

Wstęp do sztucznej inteligencji

Regresja i klasyfikacja

Jakub Robaczewski

Algorytm klasyfikacji:

Zaimplementowany przeze mnie algorytm klasyfikacji wykorzystuje liniowy SVM dopuszczający pomyłki. Stworzone przeze mnie funkcje możemy podzielić na 4 grupy: zarządzanie danymi, główny algorytm, funkcje matematyczne oraz funkcję testującą.

Zarządzanie danymi:

Funkcje zarządzania danymi służą do pobrania danych z podanego pliku oraz dalszej obróbki

- `read_from_file()` – pobiera dane z podanego pliku
- `name_to_numeric()` – zamienia nazwy zbiorów na wartości liczbowe
- `split_data()` – dzieli podany zbiór na 3, wykorzystując podane proporcje. Jeżeli proporcje nie sumują się do 1, funkcja zwraca błąd.
- `group()` – dzieli zbiór na podzbiory według przynależności do klas
- `make_set()` – porównuje klasę obiektu do podanej wartości. Zwraca 1, jeśli jest taka sama, -1, jeśli inna.

Główny algorytm:

Funkcje głównego algorytmu przeprowadzają najważniejsze operacje wykorzystując bibliotekę `scipy` do minimalizacji oraz `functools` do przekazywania danych.

- `get_lambda()` – dobiera λ wykonując algorytm na podanym zbiorze (trenującym i walidacyjnym)
- `make_test()` – wykonuje testy na zbiorze trenującym: dzieli go na 3 podgrupy, a następnie tworzy 3 funkcje porównujące każdą klasę z każdą. Po czym sprawdza wyniki na zbiorze testowym.
- `SVM()` – wykonuje minimalizację funkcji kosztu przy danym zbiorze i danej λ

Funkcje matematyczne:

- `Cost()` – funkcja kosztu

$$\min \left(\|\omega\|^2 * \lambda + \sum_i \xi_i \right)$$

- `Ksi()` – funkcja ξ_i

$$\xi_i = \max(1 - f(x_i)y_i, 0)$$
$$y_i = f(x_i) = \omega^T x - b$$

- `decision_function()` – funkcja decyzyjna f

$$f(x) = \begin{cases} -1; & x \leq 0 \\ 1; & x > 0 \end{cases}$$

Funkcja testująca:

Funkcja testująca przyjmuje 3 funkcje rozdzielające klasy i na podstawie ich wyników dopasowuje klasę obiektu.

Wyniki:

Test 0 Lambda 0.5 Iris-setosa: 100.00% Iris-versicolor: 83.87% Iris-virginica: 100.00% Total accuracy: 94.44%	Test 1 Lambda 0.001 Iris-setosa: 100.00% Iris-versicolor: 90.91% Iris-virginica: 91.43% Total accuracy: 93.33%	Test 2 Lambda 1 Iris-setosa: 100.00% Iris-versicolor: 96.55% Iris-virginica: 100.00% Total accuracy: 98.89%	Test 3 Lambda 0.001 Iris-setosa: 100.00% Iris-versicolor: 93.33% Iris-virginica: 100.00% Total accuracy: 97.78%	Test 4 Lambda 0.001 Iris-setosa: 100.00% Iris-versicolor: 92.31% Iris-virginica: 91.43% Total accuracy: 94.44%
Test 5 Lambda 0.001 Iris-setosa: 100.00% Iris-versicolor: 93.33% Iris-virginica: 100.00% Total accuracy: 97.78%	Test 6 Lambda 1 Iris-setosa: 100.00% Iris-versicolor: 96.88% Iris-virginica: 83.33% Total accuracy: 93.33%	Test 7 Lambda 0.001 Iris-setosa: 100.00% Iris-versicolor: 90.32% Iris-virginica: 100.00% Total accuracy: 96.67%	Test 8 Lambda 0.001 Iris-setosa: 100.00% Iris-versicolor: 89.29% Iris-virginica: 90.62% Total accuracy: 93.33%	Test 9 Lambda 0.001 Iris-setosa: 100.00% Iris-versicolor: 93.33% Iris-virginica: 93.55% Total accuracy: 95.56%

Jak zauważamy na powyższych testach, algorytm osiąga niemal 95% procentową skuteczność, co jest bardzo dobrym wynikiem, dodatkowo dla klasy Iris-setosa, algorytm nigdy się nie myli, a największe problemy występują z rozróżnieniem Iris-versicolor od Iris-virginica.

Wynika to z faktu, iż Iris-setosa posiada płatki o maksymalnej szerokości 0.6 cm i długości maksymalnie 2 cm, a płatki pozostałych klas posiadają znacznie większe rozmiary, dlatego dane te są liniowo separowane i oddzielenie ich od reszty nie stanowi dla algorytmu większego problemu.

