Uczenie Maszynowe - projekt

Tomasz Owienko Wojciech Zarzecki

18.11.2023

1 Cel projektu

Celem projektu jest implementacja zmodyfikowanej wersji algorytmu generowania lasu losowego, w której do generowania kolejnych drzew losowane są częściej elementy ze zbioru uczącego, na których dotychczasowy model się mylił.

Istotą metody lasu losowego w problemach regresji i klasyfikacji jest redukcja wariancji i nadmiernego dopasowania osiąganego przez pojedyncze drzewa decyzyjne. W klasycznych algorytmach generowania lasu losowego każde z B drzew generowane jest na podstawie $\sqrt{B}$ przykładów ze zbioru trenującego wylosowanych ze zwracaniem, zazwyczaj ograniczonych (w problemach klasyfikacji) do $\sqrt{|D|}$ atrybutów wylosowanych bez zwracania, gdzie D jest zbiorem atrybutów. Proces ten odbywa się w jednej iteracji - algorytm kończy pracę po wygenerowaniu B drzew. Taka koncepcja to tzw. *bagging*, który ma na celu ograniczenie wariancji modelu po przez agreację wielu prostszych modeli - w tym przypadku drzew decyzyjnych.

Realizowany projekt zakłada zbadanie możliwości zastosowania metod *boostingu* oraz *stackingu*. Wdrożenie metod boostingowych polega na uwzględnianiu błędów poprzedniego drzewewa decyzyjnego, przy budowanie kolejnego. W takim podejściu nowe drzewa są nieustannie dołączane są do istniejącego już lasu. Natomiast stacking zakłada utworzenie nowego zbioru danych na bazie wyników wytrenowanego modelu oraz wytrenowanie na nowym zbiorze *metamodelu*. Co istotne, w podejściu stackingowym las wygenerowany w iteracji $n+1$ w pełni zastępuje las z iteracji n.

2 Opis algorytmu

Algorytm korzystający z boostingu - kolejne drzewa dołączane są cały czas do tego samego zbioru:

Algorithm 1 Boosted Random Forest

Input: $U \neq \emptyset$: zbiór przykładów trenujących, C: klasy przykładów, D: zbiór atrybutów wejściowych

// przy założeniu, że każda iteracja to nowy las
$F \leftarrow \emptyset$
$\hat{C} \leftarrow \emptyset$
for $i \leftarrow 1$ to N **do**
  $F_i \leftarrow \emptyset$
  for $b \leftarrow 1$ to B **do**
    $U_b \leftarrow$B elementów wylosowanych z U z preferencją dla $\{u_j \in U : \hat{C}(u_j) \neq C(u_j)\}$
    $D_b \leftarrow \sqrt{|D|}$ atrybutów wylosowanych z D bez zwracania
    $f_b \leftarrow$ drzewo decyzyjne wygenerowane na podstawie C, U_b i D_b
    $F_i \leftarrow F_i \cup \{f_b\}$
  end for
  $F \leftarrow$ las losowy F ulepszony za pomocą F_i
  for $u \in U$ **do**
    $\hat{C} \leftarrow \hat{C} \cup$ PredictRandomForest(u, F)
  end for
end for
return F

Algorytm korzystający ze stackingu:

Algorithm 2 Stacked Random Forest with Iterative Retraining

Input: $U \neq \emptyset$: zbiór przykładów trenujących, C: klasy przykładów, D: zbiór atrybutów wejściowych

$OriginalData \leftarrow U$
for $m \leftarrow 1$ to M **do**
  // Trenowanie N niezależnych lasów losowych
  $Forests \leftarrow \emptyset$
  for $i \leftarrow 1$ to N **do**
    $F_i \leftarrow \text{RandomForest}(OriginalData, C, D)$
    $Forests \leftarrow Forests \cup \{F_i\}$
  end for
  // Tworzenie nowego zestawu danych na podstawie wyników lasów
  $NewData \leftarrow \emptyset$
  for $u \in OriginalData$ **do**
    $Results \leftarrow []$
    for $F_i \in Forests$ **do**
      $Results \leftarrow Results \cup \text{PredictRandomForest}(u, F_i)$
    end for
    $NewData \leftarrow NewData \cup \{(Results, C(u))\}$
  end for
  $OriginalData \leftarrow NewData$
end for
return $Forests$

Predykcja za pomocą wygenerowanych lasów losowych - bez różnic względem podejścia klasycznego:

Algorithm 3 PredictRandomForest

Input: x: wektor wejściowy, F: nauczony las losowy

$\hat{C} \leftarrow \emptyset$
for $f_b \in F$ **do**
  $\hat{C} \leftarrow \hat{C} \cup f_b(x)$
end for
return najczęstsza klasa z $\hat{C}$

3 Planowane eksperymenty

Planujemy porównać efektywność następujących podejść:

- klasyczny Random Forest
- Boosted Random Forest
- Stacked Random Forest
- połącznie Stacked RF i Boosted RF

4 Wykorzystywane zbiory danych

- UCI Mushrooms Dataset
- UCI Breast Cancer Dataset
- Titanic Dataset on Kaggle