Zadanie klasyfikacji- klasyfikatory zespołowe, wykład 7

Joanna Jędrzejowicz

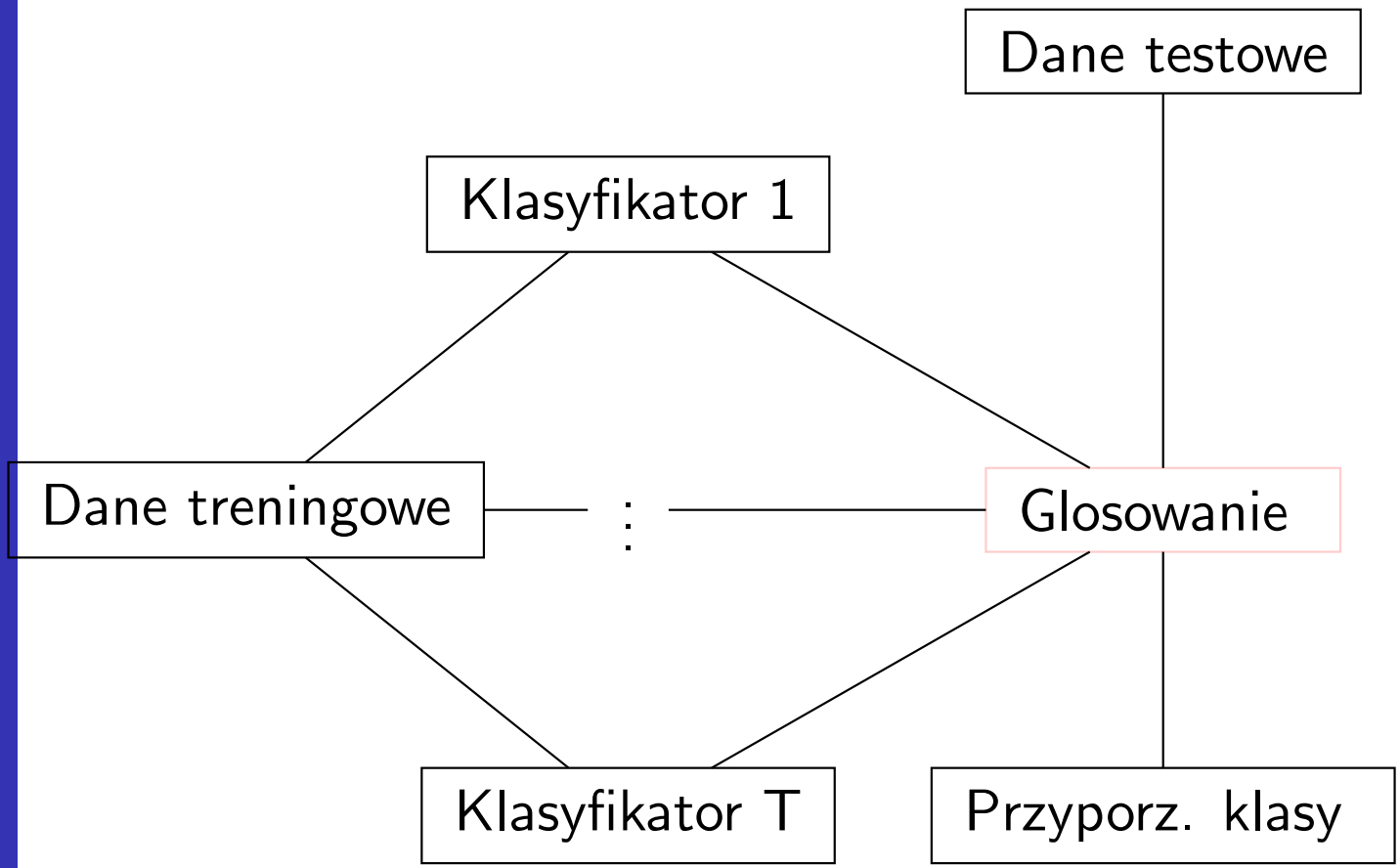
Instytut Informatyki

Dlaczego klasyfikatory zespołowe

Wiele sytuacji, w których konieczne albo opłacalne jest łączenie informacji otrzymanych z wielu klasyfikatorów:

- np. w procesie uczenia integracja danych może nie być możliwa ponieważ względy bezpieczeństwa i ochrony danych osobowych mogą wykluczać ich łączenie,
- lub złożoność obliczeniowa problemu klasyfikacji może wymagać partycji danych i uczenia odrębnego klasyfikatora na każdym z podzbiorów oryginalnego zbioru danych.
- porównanie z życia codziennego - potwierdzanie diagnozy u kilku specjalistów

Jak działają klasyfikatory zespołowe



Rodzaje klasyfikatorów zespołowych

Techniki oparte na łączeniu klasyfikatorów:

- bagging - wielokrotne losowanie ze zwracaniem elementów z oryginalnego zbioru treningowego,
- wzmacnianie (ang.: boosting), w szczególnosci AdaBoost,

Bagging i boosting

Obie metody budowy klasyfikatora zespołowego wykorzystują głosowanie indywidualnych klasyfikatorów (tzw. słabe klasyfikatory) oraz wszystkie klasyfikatory w zespole są tego samego typu, np. drzewa decyzyjne.
W przypadku bagging wagi każdego klasyfikatora są takie same. W przypadku boosting większy wpływ na wynik głosowania mają klasyfikatory z mniejszym błędem.

Bagging - generowanie modelu


```
Niech n oznacza liczbę rekordów w zbiorze treningowym
t - liczbę klasyfikatorów (słabych)
Powtórz t razy następujace czynności:
1. wybierz losowo n rekordów ze zbioru treningowego,
2. zastosuj procedurę uczenia dla wyznaczonego zbioru
3. zachowaj uzyskany klasyfikator
```

Uwaga - dane w zbiorze treningowym mogą się powtarzać!

Bagging cd - klasyfikacja


```
Dany rekord r ze zbioru testowego
Dla każdego   i=1,..,t
    Zastosuj klasyfikator Ci do rekordu r,
    wyznacz klasę Ci(r).
Przyjmij jako wynik klasyfikacji klasę, która
wśród C1(r), ... , Ct(r) wystąpiła największą liczbę
```

Przykład: metoda Random Forest (losowe lasy) wykorzystująca drzewa decyzyjne

Boosting - wzmocnienie

Wzmocnienie AdaBoost polega na iterowaniu następującego postępowania:
- zgodnie z aktualnym rozkładem, który jest jednostajny w pierwszym kroku, losuje się próbkę danych,
- dla tej próbki znajduje się najlepszy słaby klasyfikator i oblicza się wartosć błędu na całym zbiorze treningowym oraz modyfikuje się aktualny rozkład.

Boosting - wzmocnienie

- modyfikuje się aktualny rozkład tak, że wagi odpowiadające przykładom dobrze sklasyfikowanym są zmniejszane proporcjonalnie do błędu, zaś wagi dla błędnie sklasyfikowanych przykładów pozostają bez zmiany,
- po wygenerowaniu TT klasyfikatorów wykonuje się test na zbiorze testowym, gdzie zespół klasyfikatorów dokonuje klasyfikacji przez 'ważone głosowanie większościowe'.
- promowane są te klasyfikatory, które w trakcie uczenia lepiej się zachowywały – generowały mniejszy błąd.

Algorytm AdaBoost


```
Wejście: zbior n danych trening. pochodzacych z C kla
TD = {(xi, yi): yi - klasa xi} dla i=1,...,n,
zbior danych testowych TS,
liczba całkowita TT  okreslajaca liczbe iteracji,
M - rozmiar losowo wybranego podzbioru TD
Wyjscie: liczba qc -jakosc  klasyfikatora.
1. inicjalizuj rozklad  D1(i)
2 . dla t = 1,...,TT wykonuj
2.1 wylosuj zbior danych uczacych ST rozmiaru M
ze zbioru TD zgodnie z aktualnym rozkladem,
2.2 dla zbioru danych ST  znajdz klasyfikator Ct,
2.3 wylicz wartosc bledu dla klasyfikatora Ct
Jeśli Et > 0.5 to przerwanie, inaczej wylicz Bt
2.4  modyfikuj rozklad
2.5 znormalizuj rozklad
```

AdaBoost - jakość klasyfikacji

Rozkład początkowy

$$D_1(i) = \frac{1}{n},\ i = 1, \ldots, n$$

Dla klasyfikatora Ct wyliczamy błąd:

$$E_t = \sum_{C_t(x_i) \neq y_i} D_t(i)$$

Korekta rozkładu

$$B_t = \frac{E_t}{1 - E_t}$$

Jeśli $C_t(x_i) = y_i$, to modyfikuj $D_t(i) := D_t(i) \times B_t$
Normalizacja rozkładu:

$$D_{t+1}(i) := \frac{D_t(i)}{\sum_i D_t(i)}$$

Algorytm AdaBoost cd, testowanie


```
przestestuj zespół klasyfikatorów
C1, ..., CTT na zbiorze testowym TS,
3.1 qc:=0;
3.2 dla każdego przykładu (x,y) ze zbioru TS
3.2.1 wylicz Vi
3.2.2 przyjmij c:=arg max Vi dla  i=1,...,|C|,
3.2.3 jeśli c = y, to qc:= qc + 1,
3.3 wylicz qc
```

AdaBoost - wzory cd

Głosowanie

$$V_i = \sum_{C_t(x)=i} \log \frac{1}{B_t} \; i = 1, \ldots |C|$$

Jakość klasyfikacji:

$$qc := \frac{qc}{|TS|}$$

Uwagi

Warto zauważyć, że wagi rozkładu są redukowane przez czynnik Bt dla poprawnie sklasyfikowanych przykładów, a pozostałe pozostają bez zmiany.
Można sprawdzić, że po normalizacji współczynniki dla niepoprawnie sklasyfikowanych danych są zwiększone i dają w sumie 0.5, zaś współczynniki dla poprawnie sklasyfikowanych danych są zmniejszone i także sumują się do 0.5.

Uwagi

Zakładamy, że słaby klasyfikator ma błąd mniejszy niż 0.5 (w przeciwnym przypadku następuje przerwanie obliczenia), zatem w każdym kroku przynajmniej jeden przykład źle sklasyfikowany będzie dobrze sklasyfikowany w następnym kroku.

Rotation Forest

Główna idea: zapewnić jednocześnie różnorodność i dokładność

- różnorodność: używa się rozkładu PCA (Principal Components Analysis) do dokonania wyboru atrybutów dla każdego klasyfikatora bazowego,
- dokładność: zachowuje się wszystkie składowe rozkładu PCA oraz cały zbiór treningowy jest używany do uczenia każdego klasyfikatora bazowego.

PCA

Algorytm PCA:
dane: macierz X zawierająca M wierszy i N kolumn (kolumny odp. atrybutom)
wyjście: macierz przekształcenia wymiaru NxN

1. normalizacja macierzy X : dla każdego atrybutu policz wartość średnią (wartości w kolumnie) i odejmij tę wartość od każdego wiersza,
2. znajdź macierz kowariancji Cov,
3. znajdż wartości własne i wektory własne macierzy kowariancji - wektor własny odpowiadający największej wartości własnej jest tzw. główną składową,
4. utwórz macierz przekształcenia

Rotation Forest

1. Dla każdego klasyfikatora bazowego tworzy się zbiór treningowy dzieląc losowo zbiór atrybutów na K podzbiorów (K jest parametrem)
2. stosuje się rozkład PCA do każdego podzbioru atrybutów, znajduje się odpowiednie wektory własne,
3. otrzymane wektory własne są wstawiane do rzadkiej macierzy 'rotation' , która jest porządkowana zgodnie z porządkiem atrybutów,
4. klasyfikator bazowy jest uczony na macierzy $X \cdot R$, gdzie X jest macierzą początkową, a R macierzą 'rotation uporządkowaną.

WEKA

- klasyfikatory AdaBoost i Rotation Forest są zaimplementowane i można znaleźć w Classifier-Classifiers-Meta
- Random Forest w Classifier-Classifiers-Trees

Zbiory danych - benchmark datasets, w szczególności dane medyczne

http://www.ics.uci.edu/ mlearn/MLRepository.html,

nazwa zbioru	liczba rekordow	liczba atryb.	liczba klas
WBC	569	9	2
Heart	303	14	2
Acute Inflamm.	120	8	2
CMC	1473	9	3
Ecoli	336	8	8
PIMA	768	8	2

Ocena jakości klasyfikatora - przypadek dwóch klas, macierz błędu klasyfikacji - stan faktyczny i wskazanie modelu

Macierz błędu klasyfikacji (ang. confusion matrix)

stan faktyczny	przewidywana klasa		suma
	tak	nie	
tak	TP	FN	P
nie	FP	TN	N

Tabela podsumowuje wyniki klasyfikacji dla danej reguły decyzyjnej, porównując stan faktyczny ze wskazaniem modelu - np. wynikiem testu diagnostycznego. Wartości na przekątnej dotyczą prawidłowej klasyfikacji, dobry model minimalizuje FP i FN. W niektórych zastosowaniach błędne klasyfikacje mogą mieć różny koszt.

Przykład 1

stan faktyczny	przewidywana klasa		suma
	tak	nie	
tak	8 000	1 000	9 000
nie	2 000	8 000	10 000

TP rate: $\frac{TP}{TP+FN} = \frac{8000}{9000} = 0.89$

FP rate: $\frac{FP}{TN+FP} = \frac{2000}{10000} = 0.2$

dokładność: $\frac{TP+TN}{P+N} = \frac{16000}{19000} = 0.842$

Przykład 2

stan faktyczny	przewidywana klasa		suma
	tak	nie	
tak	80 000	10 000	90 000
nie	2 000	8 000	10 000

TP rate: $\frac{TP}{TP+FN} = \frac{80000}{90000} = 0.89$

FP rate: $\frac{FP}{TN+FP} = \frac{2000}{10000} = 0.2$

dokładność: $\frac{TP+TN}{P+N} = \frac{88000}{100000} = 0.88$