MED Projekt Kryszkiewicz 17Z

Artur M. Brodzki, Mateusz Orzoł

28 grudnia 2017

1 Ogólne zasady

Program ma działać w trybie wsadowym, z linii komend i mieć możliwość załadowania każdego z trzech zbiorów danych z repozytorium UCI:

- Adult: `http://archive.ics.uci.edu/ml/datasets/Adult`
- Flags: `http://archive.ics.uci.edu/ml/datasets/Flags`
- PrimaryTumor: `http://archive.ics.uci.edu/ml/datasets/Primary+Tumor`

Dla zadanego zbioru danych, program ma wypisać wagę każdego atrybutu w zbiorze - im większa waga, tym lepiej używać danego atrybutu do klasyfikacji.

Wszystkie te zbiory mają tylko po kilkanaście atrybutów, czyli nie za wiele jak na problem wyboru optymalnych atrybutów. Była prośba prof. Kryszkiewicz, żebyśmy znaleźli sobie sami jeszcze jakiś jeden dodatkowy zbiór który ma wiele atrybutów, ze 100 chociaż i też wsadzić go w ramach testów do naszego programu.

Program ma mieć opcję uruchamiania pozwalającą wybrać jeden z dwóch algorytmów działania:

- Algorytm Koronackiego
- Algorytm Boruty

Algorytmy opisuję w następnej sekcji.

Program ma się kompilować i uruchamiać na Windowsie więc chyba możemy go pisać w Visual Studio. Ogólnie rzecz biorąc ważna jest wydajność programu.

2 Algorytmy

Wybór optymalnego atrybutu w ogólności realizuje się poprzez zbudowanie lasu losowego zawierającego wiele drzew. Cały zbiór danych dzieli się na podzbiory i tworzy osobne drzewo dla każdego podzbioru.

Każde drzewo z osobna tworzone jest algorytmem SPRINT, opisanym szczegółowo na notatkach z wykładu o klasyfikacji, w wersji z wykorzystaniem współczynnika Giniego lub entropii (do wyboru).

Z kolei sposób wyznaczania najlepszych atrybutów realizowany jest inaczej w zależności od algorytmu.

2.1 Oznaczenia

Niech całkowita liczba rekordów w zbiorze danych wynosi T, a liczba atrybutów w tym zbiorze wynosi A.

2.2 Algorytm Koronackiego

1. Podział na podzbiory: decydujemy się na A drzew w lesie. Do każdego drzewa przydzielamy T/A rekordów wybieranych drogą losowania ze zwracaniem (czyli rekordy dla jednego drzewa mogą się powtarzać). Ponieważ rekordy się powtarzają, to niektóre nie zostaną wybrane do żadnego drzewa, za to użyjemy ich do testowania.

2. Każde drzewo uczymy z wykorzystaniem przydzielonych mu rekordów oraz testujemy za pomocą (tego samego dla każdego drzewa) zbioru rekordów testujących. Zapamiętujemy dokładność (ang. *accuracy*) klasyfikacji drzewa $i : \phi(i)$.

3. Na każdym z A drzew wybieramy jeden z A atrybutów. Wartości tego atrybutu na zbiorze rekordów uczących są losowo permutowane i na tak zmienionym zbiorze ponownie uczymy drzewo i wykonujemy testowanie. Zapisujemy nową (niższą, bo zbiór uczący został zepsuty) dokładność klasyfikacji $\phi'(i)$.

4. Różnica $w(i) = \phi(i) - \phi'(i)$ oznacza wagę danego atrybutu.

2.3 Algorytm Boruty

1. Podział na podzbiory: identycznie jak dla algorytmu Koronackiego.

2. Replikacja: wszystkie kolumny - atrybuty w danych sa replikowane. W nowym zbiorze danych mamy więc $2A$ atrybutów: każdy atrybut a_i ma swoją kopię a'_i.

3. Kopie atrybutów mają na samym wstępie losowo permutowane wartości.

4. Następnie obliczamy wagę każdego atrybutu $w(i) = \phi(i) - \phi'(i)$ (replikowanych i niereplikowanych) tak samo jak w algorytmie Koronackiego.

5. Jako wagę atrybutu i przyjmujemy różnicę pomiędzy wagą Koronackiego jego zwykłej wersji $w(i)$ a wagą jego zreplikowanej wersji $w'(i)$.

3 Podział zadań

Proponowany przez prof. Kryszkiewicz podział zadań pomiędzy osoby:

1. Pierwsza osoba implementuje algorytm SPRINT uczący pojedyncze drzewo na zadanych danych oraz metodę testowania klasyfikacji.
2. Druga osoba implementuje algorytmy Koronackiego i Boruty tworzenia lasu losowego.

Oprócz tego zostaje kwestia obsługi wejścia - wyjścia (parsowanie plików z danymi, obsługa linii poleceń).