Uczenie się drzew decyzyjnych

Bartłomiej Śnieżyński
KI, AGH

Plan

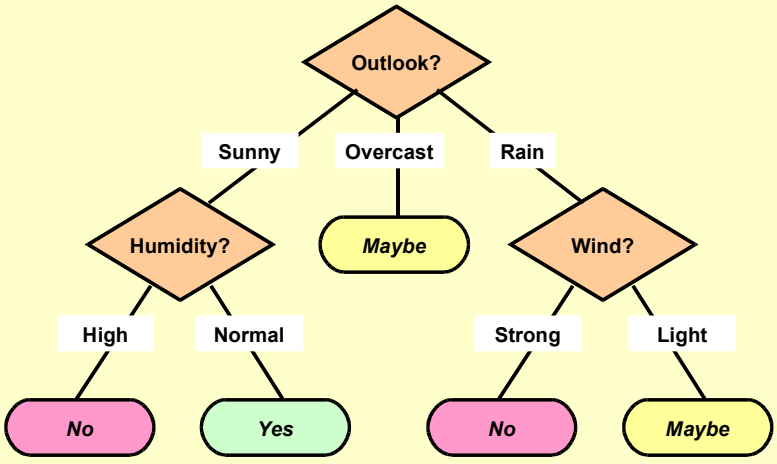
- Definicja
- Przykład
- Drzewo a reguły
- Rodzaje testów
- Wady i zalety drzew
- Uczenie się drzew – algorytm ID3
- Modyfikacje ID3 → C4.5
  - Unikanie nadmiernego dopasowania
  - Atrybuty ciągłe
  - Modyfikacja kryterium wyboru atrybutu
  - Dane z brakującymi wartościami
  - Uwzględnienie kosztu pomiaru atrybutów
- Złożoność
- Podsumowanie

Drzewo decyzyjne – definicja

- Dane: dziedzina X, $A=\{a_1, a_2, ..., a_n\}$, pojęcie $c: X \rightarrow C$
- Testem nazywamy funkcję $t:X \rightarrow R_t=\{r_1, r_2, ..., r_m\}$
- Syntaktyka: Drzewem decyzyjnym nazywamy:
  - liść etykietowany dowolnym pojęciem z C
  - wierzchołek etykietowany testem t o m gałęziach prowadzących do drzew decyzyjnych $\{T_1, T_2, ..., T_m\}$; krawędź prowadząca do T_i jest etykietowana wartością r_i
- Semantyka: Dla drzewa T: $h_T(x)=$
  - etykieta T jeśli T jest liściem,
  - $h_{Ti}(x)$, w pp., gdzie t jest testem związanym z korzeniem T, $t(x)=r_i$

Przykład drzewa decyzyjnego

- Drzewo decyzyjne dla problemu „czy grać w tenisa”



Drzewo decyzyjne a reguły

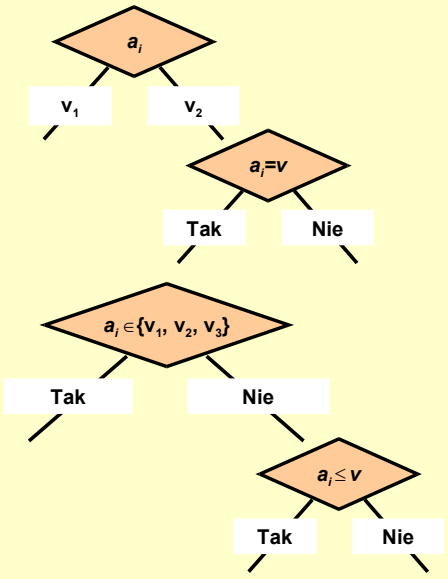
- Reguły: warunki → kategoria
- Konwersja drzewo → reguły
  - Każda ścieżka o wierzchołkach etykietowanych kolejno testami $t_1, t_2, ..., t_n, c$, i krawędziach etykietowanych wartościami $r_1, r_2, ..., r_n$ tworzy regułę

$$t_1(x)=r_1 \ \& \ t_2(x)=r_2 \ \& \ ..., \ t_n(x)=r_n \rightarrow c$$

  - Przykład: pierwsza od lewej ścieżka z poprzedniej str. *outlook*(*x*)=*sunny* & *humidity*(*x*)=*high* → *no*
- Konwersja reguły → drzewo
  - Por. Imam, Michalski
  - Algorytm – uczenie drzew z przykładów które są regułami

Rodzaje testów

- Testy tożsamościowe
- Testy równościowe
- Testy przynależnościowe
- Testy nierównościowe

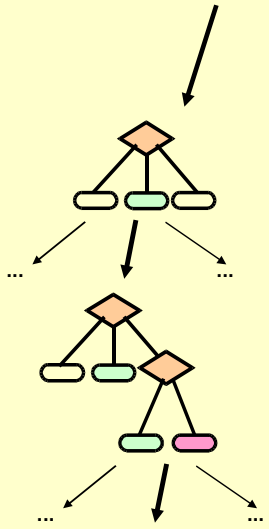


Wady i zalety drzew decyzyjnych

- Zalety:
  - Duża siła wyrazu (każda funkcja dyskretna jest reprezentowalna; DNF)
  - Reprezentacja jest zwarta
  - Szybka klasyfikacja
  - Stosunkowo czytelna dla człowieka
- Wady
  - Drzewa mogą być złożone:
    - testy na pojedynczych atrybutach – traci się zależności między atrybutami
    - reprezentacja alternatywy warunków powoduje rozrost drzewa
  - Trudne uczenie inkrementacyjne

Uczenie drzew jako przeszukiwanie

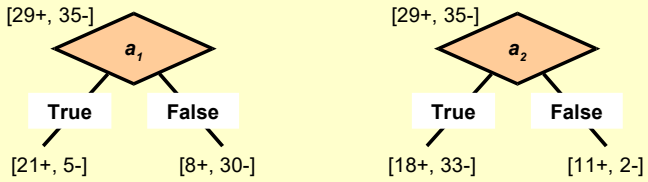
- Problem:
  - Przeszukaj przestrzeń drzew decyzyjnych, które reprezentują wszystkie możliwe funkcje dyskretne
  - Za: siła wyrazu, uniwersalność
  - Przeciw: złożoność, wielkie, niezrozumiałe drzewa
- Cel: znaleźć najlepsze drzewo decyzyjne (minimalne, zgodne z przykładami)
- Przeszkoda: problem NP-trudny
- Rozwiązania:
  - Użyć heurystyki do kierowania przeszukiwaniem
  - Użyć algorytmu zachłannego (greedy) (w optymalizacji jest to odpowiednik Hill-climbing bez nawracania)



Algorytm ID3 (Quinlan)

ID3 (T, Atr): drzewo
IF wszystkie przykłady w T mają taką samą etykietę THEN
  RETURN (liść z tą etykietą)
ELSE
  IF Atr=∅ THEN RETURN (liść z najczęstszą etykietą w T)
  ELSE
   a_i = najlepszy atrybut z Atr
   Utwórz korzeń k z testem na a_i
   Atr = Atr \ $\{a_i\}$
   FOR each $v \in A_i$
      Utwórz gałąź g od korzenia z etykietą v
      IF T_{aiv}=∅ THEN
        Do k podłącz liść z najczęstszą etykietą w T
      ELSE Do k podłącz ID3(T_{aiv}, Atr)

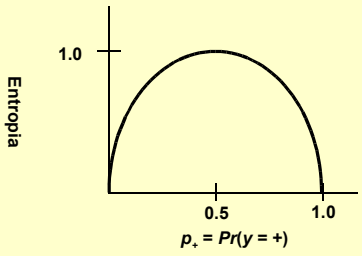
Wybór atrybutu



- Cel: atrybut który dzieli przykłady na podzbiory o dominującej jednej etykiecie
- Rezultat: bliżej do liścia
- Najbardziej popularna heurystyka:
  - Wymyślona przez Quinlan-a
  - Oparta na przyroście informacji

Entropia

- Miara nieuporządkowania, niepewności, nieregularności
- Przykład
  - C = {0, 1}, rozkład Pr(C)
  - Optymalne uporządkowanie:
    - Pr(c = 0) = 1, Pr(c = 1) = 0
    - Pr(c = 1) = 1, Pr(c = 0) = 0
  - Największy bałagan:
    - Pr(c = 0) = 0.5, Pr(c = 1) = 0.5

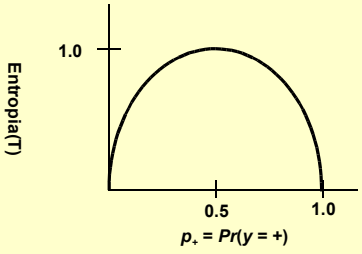


Entropia – definicja

- 2 klasy, T^c_A – przykłady,
  $p_+ = Pr(c(x) = 1)$, $p_- = Pr(c(x) = 0)$:

$$Entropia(T^c_A) = -p_+ \log_2 (p_+) - p_- \log_2 (p_-)$$

- Entropia(9+, 5-) = 0.940



- Wiele klas:

$$Entropia(T^c_A) = \Sigma_c -p_c \log_2 (p_c)$$

Przyrost informacji

- Cel: miara reprezentująca redukcję entropii po wyborze atrybutu a_i
- Definicja: dla przykładów S i atrybutu a_i

$$\text{Gain}(S, a_i) = \text{Etropia}(S) - \sum_{v \in A_i} \frac{|S_{aiv}|}{|S|} \text{Entropia}(S_{aiv})$$

- Idea: skalujemy entropię do rozmiaru każdego podzbioru
- Najlepszy atrybut:

$$\arg\max_{ai} \text{Gain}(S, a_i)$$

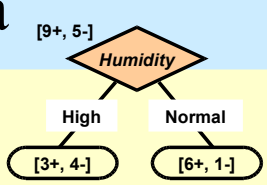
Przykład

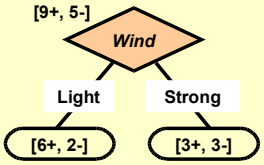
- Gra w tenisa

Day	Outlook	Temperature	Humidity	Wind
1	Sunny	Hot	High	Light
2	Sunny	Hot	High	Strong
3	Overcast	Hot	High	Light
4	Rain	Mild	High	Light
5	Rain	Cool	Normal	Light
6	Rain	Cool	Normal	Strong
7	Overcast	Cool	Normal	Strong
8	Sunny	Mild	High	Light
9	Sunny	Cool	Normal	Light

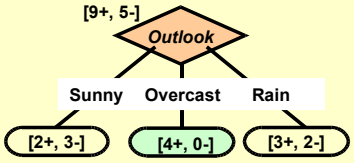
Wybór korzenia

Day	Outlook	Temperature	Humidity	Wind
1	Sunny	Hot	High	Light
2	Sunny	Hot	High	Strong
3	Overcast	Hot	High	Light
4	Rain	Mild	High	Light
5	Rain	Cool	Normal	Light
6	Rain	Cool	Normal	Strong
7	Overcast	Cool	Normal	Strong
8	Sunny	Mild	High	Light
9	Sunny	Cool	Normal	Light



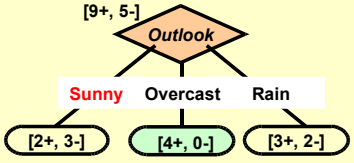


- Rozkład apriori: 9+, 5-
- Gain(T, Humidity) = 0.151
- Gain (T, Wind) = 0.048
- Gain(D, Temperature) = 0.029
- Gain(D, Outlook) = **0.246**



Następny wierzchołek

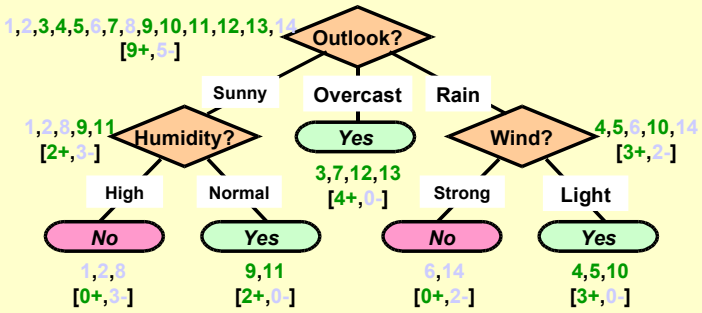
Day	Outlook	Temperature	Humidity	Wind
1	Sunny	Hot	High	Light
2	Sunny	Hot	High	Strong
3	Overcast	Hot	High	Light
4	Rain	Mild	High	Light
5	Rain	Cool	Normal	Light
6	Rain	Cool	Normal	Strong
7	Overcast	Cool	Normal	Strong
8	Sunny	Mild	High	Light
9	Sunny	Cool	Normal	Light
10	Rain	Mild	Normal	Light



- Gain($T_{\text{Outlook Sunny}}$, Humidity) = 0.97 - (3/5) * 0 - (2/5) * 0 = **0.97**
- Gain($T_{\text{Outlook Sunny}}$, Wind) = 0.97 - (2/5) * 1 - (3/5) * 0.92 = 0.02
- Gain($T_{\text{Outlook Sunny}}$, Temperature) = 0.57

Całe drzewo

Day	Outlook	Temperature	Humidity	Wind
1	Sunny	Hot	High	Light
2	Sunny	Hot	High	Strong
3	Overcast	Hot	High	Light
4	Rain	Mild	High	Light
5	Rain	Cool	Normal	Light
6	Rain	Cool	Normal	Strong
7	Overcast	Cool	Normal	Strong
8	Sunny	Mild	High	Light
9	Sunny	Cool	Normal	Light



Cechy ID3

- H – przestrzeń zupełna
- Zwracanie (przechowywanie) tylko jednej hipotezy (częściowej)
- Brak nawracania (optimum lokalne) – rozwiązanie – przycinanie drzewa

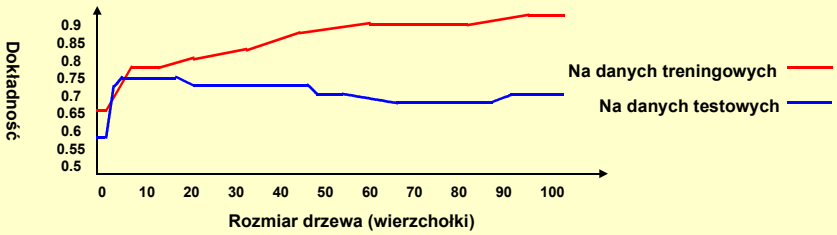
Porównanie obciążenia

- ID3
  - H kompletna
  - przeszukiwanie niekompletne
  - obciążenie preferencji
- Algorytm eliminacji kandydatów
  - H niekompletna
  - przeszukiwanie kompletne
  - obciążenie reprezentacji

Kierunki rozbudowy ID3

- Unikanie nadmiernego dopasowania
- Atrybuty ciągłe
- Modyfikacja kryterium wyboru atrybutu
- Dane z brakującymi wartościami
- Uwzględnienie kosztu pomiaru atrybutów
- ID3 + powyższe modyfikacje = C4.5

Unikanie nadmiernego dopasowania



- Przyczyna: szum powoduje błędy w generalizacji
- Rozwiązania:
  - Wcześniej przestać budować drzewo (ale kiedy?)
  - Zbudować duże, a potem je przyciąć (częściej stosowane)

Reduced-Error-Pruning

- Metoda przycinania z osobnym zbiorem do walidacji V
- Reduced-Error-Pruning (T)
  Podziel T na T_1 i V
  t = ID3(T_1, A)
  WHILE poprawność t na V się nie zmniejsza DO
   FOR each n – nie liścia t
      ttemp[n] = Prune (T, n)
      accuracy[n] = Test (ttemp[n], V)
   t = ttemp[arg $\max_n$ (accuracy[n])]
  RETURN t
- Prune(t, *node*)
  Zastąp poddrzewo t liściem o najczęstszej etykiecie

Rule Post-Pruning (C4.5)

- Rule-Post-Pruning (D)
  - t = ID3(D, A)
  - Przekształć t w zbiór reguł
  - Przytnij (uogólnij) każdą regułę oddzielnie przez usunięcie przesłanek, które powoduje wzrost estymowanej poprawności (por. literatura)
  - Posortuj przycięte reguły wg estymowanej poprawności

Atrybuty ciągłe

- Dwie metody
  - Dyskretyzacja
    - Podział wartości na przedziały przed uczeniem
    - {high = Temp > 35º C, med = 10º C < Temp ≤ 35º C, low = Temp ≤ 10º C}
  - Testy nierównościowe
    - $a_i \leq v$ tworzy dwa podzbiory A_i
    - Przyrost informacji obliczany jest tak samo jak dla testów tożsamościowych

Algorytm znajdowania punktu podziału

- Znajdź test nierównościowy (T, a_i): Test
  - L = uporządkowana lista wartości a_i występujących w T
  - FOR each (l, u) – pary sąsiednich wartości z L z różnymi etykietami
    oblicz *Gain* dla testu $a_i \leq (l+u)/2$?
    RETURN test o najwyższym *Gain*
- Przykład
  - a_i = *Length*: 10 15 21 28 32 40 50
  - *Class*: - + + - + + -
  - Sprawdź testy:
    Length ≤ 12.5?, *Length* ≤ 24.5?, *Length* ≤ 30?, *Length* ≤ 45?

Współczynnik przyrostu informacji

- Problem: Jeśli atrybut ma dużo wartości, to funkcja Gain go faworyzuje
- Rozwiązanie: GainRatio

$$\text{Gain}(D,a_i) = -\text{Entropia}(D) - \sum_{v \in A_i} \left[\frac{|D_{a_i v}|}{|D|} \text{Entropia}(D_{a_i v}) \right]$$

$$\text{GainRatio}(D,a_i) = \frac{\text{Gain}(D,a_i)}{\text{SplitInformation}(D,a_i)}$$

$$\text{SplitInformation}(D,a_i) = -\sum_{v \in A_i} \left[\frac{|D_{a_i v}|}{|D|} log_2 \frac{|D_{a_i v}|}{|D|} \right]$$

- SplitInformation rośnie z $|A_i|$ więc atrybuty o wielu wartościach są karane

Koszty testów

- Zastosowania w praktyce – np. medycyna:
  - Temperatura kosztuje 1zł, test krwi 70zł, RM 500zł
  - Inwazyjność badań
  - Ryzyko dla pacjenta
  - Czas wykonywania pomiaru
- Jak tworzyć drzewa z niskim oczekiwanym kosztem testów?
- Zamiana Gain na Cost-Normalized-Gain
  - [Nunez, 1988]:

$$\text{Cost-Normalized-Gain}(D,A) = \frac{\text{Gain}^2(D,A)}{\text{Cost}(D,A)}$$

  - [Tan and Schlimmer, 1990]:

$$\text{Cost-Normalized-Gain}(D,A) = \frac{2^{\text{Gain}(D,A)}-1}{(\text{Cost}(D,A)+1)^w} \quad w \in [0,1]$$

w – waga kosztu

Brakujące wartości atrybutów

- Czasem wartości nie znane, czasem zbyt drogie
- Podczas uczenia – trzeba obliczyć Gain(T, a_i) w przypadku gdy dla pewnych przykładów nie znamy wartości a_i
- Podczas używania klasyfikatora
  - Którą krawędź wybrać?
  - Metoda analogiczna do uczenia

Metody uczenia z przykładów z brakującymi danymi

- Pomijanie przykładów
- Redukcja – *Gain* liczony dla przykładów o znanych wartościach, wymnażany przez stosunek znanych do wszystkich
- Wypełnianie
  - najczęściej występującą wartością
  - najczęściej występującą wartością w przykładach o tej samej etykiecie
  - wartością obliczoną z innych atrybutów
  - wylosowana wartość
- Podział – zastąpienie przez przykłady o różnych wartościach i częstościach będących popularnością tych wartości (przykłady bez wartości nieznanych mają częstość = 1)
- Oddzielna gałąź

Złożoność

- Testy tożsamościowe, n atrybutów nominalnych
- Operacje:
  - wyznaczenie rozkładów: $O(n|T|)$
  - ocena jakości testów: $O(n|C|)$
  - podział na podzbiory: $O(n|T|)$
- Koszt całkowity tworzenia wierzchołka:

$$O(n|T|)$$

- Bardzo dobry stosunek kosztu do jakości

Podsumowanie

- Drzewo a reguły
- Wady i zalety drzew
- Algorytm ID3
- Przyrost informacji
- Obciążenie ID3
- Różnice pomiędzy ID3 i C4.5
- C4.5 – bardzo popularny (w Weka: J48)
- Następny wykład – uczenie reguł

Slajdy przygotowano na podstawie

1. P. Cichosz, Systemy uczące się, WNT, Warszawa, 2000.
2. William H. Hsu, Slajdy (stąd skopiowano większość przykładów).