Artur Dobrogowski
Arkadiusz Kalinowski
Alan Rashid

Opiekun projektu: dr inż. Paweł Cichosz

MOW

Projekt - eXtreme gradient boosting

Dokumentacja wstępna

1. Opis tematu projektu

Projekt obejmuje implementację algorytmu eXtreme gradient boosting w języku R. Wykorzystywany algorytm zostanie wsparty budową drzew wykonaną za pomocą pakietu rpart. Podczas prac wykorzystane zostaną trzy zbiory danych na których, w zależności od rodzaju zbioru, wykonana zostanie klasyfikacja oraz regresja zbiorów.

2. Opis algorytmu eXtreme gradient boosting

Wynikiem algorytmu jest model złożony z ważonej sumy modeli podstawowych. Model podstawowy może stanowić dowolny klasyfikator, ale w niniejszym projekcie będą to drzewa decyzyjne. Algorytm postępuje iteracyjnie, gdzie w każdej iteracji jest budowany kolejny model podstawowy którego zadaniem jest podać poprawki do dotychczasowego modelu. Przy uczeniu modelu podstawowego celem uczenia są residua z poprzedniej iteracji algorytmu - czyli ujemny gradient funkcji błędu po wyjściu dotychczasowego modelu. Waga kolejnego modelu jest dobierana tak aby zminimalizować błąd modelu złożonego.

Terminologia:

zadanie aproksymacji: $F(\overline{x}) = \overline{y}$, gdzie $\overline{x}$ i $\overline{y}$ to wektory wejściowe i wyjściowe

model podstawowy m-tej iteracji : $\overline{f_m} : R^{n_x} \rightarrow R^{n_y}$

waga m-tego modelu: $\gamma_m \in R$

model złożony, wynikowy m-tej iteracji: $\overline{F_m}(\overline{x}) = \sum \gamma_m \cdot \overline{f_m}(\overline{x}) \approx \overline{y}$

funkcja błędu: $E : R^{n_y} \rightarrow R$, np. $E(w) = 0.5 \| y - w \|^2$

Algorytm:

1. Inicjalizacja wartością stałą $\overline{F_0}(\overline{x}) = argmin\ \gamma : \Sigma_i E(\gamma)$ albo modelem wejściowym (np. pełnym drzewem)
2. Iteracyjnie:
   - 2.1. Reziduum poprzedniego modelu: $\overline{r_m} = -\left[\frac{\delta E(F_{m-1}(\overline{x}))}{\delta F_{m-1}(x)}\right]$
   - 2.2. Budowa kolejnego modelu podstawowego $\overline{f_m}$ używając $\overline{x}$ by przewidzieć $\overline{r_m}$

2.3. Wyznaczyć $\gamma_m = argmin\ \gamma : \sum E(\overline{F_{m-1}}(\overline{x}) + \gamma \overline{f_m}(\overline{x})$ jako czynnik minimalizujący błąd $\overline{F_m}$

3. Zwróć jako model ostatni z modeli $\overline{F}$

Zadanie klasyfikacji można sprowadzić do zadania regresji modelując prawdopodobieństwo przynależności do danej klasy.

Jako parametry algorytmu przyjmujemy:

- liczbę iteracji budowy modeli,
- stopień złożoności modeli podstawowych (głębokości drzew, metody wykorzystania pakietu rpart),

Za budowę drzew odpowiadać będzie funkcja:

```
rpart(formula, data, weights, subset, na.action = na.rpart,
  method,model = FALSE, x = FALSE, y = TRUE, parms, control,
                            cost, ...)
```

Najważniejszymi parametrami, z perspektywy wykonania eksperymentów, są `method` oraz `parms`. `method` odpowiada za wybranie metody budowania drzewa decyzyjnego, wykorzystany zostanie atrybut ‘anova’ (tworzone będą drzewa regresji) oraz ‘class’ (tworzone będą drzewa klasyfikacji). W przypadku drzew klasyfikacji atrybut `parms` przyjmuje, macierz prawdopodobieństwa a-priori, macierz błędu oraz indeks podziału.

3. Plan badań

Przedmiotem badań będzie analiza działania zaimplementowanego algorytmu na podstawie trzech różnych zbiorów danych.

Walidacja działania algorytmu oraz budowy drzew będziemy oceniać przy pomocy walidacji krzyżowej oraz gdy dane pozwolą - wydzielenie do 15% danych na zbiór walidacyjny.

Może się okazać, że walidacja krzyżowa będzie zbyt wolna przy dużej liczbie danych - wówczas posłużymy się tylko zbiorem walidacyjnym.

Zbiór danych		Oszustwa w płatnościach kartami kredytowymi.	Zachowania abonentów GSM	Zużycie energii
Liczba rekordów		285k	50k	20k
Liczba atrybutów	Dyskretnych\ Kategorycznych	-	40	-
	Ciągłych	30	190	29
Liczba możliwych klasyfikacji\regresji		2 klasy (klasyfikacji)	3 po 2 klasy	regresja

W przypadku danych zawierających oszustwa w płatnościach kartami kredytowymi należy przyjąć niezerową macierz błędu. Wynika to z faktu, że pomyłka działania algorytmu może mieć konkretne straty finansowe oraz z dysproporcji częstości wydarzenia oszustwa na tle normalnego użytkowania.

Każdy ze zbiorów danych będzie przez nas przeanalizowany w poszukiwaniu wartości odstających lub błędnych oraz ewentualnej selekcji części atrybutów.

Oceny jakości działania:

W projekcie przyjmujemy następujące wskaźniki oceny działania algorytmu:

- Klasyfikatory oceny jakości poznane na wykładzie
- Czas działania algorytmu
- Liczba iteracji wykonywania się algorytmu

Przeprowadzone eksperymenty wykorzystamy jako możliwość sprawdzenia zachowania się miar jakości (pola pod krzywą ROC, FP_{rate}, TP_{rate} itp.) w przypadku klasyfikacji binarnych (oszustwa i zachowania abonamentów, oraz wartością błędu średniokwadratowego w danych o zużyciu energii.

Będziemy zapoznawać się i badać poznane na wykładzie biblioteczne metody budowania drzew przed przystąpieniem do metody gradient boosting.

Główne badania metody gradient boosting:

Algorytm jest iteracyjny więc będziemy badać jakość modelu w funkcji liczby iteracji (posługując się powyższymi wskaźnikami).

Do badań zaliczamy testowanie wpływu złożoności modeli podstawowych na jakość modelu złożonego.

4. Otwarte kwestie

Jako otwartą kwestię przyjmujemy możliwość budowy lasu losowego w celu porównania wyników działania zaimplementowanego algorytmu oraz porównanie z pojedynczym drzewem.

W trakcie wykonywania projektu może się okazać, że funkcje budowy drzew zaimplementowane w pakiecie rpart nie będą w stanie poprawnie współpracować z algorytmem, istnieje szansa, że konieczne będzie korzystanie z innego pakietu.

Jesteśmy ciekawi jak sprawdza się metoda oceny bootstrapping oprócz walidacji krzyżowej w zastosowaniu do gradient-boostingu.