ZUM 24L

Projekt - założenia wstępne

Amelia Tabor, Wojciech Gierulski

Tworzenie zespołu modeli klasyfikacji przez wielokrotne stosowanie dowolnego algorytmu klasyfikacji do bootstrapowych prób ze zbioru trenującego z losowo wybranym podzbiorem atrybutów. Porównanie z wybranymi algorytmami klasyfikacji tworzącymi pojedyncze modele oraz implementacjami algorytmów bagging i random forest dostępnymi w środowisku R lub Python.

1. Interpretacja tematu

Bagging [1] jest metodą polegającą na generowaniu wielu wersji tego samego modelu i wykorzystywaniu ich do uzyskania zespołu modeli. W przypadku zadania klasyfikacji agregacja przeprowadzana jest za pomocą głosowania. Każda wersja modelu posiada swój własny zbiór treningowy - próbę bootstrapową wygenerowaną przez losowanie ze zwracaniem ze zbioru treningowego próby o liczności takiej jak liczność zbioru treningowego.

Random Subspaces [2] to metoda polegająca na tworzeniu zespołu modeli, gdzie dla każdego modelu wybierany jest losowo tylko pewien podzbiór wszystkich atrybutów.

W projekcie wykorzystujemy połączeniu obydwu metod.

2. Opis implementacji

Algorytm opisany jest następującymi krokami:

1. Wylosować ze zwracaniem *D* próbek ze zbioru treningowego, gdzie *D* to rozmiar zbioru treningowego
2. Wylosować *n* atrybutów ze zbioru wszystkich atrybutów.
3. Dopasować klasyfikator do wylosowanych danych treningowych, uwzględniając tylko wylosowane atrybuty.
4. Powtarzać kroki 1-3 *k* razy, do momentu stworzenia zespołu modeli klasyfikacji.

W fazie predykcji klasa jest wybierana na podstawie głosowania zespołu modeli.

Jako model klasyfikacji zostanie wykorzystane drzewo decyzyjne bazujące na CART. Zgodnie z publikacją [1], metoda bagging daje najlepsze rezultaty w przypadku algorytmów niestabilnych takich jak m.in. drzewa decyzyjne.

3. Plan badań

a. Cel eksperymentów

- Określenie ile modeli powinno być w zespole, żeby obserwować wzrost jakości predykcji względem pojedynczych modeli
- Określenie do jakiego momentu opłaca się zwiększać liczbę modeli w zespole
- Określenie optymalnej liczby losowanych atrybutów
- Porównanie testowanej metody z gotowymi implementacjami pojedynczych modeli, Random Forest i Bagging.

b. Zbiory danych

1. Wine quality dataset
https://archive.ics.uci.edu/dataset/186/wine+quality

typ atrybutów	liczba atrybutów	liczba klas	liczba próbek
numeryczne i dyskretne	4898	6	4 898

2. Adult dataset
https://archive.ics.uci.edu/dataset/2/adult

typ atrybutów	liczba atrybutów	liczba klas	liczba próbek
numeryczne i dyskretne	14	2	48 842

W ramach wstępnego przetwarzania danych usunięte zostaną próbki z brakującymi wartościami atrybutów. Atrybuty dyskretne zostaną zakodowane metodą one-hot.

c. **Parametry algorytmów, których wpływ będzie badany**

- liczba modeli w zespole klasyfikacji
- liczba losowanych atrybutów

d. **Miary jakości i procedury oceny**

Do ewaluacji modeli zostanie wykorzystany out-of-bag error - predykcja uzyskiwana przez głosowanie modeli, które nie miały danego przykładu w bootstrapowej próbie treningowej

Miara oceny jakości to krzywa PR oraz pole pod krzywą. W przypadku klasyfikacji wieloklasowej - mikro i makro uśrednianie w podejściu one vs rest. Prawdopodobieństwa przynależności do klas potrzebne do wykreślenia krzywej będą szacowane na podstawie stosunków głosów zespołu modeli.

4. Lista algorytmów

- drzewo decyzyjne - *sklearn.tree.DecisionTreeClassifier*
- procedura Bagging i losowanie podzbiorów atrybutów - implementacja własna
- Pojedyncze modele klasyfikacji do porównania wyników:
  - *sklearn.linear_model.LogisticRegression*
  - *sklearn.svm.SVC*
- Modele bagging do porównania wyników - *sklearn.ensemble.BaggingClassifier*
- Model Random Forest do porównania wyników - *sklearn.ensemble.RandomForestClassifier*

5. Literatura

[1] Breiman, L. Bagging predictors. Mach Learn 24, 123–140 (1996). https://doi.org/10.1007/BF00058655

[2] Tin Kam Ho, "The random subspace method for constructing decision forests," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 8, pp. 832-844, Aug. 1998, doi: 10.1109/34.709601.