Maszyny wektorów nośnych

Część I

Aleksandra Steiner

1 Klasyfikator maksymalnego marginesu (Maximal Margin)

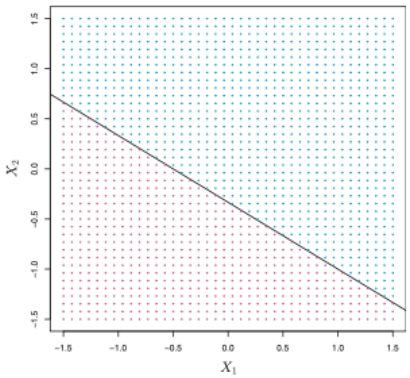
1.1 Hiperpłaszczyzna

Hiperpłaszczyzna w przestrzeni p-wymiarowej to jej afiniczna podprzestrzeń wymiaru $p-1$.
Hiperpłaszczyzna ta może być zdefiniowana przez poniższe równanie:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0. \tag{1}$$

Zatem punkt $X = (X_1, X_2, \ldots, X_p)^T$ w p-wymiarowej przestrzeni spełniający równanie (1) leży na hiperpłaszczyźnie zdefiniowanej przez nie.

O hiperpłaszczyźnie możemy myśleć jako o dzielącej p-wymiarową przestrzeń na dwie połowy. Łatwo określić, po której stronie hiperpłaszczyzny leży dany punkt przez policzenie, czy dany $X = (X_1, X_2, \ldots, X_p)^T$ spełnia $\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_p X_p > 0$ czy też $\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_p X_p < 0$.



Rysunek 1: Przykład hiperpłaszczyzny $1+2X_1+3X_2 = 0$. Niebieski obszar to punkty, dla których $1+2X_1+3X_2 > 0$, różowy to rzecz jasna $1+2X_1+3X_2 < 0$

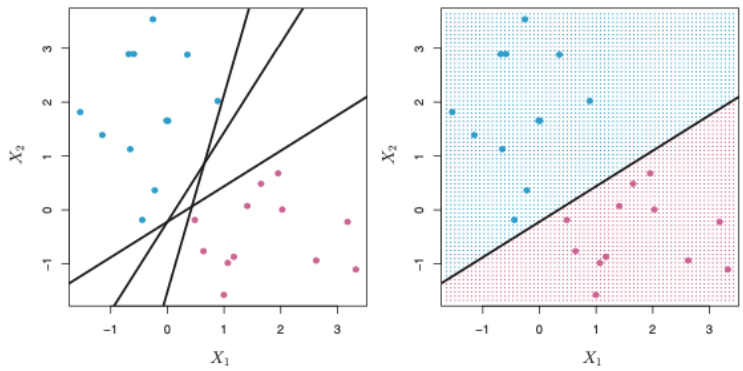
1.2 Klasyfikacja z użyciem hiperpłaszczyzny separującej

Załóżmy, że dysponujemy macierzą danych X wymiaru $n \times p$, która zawiera n obserwacji treningowych w p-wymiarowej przestrzeni:

$$x_1 = \begin{pmatrix} x_{11} \\ . \\ . \\ . \\ x_{1p} \end{pmatrix}, \ldots, x_n = \begin{pmatrix} x_{n1} \\ . \\ . \\ . \\ x_{np} \end{pmatrix}. \tag{2}$$

Załóżmy również, że obserwacje te wpadają do jednej z dwóch klas, to znaczy $y_1, \ldots, y_n \in \{-1, 1\}$, zatem mamy klasę reprezentowaną przez -1 oraz 1. Mamy tutaj również obserwację testową, to znaczy wektor p-elementowy zaobserwowanych cech $x^* = (x_1^*, \ldots, x_p^*)^T$. Cel jaki mamy to opracowanie klasyfikatora opartego na danych treningowych, który poprawnie klasyfikuje obserwację testową znając wartości cech, jakie ona przyjmuje. Znanych jest oczywiście wiele podejść do takiego problemu, my pokażemy sposób oparty na pojęciu hiperpłaszczyzny separującej.

Załóżmy, że możliwe jest skonstruowanie hiperpłaszczyzny, która idealnie separuje obserwacje treningowe zgodnie z ich klasami.



Rysunek 2: Widzimy dwie klasy obserwacji, każda z nich przyjmuje wartości dla dwóch zmiennych: X_1 i X_2. Po lewej zaprezentowano trzy różne separujące hiperpłaszczyzny, po prawej umieszczono jedną hiperpłaszczyznę separującą.

Przy powyższej sytuacji możemy przyjąć oznaczenie obserwacji z klasy niebieskiej jako $y_i = 1$ oraz tych z klasy różowej jako $y_i = -1$. Wtedy dla hiperpłaszczyzny separującej wyrażenie $\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}$ jest dodatnie jeśli $y_i = 1$ oraz jest ujemne, gdy $y_i = -1$.

Równoważnie, można zapisać, że dla tej hiperpłaszczyzny separującej zachodzi:

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) > 0 \text{ dla } i \in 1, \ldots, n \tag{3}$$

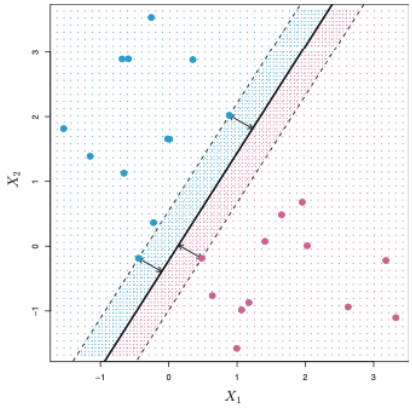
Zatem jeśli hiperpłaszczyzna separująca istnieje, możemy jej użyć do skonstruowania bardzo naturalnego klasyfikatora, który przypisuje daną obserwację do klasy zależnie po której strony hiperpłaszczyzny się ona znajduje. Tak więc klasyfikujemy obserwację x^* zależnie od znaku $f(x^*) = \beta_0 + \beta_1 x_1^* + \beta_2 x_2^* + \cdots + \beta_p x_p^*$. Można tutaj również dokonywać oceny w takiej postaci: jeśli $f(x^*)$ jest daleka od 0, to znaczy, że x^* leży daleko od hiperpłaszczyzny, czyli możemy być pewni co do przypisania x^*. Z drugiej strony, gdy $f(x^*)$ jest bliskie 0, nie mamy już tak dużej pewności co do klasyfikacji obserwacji x^*.

1.3 Klasyfikator maksymalnego marginesu

Jeśli istnieje hiperpłaszczyzna, która idealnie separuje nasze dane, to znaczy, że takich hiperpłaszczyzn jest nieskończenie wiele. Aby skonstruować klasyfikator oparty na hiperpłaszczyźnie separującej należy określić w odpowiedni sposób, którą z takich nieskończenie wielu hiperpłaszczyzn użyć.

Jak się okazuje, naturalnym wyborem jest hiperpłaszczyzna maksymalnego marginesu, która to znajduje się najdalej od obserwacji treningowych. Wyraz *margin* odnosi się tutaj do najmniejszej z odległości pomiędzy obserwacjami a hiperpłaszczyzną. Zatem hiperpłaszczyzna maksymalnego marginesu jest hiperpłaszczyzną separującą, dla której *margin* jest największy. Bazując na takiej hiperpłaszczyźnie, rozważamy klasyfikator maksymalnego marginesu, przy którym klasyfikacja wygląda tak, jak opisaliśmy to wyżej w podrozdziale (1.2). Recz jasna pojawia się tutaj zawsze nadzieja, że klasyfikator z dużą wartością *margin* dla danych treningowych będzie miał również duży *margin* w przypadku danych testowych, dzięki czemu klasyfikacja będzie poprawna. Jednak pomimo faktu, że klasyfikator maksymalnego marginesu jest często skuteczny, to niestety w przypadku, gdy liczba predykatorów jest duża (p), obserwujemy tzw. *overfitting*.

Analizując poniższą ilustrację widzimy, że trzy z obserwacji treningowych są tak samo oddalone od hiperpłaszczyzny. W tym przypadku te trzy obserwacje nazywamy *wektorami nośnymi*, ponieważ są one wektorami w dwuwymiarowej przestrzeni i niejako „podpierają" one hiperpłaszczyznę maksymalnego marginesu (przy zmianie ich położenia zmienia się również hiperpłaszczyzna). Co więc ważne, hiperpłaszczyzna maksymalnego marginesu zależy bezpośrednio wyłącznie od małego podzbioru obserwacji. Jest to istotna własność, która pojawi się poniżej w kontekście klasyfikatora i maszyn wektorów nośnych.



Rysunek 3: Dwie te same grupy obserwacji x poprzedniego przykładu z hiperpłaszczyzną maksymalnego marginesu.

1.4 Konstrukcja klasyfikatora maksymalnego marginesu

Można się teraz zastanowić, jak skonstruować hiperpłaszczyznę maksymalnego marginesu mając zestaw n obserwacji treningowych $x_1, \ldots, x_n \in \mathbb{R}^p$ oraz odpowiadające klasy $y_1, \ldots, y_n \in \{-1, 1\}$. Otóż taka hiperpłaszczyzna jest rozwiązaniem problemu optymalizacyjnego:

$$\underset{\beta_0,\ldots,\beta_p}{\text{maximize}} M \tag{4}$$

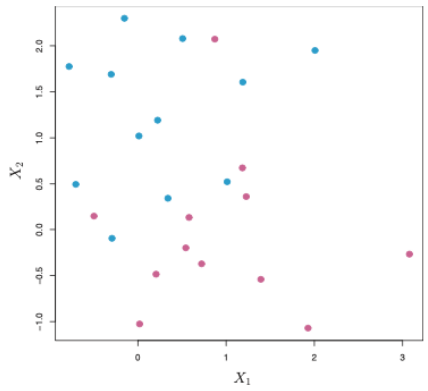
$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1, \tag{5}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geq M \ \forall i = 1, \ldots, n \tag{6}$$

Część zadanego problemu optymalizacyjnego w (6) daje nam gwarancję, że każda obserwacja zostanie przyporządkowana odpowiedniej stronie hiperpłaszczyzny. Wzór na odległość punktu od hiperpłaszczyzny $\beta x + \beta_0$, gdzie $\beta = (\beta_1, \ldots, \beta_p)$ dany przez $(\beta x_i + \beta_0)/||\beta||$ oraz warunek $\sum_{j=1}^{p} \beta_j^2 = 1$ pozwalają na stwierdzenie, że wzór na odległość i-tej obserwacji od hiperpłaszczyzny można również opisać wzorem $y_i(\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\cdots+\beta_p x_{ip})$. Zatem M stanowi tutaj *margin* hiperpłaszczyzny i problem optymalizacyjny sprowadza się do znalezienia takich $\beta_0, \ldots, \beta_p$, aby M było jak największe.

1.5 Przypadek nieseparowalny

Kolejnym pytaniem, jakie pojawia się przy tych rozważaniach jest: co w sytuacji, gdy hiperpłaszczyzna separująca nie istnieje? Dzieje się tak w wielu przypadkach, tak więc nie może być wtedy użyty klasyfikator maksymalnego marginesu.



Rysunek 4: Dwie klasy obserwacji, które nie są separowalne przez hiperpłaszczyznę.

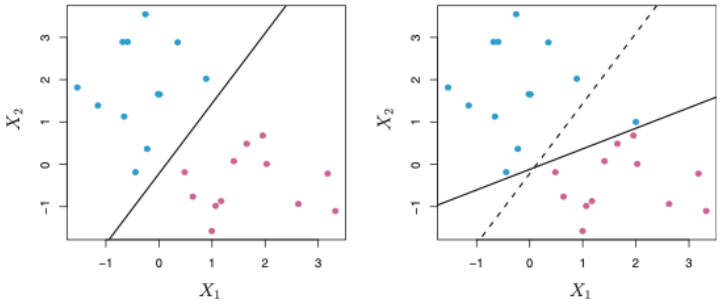
W sytuacji jak powyższa nie jesteśmy w stanie dokładnie odseparować dwóch klas. Jednak jak zaraz się dowiemy, istnieje rozszerzenie pojęcia hiperpłaszczyzny separującej do hiperpłaszczyzny, która separuje klasy *niemalże* dokładnie przy użyciu tzw. miękkiego marginesu. Uogólnienie klasyfikatora maksymalnego marginesu na przypadek nieseparowalny znany jest pod pojęciem *klasyfikatora wektorów nośnych*.

2 Klasyfikatory wektorów nośnych

2.1 Omówienie

Nawet w sytuacjach, gdy hiperpłaszczyzna separująca istnieje, zdarzają się przypadki, w których klasyfikator oparty na hiperpłaszczyźnie wydaje się nie być odpowiednim. Może on świetnie separować klasy dla obserwacji treningowych, jednak już nie radzić sobie zupełnie przy zbiorze testowym. Taki klasyfikator może wykazywać wrażliwość na pojedyncze obserwacje.

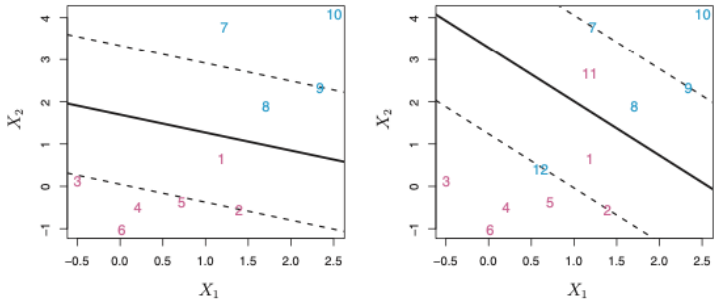
Pojawienie się nowej, pojedynczej obserwacji sprawiło (Rysunek 5), że nowa hiperpłaszczyzna maksymalnego marginesu musiała ulec znaczącej zmianie. Wynikowa, nowa hiperpłaszczyzna posiada już dużo mniejszy margines, co jest problematyczne, ponieważ jest to w jakiś sposób miara pewności, że dana obserwacja została poprawnie zaklasyfikowana. Co więcej, sam fakt, że hiperpłaszczyzna maksymalnego marginesu jest bardzo wrażliwa na zmianę pojedynczej obserwacji sugeruje, że może ona wykazywać nadmierne dopasowanie względem danych treningowych.



Rysunek 5: Dane odseparowane hiperpłaszczyzną maksymalnego marginesu przed dodaniem nowej obserwacji (po lewej) oraz po pojawieniu się nowej obserwacji w klasie niebieskiej (po prawej)

Zatem chcielibyśmy rozważyć klasyfikator oparty na hiperpłaszczyźnie, który mimo, że nie separuje dwóch klas idealnie to wykazuje większą odporność na indywidualne obserwacje. Być może warto byłoby „poświęcić" poprawną klasyfikację kilku obserwacji na rzecz lepszego dopasowania do pozostałych danych. **Klasyfikator wektorów nośnych** (z ang. *support vector classifier*) wykazuje właśnie takie cechy. Nie znajduje on największego możliwego marginesu, pozwala pewnym obserwacjom nie być po prawidłowej stronie marginesu, a nawet samej hiperpłaszczyzny.

Na poniższej ilustracji po lewej widzimy przykład użycia klasyfikatora wektorów nośnych, w którym większość obserwacji jest po właściwej stronie marginesu. Po prawej widoczny jest scenariusz, w którym nie istnieje idealna hiperpłaszczyzna m. m., tak więc sytuacja, w której obserwacje znajdą się po niewłaściwej stronie hiperpłaszczyzny jest nieunikniona, jednak taki klasyfikator zadziała lepiej dla danych testowych.



Rysunek 6: Działanie klasyfikatora wektorów nośnych na małym zestawie danych przed dodaniem (po lewej) i po dodaniu (po prawej) punktów 11 i 12

2.2 Szczegóły

Klasyfikator wektorów nośnych również przypisuje obserwacji pewną klasę zależnie od tego, po której stronie hiperpłaszczyzny się ona znajduje. Sama hiperpłaszczyzna wybrana jest tak, aby separowała ona większość treningowych obserwacji na dwie klasy, jednak dopuszcza błędną klasyfikację małej części obserwacji.
Jest ona ponownie rozwiązaniem problemu optymalizacyjnego:

$$\underset{\beta_0,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n}{\text{maximize}} \; M \tag{7}$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1, \tag{8}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geq M(1 - \epsilon_i), \tag{9}$$

$$\epsilon_i \geq 0, \; \sum_{i=1}^{n} \epsilon_i \leq C. \tag{10}$$

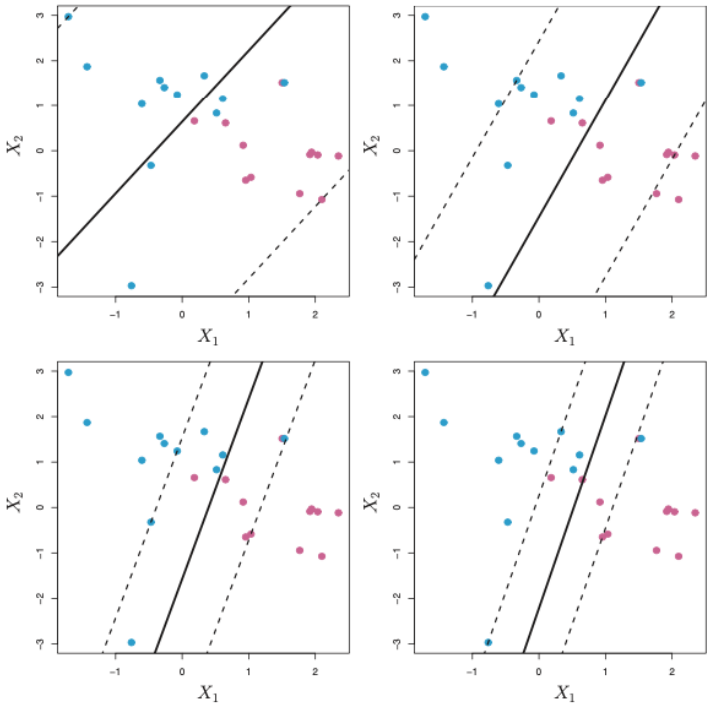
C w powyższym problemie jest nieujemnym parametrem regulującym, M to znów szerokość marginesu i chcemy go maksymalizować. $\epsilon_1, \ldots, \epsilon_n$ to tzw. zmienne luzu, które pozwalają pojedynczym obserwacjom znaleźć się po złej stronie marginesu czy hiperpłaszczyzny. Ponownie, klasyfikujemy obserwację x^* bazując na znaku $f(x^*) = \beta_0 + \beta_1 x_1^* + \cdots + \beta_p x_p^*$.

Podobnie jak wcześniej, opisany wyżej problem optymalizacyjny wydaje się być złożony, jednak może być rozwiązany z użyciem kilku prostych obserwacji.

Po pierwsze, zmienne luzu ϵ_i mówią nam o tym gdzie położona jest i-ta obserwacja względem hiperpłaszczyzny czy też marginesu. Zauważmy, że jeśli $\epsilon_i = 0$, wtedy i-ta obserwacja znajduje się po poprawnej stronie marginesu. Gdy $\epsilon_i > 0$, to znaczy, że i-ta obserwacja jest na pewno po złej stronie marginesu, jednak gdy $\epsilon_i > 1$, wtedy widzimy, że i-ta jest już po złej stronie hiperpłaszczyzny.

Jeśli chodzi o parametr regulujący C, można zauważyć, że zgodnie z (10) ogranicza on sumę parametrów luzu, czyli niejako „naruszeń" marginesu przez n obserwacji. Jeśli $C = 0$, to znaczy, że $\epsilon_1 = \cdots = \epsilon_n = 0$ i problem sprowadza się do problemu optymalizacyjnego hiperpłaszczyzny maksymalnego marginesu. W sytuacji, gdy $C > 0$ dopuszczamy maksymalnie C obserwacji, które mogą być po złej stronie hiperpłaszczyzny. Jeśli C rośnie, jesteśmy bardziej tolerancyjni na naruszenia marginesu, czyli margines poszerza się.

W praktyce, zazwyczaj regulujący parametr C wybierany jest z użyciem walidacji krzyżowej. Parametr ten kontroluje kompromis między obciążeniem a wariancją. Większe wartości C pozwalają na zwiększenie marginesu, co jest równoznaczne z otrzymaniem klasyfikatora, który potencjalnie będzie bardziej obciążony, jednak może mieć mniejszą wariancję.



Rysunek 7: Użycie klasyfikatora wektorów nośnych dla czterech różnych wartości parametru C. Ilustracja dla największej wartości C znajduje się po lewej na górze, kolejne, mniejsze wartości C mamy po prawej na górze, po lewej i prawej na dole.

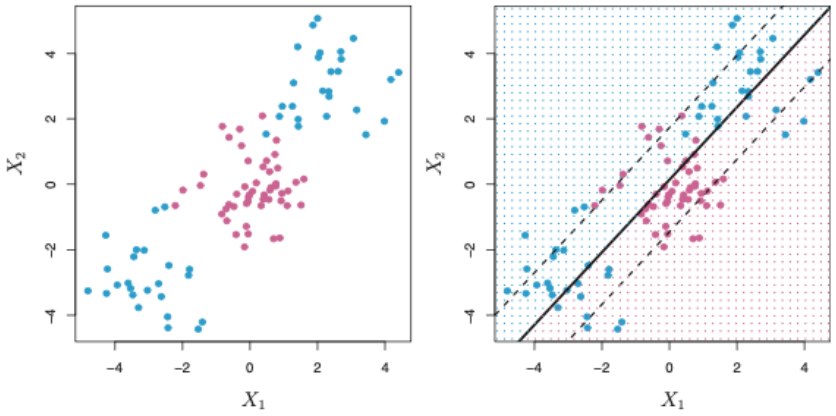
Dyskutowany problem optymalizacyjny ma bardzo interesującą własność: okazuje się, że jedynie obserwacje, które leżą na marginesie lub naruszają go mają wpływ na hiperpłaszczyznę. To znaczy, że zmiana położenia obserwacji, które leżą po dobrej stronie marginesu (na położenie, które będzie wciąż po dobrej stronie marginesu) nie ma żadnego wpływu na klasyfikator. Dlatego też, obserwacje, które leżą bezpośrednio na marginesie czy też po złej jego stronie nazywa się wektorami nośnymi.

Fakt, że wyłącznie wektory nośne mają wpływ na klasyfikator jest zgodny ze stwierdzeniem, że C kontroluje jego tzw. *bias-variance trade-off* (Rysunek 8). Ponieważ klasyfikator wektorów nośnych jest oparty wyłącznie na potencjalnie małym podzbiorze obserwacji treningowych (wektorów nośnych), oznacza to, że jest całkiem odporny na zachowanie obserwacji, które są daleko od hiperpłaszczyzny (inaczej niż przy LDA, podobnie jak przy regresji logistycznej).

3 Maszyny wektorów nośnych

3.1 Klasyfikacja z nieliniową granicą decyzyjną

Klasyfikator wektorów nośnych jest naturalnym podejściem przy klasyfikacji do dwóch klas, jeśli granica pomiędzy klasami jest liniowa. Jednakże, w praktyce mierzymy się czasami z nieliniowymi granicami między klasami.



Rysunek 8: Po lewej: Obserwacje z dwóch klas, z nieliniową granicą. Po prawej: Te same obserwacje z zastosowaniem klasyfikatora wektorów nośnych znajdującego liniową granicę między klasami.

Klasyfikator wektorów nośnych użyty dla powyższych danych (po prawej) staje się bezużyteczny. W takiej sytuacji, zamiast dopasowywać klasyfikator wektorów nośnych używając p cech $X_1, X_2, \ldots, X_p$, możemy na przykład dopasować ten klasyfikator używając $2p$ cech $X_1, X_1^2, X_2, X_2^2 \ldots, X_p, X_p^2$.

Rozszerzanie przestrzeni cech w sposób umożliwiający skuteczne wykonywanie obliczeń jest możliwe dzięki *maszynie wektorów nośnych*.