



SYSTEMY UCZĄCE SIĘ

Indukcja drzew decyzyjnych

Halina Kwaśnicka

Politechnika Wrocławska

halina.kwasnicka@pwr.edu.pl



Maszynowe uczenie się, systemy uczące się, ...

- **Dziedzina *Machine Learning* (ML):**
- Szukanie odpowiedzi na pytanie, w jaki sposób konstruować programy komputerowe, które potrafią automatycznie polepszać swoje działanie w miarę 'doświadczenia'.
- Jest interdyscyplinarna (statystyka, sztuczna inteligencja, filozofia, teoria informacji, biologia, i in.)
- **Definicja:**
Mówimy, że program komputerowy jest zdolny do automatycznego uczenia się, z doświadczeń E , według miary jego działania P , zadań należących do klasy T , jeśli skutek doświadczenia E rozwiązuje lepiej zadania z klasy T , wg miary P .

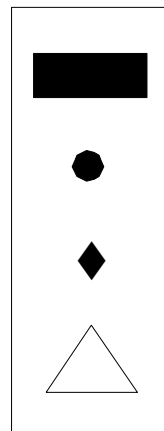
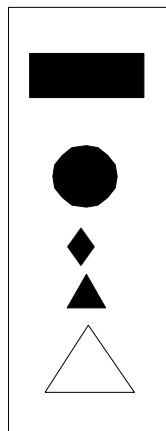
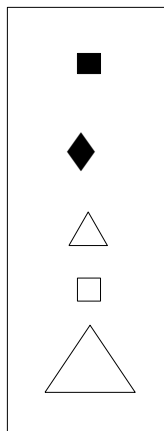


Przykłady systemów uczących się

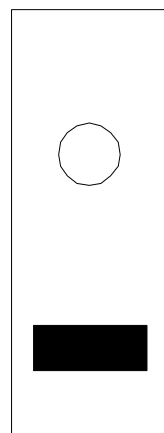
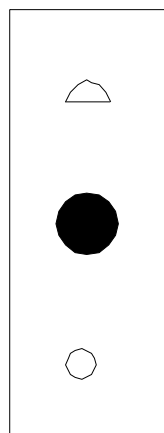
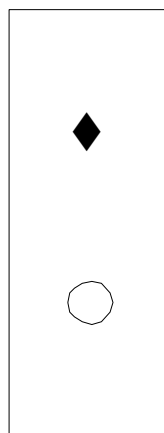
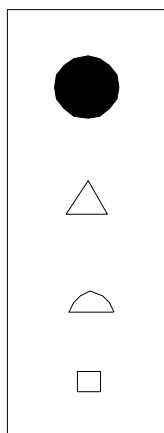
- Problem gry w szachy
 - Klasa zadań T : gra w szachy
 - Miara działania P : procent wygranych gier
 - Doświadczenia uczące E : praktyczna gra z przeciwnikami
- Problem sterowania pojazdem
 - Klasa zadań T : poruszanie się po publicznych drogach w czterech kierunkach używając sensorów wizyjnych
 - Miara działania P : średnia długość drogi przebyta bez błędu
 - Doświadczenia uczące E : sekwencja obrazów i poleceń jakie stosuje człowiek (zebrane podczas obserwacji kierowcy)

Uczenie indukcyjne – co to jest?

Klasa 1



Klasa 2





Możliwy efekt indukcji

- **Jeśli** prostokąt zawiera mały czarny kształt **to** jest to klasa 1.
- **Jeśli** prostokąt zawiera pięć figur **lub** małe czarne kółko **to** jest to klasa 1.
- **Jeśli** prostokąt zawiera gwiazdę nad trójkątem **to** jest to klasa 1.
- **Jeśli** prostokąt zawiera gwiazdę **i** nie jest ona na samej górze **to** jest to klasa 1.
- **Jeśli** prostokąt zawiera dwie figury **lub** jasny księżyc **to** jest to klasa 2.
- **Jeśli** prostokąt zawiera jasne kółko **lub** jasny księżyc **to** jest to klasa 2.
- **Jeśli** prostokąt zawiera duży trójkąt na dole **to** jest to klasa 1.



Wstęp – przypomnienie: dedukcja

- Wydobywanie wiedzy (*knowledge elicitation*) – ważny i trudny problem
- Uczenie się maszyn (*machine induction*) – kontrowersje odnośnie jego roli
- Dedukcja – wnioskowanie od ogółu do szczegółu (top-down)
- Mamy: Wszystkie dzieci rozpoczynają szkołę tak szybko jak to możliwe po skończeniu siedmiu lat
- Wiemy, że: Ala jest w szkole
- Dedukujemy: Ala ma przynajmniej siedem lat.
- Wiemy: Olek ma pięć lat, Dedukujemy: Olek nie jest w szkole.



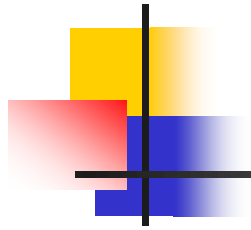
Wstęp – indukcja

- Indukcja to podejście z dołu do góry (bottom-up)
- Mamy do dyspozycji zbiór przykładów
- Indukujemy reguły opisujące przykłady
- Mamy:
 - Ala chodzi do szkoły, Ala ma 8 lat
 - Olek nie chodzi do szkoły, Olek ma pięć lat
 - Basia chodzi do szkoły, Basia ma 10 lat
- Indukujemy: Wszystkie dzieci, które mają przynajmniej osiem lat chodzą do szkoły



Maszynowe uczenie – definiowanie

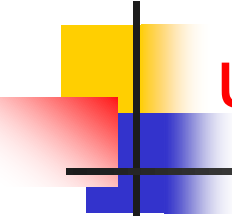
- Kluczowy problem ML: indukcja ogólnych funkcji ze specyficznych przykładów uczących
- Uczenie pojęć (***concept learning***): pozyskanie definicji ogólnej kategorii mając próbkę pozytywnych i negatywnych przykładów
- Uczenie pojęć: przeszukiwanie przestrzeni predefiniowanych hipotez w celu znalezienia hipotez najbardziej pasujących do przykładów uczących.
- Uczenie pojęć: aproksymacja wartości funkcji logicznej z przykładów jako należących bądź nie należących do uczonego pojęcia
- Uczenie pojęć: Wnioskowanie wartości funkcji logicznej z przykładów zawierających wejścia i wyjścia.





Podstawowe pojęcia

- Paradygmat uczenia:
 - *Uczenie indukcyjne* vs *Uczenie na podstawie przypadków (Case Based Reasoning)*
 - Przykłady ...
- Postać danych wejściowych:
- *Nadzorowane (z nauczycielem)* vs *Nienadzorowane uczenie* vs *Uczenie ze wzmocnieniem*
 - Główne zadania
 - Przykłady ...
 - <http://sysplan.nams.kyushu-u.ac.jp/gen/papers/JavaDemoML97/robodemo.html>
- Wybrane problemy z danymi



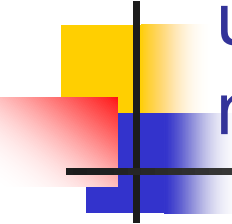
Główne rodzaje indukcyjnego uczenia się: **uczenie się pojęć** (uczenie z nadzorem)

- Forma indukcyjnego uczenia się zależy od
 - charakteru wiedzy, która ma być pozyskana
 - postaci informacji trenującej
- **Uczenie się pojęć**
 - Pojęcia służą do klasyfikacji obiektów na grupy (kategorie)
 - Podstawowa wersja: podział na dwie grupy – obiekty należące do danego pojęcia (pozytywne przykłady) i nie należące do niego (negatywne przykłady)
 - Pojęcia – funkcje przekształcające dziedzinę (obiekty) w zbiór kategorii: dwuelementowy dla pojęć pojedynczych i może mieć więcej pojęć w przypadku pojęć wielokrotnych (np. alfabet polski to jedno pojęcie o 33 kategoriach – literach)



Główne rodzaje indukcyjnego uczenia się: **tworzenie pojęć** (uczenie bez nadzoru)

- Nie zawsze kategorie pojęć są znane – uczeń obserwuje nieetykietowane przykłady (opisy obiektów)
- Uczeń, na podstawie obserwacji, grupuje obiekty w kategorie zgodnie z pewnymi kryteriami podobieństwa
- **Tworzenie pojęć** łączy dwa podzadania:
 - podział przykładów trenujących na grupy, które odpowiadają kategoriom
 - nauczanie się pojęć odpowiadających tworzonym kategoriom, aby było możliwe klasyfikowanie nowych przykładów



Główne rodzaje indukcyjnego uczenia się: uczenie się **aproksymacji funkcji** (uczenie z nadzorem)

- Uczenie pojęć – uczenie się funkcji odwzorowującej przykłady na skończony i niewielki zbiór kategorii
- Uczenie się **aproksymacji funkcji** – zbiór wartości uczonej funkcji to zbiór liczb rzeczywistych
- Przykłady – pary składające się z argumentu funkcji, reprezentowanego zazwyczaj przez wektor liczb rzeczywistych i jej wartości dla tego argumentu
- Uczeń (system) ma wygenerować funkcję dobrze przybliżającą funkcję docelową
- Wartość funkcji ma być obliczona z dużą dokładnością nie tylko dla przykładów, ale – indukcyjne uogólnienie – dla dowolnych innych argumentów z dziedziny



Obciążenie indukcyjne

- Obciążenie – właściwości algorytmu decydujące o wyborze hipotezy gdy wiedza wrodzona i informacja trenująca nie wyznaczają jej jednoznacznie
- **Definicja:**

Obciążeniem algorytmu indukcyjnego uczenia się nazywamy czynniki, które decydują o wyborze jednej hipotezy spośród zbioru hipotez dopuszczalnych ze względu na cel uczenia się



Dlaczego stosujemy zbiór trenujący i testowy

- Założenie, że zbiory przykładów są niezależne od hipotez jest niespełnione, bo hipotezy generowane są na podstawie przykładów
- Dlatego oddzielny zbiór trenujący i testowy (walidujący)
- Taki proces to *krzyżowa walidacja* hipotezy
- *k-krotna walidacja krzyżowa*
- Otrzymane błędy próbki są uśredniane



Spójna hipoteza

- Hipoteza h jest spójna z pojęciem c na zbiorze przykładów $P \subseteq X$, jeśli:

$$(\forall x \in P) h(x) = c(x)$$

- Pytanie: czy w przestrzeni hipotez \mathbf{H} może być wiele hipotez spójnych z docelowym pojęciem c ?



Postać hipotezy

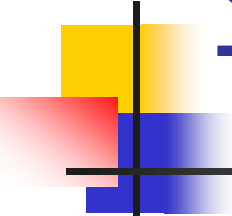
- Hipoteza: koniunkcja ograniczeń na wartości atrybutów przykładów (selektorów)
- Dla każdego atrybutu hipoteza:
 - wskazuje, że każda wartość danego atrybutu jest akceptowalna (np. symbol ?)
 - określa pojedynczą, dopuszczalną wartość atrybutu
 - wskazuje, że żadna wartość danego atrybutu nie jest akceptowalna (np. symbol ☐).
- Jeśli przykład spełnia ograniczenia hipotezy, jest on klasyfikowany jako należący do uczonego pojęcia.
- ma_4_kółka, jeździ_po_drogach, nie_ma_skrzydeł ☐ auto.



Hipoteza indukcyjnego uczenia

- Uczenie – dobranie takiej hipotezy, która najlepiej pokrywa zbiór uczący,
- *Indukcyjne algorytmy uczące* nie mogą gwarantować, że znaleziona hipoteza pokryje również przykłady spoza zbioru uczącego.
- Podstawowe założeniem to **Hipoteza indukcyjnego uczenia się**:

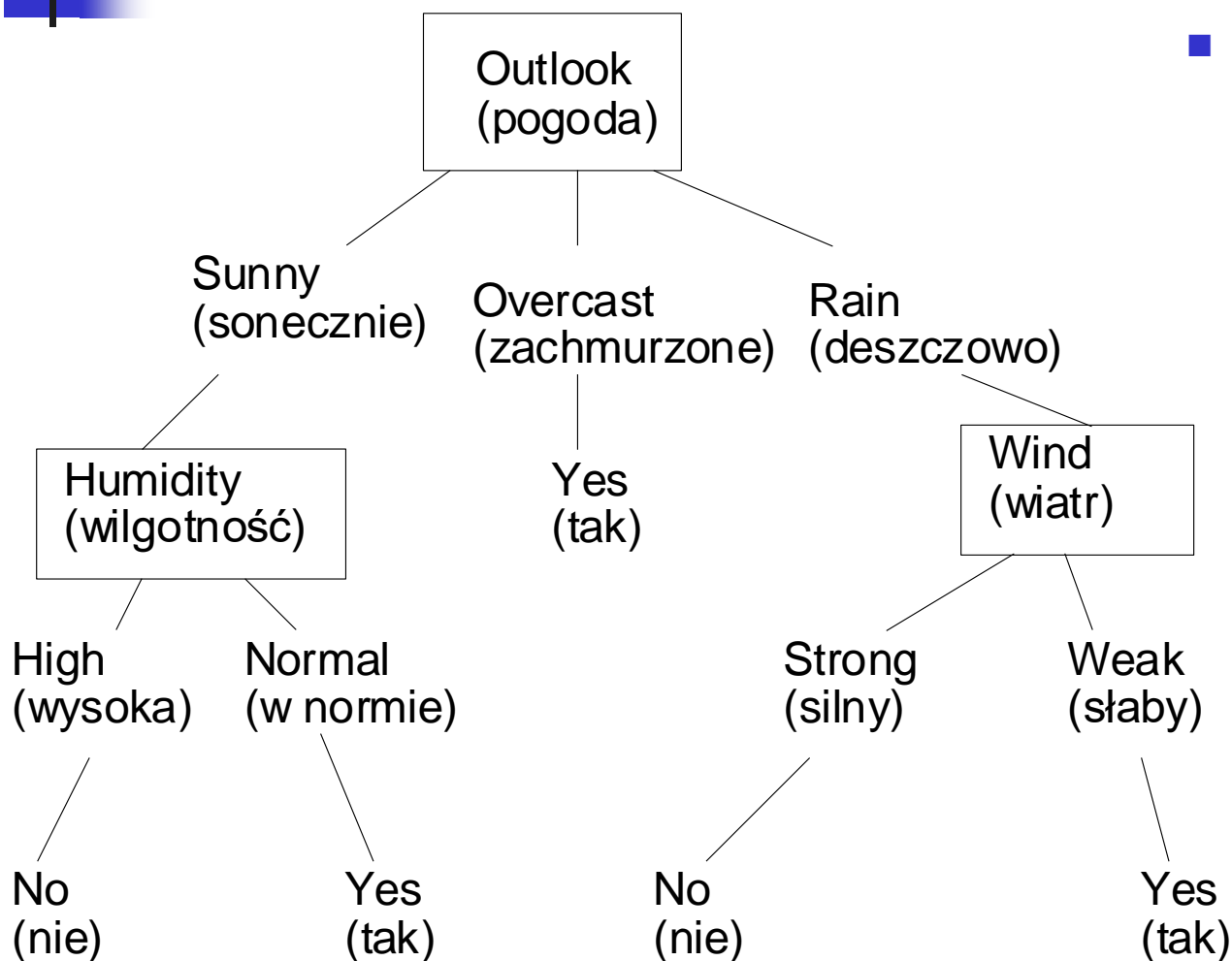
Każda znaleziona hipoteza dobrze aproksymująca docelową funkcję na dużej części zbioru uczącego będzie również aproksymować funkcję docelową dla nieobserwowanych przykładów.



Uczenie drzew decyzyjnych (**Decision Tree Learning**) – wprowadzenie

- Popularna metoda wnioskowania indukcyjnego
- Służy do aproksymacji docelowej funkcji o wartościach dyskretnych
- Uczona funkcja jest reprezentowana w postaci drzewa
- Nauczone drzewo może być przedstawione w postaci reguł *if-then* (czytelność)
- Odpowiednia dla zaszumionych danych
- Ma możliwości uczenia wyrażeń dysjunkcyjnych (alternatyw).
- Do tej rodziny należą ID3, ASSISTANT, C4.5.
- Ich obciążenie to preferencja małych drzew.

Przykład drzewa decyzyjnego



- Ogólnie, drzewo reprezentuje koniunkcje i dysjunkcje ograniczeń nałożonych na wartości atrybutów



Wnioskowanie w formalizmie drzewa

- Cel wnioskowania – określenie przynależności obiektu do klasy
- 1. Rozpatrujemy atrybut będący korzeniem k drzewa, porównujemy wartość tego atrybutu dla danego obiektu z etykietami gałęzi wychodzących z k – niech będzie to (k, l)
- 2. Z drzewa ‘wycinamy’ poddrzewo złożone z węzła l i jego następników (l jest jego korzeniem)
- 3. Jeśli nowe poddrzewo składa się z korzenia, to wynikiem wnioskowania jest klasa, która jest jego etykietą
- 4. Jeśli nie, przechodzimy do punktu 1.



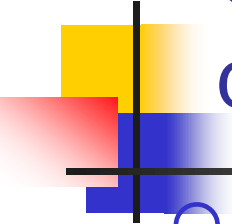
Problemy odpowiednie dla drzew decyzyjnych

- Przykłady są reprezentowane przez pary *atrybut-wartość*. Przykłady są opisane przez stały zbiór atrybutów i ich wartości, najłatwiejsza sytuacja – kiedy atrybuty mają niewielkie zbiory możliwych wartości.
- Funkcja docelowa ma dyskretne wartości (można rozszerzyć do rzeczywistych wartości).
- Jeśli możliwe jest wymaganie stosowania dysjunkcji.
- Ciąg uczący może zawierać błędy
- Ciąg uczący może zawierać brakujące wartości atrybutów
- *Problem klasyfikacji* – zadanie polega na sklasyfikowaniu przykładu do jednej kategorii (z dyskretnego zbioru możliwych)



Niedoskonałości danych zaburzające pracę algorytmów uczących się drzew

- Braki danych
- Błędy (szum) danych
- Zbyt mała próbka
- Zbyt duża próbka



Uczenie się drzewa decyzyjnego z danych – zasady ogólne

- Wstęp – dobór zbioru atrybutów, aby były istotne dla klasyfikacji.
- 1. Najpierw tworzymy jednowęzłowe drzewo skierowane i przypisujemy mu zbiór wszystkich zaobserwowanych obiektów.
- 2. Dla każdego, jeszcze nie rozpatrzonego liścia konstruowanego drzewa sprawdzamy, czy przypisany mu zbiór obiektów należy do jednej klasy. Jeśli tak – etykietujemy ten liść nazwą tej klasy. Więcej tego liścia nie rozpatrujemy.
- 3. W przeciwnym przypadku, dobieramy atrybut A , który najbardziej „optymalnie” dzieli zbiór Z obiektów przypisanych temu węzłowi w .

Uczenie się drzewa decyzyjnego z danych – zasady ogólne, cd.

- Niech $\{Z_1, Z_2, \dots, Z_n\}$ oznacza n wzajemnie rozłącznych podzbiorów zbioru Z takich, że ich suma równa jest Z .
 - Niech $\{W_1, W_2, \dots, W_n\}$ – zbiór n wzajemnie rozłącznych podzbiorów dziedziny wartości W optymalnego atrybutu A (ich suma jest równa W), uzyskanych przy wspomnianym optymalnym podziale.
 - Węzeł w etykietujemy atrybutem A
 - Tworzymy n nowych węzłów w_1, w_2, \dots, w_n i n nowych luków (w, w_i)
 - Węzłowi w_i przypisujemy zbiór obiektów Z_i , łuk (w, w_i) przypisujemy zbiór obiektów Z_i dla każdego $i=1, \dots, n$.
4. Po zakończeniu powyższego postępowania zwykle następuje krok ‘obcinania’ (pruning) – nadmiernie rozrośniętych gałęzi drzewa



Rozwiązania szczegółowe, prowadzące do różnicowania algorytmów

- Właściwy dobór kryterium optymalnego doboru atrybutu w kroku 3
 - Unikanie nadmiernego wpływu jednych trybutów kosztem innych
 - Redukcja wpływu atrybutów wielowartościowych
 - Redukcja wpływu atrybutów zaszumionych
 - Redukcja wpływu atrybutów nierelevantnych
- Właściwy dobór kryterium zakończenia konstrukcji drzewa (zwłaszcza dla danych z podejrzeniem o zaszumienie)
- Dobór metody ‘obcinania’ gałęzi
 - Usuwanie gałęzi zbyt słabo uzasadnionych statystycznie
 - Ewentualne przekształcenie drzewa i/lub przestrzeni atrybutów przy obserwacji podobnych podstruktur



Algorytm ID3 (1)

ID3 (Examples, Target_attribute, Attributes)

{Examples – ciąg uczący, Target_attribute – atrybut klasy, Attributes – lista innych atrybutów, testowanych przez uczone drzewo}

- Utwórz *Root* korzeń dla drzewa
- Jeśli wszystkie przykłady *Examples* są pozytywne, zwróć jednowęzłowe drzewo *Root* z etykietą = **+**
- Jeśli wszystkie przykłady *Examples* są negatywne, zwróć jednowęzłowe drzewo *Root* z etykietą = **-**
- Jeśli zbiór atrybutów *Attributes* jest pusty, zwróć jednowęzłowe drzewo *Root* z etykietą = najczęstszej wartości *Target_attribute* w *Examples*



Algorytm ID3 (2)

- W przeciwnym przypadku
 - $A \leftarrow$ atrybut ze zbioru *Attributes* **najlepiej*** klasyfikujący *Examples*
 - Decyzyjny atrybut dla korzenia $Root \leftarrow A$
 - Dla każdej możliwej wartości v_i atrybutu A ,
 - Dodaj nową gałąź poniżej korzenia $Root$ odpowiadającą testowi $A=v_i$
 - Niech $Examples_{v_i}$ będzie podzbiorem *Examples* przykładów mających wartość A równą v_i
 - Jeśli $Examples_{v_i}$ jest pusty
 - To poniżej tej nowej gałęzi dodaj liść z etykietą = najczęstszej wartości *Target_attribute* w *Examples*
 - W przeciwnym przypadku poniżej tej nowej gałęzi dodaj poddrzewo $ID3(Examples_{v_i}, Target_attribute, Attributes - \{A\})$
- End
- Zwróć $Root$



Który atrybut jest najlepszym klasyfikatorem?

- Wprowadzenie pojęcia **zysk informacyjny**
- Entropia (miara w teorii informacji, charakteryzuje ‘czystość’ kolekcji przykładów)
- Niech **S** zawiera pozytywne i negatywne przykłady
- **Entropia** (zawartość informacyjna) kolekcji **S** w odniesieniu do boolowskiej klasyfikacji wynosi:

$$\text{Entropy}(S) \equiv -p_+ \log_2 p_+ - p_- \log_2 p_-$$

- gdzie **p₊** jest proporcją pozytywnych przykładów w **S**,
- **p₋** jest proporcją negatywnych przykładów w **S**.
- Zawsze (z definicji), w odniesieniu do entropii:

0·log0 jest równe **0**.



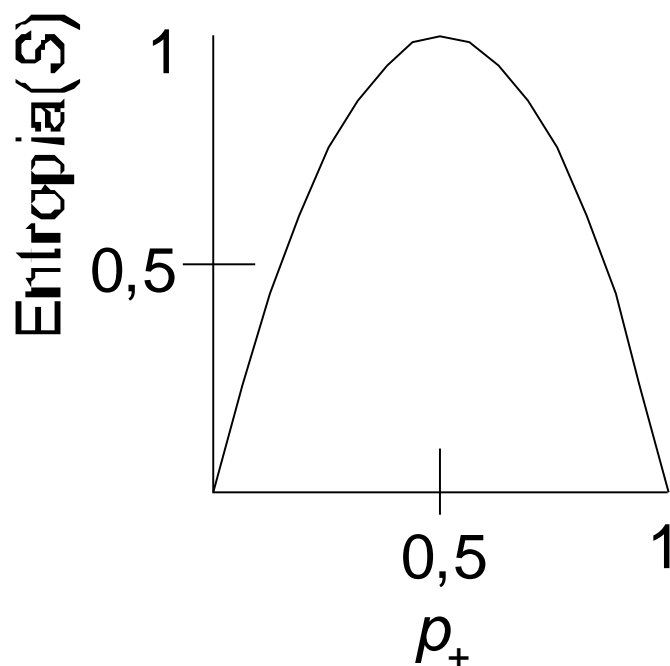
Entropia, cd.

- Bardziej ogólnie (nie dla logicznych funkcji), jeśli docelowy atrybut może przyjmować c klasyfikacji, entropia zbioru S jest definiowana:

$$Entropy(S) \equiv \sum_{i=1}^c -p_i \log_2 p_i$$

- p_i jest proporcją przykładów z S należących do klasy i .
- Podstawa logarytmu jest wciąż dwa, ponieważ entropia jest miarą oczekiwanej długości kodu mierzonej w bitach.
- Dla c możliwych klas, największa entropia może wynosić $\log_2 c$.

Entropia



Rysunek. Entropia dla funkcji logicznej

- Entropia jest zero, jeśli wszystkie przykłady z S należą do tej samej klasy.
- Entropia wynosi 1 jeśli S zawiera jednakową liczbę pozytywnych i negatywnych przykładów.
- Entropia jest mniejsza od 1 jeśli liczba pozytywnych przykładów jest różna od liczby negatywnych przykładów.

Zysk informacyjny (information gain)

- **Zysk informacyjny** to zmniejszenie entropii spowodowane podzieleniem przykładów zgodnie z wartościami rozpatrywanego atrybutu

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

- $Values(A)$ – zbiór wszystkich możliwych wartości atrybutu A ,
- S_v – podzbiór przykładów z S dla których A ma wartość v .
- Drugie wyrażenie w wzorze – oczekiwana wartość entropii po podzieleniu S używając atrybutu A (*Cichosz: zawartość informacyjna drzewa złożonego z węzła A , $B(Z, A) = \sum_{i=1}^n p_{A_i} \cdot M(Z_i)$*)
- Jest to suma każdego podzbioru, ważonego przez frakcję przykładów należących do tego podzbioru.
- Z drugiej strony, $Gain(S, A)$ jest też liczbą uzyskanych bitów kiedy kodujemy docelową wartość dowolnego członka S przy znanej wartości atrybutu A .



ID3 – przykład (1)

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

ID3 – przykład (2)

- S zawiera 14 przykładów funkcji logicznej: 9 pozytywnych i 5 negatywnych (oznaczamy [9+, 5-]).

- Entropia S :

$$Entropy([9+, 5-]) = - (9/14) \cdot \log_2(9/14) - (5/14) \cdot \log_2(5/14) = \mathbf{0,940}$$

- Zysk informacyjny spowodowany posortowaniem 14 przykładów przez atrybut *Wind* wynosi:

- $Values(Wind) = Weak, Strong$

- $S = [9+, 5-], \quad S_{Weak} \leftarrow [6+, 2-], \quad S_{Strong} \leftarrow [3+, 3-]$

$$Gain(S, Wind) = Entropy(S) - \sum_{v \in \{Weak, Strong\}} \frac{|S_v|}{|S|} Entropy(S_v) =$$

$$= Entropy(S) - [(8/14) \cdot Entropy(S_{Weak}) + (6/14) \cdot Entropy(S_{Strong})] = 0,940 - [(8/14)0,811 + (6/14)1,00] = 0,048$$



ID3 – przykład (3)

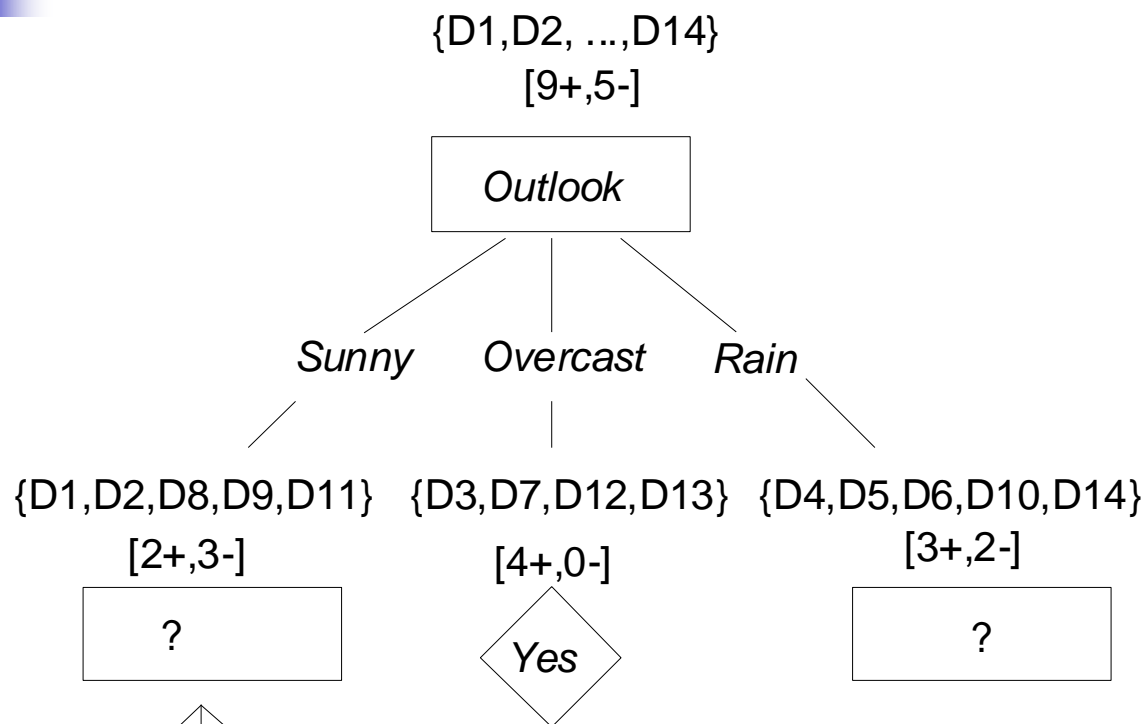
- Posortowany zysk informacyjny dla poszczególnych atrybutów:
 - $Gain(S, Outlook) = 0,246$
 - $Gain(S, Humidity) = 0,151$
 - $Gain(S, Wind) = 0,048$
 - $Gain(S, Temperature) = 0,029$
- Korzeniem zostaje atrybut *Outlook*
- Każdy przykład z *Outlook=Overcast* jest pozytywny dla *PlayTennis*, zatem jest to liść drzewa z klasyfikacją *PlayTennis=Yes*.
- Potomki dla pozostałych dwóch wartości korzenia mają entropię różną od zera, nie mogą stanowić liści.
- Dla każdego z tych dwóch węzłów powtarza się proces liczenia zysku informacji i wyboru następnika, aż będą same liście.



Dane posortowane wg Outlook

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
D3	Overcast	Hot	High	Weak	Yes
D7	Overcast	Cool	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D10	Rain	Mild	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	NORMAL	WEAK	YES
D11	Sunny	Mild	NORMAL	STRONG	YES

ID3 – przykład (4)




Który atrybut testować teraz?

$S_{sunny} = \{D1, D2, D8, D9, D11\}$

$\text{Gain}(S_{sunny}, \text{Humidity}) = 0,970 - (3/5)0,0 - (2/5)0,0 = 0,970$

$\text{Gain}(S_{sunny}, \text{Temperature}) = 0,970 - (2/5)0,0 - (2/5)0,0 - (1/5)0,0 = 0,570$

$\text{Gain}(S_{sunny}, \text{Wind}) = 0,970 - (2/5)1,0 - (3/5)0,918 = 0,019$



Przeszukiwanie przestrzeni hipotez w uczeniu drzewa decyzyjnego

- Przestrzeń hipotez – zbiór możliwych drzew
- ID3 rozpoczyna od pustego drzewa, rozpatrując kolejno więcej wypracowanych hipotez poszukując drzewa, które poprawnie klasyfikuje dane uczące – jest to przeszukiwanie *Hill-climbing*.
- Funkcją oceny ukierunkowującą przeszukiwanie jest zysk informacyjny.



Sortowanie wg Humidity dla Outluk=Sunny

- Przykłady do rozpatrzenia dla gałęzi Outlook=Sunny, biorąc następny węzeł Humidity

Nr	Outlook	Temperature	Humidity	Windy	Class
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No

- Dla tej części: suma $p_{A_i} * M(Z_i) = 2/5 * 0 + 3/5 * 0$ dlatego zysk informacyjny będzie większy niż gdyby wziąć Temperature lub Windy.



Możliwości i ograniczenia (1)

- Każda skończona funkcja o wartościach dyskretnych może być reprezentowana przez pewne drzewo decyzyjne, ID3 unika dużego ryzyka przeszukiwania niekompletnych przestrzeni hipotez (np. metody rozpatrujące tylko koniunkcje atrybutów)
- ID3 pracuje tylko z jedną hipotezą w czasie przeszukiwania – traci możliwości wynikające z reprezentowania wszystkich zgodnych hipotez (nie określa, jak dużo alternatywnych drzew jest zgodnych z ciągiem uczącym)
- ID3 (w czystej postaci) nie pozwala na powroty – wybrany atrybut na poszczególnym poziomie drzewa nie będzie więcej rozpatrywany. Jak w hill-climbing, może to doprowadzić do zbieżności do lokalnego optimum (post-pruning dodaje backtraking)



Możliwości i ograniczenia (2)

- ID3 wykorzystuje wszystkie uczące przykłady na każdym etapie. Jedną z przewag stosowania statystycznych własności wszystkich przykładów (tzn. zysku informacyjnego) jest mniejsza wrażliwość na błędy w pojedynczych przykładach.
- ID3 może być łatwo rozszerzony do postaci obejmującej zaszumione dane (przez kryterium zatrzymania).
- Obciążenie: Preferowane są krótsze drzewa oraz takie, które atrybuty o wysokim zysku informacyjnym plasują blisko korzenia.



Problemy i modyfikacje ID3

1. Atrybuty nie determinują klasyfikacji
 - Skutek – Warunek zakończenia algorytmu nie jest nigdy spełniony
 - Działanie – Zmienić kryterium zakończenia na: osiągnięcie jednorodnych klas w liściach lub niemożność obniżenia entropii $B(Z_i, A)$ (*zawartość informacyjna drzewa złożonego z węzła A*)
2. Istnieje szereg obiektów o takich samych atrybutach i klasie
 - Skutek – nadmiar nic nie wnoszących danych
 - Działanie – wyeliminować na wstępie nadmiarowe dane
3. Istnieją obiekty o identycznych wartościach atrybutów ale należące do różnych klas
 - Skutki i działanie – jak w punkcie 1.



Problemy i modyfikacje ID3, cd.

4. Istnieją obiekty z brakującymi wartościami atrybutów
 - Skutek – w niektórych przypadkach nie można policzyć entropii
 - Działanie – na wstępie, obiekt z brakującymi wartościami zastąpić obiektami o wszelkich możliwych kombinacjach wartości atrybutów w miejscach braków danych
5. Relewantne atrybuty nie determinują klasyfikacji, istnieją atrybuty nierelewantne
 - Skutek – nadmierny rozrost drzewa
 - Działanie – obcinanie drzewa. Zaczynamy od węzłów najbliższych liściom i testujemy, czy istnieje statystycznie ważna zależność między klasyfikacją a atrybutem decyzyjnym. Jeśli nie, obcinamy wychodzące z węzła gałęzie.
6. Szumy danych: skutki i działanie jak wyżej



Problemy i modyfikacje ID3, cd.

7. Nadmierny rozrost drzewa – działanie:

- Wariant A: zmienić reprezentację z drzewa na reguły i obciąć termy przesłanek reguł
- Wariant B: Przy badaniu zmniejszenia entropii przez dany atrybut badać ‘wyprzedzająco’ następny atrybut (decydujący na następnym poziomie drzewa decyzyjnego)
- Wariant C: Sprawdzić drzewo pod kątem występowania podobnych struktur i ‘obrócić’ przestrzeń atrybutów.

8. Bardzo duża baza danych

- Skutek – spowolnienie działania programu
- Działanie – pobrać z bazy losową próbkę i na niej stosować algorytm

9. W czasie stosowania algorytmu – ryzyko braków danych w wartościach atrybutów

- Skutek – nie można podjąć decyzji dla obiektu
- Działanie – generowanie tzw. probabilistycznego drzewa – gdy brak w ścieżce atrybutu, śledzi się wszystkie możliwe ścieżki następników obliczając prawdopodobieństwo warunkowe końcowej klasyfikacji – wybiera się najbardziej prawdopodobną klasę.



Techniki walidacji drzew decyzyjnych

- Można losowo podzielić próbkę na dwie (lub trzy) części:
- Pierwsza służy do generowania drzewa
- Druga – do zbadania, czy drzewo poprawnie klasyfikuje populacje obiektów
- Trzecia (ewentualna) – do operacji obcinania drzewa
- Stosuje się też technikę kroswalidacji

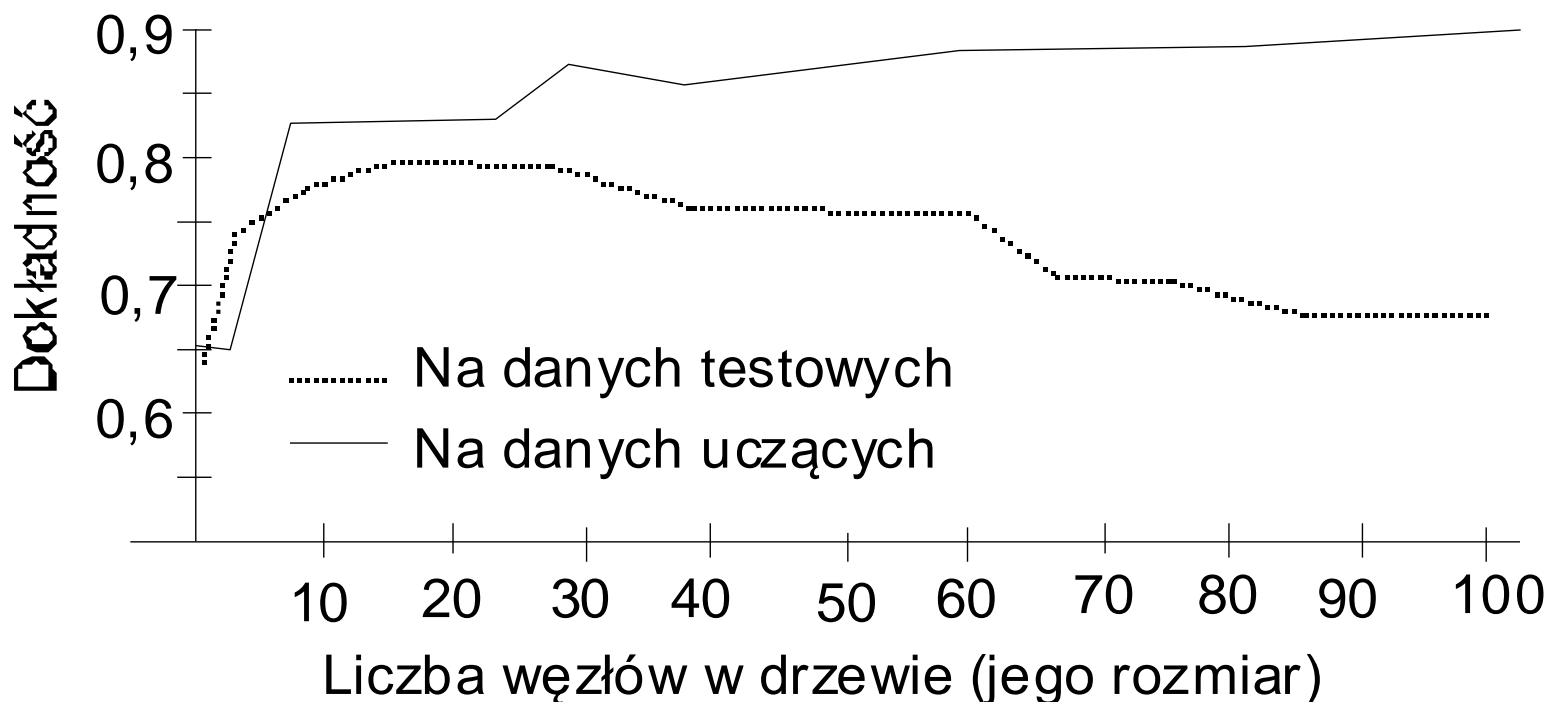


Problem zbytniego dopasowania do danych

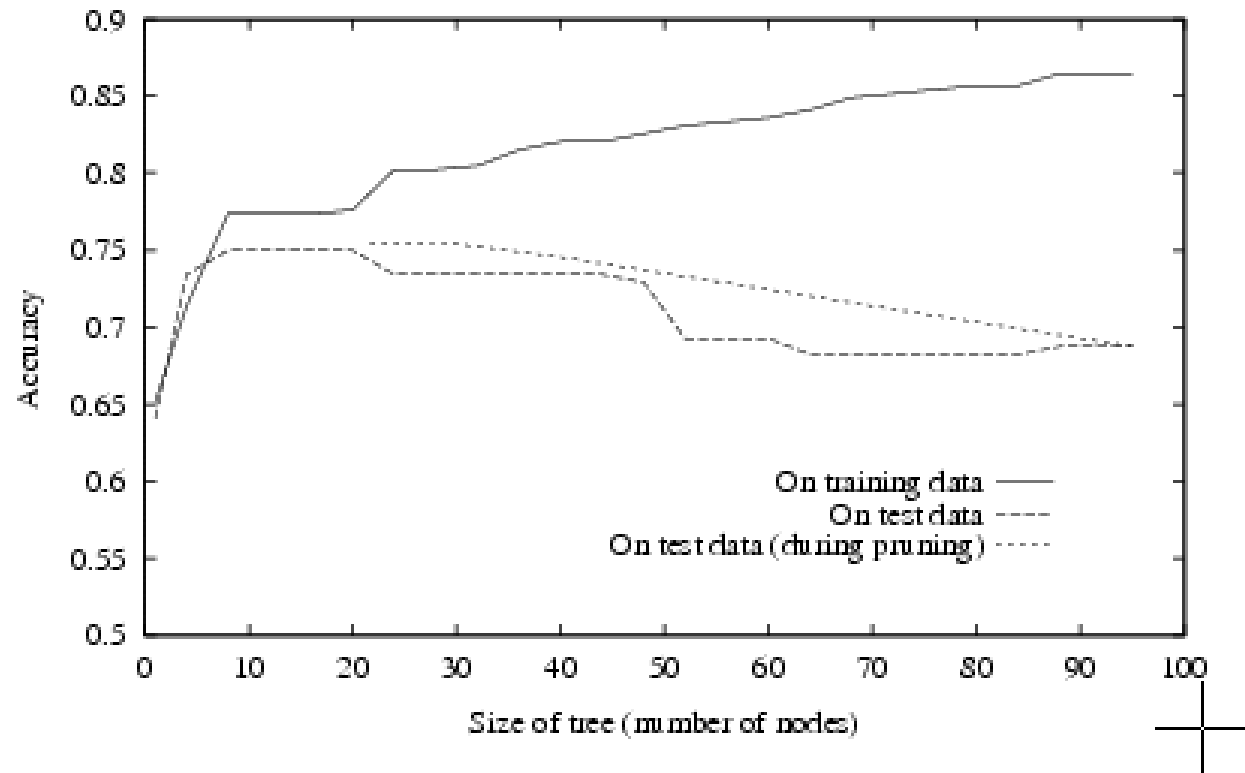
- $h \in \mathbf{H}$ nazywana jest **zbyt dopasowaną** (**overfit**) do ciągu uczącego, jeśli istnieje alternatywna hipoteza $h' \in \mathbf{H}$, taka że h ma mniejszy błąd niż h' na zbiorze uczącym, ale h' ma mniejszy błąd niż h na wejściowym rozkładzie przypadków
- W miarę dodawania nowych węzłów, monotonicznie poprawia się jego jakość drzewa na danych uczących. Jednakże, dla danych testowych, na początku dokładność drzewa również się zwiększa, ale później zaczyna spadać.

Nadmierne dopasowanie – ilustracja

- 'Naddopasowanie' w drzewie decyzyjnym (software i dane pod adresem: <http://www.cs.cmu.edu/~tom/mlbook.html>)




Effect of Reduced-Error Pruning





Zapobieganie nadmiernemu dopasowaniu

- Zatrzymać wzrost drzewa zanim osiągnie punkt, w którym perfekcyjnie klasyfikuje dane uczące
- pozwolić drzewu ‘naddopasować’ dane, a później dokonać post-pruning drzewa.
- Kryterium ustalania właściwego rozmiaru końcowego drzewa. Stosowane są trzy podejścia:
 - oddzielić ciągi uczący i testowy
 - wszystkie przykłady wykorzystać do uczenia, a statystyczne testy stosować do ustalenia, czy powiększać drzewo (np. Quinlan stosował test chi-kwadrat do ustalania, czy dalsza rozbudowa drzewa poprawi jego jakość na wejściowym rozkładzie przykładów czy tylko na zbiorze danych uczących)



Zapobieganie nadmiernemu dopasowaniu, cd.

- stosować jawną miarę złożoności (dla zakodowanych danych uczących i drzewa decyzyjnego, zatrzymując jego wzrost, kiedy rozmiar kodowania jest minimalny: zasada minimalnej długości opisu *Minimum Description Length principle*).
- Zapobieganie ‘naddopasowaniu’ – podejście *reduced-error pruning*: każdy węzeł w drzewie jest kandydatem do pruningu.
- Pruning węzła to:
 - usunięcie poddrzewa zaczynającego się w tym węźle,
 - zrobienie go liściem,
 - przypisanie mu najczęstszej klasy związanej z tym węzłem.
- Węzeł jest usuwany, jeśli otrzymane drzewo (po usunięciu analizowanego węzła) klasyfikuje nie gorzej niż wyjściowe.



Rule post-pruning

- Całkiem dobra metoda dla szukania hipotez o dużej dokładności, jej wariant jest stosowany przez C4.5 Quinlan'a – algorytm wyrosły na bazie ID3:
 - Generuj drzewo decyzyjne ze zbioru uczącego, rozbudowuj drzewo aż dane uczące są dopasowane najlepiej jak to tylko możliwe, pozwalając na 'naddopasowanie'
 - Konwertuj drzewo na zbiór reguł, tworząc regułę dla każdej ścieżki w drzewie
 - Wykonaj pruning (uogólnij) każdą regułę przez usunięcie wszystkich warunków, które prowadzi do polepszenia (nie pogarsza) estymowanej dokładności.
 - Sortuj poprawione (pruned) reguły zgodnie z ich estymowaną dokładnością, i rozpatruj je w tej kolejności kiedy klasyfikujesz nadchodzące przykłady



Przycinanie reguł – algorytm

- Funkcja *przytnij-reguły* (R, P)
 - Argumenty wejściowe
 - R – zbiór reguł do przycięcia
 - P – zbiór przycinania
 - **Zwraca:** zbiór reguł R po przycięciu
- 1. **Dla** wszystkich $r \in R$ **wykonaj**
- 2. **Dla** wszystkich selektorów $s \neq ?$ **wykonaj**
- 3. Zastąp s przez $?$ jeśli nie powiększy to szacowanego
- 4. na podstawie P błędu rzeczywistego reguły
- 4. **Koniec dla**
- 5. **Koniec dla**
- 6. **Zwróć** R
- **Selektor** – warunek
- **Kompleks** – koniunkcja **selektorów**



Zalety przekształcenia drzewa na zbiór reguł przed pruningiem

- Pozwala na rozróżnienie pomiędzy różnymi kontekstami, w których węzeł decyzyjny jest użyty, bo każda ścieżka prowadząca przez ten węzeł produkuje oddzielną regułę.
- Usuwa podział na atrybuty występujące blisko korzenia i te blisko liści (trudno usunąć korzeń w drzewie podczas pruningu).
- Poprawia czytelność.



Włączenie atrybutów o wartościach ciągłych

- Przykład:

temperatura:	40	48	60	72	80	90
graj_tenis:	No	No	Yes	Yes	Yes	No

- Wyznacza się wartość graniczną dającą największy zysk informacyjny.
- W przykładzie wyżej, są dwie kandydatki: $(48+60)/2$ oraz $(80+90)/2$.
- Większy zysk informacyjny ma wartość 54.
- Są też podejścia, w których dzieli się na kilka przedziałów, nie tylko na dwa.



C4.5 is a software extension of the basic ID3 algorithm designed by Quinlan to address the following issues not dealt with by ID3

- Avoiding overfitting the data
- Determining how deeply to grow a decision tree.
- Reduced error pruning
- Rule post-pruning
- Handling continuous attributes e.g., temperature
- Choosing an appropriate attribute selection measure
- Handling training data with missing attribute values
- Handling attributes with differing costs
- Improving computational efficiency

C4.5 is an extension of ID3 that accounts for unavailable values, continuous attribute value ranges, pruning of decision trees, rule derivation, and so on.



Numeryczne wartości w C4.5

- C4.5 umożliwia stosowanie nierówności – tzw. testy nierównościowe, np. $A_i \leq N$ oraz $A_i > N$ z dwiema wychodzącymi gałęziami
- Zysk informacyjny wyliczany jest następująco:
 - Sortowanie przykładów wg wartości rozpatrywanego atrybutu – jest skończona liczba wartości tego atrybutu: $\{v_1, v_2, \dots, v_m\}$. Istnieje $m-1$ możliwych podziałów. Wydaje się zbyt kosztowne testowanie wszystkich $m-1$ przypadków, ale: można to zrobić w jednym przebiegu; jeśli dla dwóch kolejnych v_i, v_{i+1} przykłady należą do tej samej klasy, to ‘optymalny’ podział nie będzie między tymi wartościami



Grupowanie wartości atrybutów w C4.5

- Możliwość testowania, czy wartość danego atrybutu należy do danego zbioru wartości: $A_i \in \{v_1, v_2, \dots, v_m\}$. Węzeł jest etykietowany atrybutem a gałąź – zbiorem wartości.
- Nie każda z wielu wartości atrybutu prowadzi do podzbioru przykładów istotnie różniącego się od pozostałych podzbiorów ze względu na rozkład kategorii i wartości innych atrybutów. Powstaną podobne poddrzewa (*fragmentacja* – zwielokrotnienie poddrzew).
 - Dla m różnych wartości istnieje $2^m - 1$ różnych binarnych podziałów (które wykluczają możliwość pełnego przeszukiwania w celu znalezienia najlepszego podziału)
 - C4.5 stosuje nieodwołalne (irrevocable) przeszukiwanie ‘bottom up’ – iteracyjne dołączanie grup
 - Początkowe grupy – indywidualne wartości rozpatrywanego właśnie atrybutu, w każdym cyklu c4.5 ocenia konsekwencję dołączenia każdej pary grup.
 - Proces jest kontynuowany aż zostanie właśnie dwie grupy wartości lub nie można poprawić zysku informacyjnego przez dalsze łączenie.



Modyfikacje ID3 w C4.5

1. Modyfikacja miary niejednorodności węzłów
2. Wprowadzenie możliwości wykorzystania atrybutów ciągłych
3. Włączenie metody oczyszczania drzewa – przycinanie drzewa
4. Umożliwienie klasyfikacji danych brakującymi wartościami

Ad1.: cel: uniknięcie efektu preferowania atrybutów o dużej liczbie możliwych wartości

Zamiast zysku informacyjnego stosuje się względny zysk informacyjny (współczynnik przyrostu informacji, ang. gain ratio):



Różnice C4.5 i ID3

- $$GainRatio(X) = \frac{(Gain(X))}{(SplitInfo(X))}$$

gdzie $SplitInfo(X)$ jest informacją uzyskaną poprzez podział obiektów wg wartości X :

$$SplitInfo(X) = - \sum_{i=1}^n \left(\frac{|C_i|}{|C|} \right) \log_2 \left(\frac{|C_i|}{|C|} \right)$$

- Ad. 2. C4.5, dla atrybutów ciągłych rozpatruje wszystkie możliwe podziały na dwa podzbiory (jak CART) na dwa podzbiory (wybierając punkt podziału w) Atrybuty ciągłe (inaczej niż dyskretne) mogą pojawiać się na wielu poziomach tej samej gałęzi drzewa; dla każdego możliwego podziału liczy się wartość względnego zysku informacyjnego i wybiera ten o maksymalnej wartości zysku



Różnice C4.5 i ID3

- Ad.3. W przycinaniu wykorzystuje się statystyczną ocenę istotności różnicy błędu klasyfikacji dla danego węzła i jego podwęzłów
- Przy założeniu dwumianowego rozkładu liczby błędów, ocenia się prawdopodobieństwo zmniejszenia błędu w badanym węźle; obcina się te, dla których to prawdopodobieństwo nie przekracza żądanego progu, lub zamienia się poddrzewo o korzeniu w danym węźle jego najlepszym poddrzewem



Różnice C4.5 i ID3

- Ad. 4. Brakujące dane: podczas budowy drzewa nie uwzględnia się danych z brakującym atrybutem, a wyliczony zysk skaluje się mnożąc przez częstość występowania wartości tej cechy w próbie treningowej – podział danych na podwężły wprowadza wagi dla wektorów treningowych, które dla wektora z brakującą wartością atrybutu decyzyjnego odpowiadają rozkładowi pozostałych danych w podwężłach
- Modyfikacji ulegają współczynniki p_i w wzorze na zysk – zamiast mocy zbiorów bierze się sumy wag elementów tych zbiorów
- Współczynniki p_i są też używane przy podejmowaniu decyzji, by wyliczyć prawdopodobieństwa wpadania do poszczególnych węzłów oraz przynależenia do poszczególnych klas
- C4.5 umożliwia przejście na zbiór reguł, które podlegają pruningowi – z każdej reguły usuwane są przesłanki, których usunięcie nie powoduje spadku jakości klasyfikacji