty (odstające od klas) wpadały w pas marginesowy lub nawet były po złej stronie płaszczyzny o równaniu $w_0^* + \langle \mathbf{w}^*, \mathbf{x} \rangle = 0$. Vapnik zaproponował odpowiednie rozszerzenie.

4.1 Dopuszczenie punktów wpadających w margines i błędów

Skojarzmy z każdym punktem danych błąd metryczny $\xi_i' \geqslant 0$, który mówi, na jaką odległość dany punkt wpada w pas marginesowy. Jeżeli punkt $\mathbf{x}_i$ nie wpada w ten pas, to $\xi_i' = 0$. Jeżeli $\mathbf{x}_i$ wpada w pas, ale



Rysunek 7: Przykład zbioru danych nieseparowalnego liniowo.

jest jeszcze po dobrej stronie płaszczyzny klasyfikacji (jest jeszcze dobrze klasyfikowany), to $0 < \xi'_i \leqslant \tau$. Jeżeli zaś $\mathbf{x}_i$ jest źle klasyfikowany, to $\xi'_i > \tau$. Patrz rys. 8.

Mając na uwadze zapis ograniczeń z oryginalnego postawienia problemu dla przypadku separowalnego, patrz (4), możemy zapisać ograniczenia uwzględniając błędy ξ'_i następująco:

$$\forall i \quad y_i(w_0 + \langle \mathbf{w}, \mathbf{x}_i \rangle) + \|\mathbf{w}\|\xi'_i \geqslant \|\mathbf{w}\|\tau, \tag{16}$$

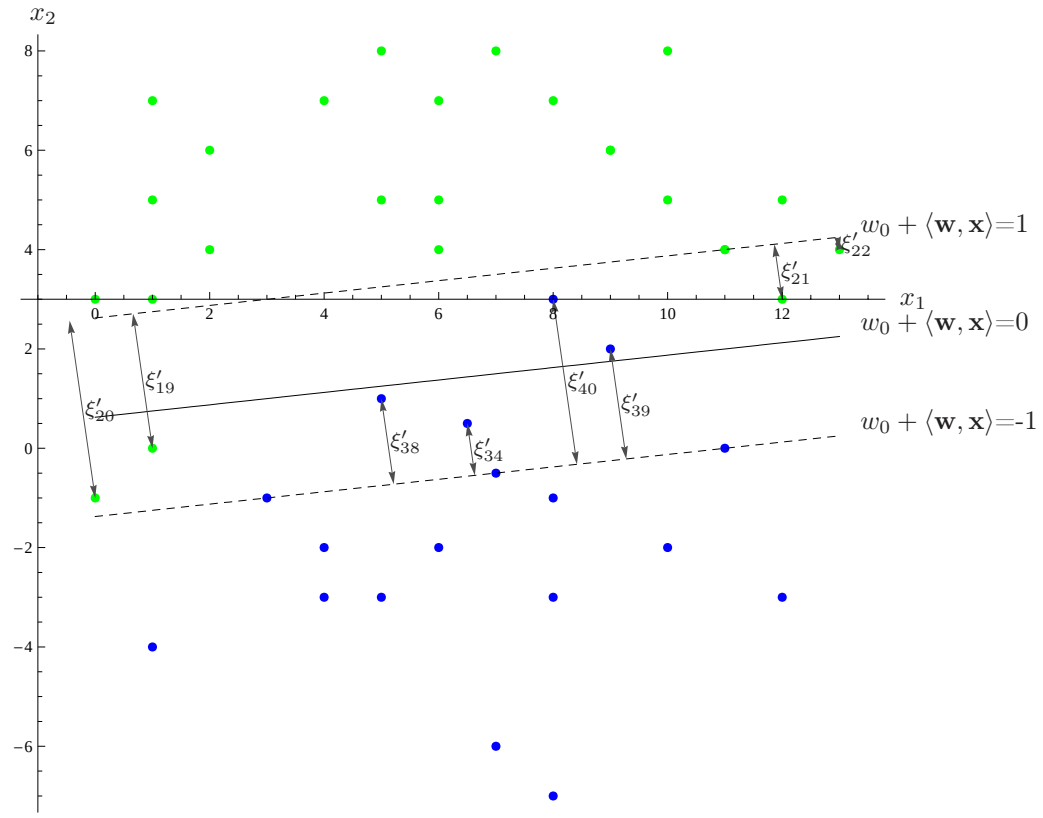
czyli po dołożeniu ewentualnych poprawek ξ'_i oryginalne ograniczenia mają być spełnione[4]. Uwzględniając teraz unormowanie poprzez więz $\|\mathbf{w}\| \cdot \tau = 1$, ograniczenia przybierają postać

$$\forall i \quad y_i(w_0 + \langle \mathbf{w}, \mathbf{x}_i \rangle) + \|\mathbf{w}\|\xi'_i \geqslant 1. \tag{17}$$

Dla uproszczenia oznaczmy iloczyny $\|\mathbf{w}\| \cdot \xi'_i$ jako ξ_i (bez primów). Nowe ξ_i nie mają już co prawda sensu metrycznego (mają zaś sens analogiczny do sensu wyrazu wolnego w_0, który wyraża przesunięcie), ale pozwolą na dogodne dla optymalizacji zapisanie naszego problemu. Wystarczy pamiętać, że każde ξ_i jednoznacznie przekłada się na metryczne ξ'_i.

Zadanie optymalizacji zaproponowane przez Vapnika dla przypadku nieseparowalnego może być opisane tak: chcemy nadal maksymalizować margines, co jest równoważne minimalizowaniu $\|\mathbf{w}\|$, ale będziemy również chcieli minimalizować błędy ξ_i, na które się zgadzamy. I to wszystko przy poluzowanych ograniczeniach (17).

[4]Jeszcze inaczej, tłumacząc wzór (16), po dołożeniu do iloczynu skalarnego (ze znakiem) $y_i(w_0 + \langle \mathbf{w}, \mathbf{x}_i \rangle)$ składnika ewentualnego błędu/poprawki $\|\mathbf{w}\|\xi'_i$, wielkość $\|\mathbf{w}\|\tau$ ma zostać osiągnięta. Pierwotnie w przypadku separowalnym, iloczyn skalarny (ze znakiem) od razu musiał osiągać przynajmniej wartość $\|\mathbf{w}\|\tau$, bez dodawania składnika błędu/poprawki.



Rysunek 8: Przypadek nieseparowalny liniowo — wprowadzenie wielkości ξ'_i wyrażających błędy, na które się zgadzamy: wpadanie punktów w pas marginesowy lub nawet położenie po złej stronie płaszczyzny.

Formalnie wyrażamy to następująco. Minimalizuj ze względu na zmienne $\mathbf{w}$, ξ_1, ..., ξ_I wyrażenie

$$Q(\mathbf{w}, \xi_1, \ldots, \xi_I) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{I} \xi_i,$$

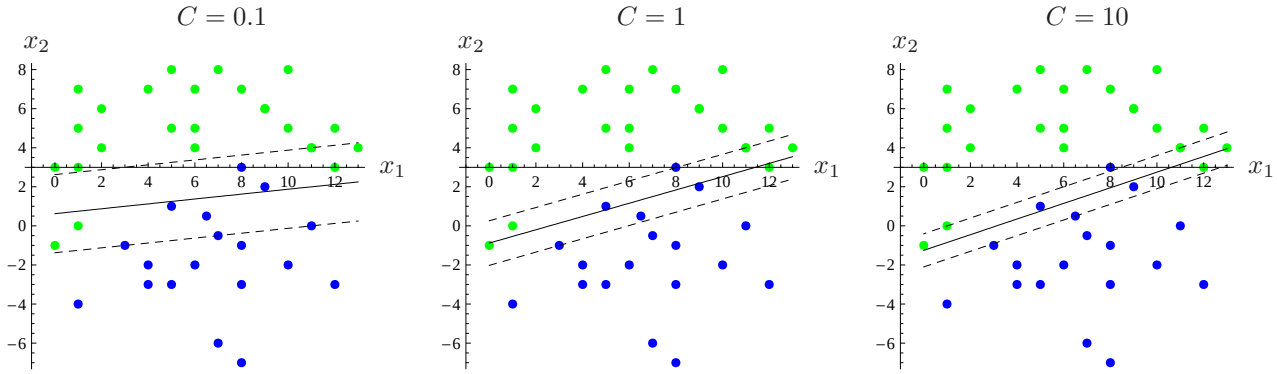
$$\text{przy ograniczeniach: } \forall i \quad y_i(w_0 + \langle \mathbf{w}, \mathbf{x}_i \rangle) + \xi_i \geqslant 1,$$

$$\xi_i \geqslant 0, \tag{18}$$

gdzie $C > 0$ jest dobieralną (niestety) przez nas stałą — heurystyka. C wyraża kompromis pomiędzy dużym marginesem a dopuszczaniem do błędów ξ_i. Im C ustalimy na mniejsze, tym mniejszy będzie wpływ drugiego składnika w Q i rozwiązanie — położenie płaszczyzny klasyfikacji — będzie preferowało większy margines pomiędzy klasami nawet za cenę istnienia dużych błędów ξ_i dla punktów trudnych (odstających). Im C ustalimy na większe, tym drugi składnik będzie miał większe znaczenie w wyrażeniu Q i minimalizacja będzie nastawiona na eliminowanie błędów ξ_i kosztem mniejszego marginesu.

Mówiąc jeszcze inaczej, ustawienie dużego C spowoduje, że procedura minimalizacji będzie bardziej

czuła na trudne punkty danych i wynikowe położenie płaszczyzny będzie w dużej mierze powodowane właśnie przez te punkty. Ustawienie małego C spowoduje, że minimalizacja nie będzie tak czuła na kilka punktów najbardziej trudnych, a wynikowe położenie płaszczyzny będzie bardziej powodowane przez ogół bardziej typowych punktów w klasach. Wpływ wyboru różnych wartości parametru C na położenie płaszczyzny klasyfikacji ilustruje rys. 9.



Rysunek 9: Ilustracja wpływu parametru C na położenie płaszczyzny klasyfikacji.

4.2 Sformułowanie problemu w terminach mnożników Lagrange'a

Podobnie jak w poprzednim rozdziale, problem optymalizacji sformułowany w postaci (18) można już rozwiązywać (np. za pomocą `quadprog`), można natomiast przekształcić go do równoważnej postaci wyrażonej w terminach mnożników Lagrange'a α_i. Dla porządku wyprowadźmy tę postać, mimo że nie jest ona koniecznie potrzebna.

Wplatamy ograniczenia do wyrażenia Q:

$$Q(\mathbf{w}, \xi_1, \ldots, \xi_I, \alpha_1, \ldots, \alpha_I) = \frac{1}{2}\|\mathbf{w}\| + C\sum_{i=1}^{I}\xi_i - \sum_{i=1}^{I}\alpha_i\big(y_i(w_0 + \langle \mathbf{w}, \mathbf{x}_i\rangle) + \xi_i - 1\big). \tag{19}$$

Z warunków punktu siodłowego $\frac{\partial Q}{\partial \mathbf{w}} = 0$, $\frac{\partial Q}{\partial w_0} = 0$ i $\frac{\partial Q}{\partial \alpha_i} = 0$ (dla ograniczeń) dostaniemy te same informacje co w poprzednim rozdziale przy liniowej separowalności. Nowym elementem są warunki

$$\frac{\partial Q}{\partial \xi_i} = 0, \tag{20}$$

dla każdego z punktów danych, które dają nam

$$C - \alpha_i = 0, \tag{21}$$

co możemy traktować jako górne ograniczenie na α_i, i będziemy wstawiać $\alpha_i = C$, tam gdzie będzie to przydatne, żeby coś uprościć.

Rozpisując Q i wykorzystując zależności (12), (13) oraz (21) otrzymujemy:

$$\begin{aligned} Q(\xi_1,\ldots,\xi_I,\alpha_1,\ldots,\alpha_I) &= -\frac{1}{2}\sum_{i=1}^{I}\sum_{j=1}^{I}\alpha_i\alpha_j y_i y_j\langle x_i,x_j\rangle - \sum_{i=1}^{I}\alpha_i(\xi_i-1) + \sum_{i=1}^{I}\underbrace{C}_{\alpha_i}\xi_i \\ &= -\frac{1}{2}\sum_{i=1}^{I}\sum_{j=1}^{I}\alpha_i\alpha_j y_i y_j\langle x_i,x_j\rangle + \sum_{i=1}^{I}(-\alpha_i\xi_i+\alpha_i+\alpha_i\xi_i). \end{aligned}$$

Ostatecznie zadanie optymalizacji ma następującą postać. Maksymalizuj

$$Q(\alpha_1,\ldots,\alpha_I) = -\frac{1}{2}\sum_{i=1}^{I}\sum_{j=1}^{I}\alpha_i\alpha_j y_i y_j\langle \mathbf{x}_i,\mathbf{x}_j\rangle + \sum_{i=1}^{I}\alpha_i$$

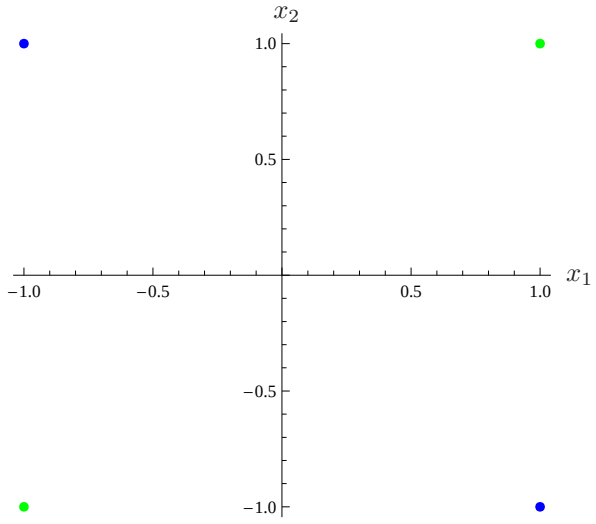
przy ograniczeniach: $\forall i \quad 0 \leqslant \alpha_i \leqslant C$,

$$\sum_{i=1}^{I}\alpha_i y_i = 0. \tag{22}$$

Jest to postać bardzo podobna do (14) z jedyną różnicą w ograniczeniach, taką że $0 \leqslant \alpha_i \leqslant C$.

5 Uogólnienie na krzywoliniowe granice klasyfikacji

Dla danych niesaprowalnych liniowo oprócz możliwości przedstawionych w poprzednim rozdziale istnieje dodatkowa możliwość zastosowania tzw. *podniesienia wymiarowości* i uzyskania krzywoliniowej granicy klasyfikacji. Prostym przykładem, który posłuży do objaśnienia tej techniki jest problem XOR, przedstawiony na rys. 10, gdzie punkty danych są rozłożone klasami „na krzyż".



Rysunek 10: Problem XOR.

5.1 Idea pomysłu podniesienia wymiarowości

Idea pomysłu z krzywoliniową granicą klasyfikacji jest następująca.

1. Jeżeli dane nie są liniowo separowalne w oryginalnej przestrzeni $\mathbb{R}^n$, to spróbujmy za pomocą pewnego przekształcenia przenieść je do wyższej przestrzeni $\mathbb{R}^m$, $m > n$, gdzie jest „luźniej" i być może tam dane będą liniowo separowalne. Innymi słowy, każdy punkt $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ odwzorujmy za pomocą pewnego przekształcenia $\phi(\mathbf{x}_i)$ w punkt $\mathbf{z}_i = (z_{i1}, z_{i2}, \ldots, z_{im})$. Tę przestrzeń nazywamy *przestrzenią cech*[5].

2. Poszukajmy optymalnej liniowej granicy decyzyjnej — płaszczyzny — w przestrzeni cech. Płaszczyzna ta będzie określona przez w_0^* i wektor normalny o większej liczbie m współrzędnych $\mathbf{w}^* = (w_1^*, w_2^*, \ldots, w_m^*)$,

3. Liniowej granicy pomiędzy klasami w przestrzeni cech będzie odpowiadała nieliniowa granica w przestrzeni oryginalnej.

Powiedzmy, że po znalezieniu optymalnej płaszczyzny chcemy sklasyfikować pewien nowo przychodzący punkt $\mathbf{x}$. Schemat postępowania jest wówczas taki:

$$\mathbf{x} \in \mathbb{R}^n \quad \overset{\phi}{\longmapsto} \quad \mathbf{z} \in \mathbb{R}^m \quad \longmapsto \quad w_0^* + \langle \mathbf{w}^*, \mathbf{z} \rangle \quad \longmapsto \quad \{-1, 1\}. \tag{23}$$

Można tu także wspomnieć, że według takiego schematu można wykreślić w przestrzeni oryginalnej krzywoliniową granicę klasyfikacji, realizując takie postępowanie dla każdego punktu siatki wykresu.

5.2 Rozwiązanie dla problemu XOR

Przyjmijmy, dla przykładu XOR, że przekształcenie ϕ działa w sposób następujący ($m = 6$):

$$\phi(\mathbf{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, {x_1}^2, {x_2}^2, \sqrt{2}x_1x_2). \tag{24}$$

Jest to tzw. *przekształcenie jądrowe o bazach wielomianowych*. Oryginalny zbiór odwzorowujemy teraz za pomocą ϕ w nowy zbiór, patrz tab. 1. Patrząc na tabelę można zauważyć, że teraz klasy stały się rozróżnialne (liniowo) choćby na podstawie samej zmiennej z_6.

[5]Przestrzeń cech może mieć nawet kilka rzędów wielkości więcej wymiarów niż przestrzeń oryginalna. W praktyce przechodzimy np. z $\mathbb{R}^2$ do $\mathbb{R}^{50}$ lub nawet $\mathbb{R}^{100}$.

i	$\mathbf{x}_i$ (x_1, x_2)	$\mathbf{z}_i$ $(1, \sqrt{2}x_1, \sqrt{2}x_2, {x_1}^2, {x_2}^2, \sqrt{2}x_1x_2)$	y_i
1	$(1,1)$	$(1, \sqrt{2}, \sqrt{2}, 1, 1, \sqrt{2})$	1
2	$(1,-1)$	$(1, \sqrt{2}, -\sqrt{2}, 1, 1, -\sqrt{2})$	-1
3	$(-1,-1)$	$(1, -\sqrt{2}, -\sqrt{2}, 1, 1, \sqrt{2})$	1
4	$(-1,1)$	$(1, -\sqrt{2}, \sqrt{2}, 1, 1, -\sqrt{2})$	-1

Tablica 1: Zbiór danych problemu XOR podniesiony do wyższej wymiarowości.

Formułujemy zadanie optymalizacji w terminach mnożników Lagrange'a:

$$Q(\alpha_1, \ldots, \alpha_4) = \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 - \frac{1}{2}\sum_{i=1}^{4}\sum_{j=1}^{4} \alpha_i \alpha_j y_i y_j \langle \underbrace{\phi(\mathbf{x}_i)}_{\mathbf{z}_i}, \underbrace{\phi(\mathbf{x}_j)}_{\mathbf{z}_j} \rangle$$

$$\begin{aligned} \text{przy ograniczeniach: } & \sum_{i=1}^{4} y_i \alpha_i = 0 \quad \Leftrightarrow \quad \alpha_1 - \alpha_2 + \alpha_3 - \alpha_4 = 0, \\ & \alpha_1 \geqslant 0, \\ & \alpha_2 \geqslant 0, \\ & \alpha_3 \geqslant 0, \\ & \alpha_4 \geqslant 0. \end{aligned} \tag{25}$$

Wszystkie potrzebne iloczyny skalarne $\langle \mathbf{z}_i, \mathbf{z}_j \rangle$ dla każdej pary i, j wynoszą (przedstawiamy je tu w formie macierzy):

$$\begin{pmatrix} 9 & 1 & 1 & 1 \\ 1 & 9 & 1 & 1 \\ 1 & 1 & 9 & 1 \\ 1 & 1 & 1 & 9 \end{pmatrix}.$$

W wyniku rozwiązania zadania programowania kwadratowego otrzymujemy optymalne rozwiązanie:

$$\alpha_1^* = \alpha_2^* = \alpha_3^* = \alpha_4^* = \frac{1}{8}.$$

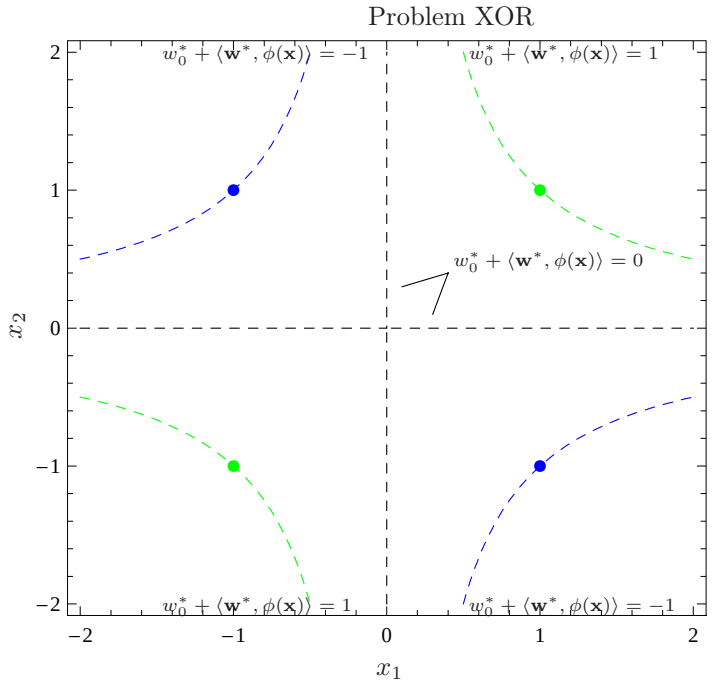
Wszystkie $\alpha_i > 0$, a więc wszystkie punkty $\mathbf{z}_i$ są punktami podparcia w przestrzeni cech. I zarazem są nimi wszystkie $\mathbf{x}_i$, tyle że w przestrzeni oryginalnej.

Wyznaczamy teraz optymalny wektor $\mathbf{w}^*$ zgodnie ze wzorem (12):

$$\mathbf{w}^* = \sum_{i=1}^{4} \alpha_i^* y_i \phi(\mathbf{x}_i) = \frac{1}{8}\sum_{i=1}^{4} y_i \phi(\mathbf{x}_i) = \Big(0, 0, 0, 0, 0, \frac{\sqrt{2}}{2}\Big),$$

oraz wyraz w_0^* na podstawie wybranego punktu podparcia np. $\phi(\mathbf{x}_1)$:

$$w_0^* = y_1 - \langle \mathbf{w}^*, \mathbf{x}_1 \rangle = 1 - \Big\langle (0, 0, 0, 0, 0, \sqrt{2}/2), (1, \sqrt{2}, \sqrt{2}, 1, 1, \sqrt{2}) \Big\rangle = 0.$$



Rysunek 11: Rozwiązanie problemu XOR — krzywoliniowa granica klasyfikacji o równaniu $w_0^* + \langle \mathbf{w}^*, \phi(\mathbf{x}) \rangle = 0$, oraz granice brzegowe $w_0^* + \langle \mathbf{w}^*, \phi(\mathbf{x}) \rangle = \pm 1$.

.

Granicę krzywoliniową w oryginalnej przestrzeni możemy narysować np. za pomocą wykresu warstwicowego poddając każdy punkt siatki wykresu postępowaniu jak na schemacie (23), patrz rys. 11.

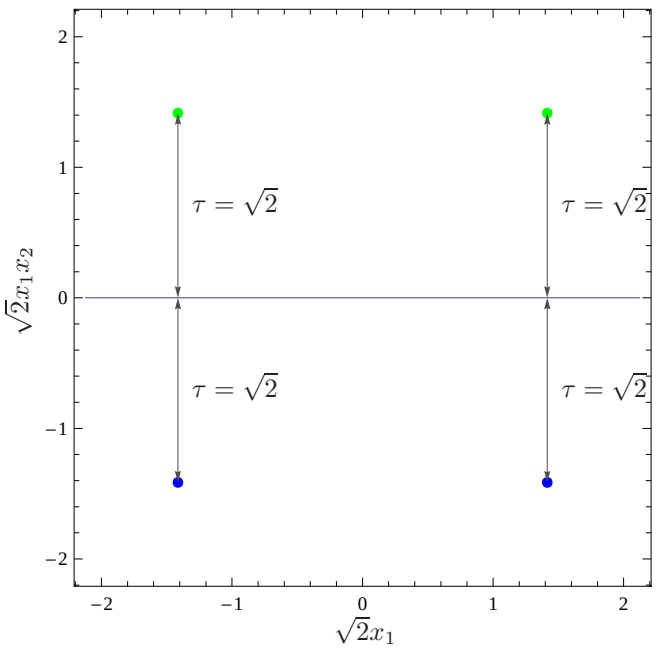
Na podstawie odwrotności normy $\|\mathbf{w}^*\|$ można też obliczyć uzyskany maksymalny margines separacji

$$\tau = \frac{1}{\|\mathbf{w}^*\|} = \frac{1}{\frac{\sqrt{2}}{2}} = \sqrt{2},$$

przy czym należy rozumieć, że jest to margines „przebywający" w przestrzeni cech po obu stronach płaszczyzny klasyfikacji w tejże przestrzeni. Rys. 12 przedstawia margines w wybranej podprzestrzeni cech: $\sqrt{2}x_1 \times \sqrt{2}x_1x_2$.

5.3 Przekształcenia jądrowe

W poprzednim rozdziale pokazano przykład przekształcenia jądrowego o bazach wielomianowych: $(1, \sqrt{2}x_1, \sqrt{2}x_2, {x_1}^2, {x_2}^2, \sqrt{2}x_1x_2)$. Patrząc na wzór (25) lub na wcześniejsze wzory (14), (22) zauważamy, że wspólnym motywem jest obliczanie iloczynów skalarnych pomiędzy wszystkimi parami punktów, bądź to w przestrzeni oryginalnej: $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$, bądź w przestrzeni cech: $\langle \mathbf{z}_i, \mathbf{z}_j \rangle = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$, gdzie operowaliśmy właśnie na bazach wielomianowych. Okazuje się, że istnieją różne dobre zestawy baz. W ogólności przekształcenia związane z tymi bazami nazywamy *przekształceniami jądrowymi* (ang. *kernel*



Rysunek 12: Uzyskany margines dla problemu XOR w wybranej podprzestrzeni cech: $\sqrt{2}x_1 \times \sqrt{2}x_1x_2$.

transformations). Funkcja realizująca te przekształcenia jest zwykle oznaczana (oprócz oznaczenia ϕ) przez K (ang. *kernel*). Najpopularniejsze dwa przekształcenia to:

- *wielomianowe* (w ogólności stopnia n):

$$K_n(\mathbf{x}_i, \mathbf{x}_j) = (1 + \langle \mathbf{x}_i, \mathbf{x}_j \rangle)^n. \tag{26}$$

- *Gaussowskie*:

$$K_\sigma(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}}. \tag{27}$$

Można łatwo sprawdzić, że w przypadku jąder wielomianowych dla $n = 2$ otrzymamy:

$$\begin{aligned} K_2(\mathbf{x}_i, \mathbf{x}_j) &= \Big(1 + \langle (x_{i1}, x_{i2}), (x_{j1}, x_{j2}) \rangle\Big)^2 \\ &= 1 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2} + {x_{i1}}^2{x_{j1}}^2 + {x_{i2}}^2{x_{j2}}^2 + 2x_{i1}x_{j1}x_{i2}x_{j2} = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle, \end{aligned}$$

czyli to, czego używaliśmy przy problemie XOR, tyle, że inaczej wyrażone.

5.4 Przykład zastosowania jądrowego przekształcenia Gaussowskiego

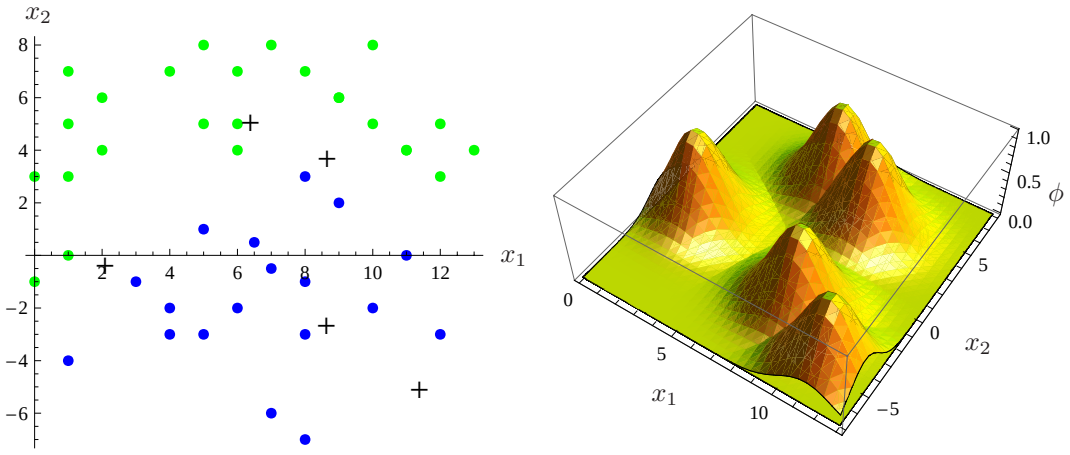
Powróćmy do zbioru danych z rys. 7. Rozwiązania liniowe dla tego przykładu tzn. płaszczyzny klasyfikacji zostały przedstawione na rys. 9. Były to płaszczyzny o różnym marginesie i różnej mierze błędu, co było konsekwencją wyboru parametru C.

Pokażemy teraz rozwiązanie tego problemu, wyznaczające krzywoliniową granicę klasyfikacji za pomocą przekształcenia jądrowego Gaussowskiego, ale w sposób nieco inny niż dane to jest wzorem (27). Po pierwsze wybierzmy najpierw m, czyli liczbę wymiarów przestrzeni, do której chcemy podnieść oryginalną przestrzeń. Powiedzmy, że ustalamy $m = 30$.[6]

Następnie umieszczamy w oryginalnej przestrzeni m punktów $\mu_k = (\mu_{k1}, \mu_{k2})$, $k = 1, \ldots, m$. Można je rozmieścić regularnie lub przypadkowo (losując ich współrzędne). Te punkty to tzw. *centra* lub właśnie *jądra*. Dla każdego z tych punktów zdefiniowana jest funkcja

$$K_\sigma(\mathbf{x}, \mu_k) = e^{-\frac{\|\mathbf{x}-\mu_k\|^2}{2\sigma^2}}, \tag{28}$$

która zwraca wartości z przedziału $[0, 1]$ i może być rozumiana jako stopień bliskości dowolnego punktu $\mathbf{x}$ do k-tego centrum/jądra. W szczególności jeżeli $\mathbf{x} = \mu_k$, to $K_\sigma(\mathbf{x}, \mu_k) = 1$. Powierzchnia generowana przez powyższą funkcję to powierzchnia o kształcie dzwonu. Na rys. 13 przedstawiono zbiór danych wraz z rozmieszczonymi losowo jądrami (czarne krzyże) oraz wizualizację funkcji dzwonowych ustawionych nad jądrami. Dla czytelności przedstawiono tylko 5 spośród $m = 30$ jąder. Parametr σ jest dobieralny i steruje szerokością dzwonów. Im większe σ tym szersze dzwony. Zaleca się wybierać σ jako odwrotnie proporcjonalne do $\frac{m}{\text{miara dziedziny}}$, tj. im większe m tym dzwony mogą być węższe, im mniejsze m tym szersze.



Rysunek 13: Zbiór danych wraz z rozmieszczonymi losowo jądrami (czarne krzyże) oraz wizualizacja funkcji dzwonowych ustawionych nad jądrami. Dla czytelności przedstawiono tylko 5 spośród $m = 30$ jąder.

W tym momencie można już podnieść zbiór danych do przestrzeni cech, w taki sposób, że każdy

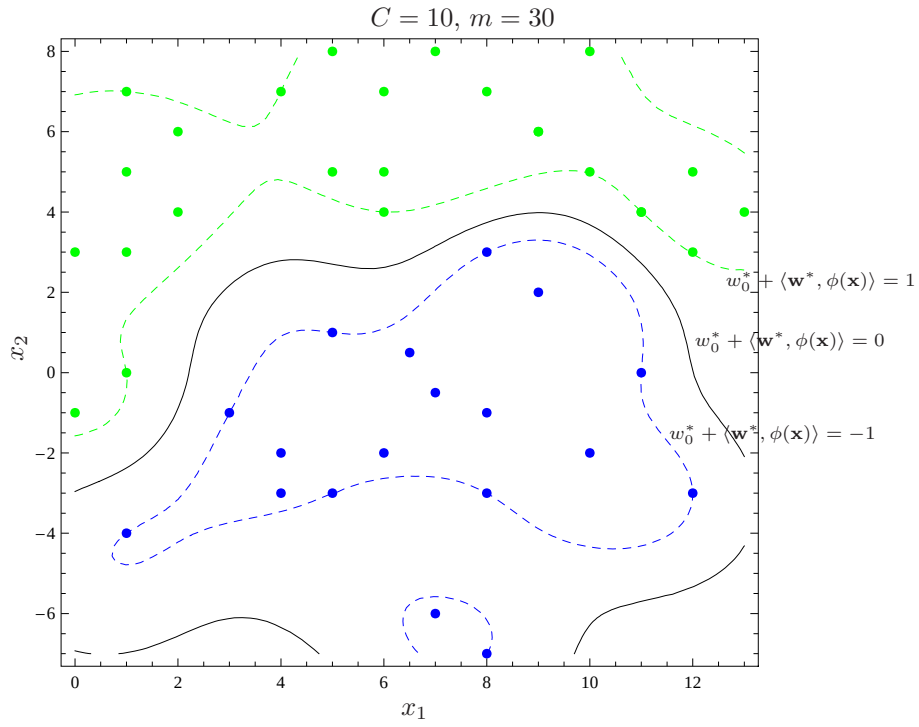
[6] Można poeksperymentować z większymi wartościami m, np. $m = 100$, $m = 1000$. Ale dla danego przykładu okaże się, że m rzędu kilkadziesiąt jest wystarczające.

punkt $\mathbf{x}_i$ odwzorowany zostanie w punkt $\mathbf{z}_i$ o m współrzędnych:

$$\mathbf{z}_i = \phi(\mathbf{x}_i) = \Big(K_\sigma(\mathbf{x}_i, \mu_1), K_\sigma(\mathbf{x}_i, \mu_2), \ldots, K_\sigma(\mathbf{x}_i, \mu_m)\Big).$$

Dobrze jest wypowiedź to sobie słownie: każdemu punktowi $\mathbf{x}_i$ przyporządkowujemy jako nowe współrzędne stopnie bliskości do poszczególnych punktów μ_k.

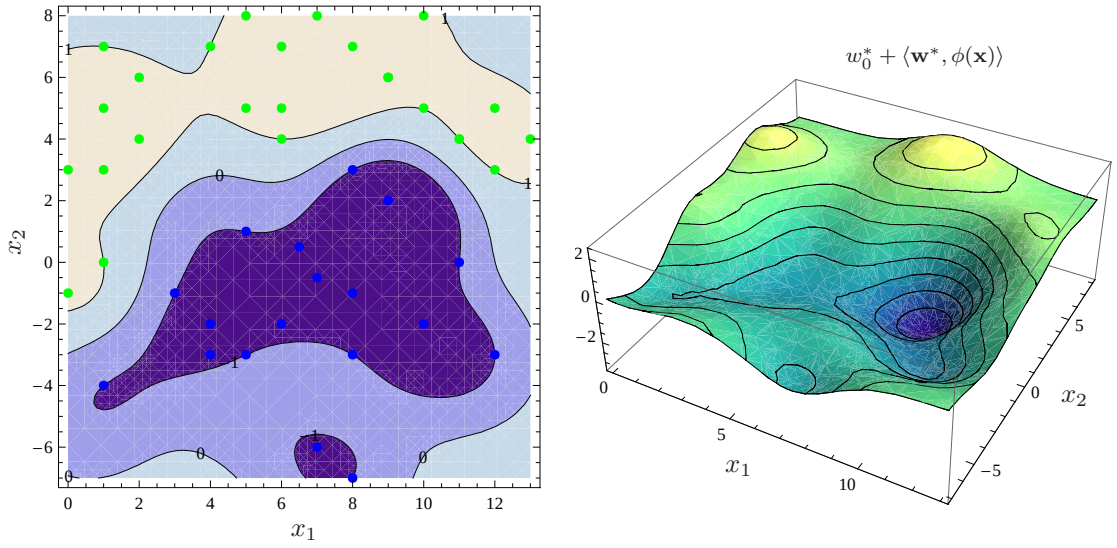
Następnie formułujemy i rozwiązujemy zadanie optymalizacji w postaci (18) lub (22), przy czym oczywiście operujemy teraz na punktach $\mathbf{z}_i$. Parametr C można ustawić na stosunkowo duży, jeżeli jednocześnie mamy duże m. Na rysunkach 14, 15 przedstawiono otrzymane rozwiązanie.



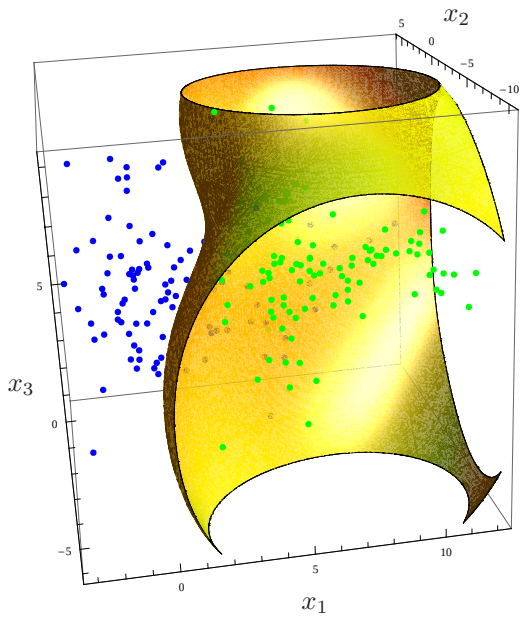
Rysunek 14: Krzywoliniowa granica decyzyjna zbudowana w oparciu o $m = 30$ jąder Gaussowskich.

5.5 Przykład krzywoliniowej granicy kasyfikacji dla zbioru danych w $\mathbb{R}^3$.

Dla lepszego wyobrażenia i utrwalenia działania metody pokazujemy dodatkowo przykładowe rozwiązanie dla klasyfikacji zbioru danych w przestrzeni $\mathbb{R}^3$ (nieseparowalnego liniowo). Patrz rys. 16.



Rysunek 15: Krzywoliniowa granica decyzyjna oraz odpowiadający jej powierzchniowy wykres wartości $w_0^* + \langle \mathbf{w}^*, \phi(\mathbf{x}) \rangle$.



Rysunek 16: Krzywoliniowa granica klasyfikacji dla zbioru danych w $\mathbb{R}^3$.