

# Sprawozdanie z Ćwiczenia 4 z WSI

Adam Szostek nr. indeksu 331443

1 stycznia 2025

## 1 Opis implementowanego algorytmu

Support Vector Machine (SVM) jest algorytmem do klasyfikacji, który polega na znalezieniu hiperpłaszczyzny, która najlepiej oddziela dane z dwóch klas. Celem SVM jest maksymalizacja marginesu — odległości pomiędzy hiperpłaszczyzną a najbliższymi punktami z każdej klasy, które nazywane są wektorami nośnymi.

### 1.1 Zastosowanie problemu dualnego

Aby rozwiązać problem optymalizacji SVM i móc wykorzystać trik z przekształceniami jądrowymi formułujemy się tzw. postać dualną problemu optymalizacji algorytmu. Wychodząc od pierwotnego problemu maksymalizacji marginesu:

$$\min_{\mathbf{w}, b, \xi} \frac{1}{2} \|\mathbf{w}\|^2$$

przy ograniczeniach:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \dots, N.$$

Aby uzyskać postać dualną, wprowadzamy funkcję Lagrange’a:

$$L(\mathbf{w}, b, \xi, \alpha, \beta) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i (y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i) - \sum_{i=1}^N \beta_i \xi_i$$

gdzie  $\alpha_i \geq 0$  i  $\beta_i \geq 0$  są mnożnikami Lagrange’a. Minimalizujemy ją względem  $\mathbf{w}$ ,  $b$  oraz  $\xi$ , aby otrzymać postać do maksymalizacji.

Po tych działaniach uzyskujemy dualną funkcję celu:

$$\max_{\alpha} \left( \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \right)$$

przy ograniczeniach:

$$0 \leq \alpha_i \leq C, \quad \sum_{i=1}^N \alpha_i y_i = 0.$$

Gdzie  $C$  to parametr regularyzacji.

## 1.2 Rozwiązywanie problemu dualnego

Na rzecz implementacji w python użyto solwera cvxopt, który rozwiązuje zadanie optymalizacyjne metodą "interior point" w postaci:

$$\min_x \quad \frac{1}{2}x^T Px + q^T x$$

przy warunkach:

$$Gx \leq h, \quad Ax = b.$$

Dualną funkcję celu przedstawiamy w postaci minimalizacji:

$$\min_{\alpha} \quad \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^N \alpha_i$$

przy warunkach:

$$\begin{aligned} \sum_{i=1}^N \alpha_i y_i &= 0, \\ 0 \leq \alpha_i &\leq C \quad \forall i = 1, \dots, N. \end{aligned}$$

Tak więc po sprowadzeniu jej do postaci odpowiedniej dla wybranego solwera dostajemy:

$$P_{ij} = y_i y_j K(x_i, x_j), \quad q_i = -1, \quad G = \begin{bmatrix} -I \\ I \end{bmatrix}, \quad h = \begin{bmatrix} 0 \\ C\mathbf{1} \end{bmatrix}, \quad A = [y_1 \quad y_2 \quad \dots \quad y_N], \quad b = 0,$$

które oznaczają:

- $P$ : Macierz Gramowa (kernel matrix), reprezentująca wewnętrzny iloczyn wektorów w przestrzeni cech, uwzględniając etykiety klas.
- $q$ : Wektor współczynników w funkcji celu, będący wektorem ujemnych jedynek, ponieważ maksymalizujemy funkcję celu.
- $G$ : Macierz nierównościowych ograniczeń, która zapewnia, że  $\alpha_i \geq 0$  oraz  $\alpha_i \leq C$  dla wszystkich próbek.
- $h$ : Wektor wartości dla ograniczeń w macierzy  $G$ , odpowiadający za  $\alpha_i \geq 0$  (0) i  $\alpha_i \leq C$  (C).
- $A$ : Macierz równościowego ograniczenia, zawierająca etykiety klas  $y_i$ , zapewniająca, że suma ważonych etykiet jest równa 0.
- $b$ : Wektor równościowych ograniczeń, zawierający wartość 0, co odpowiada warunkowi  $\sum_i \alpha_i y_i = 0$ .

Solver zwraca rozwiązanie  $\alpha$ , które minimalizuje daną funkcję celu przy spełnieniu powyższych ograniczeń.

Po rozwiązaniu problemu dualnego, uzyskujemy wartości zmiennych Lagrange'a  $\alpha_i$ . Indeksy, dla których  $\alpha_i$  są różne od zera (przyjęta tolerancja to  $10^{-5}$ ), odpowiadają wektorom nośnym.

### 1.3 Obliczanie przesunięcia $b$ , funkcji decyzyjnej i predykcji

Przesunięcie  $b$  obliczane jest jako średnia różnic pomiędzy etykietami wektorów nośnych a ich produktami skalarnymi z innymi wektorami nośnymi:

$$b = \frac{1}{|\text{support vectors}|} \sum_i \left( y_i - \sum_j \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j) \right)$$

Funkcja decyzyjna przyjmuje postać:

$$f(\mathbf{x}) = \sum_i \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b$$

Na podstawie funkcji decyzyjnej klasyfikujemy nowy punkt  $\mathbf{x}$  za pomocą:

$$\text{predykcja} = \text{sign}(f(\mathbf{x}))$$

### 1.4 Zaimplementowane jądra

Jądro RBF (Radial Basis Function) jest określane wzorem:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp \left( -\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2} \right)$$

gdzie  $\|\mathbf{x}_i - \mathbf{x}_j\|^2$  to kwadrat odległości euklidesowej między wektorami cech  $\mathbf{x}_i$  i  $\mathbf{x}_j$ , a  $\sigma$  jest parametrem jądra, który kontroluje szerokość funkcji.

Jądro liniowe określane jest wzorem:

$$K(x_i, x_j) = x_i^\top x_j$$

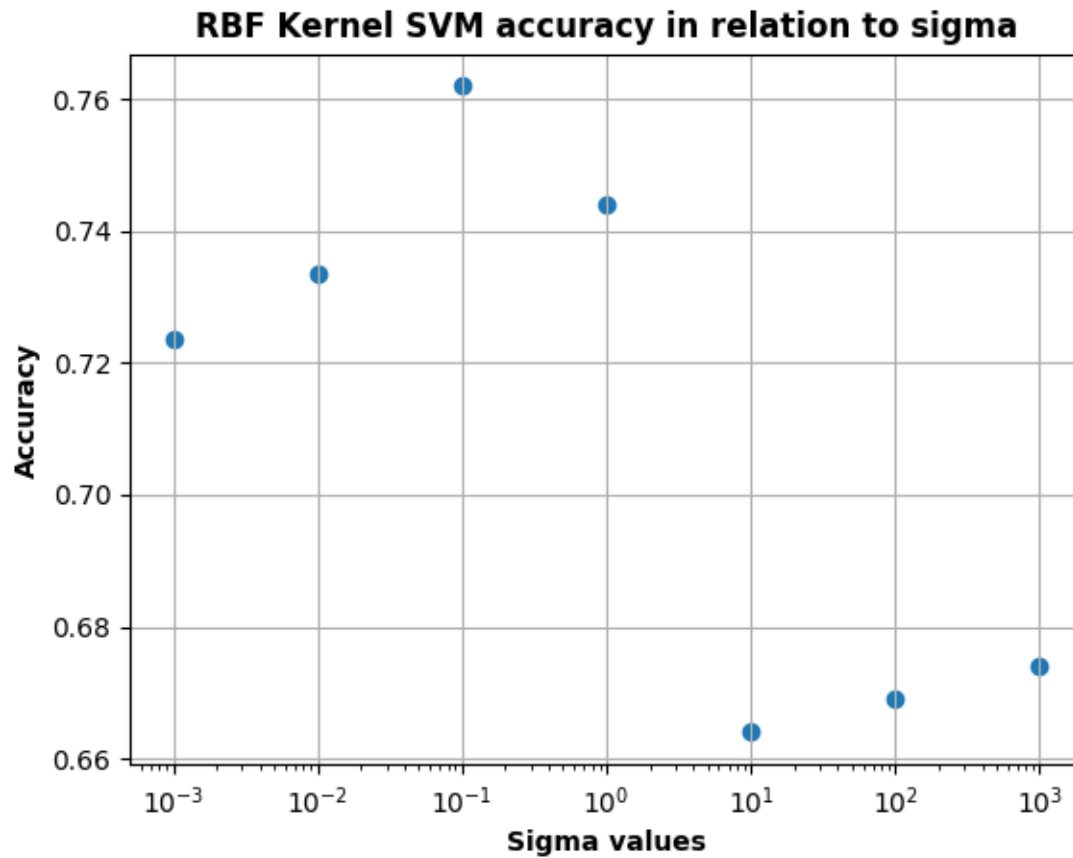
## 2 Planowane eksperymenty numeryczne

W celu przetestowania implementacji algorytmu SVM użyto zbioru danych "Wine quality". Aby przystosować zbiór danych do problemu klasyfikacji binarnej zdyskretyzowano zmienną decyzyjną quality tak, że dla każdego  $y$ :  $f(y) = \begin{cases} -1 & \text{dla } y \in [1, 5], \\ 1 & \text{dla } y \in [6, 10]. \end{cases}$

Wyniki badań uśredniono z pięciu prób trenowania modelu, przy czym każda próba korzystała z losowo wybieranej próbki 2000 par uczących. Zastosowano podział na zbiór uczący i testujący przy pomocy funkcji "train test split" z biblioteki sklearn. 80% danych stanowi część trenującą, a 20% testującą. Przeprowadzono również badanie w poszukiwaniu najlepszej wartości parametru sigma dla jądra RBF, a następnie w kolejnym eksperymencie dotyczącym parametru C, używano tej właśnie wartości.

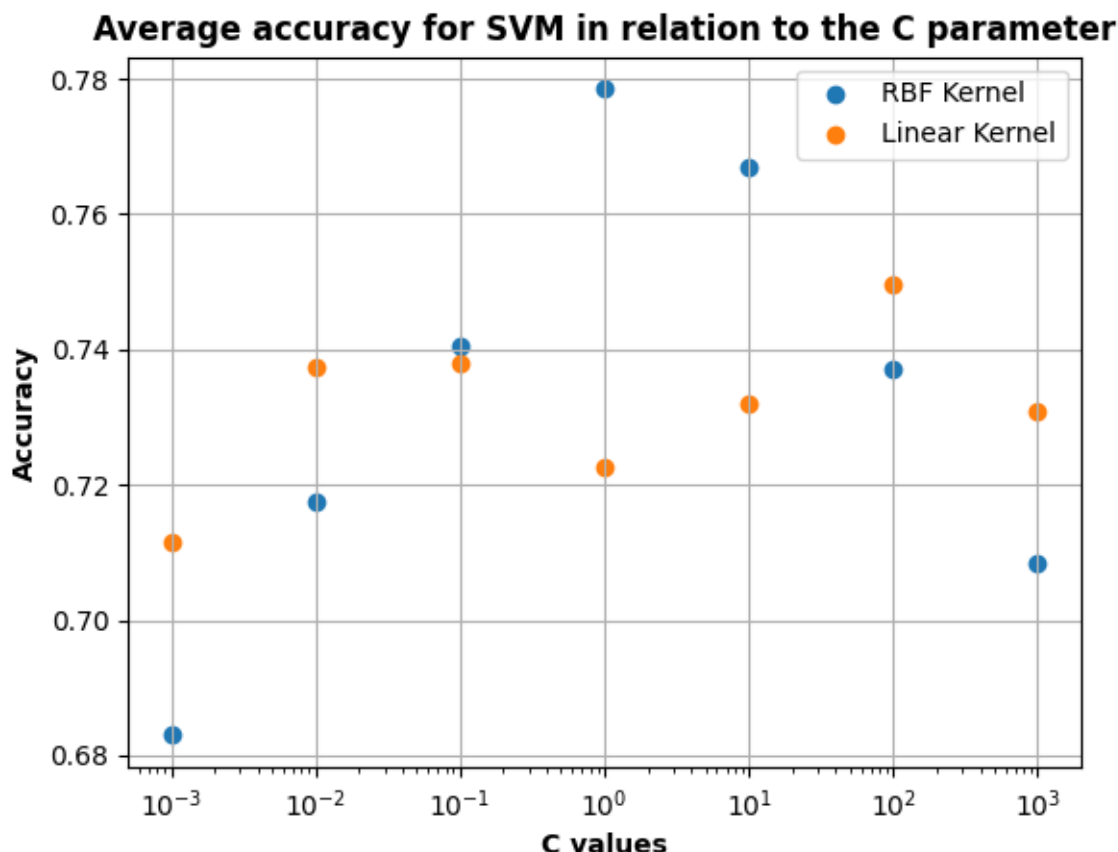
### 3 Uzyskane wyniki

#### 3.1 Parametr $\sigma$ w jądrze RBF



Analizując wpływ wartości parametru  $\sigma$  na dokładność SVM z jądrem RBF można zauważyć, że optymalny zakres to  $([10^{-3}, 10^0])$ , a najwyższe wyniki osiągają wartości  $(10^{-1}, 10^0)$ . Jednocześnie łatwo zauważyć, że od  $10^1$  algorytm osiąga znacząco słabszą dokładność. W dalszych badaniach dla jądra RBF użyto  $\sigma = 10^{-1}$ .

### 3.2 Badanie parametru C



Przy analizie dokładności SVM dla jąder RBF i liniowego widzimy, że jądro liniowe zachowuje się dużo stabilniej podczas zmian  $C$  od RBF. Jednakże ostatecznie najwyższą wartość dokładności osiąga RBF przy  $C = 10^0$ . Dla  $C = 10^1$  wynik jest zbliżony, jednak pozostałe wartości SVM z jądrem RBF są znacząco gorsze.

## 4 Wnioski z przeprowadzonych eksperymentów

Na podstawie obserwacji z badań można wyciągnąć następujące wnioski. Dobór parametru  $\sigma$  jest kluczowy w przypadku jądra RBF, a optymalna wartość może różnić się w zależności od zbioru danych. Należy jednak ograniczyć się do wyboru  $\sigma$  mniejszych od  $10^0$ , ponieważ dla większych skuteczność SVM z jądrem RBF znacznie spada.

Dla obu jąder parametr  $C$  ma znaczący wpływ na osiąganą dokładność. W przypadku jądra liniowego wyniki są dosyć stabilne, ale da się wyróżnić wartości lepsze (np.  $10^{-2}$ ,  $10^{-1}$ ,  $10^2$ ). Dla jądra RBF  $C$  mniejsze bądź równe  $10^{-2}$  i większe od  $10^3$  zapewnia gorsze wyniki co w przypadku pierwszego przedziału może być powodowane zbyt dużym uproszczaniem modelu, natomiast w drugim przedziale powodem może być nadmierne dostosowywanie się do zbioru uczącego przez model nazywane overfittingiem. Ostatecznie jądro RBF osiąga lepszą dokładność niż liniowe, ale jest znacznie bardziej wrażliwe na zmiany parametru  $C$ . Podsumowując, jądro RBF będzie lepszym wyborem dla bardziej złożonych problemów, ale wymaga dokładniejszego strojenia parametrów.