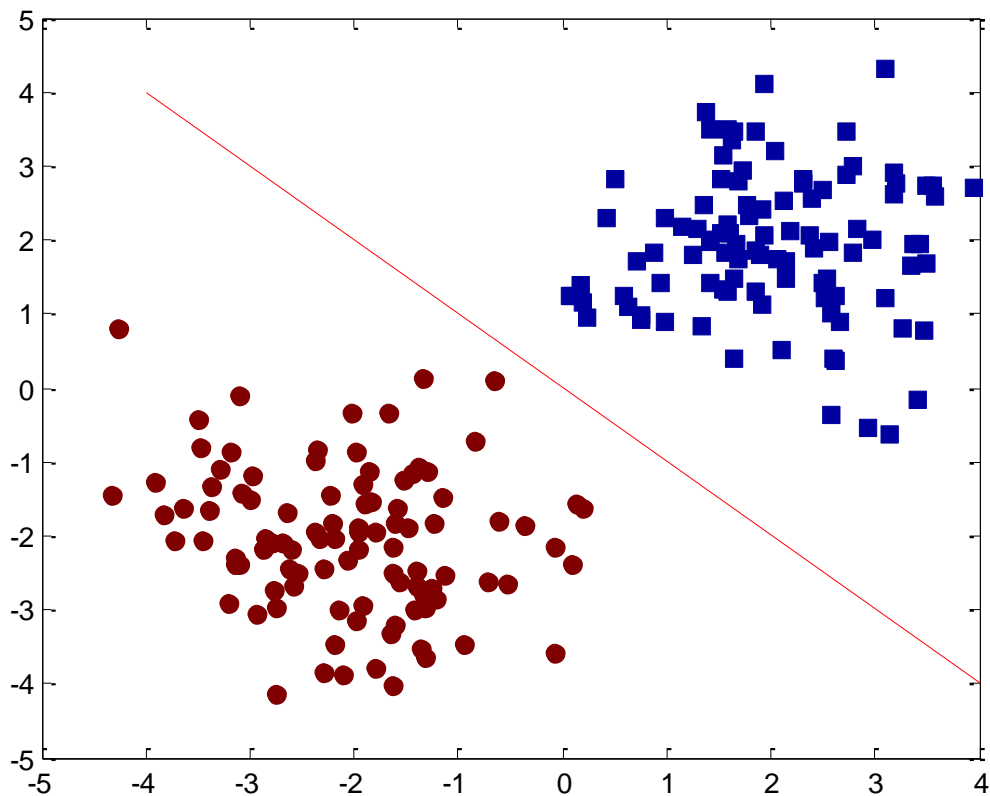


Klasyfikator liniowy

Wstęp

Klasyfikator liniowy jest najprostszym możliwym klasyfikatorem. Zakłada on liniową separację – liniowy podział dwóch klas między sobą. Przedstawia to poniższy rysunek:



Gdzie dwie klasy oddzielone są hiperpłaszczyzną, która dla przedstawionego przypadku redukuje się do linii prostej w n-wymiarowej przestrzeni. Funkcję taką nazywamy funkcją decyzyjną

$$g(\mathbf{x}) = \mathbf{w}\mathbf{x}^T = w_1x_1 + w_2x_2 + \dots + w_nx_n$$

Gdzie

\mathbf{w} – szukany wektor wag

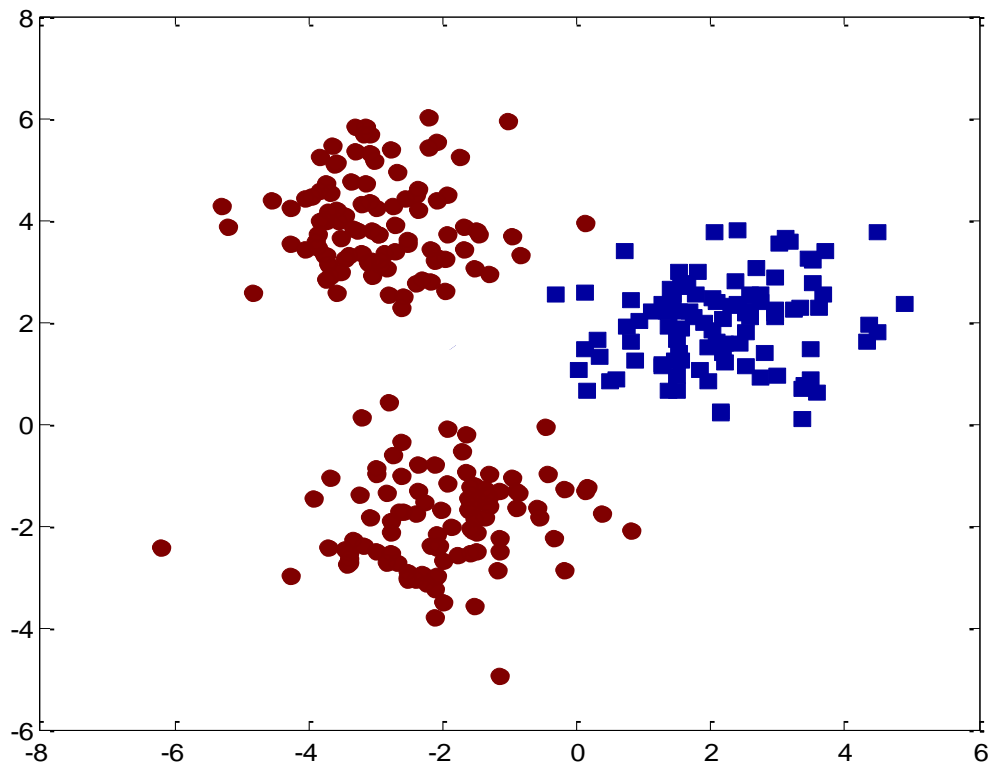
\mathbf{x} – wektor wejściowy/wektor sygnału wejściowego

Funkcja decyzyjna rozdziela więc obydwie klasy i określa wyjście modelu wg. zależności

$$g(\mathbf{x}) > 0 - \text{klasa c1}$$

$$g(\mathbf{x}) < 0 - \text{klasa c2}$$

W pewnych sytuacjach dyskryminacja liniowa nie jest możliwa do osiągnięcia, jak na poniższym obrazku,



dlatego też część modeli liniowych uwzględnia taką możliwość poprzez minimalizację liczby popełnianych błędów.

Do typowych modeli dyskryminacji liniowej można zaliczyć:

- algorytm Rosenblatta
- algorytm Fishera
- algorytm minimalizacji MSE czyli wykorzystanie regresji liniowej

Algorytm Rosenblatta

Algorytm Rosenblatta zakłada liniową separowalność klas. Stara się on minimalizować funkcję kosztu zdefiniowaną jako:

$$J(\mathbf{w}) = \sum_{\mathbf{x} \in \Omega} (-\mathbf{w}^T \mathbf{x})$$

Gdzie Ω - zbiór wektorów błędnie klasyfikowanych. Algorytm Rosenblatta można opisać w postaci minikodu

```

t – numer iteracji
w – wektor wag
C(x) – funkcja zwracająca etykietę klasy dla wektora x
m – liczba wektorów uczących
t = 0
w = generuj losowo w początkowe
b = generuj losowo
repeat
    t0=t;
    for i=1:m
        g = wTxi+b;
        if (C(xi)=1) and (g<0) =>
            w=w+xi; t=t+1
            b = b - g;
        if (C(xi)=-1) and (g>0) =>
            w=w-xi; t=t+1
            b = b - g;
    end
until t0=t

```

lub z dobną modyfikacją zapewniającą lepszą zbieżność procesu optymalizacji

```

t – numer iteracji
w – wektor wag
C(x) – funkcja zwracająca etykietę klasy dla wektora x
m – liczba wektorów uczących
η(j) – współczynnik uczenia -> funkcja malejąca w czasie np.

```

$$\eta(t) = \frac{1}{t+1}$$

```

t = 0
w = generuj losowo
b = generuj losowo
for t=1:max_iter
    for i=1:m
        g = wTxi+b;
        if (C(xi)=1) and (g<0) =>
            w=w+η(t)xi
            b = b - η(t)*g;
        if (C(xi)=-1) and (g>0) =>
            w=w-η(t)xi
            b = b - η(t)*g;
    end
end
end

```

Algorytm Fishera

Algorytm Fishera jest klasyfikatorem wywodzącym się z metod statystycznych. Zakłada on, że rozkłady poszczególnych klas mają rozkład Gaussa.

Przyjmując oznaczenia

$\vec{\mu}_{y=0}, \vec{\mu}_{y=1}$ - wektor opisujący średnie poszczególnych klas (środki rozkładów Gaussa)

$\Sigma_{y=0}, \Sigma_{y=1}$ - macierze kowariancji poszczególnych klas

I definiując funkcję kosztu jako iloraz rozproszeń międzyklasowych w stosunku do rozproszeń wewnątrz klasowych jako:

$$S = \frac{\sigma_{between}^2}{\sigma_{within}^2} = \frac{(\vec{w} \cdot \vec{\mu}_{y=1} - \vec{w} \cdot \vec{\mu}_{y=0})^2}{\vec{w}^T \Sigma_{y=1} \vec{w} + \vec{w}^T \Sigma_{y=0} \vec{w}} = \frac{(\vec{w} \cdot (\vec{\mu}_{y=1} - \vec{\mu}_{y=0}))^2}{\vec{w}^T (\Sigma_{y=0} + \Sigma_{y=1}) \vec{w}}$$

można pokazać że optymalnym rozwiązaniem tak postawionego problemu jest hiperpłaszczyzna \mathbf{w} określona zależnością:

$$\vec{w} = (\Sigma_{y=0} + \Sigma_{y=1})^{-1} (\vec{\mu}_{y=1} - \vec{\mu}_{y=0})$$

Gdzie wartość wyrazu wolnego wynosi

$$b = -\frac{\mu_{y=1} + \mu_{y=0}}{2} \mathbf{w}$$

UWAGA

Uwaga na znaki we wzorze na \mathbf{w} !!!

Wykorzystanie regresji liniowej

Jednym z często stosowanych algorytmów jest wykorzystanie regresji liniowej. W tym przypadku funkcja celu zdefiniowana jest jako minimalizacja błędu średniokwadratowego opisanego zależnością:

$$J(\mathbf{x}) = (\mathbf{w}^T \mathbf{x} - \mathbf{y}) = \sum_{i=1}^m (\mathbf{w}^T \mathbf{x}_i - y_i)^2$$

Gdzie

\mathbf{y} – por zadane wyjście

Wówczas szukając minimum tak zdefiniowanej funkcji celu uzyskujemy pochodną:

$$\nabla J = \sum_{i=1}^m 2(\mathbf{w}^T \mathbf{x}_i - y_i) \mathbf{x}_i = 2\mathbf{X}^T (\mathbf{X}\mathbf{w} - \mathbf{Y})$$

Którą minimalizując ze względu na \mathbf{w} uzyskujemy rozwiązanie:

$$\mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} = \mathbf{X}^* \mathbf{y}$$

Gdzie

$$\mathbf{X}^* = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T - \text{macierz pseudoodwrotna}$$

Zadania

Zad 1) Zaimplementuj klasyfikator liniowy Rosenblatta.

Zad 2) Zaimplementuj klasyfikator liniowy Fishera, pamiętaj o wyrazie wolnym. W tym celu doglej go do końca wektora \mathbf{w} jako $w = [w' \ b]$

Przydatne funkcje

$C = cov(x)$ – zwraca macierz kowariancji macierzy x

$M = mean(x)$ – zwraca wartość średnią dla każdej kolumny macierzy x

$S = std(x)$ – zwraca odchylenie standardowe dla każdej kolumny macierzy x

$V = var(x)$ – zwraca wariancję dla każdej kolumny macierzy x

Zad 3) Zaimplementuj klasyfikator liniowy w oparciu o regresję liniową

Uwaga:

Pamiętaj o wyrazie wolnym. Wykorzystując metodę regresji do zbioru danych uczących doglej kolejną kolumnę wypełnioną wartościami 1, pełniącą rolę wyrazu wolnego.

Np.

x - dane treningowe wejściowe

Y – etykiety klas

$n_x = [x \ ones(size(x,1),1)]$;

wówczas po wyznaczeniu wektora \mathbf{w} podczas klasyfikacji danych testowych również musisz dogleić kolumnę z wartościami 1.

Zad 4) Zwizualizuj uzyskane wyniki na przykładzie zbioru Gaus21 oraz Gaus22.

Zad 5) Porównaj dokładność zaimplementowanych metod na zbiorach danych

- WBC – Wisconsin Breast Cancer
- Pima - Pima Indian Diabetes
- Ionosphere

W tym celu podziel każdy zbiór danych wcześniej przygotowanym algorytmem podziel, dokonaj uczenia modelu na zbiorze treningowym, a następnie wykonaj testowanie wyznaczając błąd na zbiorze testowym.

UWAGA

Każdy z Algorytmów zaimplementuj jako nową funkcję: Rozenblatt_ucz, Fisher_Ucz oraz Reg_lin_ucz

Każda z funkcji powinna zwrócić parametr w opisujący równanie prostej,

Uzyskany wynik przełącz do jednej funkcji lin_test i dokonaj klasyfikacji.