Drzewa decyzyjne

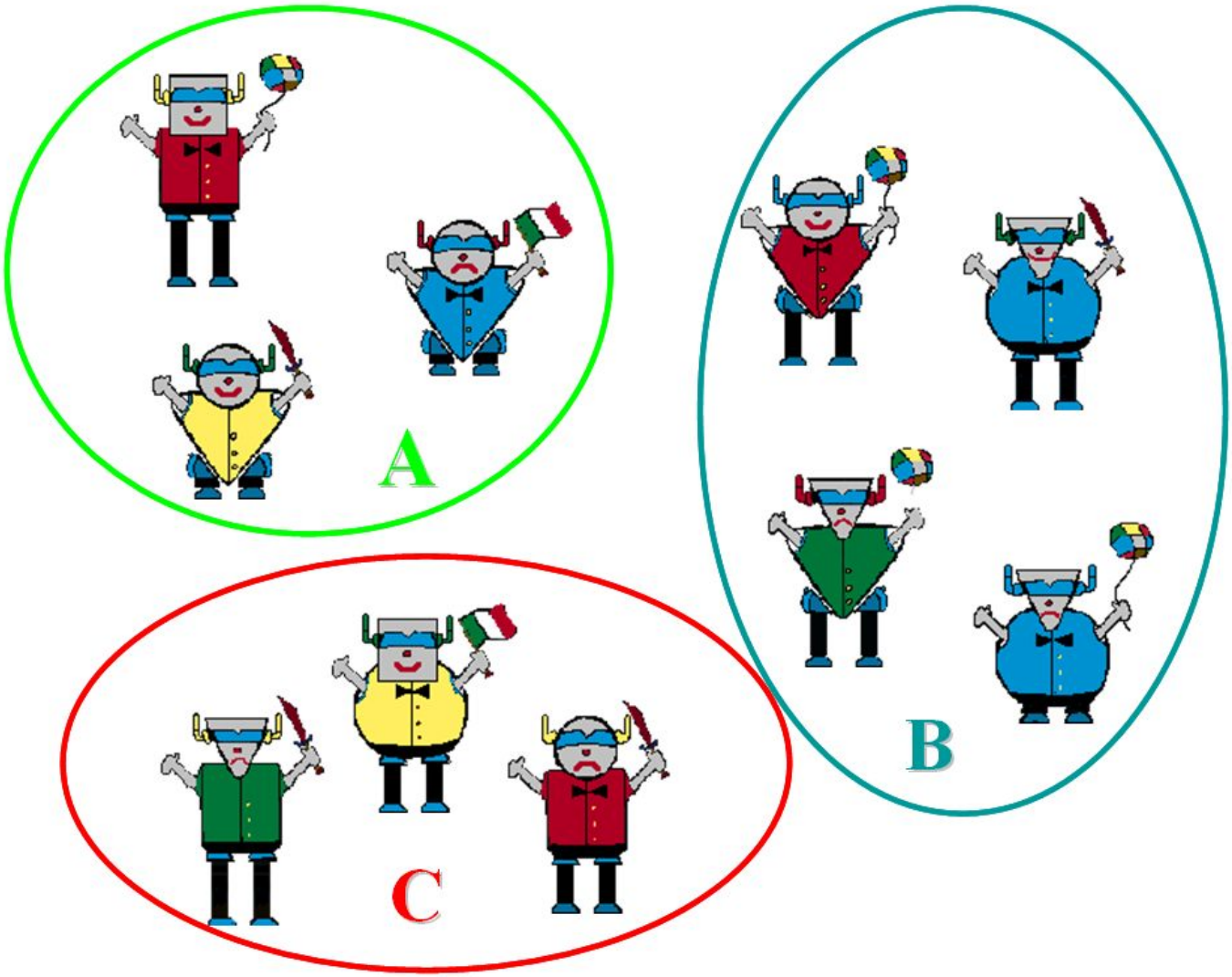
Nguyen Hung Son

Outline

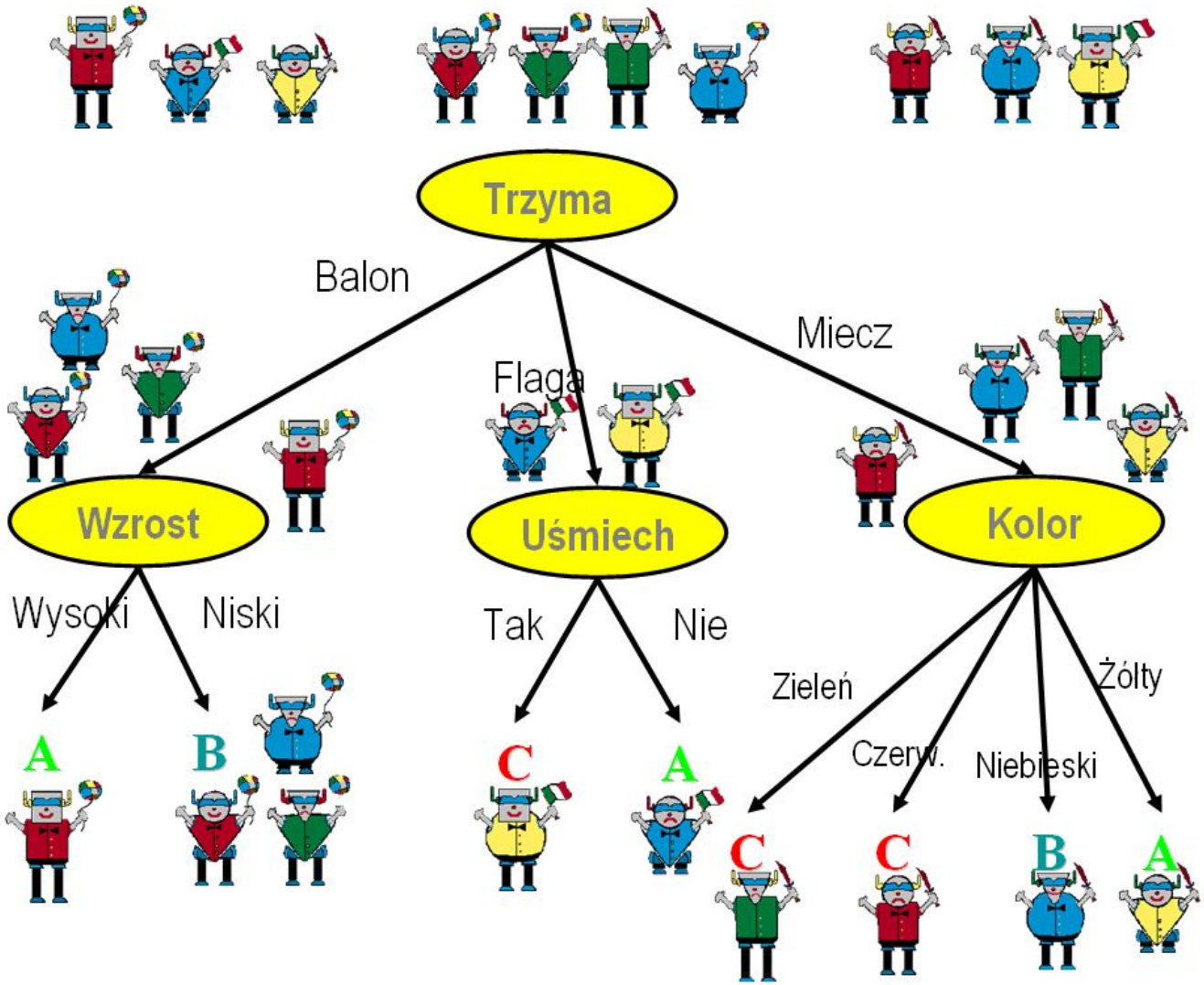
Co to jest drzewo decyzyjne

- **Jest to struktura drzewiasta, w której**
  - **węzły wewnętrzne** zawierają testy na wartościach atrybutów
  - z każdego węzła wewnętrznego wychodzi tyle **gałęzi**, ile jest możliwych wyników testu w tym węźle;
  - **liście** zawierają decyzje o klasyfikacji obiektów
- **Drzewo decyzyjne koduje program zawierający same instrukcje warunkowe**

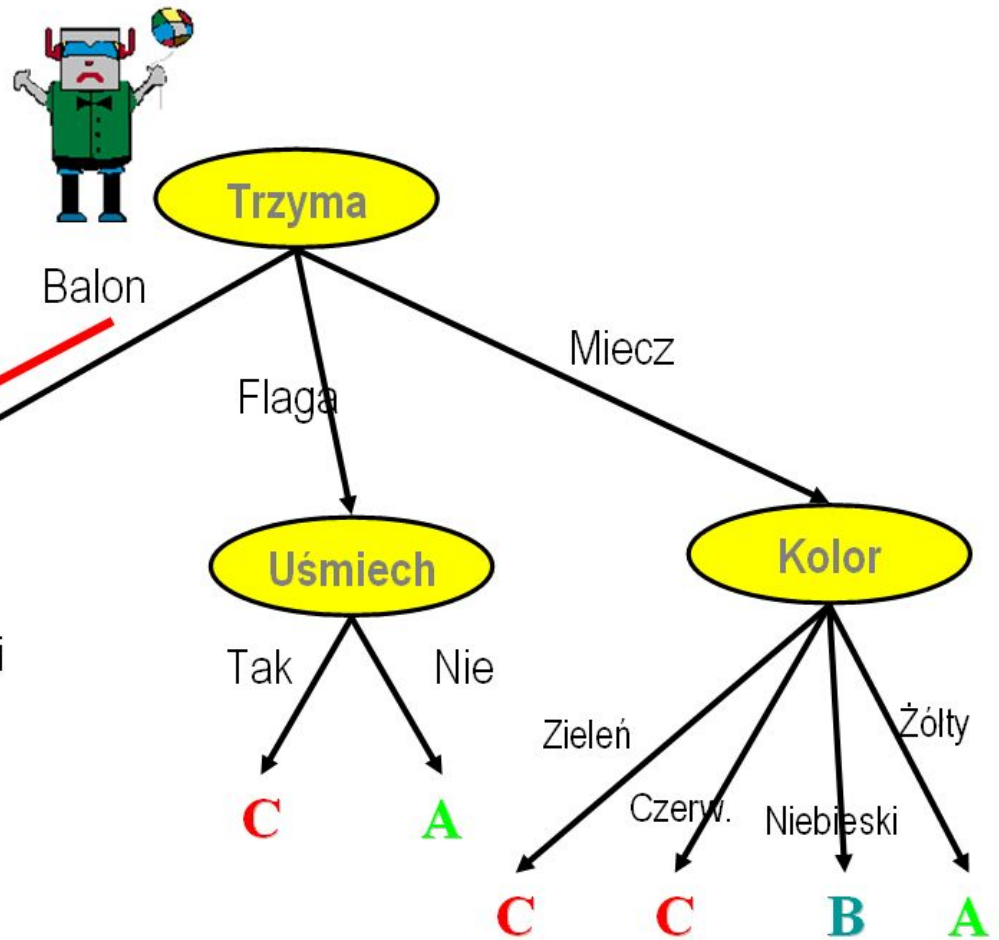
Przykład: klasyfikacja robotów



Przykład: drzewo decyzyjne



Klasyfikacja drzewem decyzyjnym



Outline

Przykład tablicy decyzyjnej

x	outlook	Temperature	humidity	wind	play(x)
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cold	normal	weak	yes
6	rain	cold	normal	strong	no
7	overcast	cold	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cold	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no

Rodzaje testów

Wyróżniamy 2 klasy funkcji testów

- Testy operują się na wartościach pojedyńczego atrybutu (univariate tree):

$$t : V_a \rightarrow R_t$$

- Testy będące kombinacją wartości kilku atrybutów (multivariate tree).

$$t : V_{a_1} \times V_{a_2} \times ... \times V_{a_k} \rightarrow R_t$$

gdzie

- Va : dziedzina atrybutu a
- Rt : zbiór możliwych wyników testu

Przykłady funkcji testu

- Dla atrybutów nominalnych a_i oraz obiekt x:
  - test tożsamościowy: $t(x) \rightarrow a_i(x)$
  - test równościowy: $t(x) = \begin{cases} 1 & \text{if } (a_i(x) = v) \\ 0 & \text{otherwise} \end{cases}$
  - test przynależnościowy: $t(x) = \begin{cases} 1 & \text{if } (a_i(x) \in V) \\ 0 & \text{otherwise} \end{cases}$
- Dla atrybutów o wartościach ciągłych:
  - test nierównościowy:

    $t(x) = \begin{cases} 1 & \text{if } (a_i(x) > c) \\ 0 & \text{otherwise, i.e., } (a_i(x) \leq c) \end{cases}$ gdzie c jest wartością progową lub cięciem

Outline

Ocena jakości drzewa

- Jakość drzewa ocenia się
  - rozmiarem: im drzewo jest mniejsze, tym lepsze
    - mała liczba węzłów,
    - mała wysokość, lub
    - mała liczba liści;
  - dokładnością klasyfikacji na zbiorze treningowym
  - dokładnością klasyfikacji na zbiorze testowym
- Na przykład:

$$\boldsymbol{Q}(\boldsymbol{T}) = \alpha \cdot \boldsymbol{size}(\boldsymbol{T}) + \beta \cdot \boldsymbol{accuracy}(\boldsymbol{T}, \boldsymbol{P})$$

gdzie α, β są liczbami rzeczywistymi
size(.) jest rozmiarem drzewa
accuracy(.,.) jest jakością klasyfikacji

Problem konstrukcji drzew optymalnych:

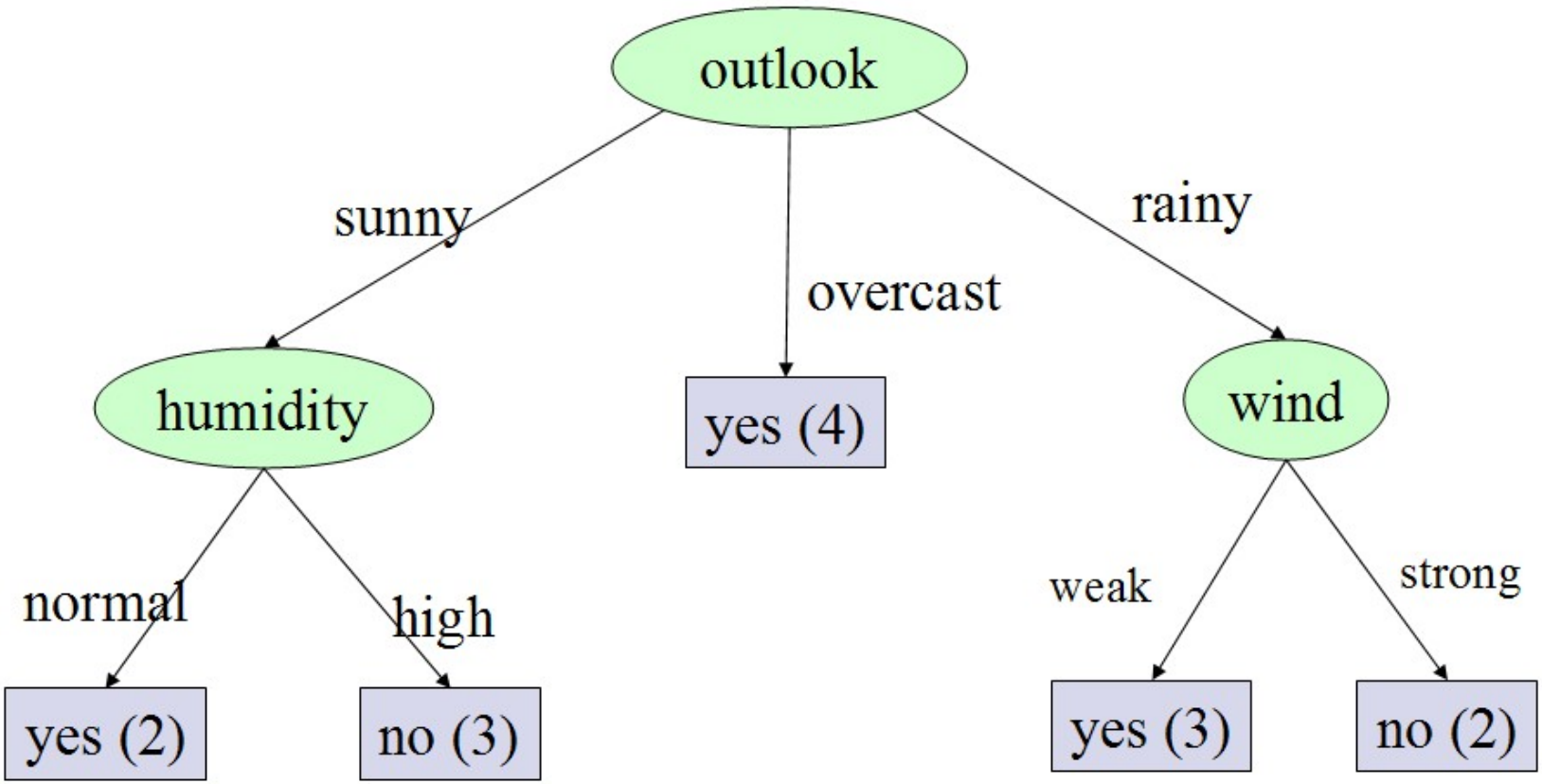
Dane są:

- tablica decyzyjna $\mathbb{S}$
- zbiór funkcji testów $\mathbf{TEST}$,
- kryterium jakości Q

Szukane: drzewo decyzyjne $\mathbf{T}$ o najwyższej jakości $Q(\mathbf{T})$.

- Dla większości parametrów, problem szukania optymalnego drzewa jest NP-trudny !
- **Wnioski:**
  Trudno znaleźć optymalne drzewo w czasie wielomianowym;
  Konieczność projektowania heurystyk.
- **Quiz:** Czy drzewo z przykładu jest optymalne?

Optymalne drzewo decyzyjne



Outline

Funkcja rekurencyjna $buduj\_drzewo(U, dec, \mathbf{T})$:

1: **if** $(kryterium\_stopu(U, dec) = \mathbf{true})$ **then**
2: $\quad \mathbf{T}.etykieta = kategoria(U, dec)$;
3: $\quad$ **return**;
4: **end if**
5: $t := wybierz\_test(U, \mathbf{TEST})$;
6: $\mathbf{T}.test := t$;
7: **for** $v \in R_t$ **do**
8: $\quad U_v := \{x \in U : t(x) = v\}$;
9: $\quad$ utwórz nowe poddrzewo $\mathbf{T}'$;
10: $\quad \mathbf{T}.$gałąź$(v) = \mathbf{T}'$;
11: $\quad buduj\_drzewo(U_v, dec, \mathbf{T}')$
12: **end for**

Funkcje pomocnicze

- Kryterium stopu: Zatrzymamy konstrukcji drzewa, gdy aktualny zbiór obiektów:
  - jest pusty lub
  - zawiera obiekty wyłącznie jednej klasy decyzyjnej lub
  - nie ulega podziale przez żaden test
- Wyznaczenie etykiety zasadą większościową:

$$kategoria(P, dec) = \arg\max_{c \in V_{dec}} |P_{[dec=c]}|$$

tzn., etykietą dla danego zbioru obiektów jest klasa decyzyjna najliczniej reprezentowana w tym zbiorze.

- Kryterium wyboru testu: heurytyczna funkcja oceniająca testy.

Outline

Miary różnorodności zbioru

Każdy zbiór obiektów X ulega podziale na klasy decyzyjne:

$$X = C_1 \cup C_2 \cup ... \cup C_d$$

gdzie $C_i = \{u \in X : dec(u) = i\}$.

Wektor $(p_1, ..., p_r)$, gdzie $p_i = \frac{|C_i|}{|X|}$, nazywamy rozkładem klas decyzyjnych w X.

$$Conflict(X) = \sum_{i<j} |C_i| \times |C_j| = \frac{1}{2}\left(|X|^2 - \sum |C_i|^2\right)$$

$$Entropy(X) = -\sum \frac{|C_i|}{|X|} \cdot \log \frac{|C_i|}{|X|}$$

$$= -\sum p_i \log p_i$$

Własności miar róznorodności

Funkcja $conflict(X)$ oraz $Ent(X)$ przyjmują

- największą wartość, gdy rozkład klas decyzyjnych w zbiorze X jest równomierny.
- najmniejszą wartość, gdy wszystkie obiekty w X są jednej kategorii (X jest **jednorodny**)

W przypadku 2 klas decyzyjnych:

$$Conflict(p, 1-p) = |X|^2 \cdot p(1-p)$$
$$Entropy(p, 1-p) = -p \log p - (1-p) \log(1-p)$$

Kryteria wyboru testu

Niech t definiuje podział X na podzbiory: $X_1 \cup ... \cup X_r$. Możemy stosować następujące miary do oceniania testów:

- liczba par obiektów rozróżnionych przez test t.

$$disc(t,X) = conflict(X) - \sum conflict(X_i)$$

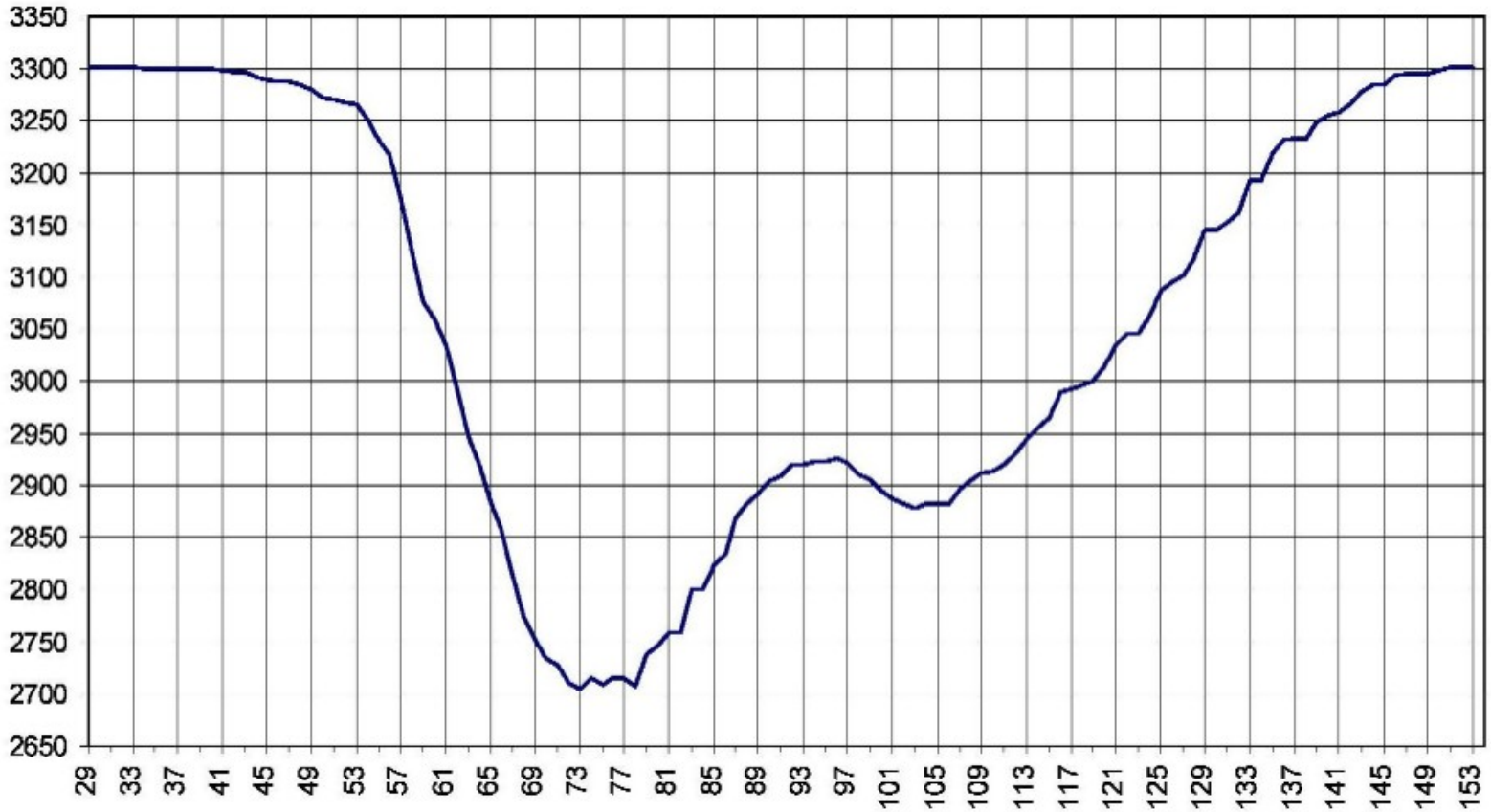
- kryterium przyrostu informacji (ang. Inf. gain).

$$Gain(t,X) = Entropy(X) - \sum_i p_i \cdot Entropy(X_i)$$

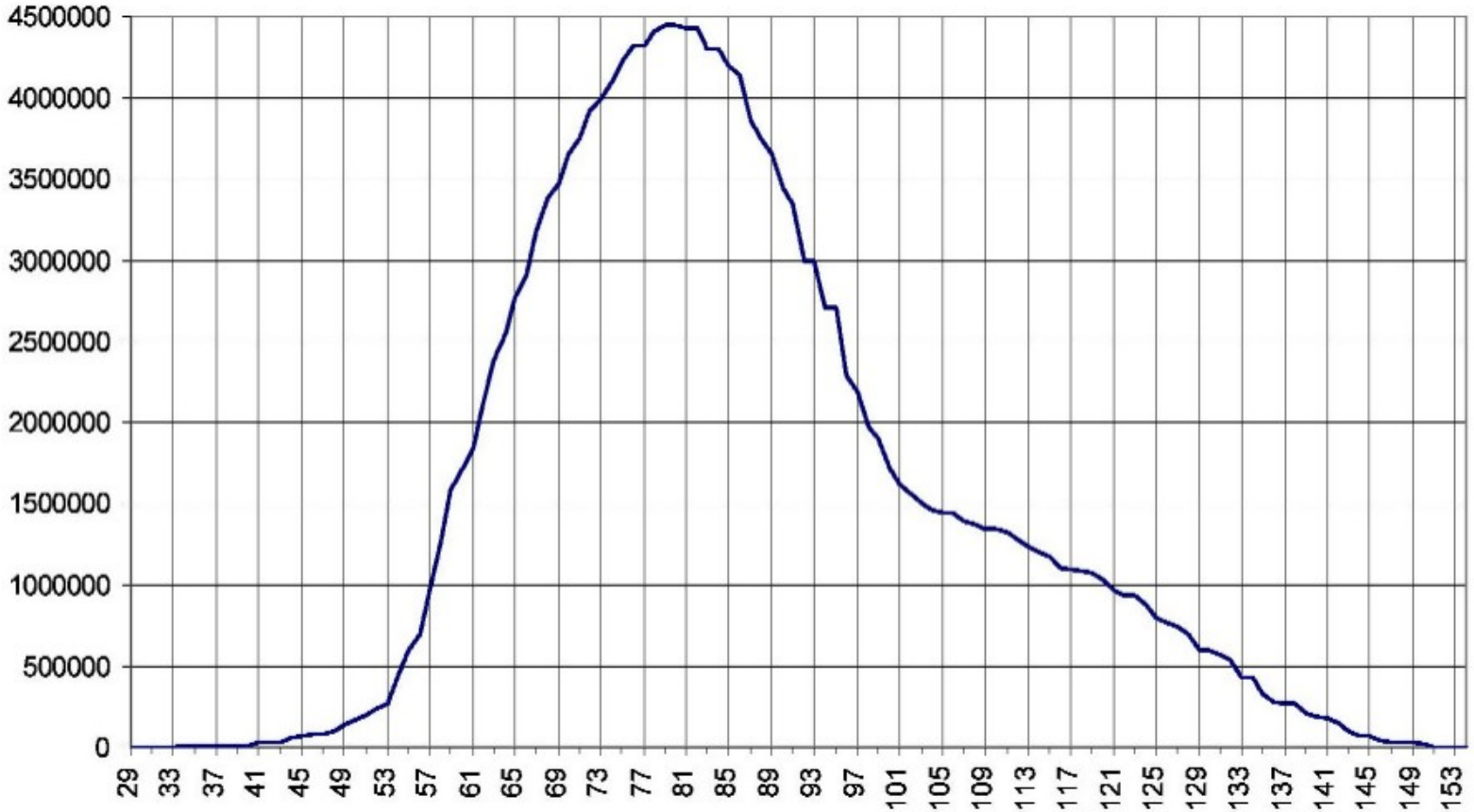
Im większe są wartości tych ocen, tym lepszy jest test.

Miara Entropii dla cięć

$$N \times \sum_i p_i \cdot Entropy(X_i)$$



Rozróżnialność dla cięć



Własności funkcji ocen:

- Monotoniczność: Jeśli t' definiuje drobniejszy podział niż t to

$$Gain(t', X) \geq Gain(t, X)$$

(analogiczną sytuację mamy dla miary $conflict()$.

- Funkcje ocen testu t przyjmują małe wartości jeśli rozkłady decyzyjne w podzbiorach wyznaczanych przez t są zbliżone.

Uniwersalne oceny

Zamiast bezwzględnego przyrostu informacji, stosujemy współczynnik przyrostu informacji

$$Gain\_ratio = \frac{Gain(t,X)}{iv(t,X)}$$

gdzie $iv(t,X)$, zwana wartością informacyjną testu t (information value), jest definiowana jak nast.:

$$iv(t,X) = -\sum_{i=1}^{r} \frac{|X_i|}{|X|} \cdot \log \frac{|X_i|}{|X|}$$

Ocena funkcji testu

- Rozróżnialność:

$$disc(t,X) = conflict(X) - \sum conflict(X_i)$$

- Przyrostu informacji (Information gain).

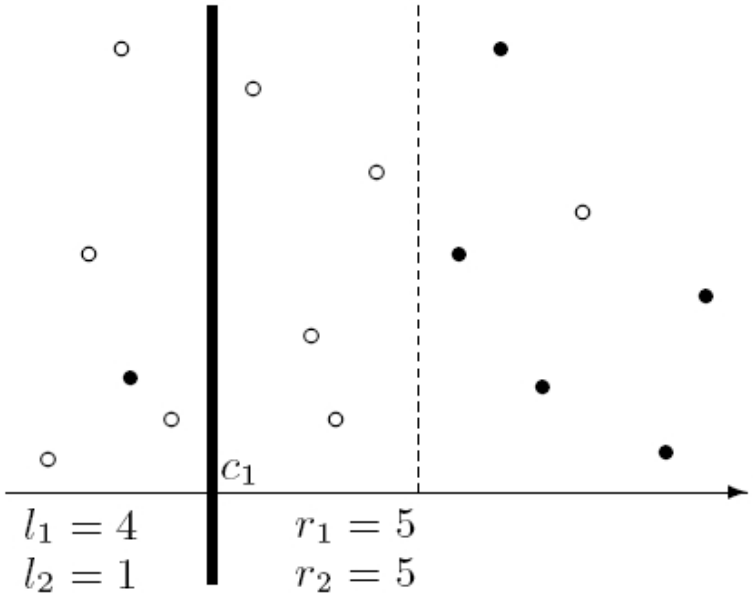
$$Gain(t,X) = Entropy(X) - \sum_i p_i \cdot Entropy(X_i)$$

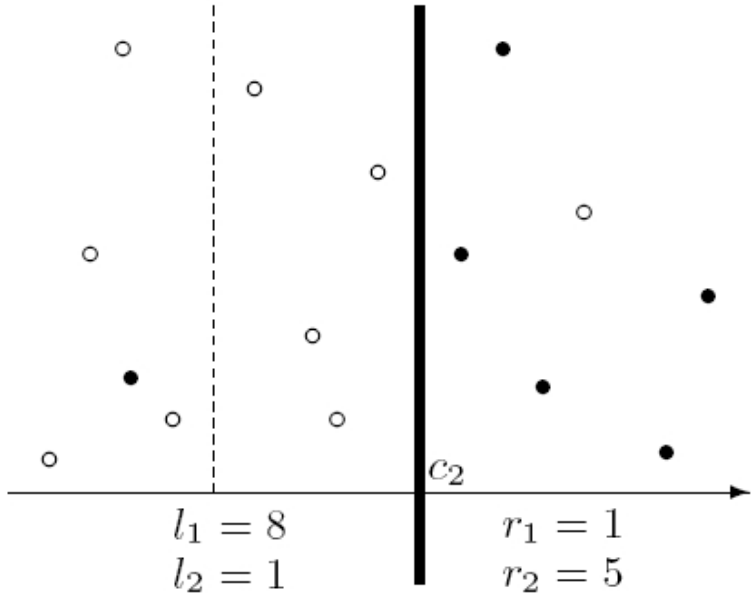
- Współczynnik przyrostu informacji (gain ratio)

$$Gain\_ratio = \frac{Gain(t,X)}{-\sum_{i=1}^{r} \frac{|X_i|}{|X|} \cdot \log \frac{|X_i|}{|X|}}$$

- Inne (np. Gini's index, test χ^2, ...)

Własności funkcji ocen: Cięcia brzegowe





Twierdzenie:

Miary $disc, Gain, Gini$ wybierają najlepsze cięcia wśród cięć brzegowych

Szkic dowodu: Wystarczy pokazać, że jeśli jakieś cięcie nie jest "brzegowe", to jeden z sąsiednich cięć okaże się lepsze.

Własności funkcji ocen: miara $disc$

Twierdzenie:

Jeśli dana tablica decyzyjna ma 2 klasy decyzyjne, i atrybut a jest funkcją różnowartościową na zbiorze obiektów:

- najlepsze cięcie według miary $disc$ rozwiązuje co najmniej połowę konfliktów.
- wysokość drzewa decyzyjnego nie przekracza $2(\log n - 1)$.

Szkic dowodu: Wystarczy korzystać z faktu, że najlepsze cięcie jest "brzegowe", następnie pokazać pewną "prostą" nierówność.

Outline

Przycinanie drzew

- Problem nadmiernego dopasowania do danych trenujących (prob. przeuczenia się).
- Rozwiązanie:
  - zasada krótkiego opisu: skracamy opis kosztem dokładności klasyfikacji w zbiorze treningowym
  - zastąpienie podrzewa nowym liściem (przycinanie) lub mniejszym podrzewem.
- Podstawowe pytania:
  - Q: Kiedy poddrzewo może być zastąpione liściem?
  - A: jeśli nowy liść jest niegorszy niż istniejące poddrzewo dla nowych obiektów (nienależących do zbioru treningowego).
  - Q: Jak to sprawdzić?
  - A: testujemy na próbce zwanej „zbiorem przycinania”!

Ogólny schemat algorytmu przycinania

Funkcja $przytnij(\mathbf{T}, P)$

1: **for all** $n \in \mathbf{T}$ **do**
2: utwórz nowy liść l etykietowany kategorią dominującą w zbiorze P_n
3: **if** (liść l jest niegorszy od poddrzewa o korzeniu w n pod względem zbioru P) **then**
4: zastąp poddrzewo o korzeniu w n liściem l;
5: **end if**
6: **end for**
7: return $\mathbf{T}$

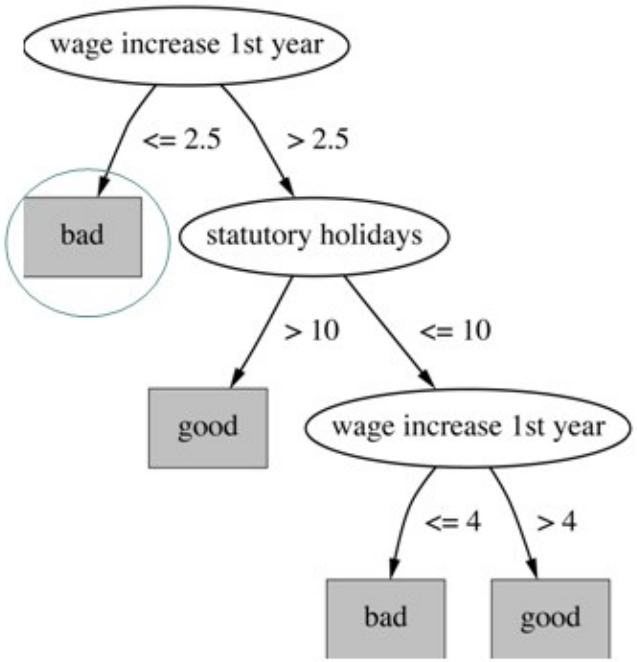
Kryterium przycinania

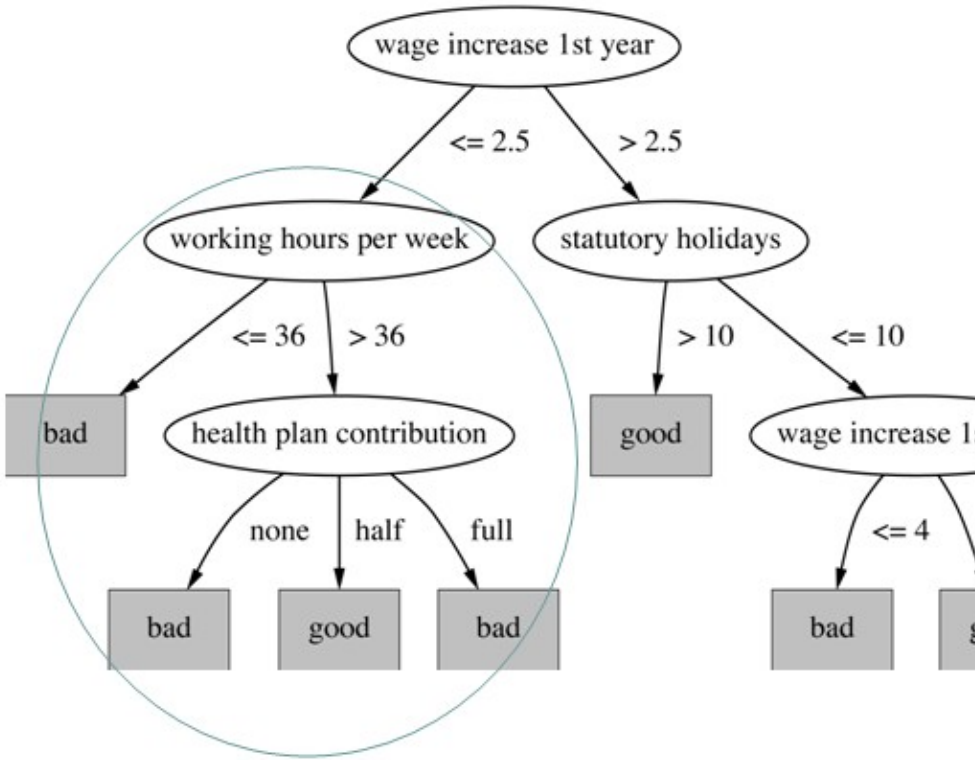
- Niech
  $e_T(l)$ - błąd klasyfikacji kandydującego liścia l,
  $e_T(n)$ - błąd klasyfikacji poddrzewa o korzeniu w n.
- Przycinanie ma miejsce, gdy

$$e_T(l) \leq e_T(n) + \mu \sqrt{\frac{e_T(n)(1 - e_T(n))}{|P_{T,n}|}}$$

na ogół przyjmujemy $\mu = 1$.

Przykład


wage increase 1st year
<= 2.5
> 2.5
bad
statutory holidays
> 10
<= 10
good
wage increase 1st year
<= 4
> 4
bad
good


wage increase 1st year
<= 2.5
> 2.5
working hours per week
statutory holidays
<= 36
> 36
> 10
<= 10
bad
health plan contribution
good
wage increase 1
none
half
full
<= 4
bad
good
bad
bad

Outline

Brakuje danych podczas uczenia się

Możliwe są następujące rozwiązania:

- Zredukowanie wartości kryterium wyboru testu (np. przyrostu informacji) dla danego testu o współczynnik równy:

$$\frac{\text{liczba obiektów z nieznanymi wartościami}}{\text{liczba wszystkich obiektów}}$$

- Wypełnienie nieznanych wartości atrybutu najczęściej występującą wartością w zbiorze obiektów związanych z aktualnym węzłem
- Wypełnienie nieznanych wartości atrybutu średnią ważoną wyznaczoną na jego zbiorze wartości.

Brakuje danych podczas klasyfikowania

Możliwe rozwiązania:

- Zatrzymanie procesu klasyfikacji w aktualnym węźle i zwrócenie większościowej etykiety dla tego węzła (etykiety, jaką ma największą liczbę obiektów trenujących w tym węźle)
- Wypełnienie nieznanej wartości według jednej z heurystyk podanych wyżej dla przypadku konstruowania drzewa
- Uwzględnienie wszystkich gałęzi (wszystkich możliwych wyników testu) i połączenie odpowiednio zważonych probabilistycznie rezultatatów w rozkład prawdopodobieństwa na zbiorze możliwych klas decyzyjnych dla obiektu testowego.