Zadanie 4

Klasyfikator SVM oraz Drzewa Decyzyjne

Kacper Kania

9.04.2024

Treść zadania

W tym zadaniu zbadacie Państwo własności klasyfikatora SVM (z ang. *Support Vector Machine*) oraz drzewa decyzyjnego.

SVM. Umiejętność posługiwania się nim jest przydatne nawet w dzisiejszych problemach, gdzie dostęp do danych jest ograniczony. SVM służy też w zadaniach analizy jakości ekstrakcji cech w modelach językowych (*linear probing*). Do głównych zalet klasyfikatora należą:

- szybkość działania,
- wydajne implementacje na CPU,
- rozbudowana infrastruktura w bibliotekach takich jak `scikit-learn` oraz `libsvm`.

Drzewa decyzyjne. Same pojedyncze drzewa decyzyjne są rzadko stosowane, ale za to są podstawowym elementem budującym algorytmy jak XGBoost czy CatBoost. Ich zastosowanie jest szczególnie przydatne w sytuacjach, kiedy jest konieczna wyjaśnialność modelu (czyli która cecha wpłynęła na decyzję) i kiedy mamy do czynienia z danymi o różnej domenie (liczby całkowite, rzeczywiste itd.).

Wykorzystacie Państwo te klasyfikatory do zadania klasyfikacji irysów[1] na podstawie ich pomiarów. Zbiór można pobrać stąd: link.

Państwa zadaniem jest:

- Implementacja ładowania zbioru danych oraz treningu i ewaluacji przy wykorzystaniu walidacji krzyżowej (proszę założyć 5 podziałów zbioru danych).
- Dla SVM—zbadanie wpływu następujących parametrów: siły regularyzacji, funkcji jądra oraz liczby iteracji (w tym celu proszę ustawić parametr `tol` na bardzo niską wartość, rzędu 10^{-16}) dla algorytmu SVM. Przy badaniu funkcji jądra i siły regularyzacji można ustawić automatyczne wyznaczanie liczby iteracji.

[1] Jest to właściwie standardowe zadanie na początku w uczeniu maszynowym, obok klasyfikacji obrazów MNIST.

- Dla drzewa—zbadanie wpływu parametrów: kryterium oceny, technika podziału węzła, maksymalna głębokość drzewa dla drzewa decyzyjna.
- W celu zbadania jakości klasyfikatorów, wykorzystajcie miary jakości klasyfikacji: *accuracy*, *precision*, *recall* oraz *F1*. Proszę zapoznać się z ich definicjami, jednak nie będę wymagał dokładnej ich interpretacji.

W tym zadaniu proszę korzystać z biblioteki `scikit-learn`[2].

Proszę zauważyć, że każda z tych klas ma swój `random_state`. Jest to atrybut odpowiadający za inicjalizację generatora liczb losowych w tych klasach. W celu przeprowadzenia eksperymentów, proszę ustawić go na 3 różne wartości (3 różne eksperymenty). Wyniki w tabelach proszę podać jako średnią z odchyleniem standardowym. Biorąc pod uwagę 5 foldów z walidacji krzyżowej, daje to łącznie 15 wyników dla każdego eksperymentu. Proszę podawać wyniki jako średnią i odchylenie standardowe.

W sprawozdaniu

Należy:

- Przedstawić wyniki eksperymentów w formie tabelarycznej. Zestawić ze sobą wyników porównujące oba algorytmy.
- Przedstawić wykresy zależności wartości parametrów o charakterze ciągłym (takich jak siła regularyzacji) i jakości klasyfikacji (dla wybranej miary).
- Napisać wnioski.

Zalecam zautomatyzować proces generowanie tabel i wykresów (najwygodniej się to robi przy wykorzystaniu Jupyter Notebook). W tym celu można wykorzystać bibliotekę `pandas`, w których tabelę `DataFrame` można zapisać do pliku `.csv` lub wygenerować wykres względem określonego zapytania. Dla piszących w LaTeX, klasa też ma funkcję `to_latex(...)`.

Jeśli okaże się, że eksperymenty uruchomiają się zbyt długo, można ograniczyć liczbę foldów.

[2]Linki do: drzewa decyzyjnego, klasyfikatora SVM.