Raport z Ćwiczenia 4

Bartłomiej Rasztabiga 304117

16 grudnia 2021

1 Treść zadania

Zaimplementuj algorytm SVM oraz zbadaj działanie algorytmu w zastosowaniu do zbioru danych Wine Quality Data Set. W celu dostosowania zbioru danych do problemu klasyfikacji binarnej zdyskretyzuj zmienną objaśnianą. Pamiętaj, aby podzielić zbiór danych na zbiór trenujący oraz uczący. Zbadaj wpływ hiperparametrów na działanie implementowanego algorytmu. W badaniach rozważ dwie różne funkcje jądrowe poznane na wykładzie.

2 Opis implementowanego algorytmu

Aby zdyskretyzować problem klasyfikacji wprowadzam zmienną objaśnianą: 1 jeżeli ocena jest wyższa od 5 oraz -1 w przeciwnym wypadku.

W celu wyznaczenia optymalnych wag w algorytmie SVM trzeba rozwiązać problem znalezienia maksymalnego marginesu hiperpłaszczyzny oddzielającej dwie klasy punktów.

$$\min \frac{1}{2}||w||^2 \quad \text{given} \quad \sum_i^m y_i(w \cdot x_i + b) - 1 \geqslant 0 \tag{1}$$

W tym celu użyję postaci dualnej z wprowadzeniem mnożników Lagrange'a.

$$L = \frac{1}{2}||w||^2 - \sum_i^m \lambda_i \left[y_i(w \cdot x_i + b) - 1\right] \quad \text{given} \quad \lambda_i \geqslant 0 \tag{2}$$

$$\text{maximize } L_D = \sum_i^m \lambda_i - \frac{1}{2}\sum_i^m \sum_j^m \lambda_i \lambda_j y_i y_j (x_i \cdot x_j) \quad \text{given} \quad \sum_i^m \lambda_i y_i = 0,\ \lambda_i \geqslant 0 \tag{3}$$

W powyższym wzorze, szukanymi wagami są λ. Po ich wyliczeniu będę w stanie obliczyć potrzebny bias i na jego podstawie granicę między klasami obiektów.

Z powodu użycia zewnętrznego solvera *cvxopt* muszę powyższe równania skonwertować do odpowiedniej postaci. Solver ten wymaga zapisania problemu w następującej formie:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}x^T P x + q^T x \\ \text{subject to} \quad & Gx \preceq h \\ & Ax = b \end{aligned} \tag{4}$$

Aby zamienić nasz problem maksymalizacji na wymagany problem minimalizacji, przemnażam równanie przez -1 otrzymując:

$$\text{minimize } L_D = \frac{1}{2}\sum_i^m \sum_j^m \lambda_i \lambda_j y_i y_j (x_i \cdot x_j) - \sum_i^m \lambda_i \quad \text{given} \quad \sum_i^m \lambda_i y_i = 0,\ \lambda_i \geqslant 0 \tag{5}$$

Następnie dokonuję serii przekształceń macierzowych przedstawionych poniżej, w celu otrzymania odpowiedniej postaci problemu:

$$\begin{aligned}\frac{1}{2}x^T P x &= \frac{1}{2}\begin{bmatrix}x_1 & x_2 & \dots & x_m\end{bmatrix} \cdot \begin{bmatrix}P_{11} & P_{12} & \dots & P_{1m} \\ P_{21} & P_{22} & & \\ \vdots & & \ddots & \\ P_{m1} & & & P_{mm}\end{bmatrix} \cdot \begin{bmatrix}x_1 \\ x_2 \\ \vdots \\ x_m\end{bmatrix} \\ &= \frac{1}{2}(x_1^2 \cdot P_{11} + x_1 x_2 \cdot P_{12} + x_2 x_1 \cdot P_{21} + x_2^2 \cdot P_{22} + \dots) \\ &= \frac{1}{2}\sum_i^m \sum_j^m x_i x_j \cdot P_{ij}\end{aligned} \tag{6}$$

$$\frac{1}{2}\sum_i^m \sum_j^m \lambda_i \lambda_j \cdot P_{ij} = \frac{1}{2}\sum_i^m \sum_j^m \lambda_i \lambda_j y_i y_j (x_i \cdot x_j) \quad \Rightarrow \quad P_{ij} = y_i y_j (x_i \cdot x_j) \tag{7}$$

$$q^T \lambda = \sum_i^m q_i \lambda_i = -\sum_i^m \lambda_i \quad \Rightarrow \quad q_i = -1 \tag{8}$$

Wyrażenie $Gx \preceq h$ reprezentuje ograniczenia w zagadnieniu optymalizacji. W naszym przypadku ograniczeniem tym jest $\lambda_i \geqslant 0$ dla każdego i od 0 do m. To ograniczenie może zostać przepisane na akceptowalną postać: $-\lambda_i \leqslant 0$. Przekształcenie to daje nam następujące macierze G i h:

$$G = \begin{bmatrix}-1 & 0 & 0 & \dots & 0 \\ 0 & -1 & 0 & & \\ 0 & 0 & -1 & & \vdots \\ \vdots & & & \ddots & \\ 0 & & \dots & & -1\end{bmatrix} \quad h = \begin{bmatrix}0 \\ 0 \\ \vdots \\ 0\end{bmatrix} \tag{9}$$

Finalnie, równanie $Ax = b$ reprezentuje kolejne ograniczenie problemu optymalizacji: $\sum_i^m \lambda_i yi = 0$. Otrzymujemy poniższe macierze A i b:

$$A = \begin{bmatrix}y_1 & y_2 & \dots & y_m\end{bmatrix} \quad b = \begin{bmatrix}0\end{bmatrix} \tag{10}$$

Stosując powyższe przekształcenia, możemy wykorzystać solver *cvxopt* do znalezienia optymalnych wag λ

3 Eksperymenty numeryczne

3.1 Porównanie funkcji jądrowych

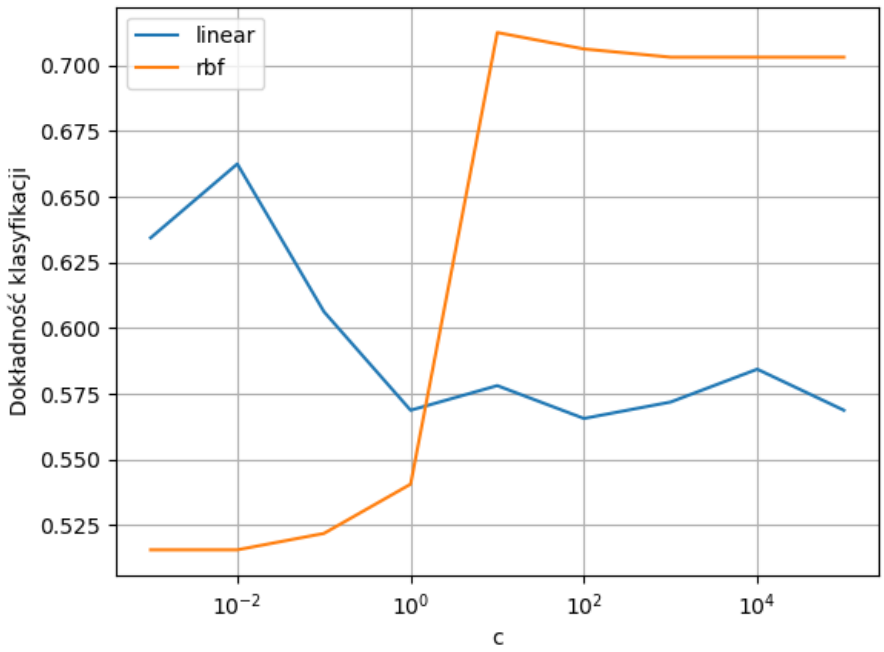
Posłużę się dwoma funkcjami jądrowymi poznanymi na wykładzie: funkcją liniową oraz RBF:

$$k(u, v) = u^T v \tag{11}$$

$$k(u, v) = \exp(\frac{-\|u - v\|^2}{2\sigma^2}) \tag{12}$$

W celu porównania wykorzystam poniższe wartości c i σ :
c = [0.001, 0.01, 0.1, 1.0, 10.0, 100.0, 1000.0, 10000.0, 100000.0]
$\sigma = 0.1$

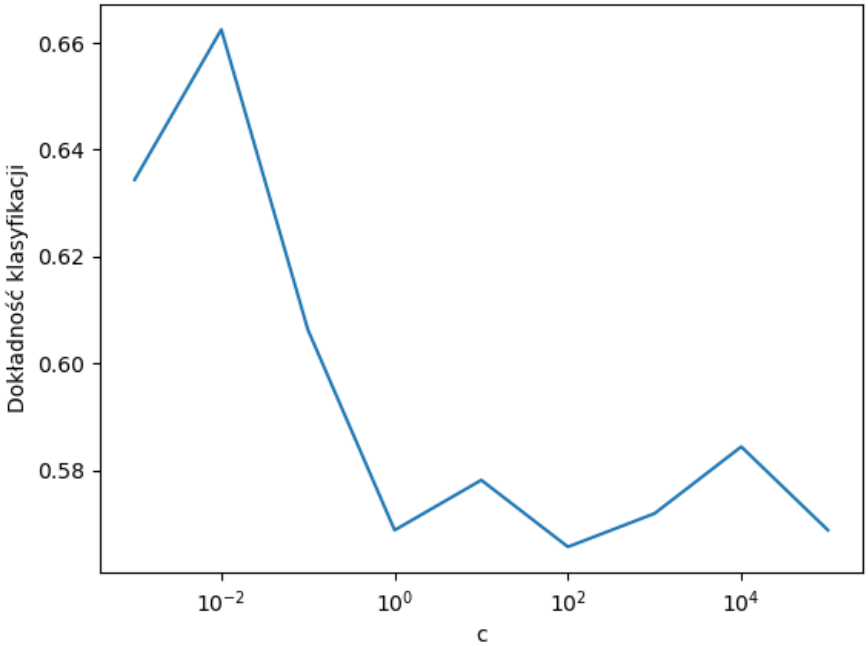
Rysunek 1: Porównanie dokładności przy użyciu różnych funkcji jądrowych



3.2 Wpływ parametrów na dokładność

Najpierw sprawdzę wpływ parametru c na dokładność dla funkcji jądrowej liniowej

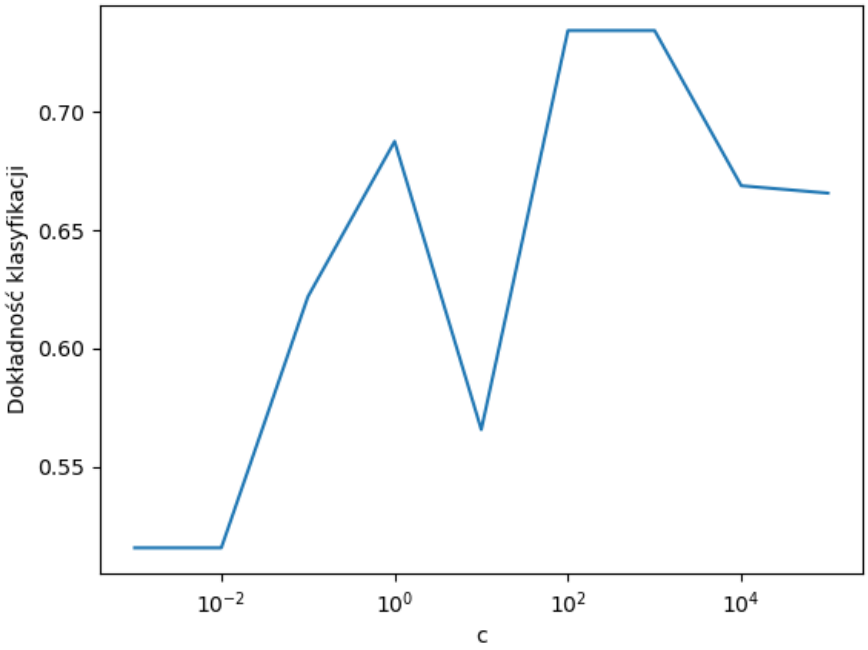
Rysunek 2: Porównanie wpływu parametru c na dokładność przy użyciu funkcji jądrowej liniowej



Algorytm uzyskuje najlepszą dokładność dla c = 0.01

Następnie sprawdzam wpływ parametru c na dokładność dla funkcji jądrowej RBF ze stałym $\sigma = 0.01$

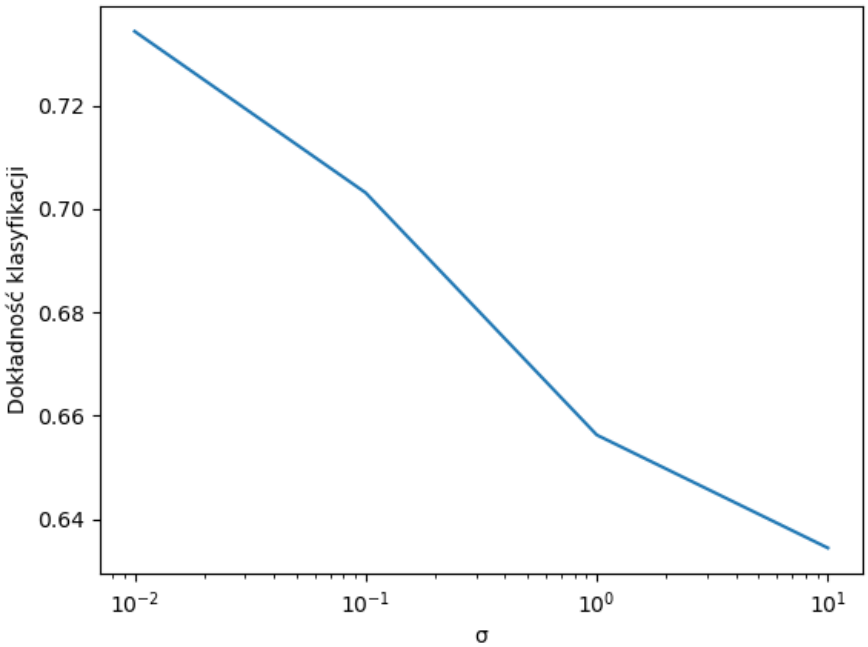
Rysunek 3: Porównanie wpływu parametru c na dokładność przy użyciu funkcji jądrowej RBF



Algorytm uzyskuje najlepszą dokładność dla c pomiędzy 100 a 1000

Na koniec sprawdzę wpływ parametru σ na dokładność dla funkcji jądrowej RBF ze stałym c=1000

Rysunek 4: Porównanie wpływu parametru σ na dokładność przy użyciu funkcji jądrowej RBF



Algorytm uzyskuje najlepszą dokładność dla $\sigma = 0.01$

4 Wnioski z przeprowadzonych badań

4.1 Porównanie funkcji jądrowych

Jak można zauważyć, od pewnego parametru c jądro RBF oferuje znacząco lepszą klasyfikację od jądra liniowego.

Tak przedstawia się tablica pomyłek dla najlepszego wyniku RBF:

$$\begin{vmatrix} 103 & 52 \\ 43 & 122 \end{vmatrix}$$

A tak przedstawia się tablica pomyłek dla najlepszego wyniku jądra liniowego:

$$\begin{vmatrix} 51 & 104 \\ 13 & 152 \end{vmatrix}$$

4.2 Wpływ parametrów na dokładność

Można zauważyć, że jądro liniowe dla osiągnięcia najwyższej dokładności preferuje niskie wartości c, w okolicach 0.01. Jego dokładność spada wraz ze wzrostem parametru c.

Dokładność jądra RBF wzrasta wraz ze wzrostem parametru c aż do okolicy 100, 1000 gdzie osiąga swoją maksymalną wartość.

To samo jądro RBF osiąga największą dokładność przy stosowaniu niskich wartośći σ, w okolicach 0.01.

Niska dokładność algorytmu w nawet najlepszym przypadku może być wytłumaczona wykorzystaniem wszystkich atrybutów win, gdzie niektóre z nich są wysoce skorelowane, co zaburza działania klasyfikatora.