

UCZENIE SIĘ MASZYN

Drzewa i lasy losowe
Dokumentacja wstępna

Autor: Krzysztof Marcinek
Prowadzący: Paweł Cichosz

1. Wprowadzenie

Jednym z najistotniejszych zagadnień z dziedziny uczenia się maszyn jest wybór metody klasyfikacji. Niniejsza praca skupia się na problemie klasyfikacji pod nadzorem jako zadania wyboru (konstrukcji) klasyfikatora (reguły/funkcji dyskryminacyjnej) na podstawie dostępnej próby uczącej. Zaprezentowane zostaną metody stabilizacji klasyfikatorów u podstaw których leży idea poprawy dokładności działania dowolnej reguły. Techniki te polegają na konstruowaniu wielu wersji jednego klasyfikatora (rodziny klasyfikatorów) na podstawie pewnych pseudoprób utworzonych z oryginalnej próby uczącej.

- Pojęcie rodziny klasyfikatorów:

$$D = \{ d_k : \mathbf{X} \rightarrow \{1, 2, \dots, g\} \}_{k=1,2,\dots,K}, \text{ gdzie } K \geq 2$$

- Liczbę głosów oddanych przez rodzinę klasyfikatorów D na to, aby obserwację $x \in \mathbf{X}$ zaklasyfikować do klasy $j \in \{1, 2, \dots, g\}$ określamy jako:

$$N_j(x) := \#\{k : d_k(x) = j\}$$

- Klasyfikatorem generowanym przez rodzinę klasyfikatorów D nazywamy klasyfikator $dD : \mathbf{X} \rightarrow \{1, 2, \dots, g\}$ wybrany regułą głosowania:

$$dD(x) := \arg \max_j N_j(x)$$

- Rodzina klasyfikatorów generuje regułę poprzez ostateczne zaklasyfikowanie x do klasy, która była najczęściej wskazywana przez poszczególne reguły z rodziny.

2. Bootstrap i metoda bagging

- Bagging (Bootstrap Aggregating) jest jedną z pierwszych rodzin klasyfikatorów, zaproponowaną przez Breimana w 1996 r.
- Załóżmy, że dany mamy zbiór uczący $L \in \mathbf{L}$ posiadający n elementów oraz klasyfikator $d : \mathbf{X} \times \mathbf{L} \rightarrow \{1, 2, \dots, g\}$. Na podstawie zbioru L tworzymy K pseudoprób L_1, L_2, \dots, L_K . Każda pseudopróba L_k powstaje w wyniku wylosowania ze zwracaniem n -elementów z wyjściowego zbioru uczącego L . Zakładamy przy tym, że wylosowanie każdego spośród n elementów jest równoprawdopodobne. Taki sposób generowania pseudoprób nazywamy metodą bootstrap. Rozważmy K reguł $d(\cdot, L_k) : \mathbf{X} \rightarrow \{1, 2, \dots, g\}$. Mówimy, że reguła $d(\cdot, L_k)$ jest k -tą wersją klasyfikatora d .
- Algorytm: Metoda bagging

Dane wejściowe : L – próba ucząca , K – liczba iteracji

for $k = 1$ **to** K **do**

- Z próby uczącej L wygeneruj pseudopróbkę L_k metodą bootstrap
- Skonstruuj regułę decyzyjną $d(\cdot, L_k)$

end for

Wyjście : Zlicz liczbę głosów $N_j(x)$, a następnie oblicz $dD(x) := \arg \max_j N_j(x)$

3. Drzewa decyzyjne

- Drzewa decyzyjne są jedną z najbardziej skutecznych i najpopularniejszych metod klasyfikacji. Podstawową zaletą drzew klasyfikacyjnych jest możliwość reprezentowania dowolnego pojęcia wynikającego z danych uczących.
- Pojęcie drzewa wywodzi się z teorii grafów. Drzewem nazywamy graf spójny i acykliczny. Wierzchołek nieposiadający rodzica nazywamy korzeniem. Krawędzie w drzewie nazywamy gałęziami. Każdy wierzchołek nie będący korzeniem a posiadający gałęzie nazywamy węzłem. Pozostałe wierzchołki nazywamy liśćmi drzewa. Korzeń w drzewie wyznacza kierunek, rozumiany jako kierunek na ścieżce łączącej korzeń z dowolnym wierzchołkiem w drzewie, w szczególności z liściem. Zgodnie z acyklicznością grafu, od korzenia do ustalonego liścia prowadzi tylko jedna droga.

4. Lasy losowe

- Załóżmy, że dany mamy zbiór uczący $L \in \mathbf{L}$ posiadający n elementów. Na podstawie zbioru L tworzymy K pseudoprób metodą bootstrap L_1, L_2, \dots, L_K . Wykorzystując każdą z utworzonych pseudoprób budujemy drzewa klasyfikacyjne, gdzie proces budowy modyfikujemy tak, aby w każdym węźle dokonać wyboru najlepszego podziału na podstawie m wylosowanych atrybutów. W ten sposób otrzymujemy K drzew t_1, t_2, \dots, t_K . Wykorzystując regułę głosowania otrzymujemy klasyfikator dF generowany rodziną F . Klasyfikator dF nazywamy klasyfikatorem otrzymanym metodą lasów losowych.

- Algorytm: Forest-RI

Dane wejściowe : L – próba ucząca , K – liczba iteracji

for $k = 1$ **to** K **do**

- Z próby uczącej L wygeneruj pseudopróbę L_k metodą bootstrap
- **Iteracja dla:** wszystkich węzłów w w drzewie t_k dopóki nie są spełnione warunki zastopowania budowy drzewa
 - Wybierz m atrybutów do podziału
 - Dla każdego z nich wybierz najlepszy podział
 - Wykonaj podział elementów znajdujących się w węźle w oparty o wcześniej wybrany podział

end for

Wyjście: Zlicz liczbę głosów $N_j(x)$, a następnie oblicz $dF(x) := \arg \max_j N_j(x)$

- Algorytm: Forest-RC

Dane wejściowe : L – próba ucząca , K – liczba iteracji

for $k = 1$ **to** K **do**

- Z próby uczącej L wygeneruj pseudopróbę L_k metodą bootstrap
- **Iteracja dla:** wszystkich węzłów w w drzewie t_k dopóki nie są spełnione warunki zastopowania budowy drzewa
 - Wybierz m atrybutów do podziału
 - Wygeneruj macierz o wymiarach $m \times f$ wypełnioną wartościami wylosowanymi z rozkładu jednostajnego na przedziale $[-1, 1]$
 - Stwórz f nowych zmiennych i przypisz ich wartości dla każdego elementu x_i w węźle w
 - Znajdź najlepsze podziały dla każdej z nowych zmiennych
 - Wykonaj optymalny podział elementów znajdujących się w węźle w

end for

Wyjście: Zlicz liczbę głosów $N_j(x)$, a następnie oblicz $dF(x) := \arg \max_j N_j(x)$

5. Badanie algorytmów

- W ramach pracy zostaną zaimplementowane oraz przetestowane zaprezentowane wcześniej algorytmy na zróżnicowanych zestawach danych dostępnych w UCI Machine Learning Repository.
 - Wykreślenie zależności błędu klasyfikacji lasu deterministycznego, lasu losowego, oraz pojedynczego drzewa wylosowanego z rodziny drzew losowych. Test ma na celu wstępne zaobserwowanie przewagi drzew losowych w stosunku do drzew deterministycznych i przewagi jaką niosą metody rodzin klasyfikatorów.
 - Zbadanie zależności błędu klasyfikacji lasu deterministycznego oraz lasu losowego w funkcji liczby drzew. test ma na celu zbadanie czy i w jaki sposób wpływa zwiększanie drzew w klasyfikatorze na finalny efekt klasyfikacji.
 - Wyznaczenie zależności błędu na próbie uczącej w stosunku do błędu na zestawie testowym w funkcji ilości drzew w klasyfikatorze. Pozwoli to nam na sprawdzenie jak uzyskane klasyfikatory poradzą sobie z klasyfikacją nowych przykładów oraz zbadanie zjawiska przeuczenia.
 - Zbadanie stabilności algorytmów budowy losów losowych. Test ma na celu zaobserwowanie jak badane algorytmy są wrażliwe na nieznacznie zmiany zbioru danych trenujących.
 - Wpływ zakłóceń w danych trenujących na sposób klasyfikacji poszczególnych algorytmów. Test pozwoli zaobserwować różnicę w błędach klasyfikacji w przypadku danych dokładnych jak i zaszumionych.

6. Bibliografia

- [1] Paweł Cichosz: Systemy uczące się, Wydawnictwa Naukowo-Techniczne, Warszawa 2000
- [2] Leo Breiman: Random Forests, Statistics Department, University of California, Berkeley, CA 94720
- [3] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, Charles J. Stone: Classification and Regression Trees, Chapman & Hall, New York, NY, 1984
- [4] Leo Breiman, Adele Cutler: Random Forest, http://stat-www.berkeley.edu/users/breiman/RandomForests/cc_home.htm
- [5] Wikipedia: Random Forest, http://en.wikipedia.org/wiki/Random_forest