

# Dokumentacja zadania laboratoryjnego - badanie algorytmu SVM

Krzysztof Wyrzykowski, nr indeksu 331455

2 stycznia 2025

## 0.1 Opis działania algorytmu

Zadaniem algorytmu jest rozwiązanie problemu klasyfikacji za pomocą hiperpłaszczyzny separującej, danej wzorem:

$$\hat{f}(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} - b$$

Przynależność do danej klasy określamy na podstawie znaku wartości funkcji.

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

Znaleziona funkcja ma spełniać dwa warunki:

1. Poprawna separacja klas.
2. Maksymalnie szeroki obszar separujący.

Szerokość obszaru wyraża się poprzez  $\frac{2}{\|\omega\|}$ . Zatem w celu uzyskanie maksymalnej szerokości należy minimalizować  $\|\omega\|$ . Parametr ten możemy przedstawić jako kombinację liniową elementów ze zbioru uczącego.

$$w = \sum_{i=1}^N \alpha_i y_i x_i$$

gdzie  $\alpha_i \in R$ . Te  $x_i$  dla których  $\alpha_i = 0$  nazywamy wektorami nośnymi. Algorytm dopuszcza pomyłki  $\xi_i$ , dlatego wprowadzamy funkcję kary

$$\xi_i = \max(1 - f(x_i)y_i, 0)$$

Zatem funkcja celu w tym zadaniu optymalizacji ma postać:

$$(w, b) = \arg \min_{w, b, \xi} \left( \sum_{i=1}^N \max(1 - f(x_i)y_i, 0) + \lambda \|w\|^2 \right)$$

Separacja danych za pomocą hiperpłaszczyzny nie jest zawsze możliwa. W celu rozwiązania tego problemu konieczne jest wykonanie przekształceń rzutujących wektory wejściowe do wyższych przestrzeni. Aby rzutować wektory należy skorzystać z przekształceń takich jak na przykład:

1. Liniowe:  $k(u, v) = u^T v$
2. Gaussowskie/RBF:  $k(u, v) = \exp\left(-\frac{\|u-v\|^2}{2\sigma^2}\right)$

Uwzględniając przekształcenia zadanie minimalizacji ma następującą postać

$$(\alpha_1, \dots, \alpha_N, b) = \arg \min_{\alpha, b} \sum_{i=1}^N \max(1 - f(x_i)y_i, 0) + \lambda \|w\|^2, \quad f(x) = \sum_{i=1}^N \alpha_i y_i k(x, x_i) - b$$

Korzystając z dualnego zadania optymalizacji i mnożników Lagrange'a możemy uzyskać nieco odmienną postać tego zadania, która jest łatwiejsza w implementacji.

$$(\alpha_1, \dots, \alpha_N, b) = \arg \min_{\alpha_1, \dots, \alpha_N} \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(x_i, x_j) - \sum_i \alpha_i$$

Z tej postaci łatwo możemy wyliczyć gradient funkcji i celu, aby wykorzystać go w metodzie gradientu prostego, w celu znalezienia optymalnych wartości parametrów  $\alpha_i$ .

$$\sum_{j=1}^N \alpha_j y_j y_i k(x_i, x_j) - 1$$

przy ograniczeniach:  $\alpha_i \in [0, C]$  dla  $i = 1, \dots, N$ . Im większa wartość współczynnika  $C$  tym większy wpływ wektorów nośnych na kształt hiperpłaszczyzny i mniejsza odporność na przeucenie.

## 0.2 Opis planowanych eksperymentów numerycznych

### 0.2.1 Cel przeprowadzenia eksperymentów

Celem eksperymentów było zbadanie wpływu wartości współczynnika  $C$  na jakość uzyskanej funkcji klasyfikującej. Jakość funkcji mierzona jest w stosunku liczby poprawnych przewidywań do rozmiaru próby testowej. Następnie zmierzyłem wpływ parametru  $\sigma$  przekształcenia RBF na jakość wyników.

### 0.2.2 Parametry eksperymentów

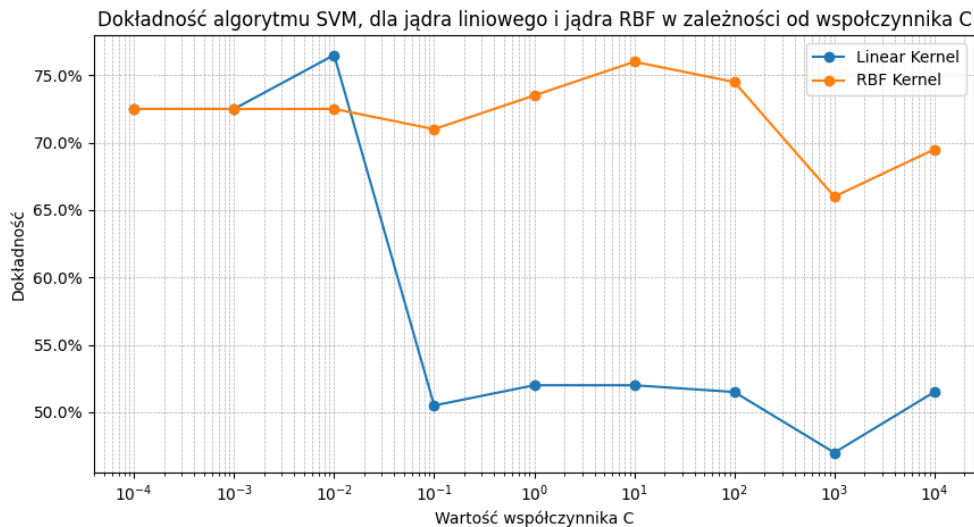
Badania zostały przeprowadzone na zbiorze danych "Wine Quality Data Set", dostępnym po adresie <https://archive.ics.uci.edu/dataset/186/wine+quality>. Aby badanie dało się przeprowadzić w rozsądnym czasie ograniczyłem zbiór danych do 200 początkowych wartości. Następnie podzieliłem wybrane dane na zbiór trenujący i testowy w stosunku 4:1. Dla każdej testowanej wartości, wykonałem test 5-krotnie. Dla każdego test podział na zbiór treningowy i testowy został wykonany ponownie.

## 0.3 Parametr C oraz porównanie przekształceń

### 0.3.1 Przebieg badań

Dla obu przekształceń sprawdziłem skuteczność dla wartości parametru C równych  $10^k$ ,  $k \in [-4, 4] \cap \mathbb{Z}$ . Przyjąłem średnią wartość parametru  $\sigma$  równą 1.

### 0.3.2 Wyniki badań



Z wykresów wynika, że dla jądra RBF algorytm uzyskał najlepsze wyniki dla wartości współczynnika C równej 10. Natomiast wyniki dla wartości mniejszych nie były znacząco gorsze. Widać jednak spadek skuteczności dla wartości większych. Dla jądra liniowego dobór parametru C miał większe znaczenie. Najlepsze działanie osiągał algorytm dla wartości 0.01 i mniejszych. Dla większych widać wyraźny spadek.

## Wnioski

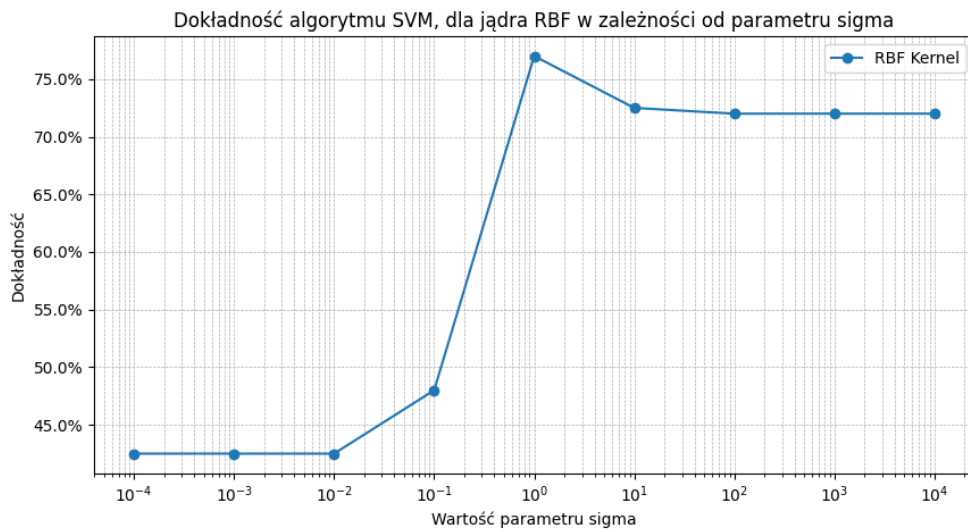
Dla jądra RBF wybór odpowiedniej wartości współczynnika C może podnieść skuteczność algorytmu w granicach 10 punktów procentowych, aczkolwiek algorytm w tej konfiguracji nie jest podatny na zmiany tego współczynnika. Dla jądra liniowego dobór odpowiedniej wartości współczynnika C jest kluczowy i decyduje o użyteczności otrzymanych wyników. Oba przekształcenia w odpowiedniej konfiguracji uzyskiwały podobne wyniki. Dla tych danych przekształcenie liniowe było w stanie uzyskać nawet lepszy wynik, lecz wymagało dokładniejszego strojenia.

## 0.4 Parametr $\sigma$

### 0.4.1 Przebieg badań

Dla wartości  $C=10$ , która dawała najlepsze rezultaty w pierwszym etapie przetestowałem wpływ parametru  $\sigma$ , który steruje zasięgiem przekształcenia.

#### 0.4.2 Wyniki badań



Z wykresu wynika, że najlepsze działanie osiąga algorytm dla parametru  $\sigma = 1$ . Dla większych wartości działanie jest gorsze o około 5 punktów procentowych. Natomiast dla wartości mniejszych działanie jest zauważalnie gorsze.

#### Wnioski

Zbyt mała wartość  $\sigma$  może powodować przeuczenie, które znacząco obniża skuteczność algorytmu. Dobranie zbyt dużej wartości również pogarsza wyniki, jednakże w badanym zakresie wartości nie miało to znaczącego wpływu na otrzymane wyniki.