

Algorytmy eksploracji danych: Wykład 8

Copyright by Wojciech Kempa

Politechnika Śląska
Wydział Matematyki Stosowanej

Indeks Gini (1)

- Inną często wykorzystywaną miarą nieuporządkowania zbiorów jest tzw. **indeks Gini (Giniego)** (Corrado Gini (1884-1965) był włoskim statystykiem i demografem).

Copyright by Wojciech Kempa

Indeks Gini (1)

- Inną często wykorzystywaną miarą nieuporządkowania zbiorów jest tzw. **indeks Gini (Giniego)** (Corrado Gini (1884-1965) był włoskim statystykiem i demografem).
- Indeks Gini wykorzystywany jest m.in. przez algorytmy klasifykacyjne CART i SPRINT.

Copyright by Wojciech Kempa

Indeks Gini (1)

- Inną często wykorzystywaną miarą nieuporządkowania zbiorów jest tzw. **indeks Gini (Giniego)** (Corrado Gini (1884-1965) był włoskim statystykiem i demografem).
- Indeks Gini wykorzystywany jest m.in. przez algorytmy klasyfikacyjne CART i SPRINT.
- Miarę tę definiuje się dla zbioru treningowego D w następujący sposób:

Copyright by Wojciech Kempa

$$Gini(D) \stackrel{def}{=} 1 - \sum_{i=1}^m p_i^2, \quad (1)$$

gdzie p_i oznacza prawdopodobieństwo, że wybrany element należy do klasy C_i atrybutu decyzyjnego.

Indeks Gini (1)

- Inną często wykorzystywaną miarą nieuporządkowania zbiorów jest tzw. **indeks Gini (Giniego)** (Corrado Gini (1884-1965) był włoskim statystykiem i demografem).
- Indeks Gini wykorzystywany jest m.in. przez algorytmy klasyfikacyjne CART i SPRINT.
- Miarę tę definiuje się dla zbioru treningowego D w następujący sposób:

Copyright by Wojciech Kempa

$$Gini(D) \stackrel{def}{=} 1 - \sum_{i=1}^m p_i^2, \quad (1)$$

gdzie p_i oznacza prawdopodobieństwo, że wybrany element należy do klasy C_i atrybutu decyzyjnego.

- Prawdopodobieństwo p_i estymujemy za pomocą częstości względnej $\frac{n_i}{n}$, gdzie n jest liczbą obiektów w bazie danych D , a n_i oznacza liczbę obiektów D reprezentujących klasę C_i .

- Sposób wykorzystania indeksu Gini do wyboru atrybutu „podziałowego” jest następujący.

Copyright by Wojciech Kempa

- Sposób wykorzystania indeksu Gini do wyboru atrybutu „podziałowego” jest następujący.
- Założymy, że atrybut warunkowy A dzieli zbiór treningowy na dwie partycje: D_1 i D_2 .

Copyright by Wojciech Kempa

- Sposób wykorzystania indeksu Gini do wyboru atrybutu „podziałowego” jest następujący.
- Założymy, że atrybut warunkowy A dzieli zbiór treningowy na dwie partycje: D_1 i D_2 .
- **Indeks podziału Gini** zbioru treningowego D , uwzględniający podział na partycje względem atrybutu A , ma postać

$$Gini_{split}^A(D_1, D_2) \stackrel{def}{=} \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2), \quad (2)$$

a zatem jest średnią ważoną indeksów Gini poszczególnych partycji.

Zysk informacyjny wynikający z zastosowania atrybutu A jako potencjalnego atrybutu „podziałowego”, uwzględniając podział zbioru D na partie D_1 i D_2 , obliczamy teraz w następujący sposób:

$$Gain_{Gini}(A) = Gini(D) - Gini_{split}^A(D_1, D_2). \quad (3)$$

Wartość powyższego indeksu określa różnicę w ilości informacji (zysk informacyjny) potrzebnej do sklasyfikowania obiektów zbioru D przed i po jego podziale na partie D_1 i D_2 względem atrybutu warunkowego A .

- Widać zatem, że w algorytmach klasyfikacyjnych opartych na indeksie Gini wybór atrybutu „podziałowego” na danym etapie klasyfikacji to za mało.

Copyright by Wojciech Kempa

- Widać zatem, że w algorytmach klasyfikacyjnych opartych na indeksie Gini wybór atrybutu „podziałowego” na danym etapie klasyfikacji to za mało.
- Konieczny jest także wybór **optymalnego podziału** zbioru treningowego względem tego atrybutu (tzn. takiego, który maksymalizuje zysk informacyjny lub, równoważnie, minimalizuje indeks podziału Gini).

- Widać zatem, że w algorytmach klasyfikacyjnych opartych na indeksie Gini wybór atrybutu „podziałowego” na danym etapie klasyfikacji to za mało.
- Konieczny jest także wybór **optymalnego podziału** zbioru treningowego względem tego atrybutu (tzn. takiego, który maksymalizuje zysk informacyjny lub, równoważnie, minimalizuje indeks podziału Gini).
- Drzewo klasyfikacyjne, którego konstrukcja oparta będzie na wykorzystaniu indeksu Gini będzie zatem **drzewem binarnym** (z każdego węzła wychodzić będą tylko dwie gałęzie drzewa, odpowiadające partycjom D_1 i D_2).

Algorytm SPRINT (1)

Ogólny schemat algorytmu klasyfikacyjnego opartego na indeksie Gini przedstawimy na przykładzie algorytmu SPRINT:

1. Określamy początkowy zbiór treningowy.
2. Dla każdego atrybutu warunkowego A i dla wszystkich możliwych punktów podziału wartości tego atrybutu obliczamy indeks podziału Gini. W przypadku atrybutu ciągłego partycje utworzone są przez warunki postaci $A \leq a$ oraz $A > a$, gdzie a jest punktem podziału. Dla pozostałych typów atrybutów partycje tworzą warunki $A = a$ oraz $A \neq a$.

3. Wybieramy punkt podziału o najmniejszej wartości indeksu

$$Gini_{split}^A(D_1, D_2)$$

lub, równoważnie, o największej wartości indeksu

$Gain_{Gini}(A)$.
Copyright by Wojciech Kempa

4. Wybrany punkt podziału włączamy do konstruowanego drzewa decyzyjnego: dzieli on zbiór D na partie D_1 i D_2 .
5. Powtarzamy procedurę poszukiwania punktu podziału i obliczania indeksu podziału Gini dla partycji D_1 i D_2 . Znalezione punkty podziału przyłączamy do drzewa decyzyjnego itd.

Optymalizacja drzewa decyzyjnego (1)

- W praktyce podczas konstrukcji drzewa klasyfikacyjnego na podstawie zbioru treningowego dochodzi czasem do zjawiska tzw. **przeuczenia klasyfikatora** (ang. *overfitting*), czyli zbyt silnego dopasowania otrzymanego drzewa decyzyjnego do zbioru treningowego (uczącego).

Copyright by Wojciech Kempa

- W praktyce podczas konstrukcji drzewa klasyfikacyjnego na podstawie zbioru treningowego dochodzi czasem do zjawiska tzw. **przeuczenia klasyfikatora** (ang. *overfitting*), czyli zbyt silnego dopasowania otrzymanego drzewa decyzyjnego do zbioru treningowego (uczącego).
- Ma to miejsce np. wówczas, gdy zbiór treningowy niezbyt dobrze reprezentuje całą populację lub też jest zbyt mało liczny.

Copyright by Wojciech Kempa

- W praktyce podczas konstrukcji drzewa klasyfikacyjnego na podstawie zbioru treningowego dochodzi czasem do zjawiska tzw. **przeuczenia klasyfikatora** (ang. *overfitting*), czyli zbyt silnego dopasowania otrzymanego drzewa decyzyjnego do zbioru treningowego (uczącego).
- Ma to miejsce np. wówczas, gdy zbiór treningowy niezbyt dobrze reprezentuje całą populację lub też jest zbyt mało liczny.
- W efekcie otrzymane drzewo decyzyjne może błędnie klasyfikować nowe obiekty (głównie takie, które są mało „podobne” do obiektów zbioru treningowego).

- W praktyce podczas konstrukcji drzewa klasyfikacyjnego na podstawie zbioru treningowego dochodzi czasem do zjawiska tzw. **przeuczenia klasyfikatora** (ang. *overfitting*), czyli zbyt silnego dopasowania otrzymanego drzewa decyzyjnego do zbioru treningowego (uczącego).
- Ma to miejsce np. wówczas, gdy zbiór treningowy niezbyt dobrze reprezentuje całą populację lub też jest zbyt mało liczny.
- W efekcie otrzymane drzewo decyzyjne może błędnie klasyfikować nowe obiekty (głównie takie, które są mało „podobne” do obiektów zbioru treningowego).
- Poziom ryzyka błędnej klasyfikacji można zmniejszyć, wykorzystując jedną z technik tzw. **przycinania drzewa**.

- Stosowanych w praktyce technik jest kilka. Jedna z nich bazuje na wykorzystaniu tzw. **zasady minimalizacji długości kodu (MDL)** (ang. *Minimum Description Length*), w myśl której optymalne drzewo powinno charakteryzować się najmniejszym możliwym „kosztem” (liczbą bitów) jego zakodowania.

- Stosowanych w praktyce technik jest kilka. Jedna z nich bazuje na wykorzystaniu tzw. **zasady minimalizacji długości kodu (MDL)** (ang. *Minimum Description Length*), w myśl której optymalne drzewo powinno charakteryzować się najmniejszym możliwym „kosztem” (liczbą bitów) jego zakodowania.
- Inna z metod opiera się na wykorzystaniu tzw. **funkcji kary**.

Optymalizacja drzewa decyzyjnego (3)

Jakość $e(D)$ klasyfikatora w postaci drzewa decyzyjnego uzyskanego dla zbioru treningowego D szacuje się wówczas z następującego wzoru (jest to tzw. **błąd generalizacji drzewa**):

$$e(D) \stackrel{\text{def}}{=} \frac{1}{|D|} \sum_{i=1}^k [e(N_i) + \Omega(N_i)], \quad (4)$$

Copyright by Wojciech Kempa

gdzie $e(N_i)$ oznacza liczbę błędnie sklasyfikowanych obiektów w liściu (wierzchołku) N_i drzewa decyzyjnego, $\Omega(N_i)$ oznacza wartość funkcji kary $\Omega(\cdot)$ dla liścia (wierzchołka) N_i , k oznacza liczbę liści drzewa, a $|D|$ – moc zbioru treningowego.

Jeżeli wartość $e(D)$ drzewa przyciętego jest mniejsza od analogicznej wartości obliczonej dla drzewa oryginalnego, drzewo opłaca się przyciąć.

Klasyfikacja bayesowska (1)

- Zupełnie inną techniką klasyfikacyjną w porównaniu do konstrukcji drzew decyzyjnych jest tzw. **klasyfikacja bayesowska**.

Copyright by Wojciech Kempa

Klasyfikacja bayesowska (1)

- Zupełnie inną techniką klasyfikacyjną w porównaniu do konstrukcji drzew decyzyjnych jest tzw. **klasyfikacja bayesowska**.
- Opiera się ona na zastosowaniu **wzoru Bayesa** znanego z rachunku prawdopodobieństwa. Jeżeli X i Y są zdarzeniami losowymi, przy czym $\mathbf{P}(X) > 0$, wówczas prawdziwa jest równość

Copyright by Wojciech Kempa

$$\mathbf{P}(Y | X) = \frac{\mathbf{P}(Y \cap X)}{\mathbf{P}(X)} = \frac{\mathbf{P}(X | Y)\mathbf{P}(Y)}{\mathbf{P}(X)}. \quad (5)$$

Klasyfikacja bayesowska (1)

- Zupełnie inną techniką klasyfikacyjną w porównaniu do konstrukcji drzew decyzyjnych jest tzw. **klasyfikacja bayesowska**.
- Opiera się ona na zastosowaniu **wzoru Bayesa** znanego z rachunku prawdopodobieństwa. Jeżeli X i Y są zdarzeniami losowymi, przy czym $\mathbf{P}(X) > 0$, wówczas prawdziwa jest równość

Copyright by Wojciech Kempa

$$\mathbf{P}(Y | X) = \frac{\mathbf{P}(Y \cap X)}{\mathbf{P}(X)} = \frac{\mathbf{P}(X | Y)\mathbf{P}(Y)}{\mathbf{P}(X)}. \quad (5)$$

- Założymy teraz, że X jest pewnym obiektem o znanych wartościach atrybutów warunkowych, którego klasyfikacji należy dokonać. Interesuje nas zatem oszacowanie prawdopodobieństwa *a posteriori* $\mathbf{P}(C = C_i | X)$, $i = 1, \dots, m$, gdzie C jest atrybutem decyzyjnym.

Klasyfikacja bayesowska (2)

Oczywiście, logika podpowiada, aby obiekt X zakwalifikować do tej spośród klas C_i atrybutu decyzyjnego, dla której prawdopodobieństwo to jest największe. Jest to tzw. **zasada maksymalizacji prawdopodobieństwa a posteriori (MAP)**. Mamy zatem ze wzoru Bayesa

$$\mathbf{P}(C = C_i | X) = \frac{\mathbf{P}(X | C = C_i) \mathbf{P}(C = C_i)}{\mathbf{P}(X)}, \quad (6)$$

gdzie $i = 1, \dots, m$. Mianownik prawej strony wzoru (6) jest identyczny dla każdego i . Wartości $\mathbf{P}(C = C_i)$ możemy oszacować za pomocą częstości względnych $\frac{n_i}{n}$.

Klasyfikacja bayesowska (3)

- Kluczowe jest zatem oszacowanie prawdopodobieństw *a priori* $\mathbf{P}(X | C = C_i)$.

Copyright by Wojciech Kempa

- Kluczowe jest zatem oszacowanie prawdopodobieństw *a priori* $\mathbf{P}(X | C = C_i)$.
- Do tego celu w praktyce stosuje się różne podejścia, na przykład sieci bayesowskie, w których zależności pomiędzy poszczególnymi zdarzeniami przedstawia się w postaci acyklicznego grafu skierowanego. Inną z metod jest tzw. „naiwny” klasyfikator bayesowski, który poniżej przedstawimy szczegółowo.

„Naiwny” klasyfikator bayesowski (1)

Istotą „naiwnego” klasyfikatora bayesowskiego jest przyjęcie założenia o tzw. **warunkowej niezależności atrybutów**. Niech X, Y oraz Z będą zdarzeniami losowymi. Mówimy, że zdarzenie X jest **warunkowo niezależne** od zdarzenia Y względem zdarzenia Z , jeżeli zachodzi następująca równość:

$$\mathbf{P}(X | Y \cap Z) = \mathbf{P}(X | Z). \quad (7)$$

Zauważmy, że

Copyright by Wojciech Kempa

$$\begin{aligned} \mathbf{P}(X \cap Y | Z) &= \frac{\mathbf{P}(X \cap Y \cap Z)}{\mathbf{P}(Z)} \\ &= \frac{\mathbf{P}(X \cap Y \cap Z)}{\mathbf{P}(Y \cap Z)} \cdot \frac{\mathbf{P}(Y \cap Z)}{\mathbf{P}(Z)} = \mathbf{P}(X | Y \cap Z) \mathbf{P}(Y | Z), \end{aligned}$$

a stąd, wykorzystując (7),

$$\mathbf{P}(X \cap Y | Z) = \mathbf{P}(X | Z) \mathbf{P}(Y | Z). \quad (8)$$

„Naiwny” klasyfikator bayesowski (2)

Weźmy pod uwagę prawdopodobieństwo *a priori*

$$\mathbf{P}(X \mid C = C_i) = \mathbf{P}(A_1 = x_1, \dots, A_s = x_s \mid C = C_i).$$

Zakładając, że atrybuty warunkowe A_1, \dots, A_s są warunkowo niezależne względem zdarzenia $C = C_i$, mamy

Copyright by Wojciech Kempa

$$\mathbf{P}(X \mid C = C_i) = \prod_{j=1}^s \mathbf{P}(A_j = x_j \mid C = C_i), \quad (9)$$

gdzie $j = 1, \dots, m$.

Powyższa równość jest podstawą „naiwnej” klasyfikacji bayesowskiej.

„Naiwny” klasyfikator bayesowski (3)

Obiekt X przyporządkowujemy do tej klasy C^* atrybutu decyzyjnego, która maksymalizuje prawdopodobieństwo (porównaj licznik wzoru (6))

$$\mathbf{P}(C = C_i) \prod_{j=1}^s \mathbf{P}(A_j = x_j | C = C_i), \quad (10)$$

a zatem

$$C^* = \arg \max \mathbf{P}(C = C_i) \prod_{j=1}^s \mathbf{P}(A_j = x_j | C = C_i). \quad (11)$$

Prawdopodobieństwo $\mathbf{P}(C = C_i)$ szacujemy za pomocą częstości względnej $\frac{n_i}{n}$, gdzie n_i oznacza liczbę obiektów zbioru treningowego o liczności n , dla których $C = C_i$. Podobnie czynimy z poszczególnymi prawdopodobieństwami $\mathbf{P}(A_j = x_j | C = C_i)$, przyporządkowując im wartości $\frac{n_{i,j}}{n_i}$, gdzie $n_{i,j}$ jest liczbą elementów zbioru treningowego, dla których równocześnie $A_j = x_j$ oraz $C = C_i$.