Analiza skupień (II)

Marcin Szeląg

Zakład ISWD, Instytut Informatyki, Politechnika Poznańska

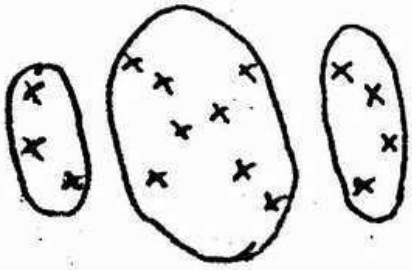
08.01.2020

Numeryczna versus konceptualna Analiza Skupień 2

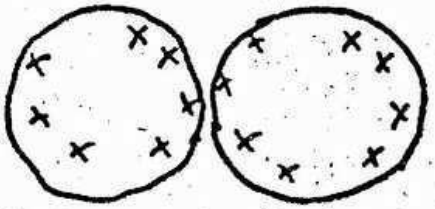
- Ograniczenia metod numerycznych Przykład.

Zbiór obserwacji

Metoda centroidalna



Właściwy rezultat (single linkage)



"odkrycie" 2 okręgów.

- Jaka jest więc rola konceptualnego budowania skupień
  - Wspomagać użytkownika we właściwym kodowaniu danych i wyrażaniu jego wiedzy dziedzinowej
  - Dostarczać nowe narzędzia analizy rezultatów i wyrażania ukierunkowań algorytmów poszukiwania skupień

Różne punkty widzenia

Grupowanie, kategoryzacja oraz tworzenie klasyfikacji

Numeryczna Analiza Danych

Automatyczne metody grupowania danych:

- podejścia matematyczne
- techniki efektywne obliczeniowo

Psychologia Poznawcza: Kategoryzacja

Analizie podlega ludzkie postrzeganie:

- notacja kategorii
- notacja podobieństwa

AI: Konceptualne Grupowanie

(*Michalski, Stepp, Fisher, Gennari*)

Nacisk na efektywność metod i zrozumiałość rezultatów końcowych

6. Motywacje metod konceptualnych

- **Dane wejściowe:**
  - Zbiór niesklasyfikowanych obserwacji
  - Wiedza o dziedzinie, atrybutach, celach uczenia ...
- **Cel - Znaleźć:**
  - zbiór skupień grupujących obserwacje
  - zrozumiałe symboliczne definicje każdego skupienia
  - hierarchiczną organizację pojęć odpowiadających skupieniom

Grupowanie pojęciowe – inspiracje psychologiczne

Dyskusja literaturowa:

Podstawa czynności poznawczych człowieka → „myślenie polega na operowaniu pojęciami i tworzeniu z nich większych struktur, np. hipotez, teorii, itd.

Pojęcie – definicja (J.Kozielecki):

Poznawcza reprezentacja skończonej liczby wspólnych cech, które w jednakowym stopniu przysługują wszystkim desygnatom (egzemplarzom) danej klasy.

Pojecie to klasa obiektów zawierająca obiekty o tych samych właściwościach.

Przykład:

„kwadrat" to nazwa wszystkich płaskich figur geometrycznych o równych bokach i czterech równych katach, tworzących krzywą zamkniętą.

Zapis w logice VL1 (R.Michalski)

[figura=płaska][krzywa=zamknięta][boki=4][długości boku=równe][liczba katów =4][kąt =90]

Taksonomia symboliczna – inspiracje psychologiczne

Trzy grupy podejść do konstrukcji algorytmów:

1. Klasyczne
2. Probabilistyczne
3. Oparte na wzorcach

Klasyczne podejście → pojęcie definiuje się jako zbiór cech, które są z osobna konieczne, a łącznie wystarczające do klasyfikacji desygnatów.

CLUSTER → pojęcia koniunkcyjne

Podejście probabilistyczne → opis za pomocą ważonych cech, jakie występują u obiektów będących desygnatami.

Każda cecha ma wagę, będącą np. częstością lub prawdopodobieństwem jej występowania, np. $P(C_i=W_{ij}|K_k)$. Przykład użycia - COBWEB.

Podejście oparte na wzorcach → wykorzystuje do reprezentacji pojęcia jeden lub kilka typowych desygnatów.

Przykłady Algorytmów Konceptualnego Tworzenia Skupień

- **Dwie cechy różnicujące algorytmy:**
  - ⇨ Nie-przyrostowy lub przyrostowy sposób pracy algorytmu
  - ⇨ Strategie "Bottom-up" lub "Top-down"

Algorytm	**Przyrostowy**	**Nieprzyrostowy**
Top-down	**COBWEB** **CALSSIT** **UNIMEM** **ADELCU**	**CLUSTER/2** **CLUSTER/S**
Bottom-up	**WITT**	**AutoClass** **Pyramid** **KBG**

(9) 3.11.2005

reguła decyzyjna *r*

Jest to wyrażenie postaci:

jeżeli $R(x)$ to $(x \in K)$

gdzie:

R (część warunkowa reguły) jest koniunkcją $s_1 \wedge s_2 \wedge \ldots \wedge s_q$, spełniającą warunek $[R]_K^+ \neq \varnothing$.

reguła dyskryminująca

Reguła jest dyskryminująca (tzn. odróżnia przykłady pozytywne od negatywnych) jeżeli część warunkowa reguły R jest:

- ☐ spójna - $[R]_K^- = \varnothing$,
- ☐ minimalna - usunięcie dowolnego warunku s_j z R spowoduje niespełnienie warunku spójności.

Tworzenie reguł decyzyjnych w logice zmienno wartościowej (VL1 - Variable-valued Logic system 1 [Michalski 75])

VL1 - logika wielowartościowa dla reprezentowania problemów decyzyjnych, gdzie zmienna decyzyjna przyjmuje wartości z pewnego zakresu.

Pojęcia podstawowe:

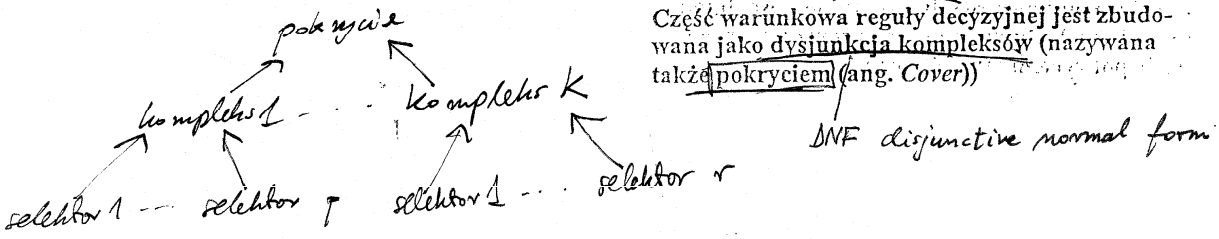
selektor (ang. selector)- **warunek elementarny wyrażony w notacji VL1, na ogół wiążący atrybut z jego wartością lub podzbiorem wartości, np.**

[a#R]

gdzie *a* jest atrybutem czy zmienną, # jest operatorem relacyjnym (takim jak $=, \neq, <, >, \leq, \geq$) and *R* jest zbiorem jednej lub wielu wartości, które *a* może przyjmować.

Complex (ang. kompleks, zespół, złożenie) - koniunkcja selektorów

Część warunkowa reguły decyzyjnej jest zbudowana jako dysjunkcja kompleksów (nazywana także pokryciem (ang. *Cover*))



DNF disjunctive normal form

Algorytm AQ (wg. wersji AQ15 [Michalski 86])

W Algorytmie przeszukuje się heurystycznie przestrzeń możliwych wyrażeń w logice VL_1 w celu odnalezienia takich pokryć klas decyzyjnych, które obejmują wszystkie przykłady pozytywne i żadnych negatywnych.

W ocenie kompleksów wykorzytuje się tzw. *kryteria preferencyjne* zależne od użytkownika i zastosowania (*zasada porządku leksykograficznego*).

W budowie reguł wykorzystuje się *przecięcie* (ang. intersection) zbioru kompleksów ze zbiorem selektorów. Przecięcie dwóch zbiorów A i B jest zbiorem wszystkich kombinacji koniunkcji elementów z obu zbiorów tj. $\{x \wedge y \mid x \in A,\ y \in B\}$

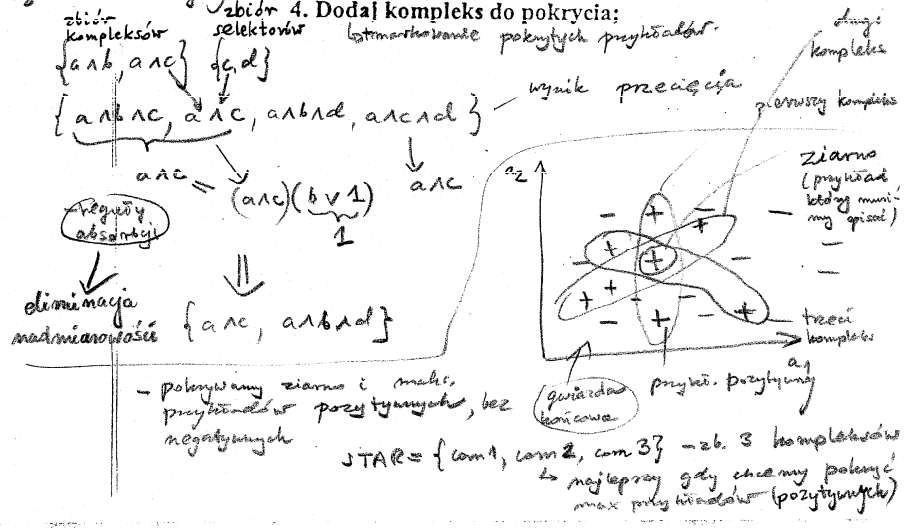
Algorytm AQ15 generuje regułę (DNF) kolejno dla każdej klasy decyzyjnej. Algorytm krokowo wykorzystuje technikę typu (ang. beam search: multiple hill climbers in parallel). W każdym kroku algorytmu najlepszy kompleks jest dodawany do pokrycia klasy decyzyjnej. Każdy krok rozpoczyna się od skupienia uwagi na wybranym przykładzie pozytywnym nazywanym ziarnem (ang. *seed*).

Algorytm generuje zbiór wszystkich kompleksów (tzw. gwiazdę - ang. *star*) który pokrywa ziarno i nie pokrywa żadnego z przykładów negatywnych. Następnie najlepszy kompleks jest wybierany.

Przebieg algorytmu AQ15

While częściowe pokrycie nie pokrywa wszystkich przykładów pozytywnych *do*

1. **Wybierz *ziarno*, tj. niepokryty przykład pozytywny;**
2. Generuj ***Gwiazdę*** - tj. zbuduj maksymalnie ogólne kompleksy pokrywające ziarno i niepokrywające żadnego z przykładów negatywnych;
3. **Wybierz najlepszy** kompleks zgodnie z **definicją funkcji jakości;**
4. **Dodaj kompleks** do pokrycia;



- pierwsza gwiazda częściowa – pusta (pokrywa wszystko)

While częściowa gwiazda pokrywa przykłady negatywne *do*

1. Wybierz jeden z pokrytych przykładów negatywnych;
2. **Zbuduj gwiazdę** częściową: **maksymalnie ogólne kompleksy pokrywające** ziarno i niepokrywające **wybranego przykładu** negatywnego;
3. Utwórz nową **gwiazdę częściową poprzez** operację przecięcia **zbudowanej w pkt. 2.** gwiazdy częściowej **z gwiazdą poprzednio** skonstruowaną;
4. Uporządkuj nową gwiazdę **częściową, tj.** zatrzymaj tylko *maxstar* **najlepszych jak dotąd** kompleksów.

określona liczba (SZEROKOŚĆ WIĄZKI)

(porządkowanych leksykograficznie)

(10a) reguły absorbcji

AQ Algorithm

e – przykład będący ziarnem
e1 – –"– negatywny
G{e|e1} – gwiazda e przeciwko e1 ⇓ dobieramy war. atrybutów

seed a positive example

selector relates a variable to a value or disjunction of values

complex is a conjunction of selectors

star is a set of all complexes describing the seed and not covering negative examples

cover is a disjunction of complexes describing all positive examples and none of negative examples

partial star $G(e \mid e_1)$, where

$e = (x_1, x_2, \dots, x_k)$, — 2 ciągi wartości atrybutów

$e_1 = (r_1, r_2, \dots, r_k)$,

the complexes of $G(e \mid e_1)$ are (all selectors with) $(x_i \neq r_i)$ (i)

star $G(e \mid F)$ is constructed by building stars $G(e \mid e_i)$ for all i and then conjuncting these partial stars by each other, using absorbtion law to eliminate redundancy.

~~G(e|N)~~

MAXSTAR (beam) — szerokość wiązki

$P \wedge (P \vee Q) \Longleftrightarrow P$

absorption law

conjunct

7. Przykład Algorytmu CLUSTER/2

Autorzy: (*R. Michalski et al. 1983*)

❒ **Cechy charakterystyczne:**

⇨ Reprezentacja przykładów w notacji atrybut-wartość (porządkowe, nominalne, strukturalizowane taksonomie)

⇨ Podejście "Top-down": zbiór przykładów stopniowo dzielony

⇨ Inspirowany metodami ***Dynamicznego Grupowania***

- Funkcja przypisywania ***F*** oparta na procedurze dopasowania
- Funkcja charakteryzująca ***G*** wykorzystuje technikę uogólniania STAR

⇨ Kryteria oceny skupień (LEF: w porządku leksykograficznym):

- Zgodność pomiędzy skupieniem a obserwacjami,
- Syntaktyczna prostota charakteryzacji skupień
- Maksymalizacja różnicy między skupieniami
- Liczba dyskryminujących atrybutów (tj. posiadających różne wartości w każdym skupieniu)
- Inne ...

CLUSTER/2 (Michalski, Stepp)

System CLUSTER/2 reprezentuje tworzone przez siebie grupowanie jako pojęcia opisane przez koniunkcyjne wyrażenia złożone z selektorów, czyli kompleksy.

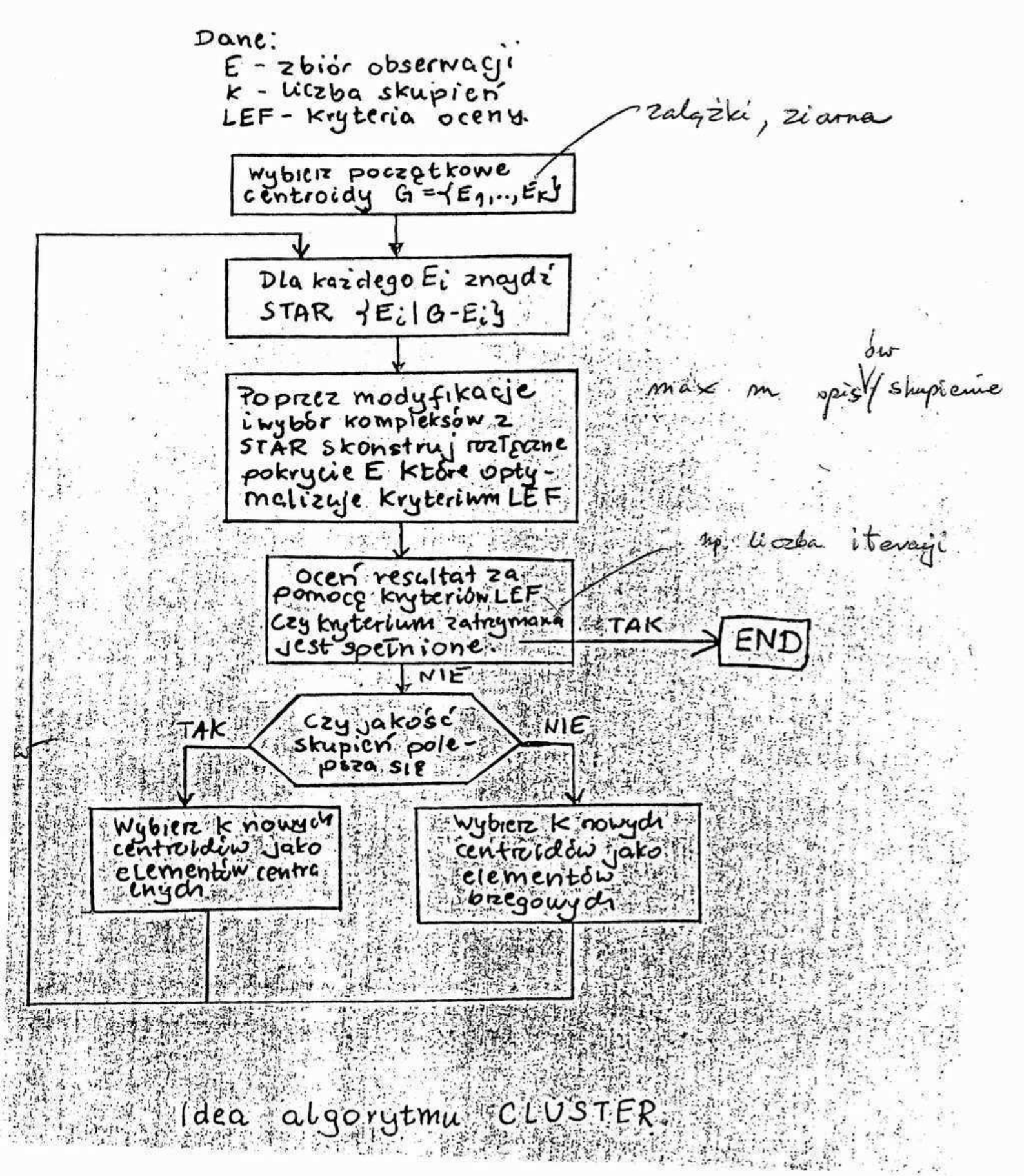
Jeden kompleks dla każdej kategorii: wszystkie przykłady pokrywane przez ten kompleks zaliczane są do związanej z nim kategorii.

Kompleksy reprezentujące kategorie są parami rozłączne, tzn. nie pokrywają żadnych wspólnych przykładów.

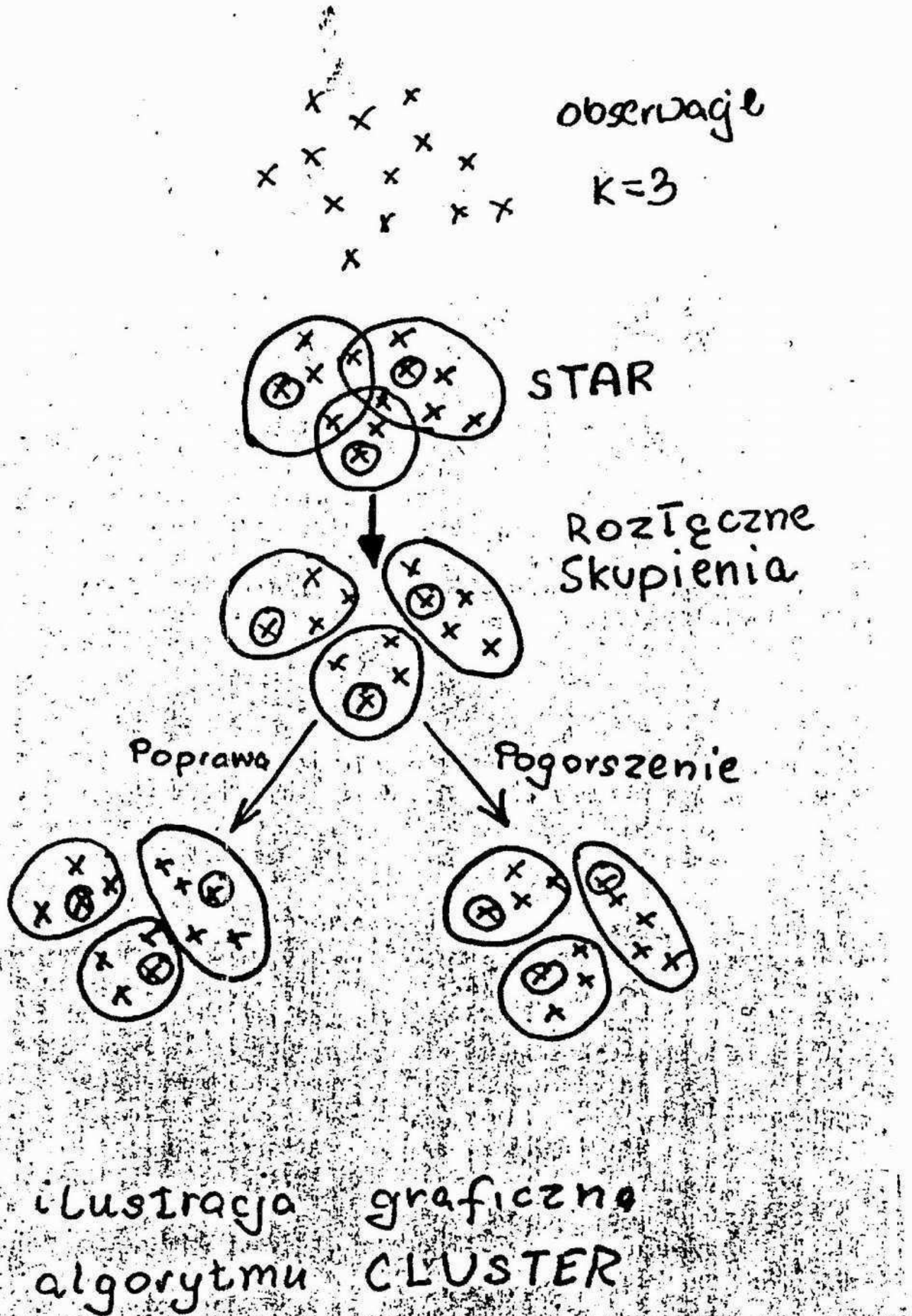
Podstawowy algorytm tworzenia kategorii w systemie CLUSTER/2.

1. ***powtarzaj pewną liczbę razy* (???)**
    1. ***wybierz k przykładów jako ziarna;***
    2. ***dla każdego ziarna x_s wykonaj***

       wygeneruj gwiazdę S jako zbiór m najlepszych maksymalnie ogólnych kompleksów pokrywających x_s i niepokrywających żadnego innego ziarna;
    3. ***wybierz po jednym kompleksie z każdej gwiazdy modyfikując je w razie potrzeby tak, aby wybrane kompleksy były parami rozłączne;***
2. ***wybierz najlepsze z uzyskanych grupowań według heurystyki oceniającej.***



Idea algorytmu CLUSTER



ilustracja graficzna algorytmu CLUSTER

The CLUSTER/2 algorithm

1. Select k seeds from the set of observed objects. This may be done randomly or according to some selection function.

2. For each seed, using that seed as a positive instance and all other seeds as negative instances, produce a maximally general definition that covers all of the positive and none of the negative instances (multiple classifications of non-seed objects are possible.)

The CLUSTER/2 algorithm

3. Classify all objects in the sample according to these descriptions. Replace each maximally general description with a maximally specific one that covers all objects in the category (to decrease the likelihood that classes overlap on unseen objects.)

4. Adjust remaining overlapping definitions.

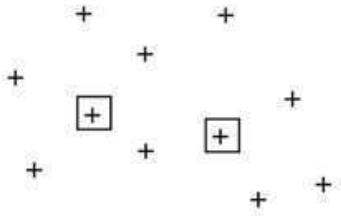
5. Using a distance metric, select an element closest to the center of each class.

6. Repeat steps 1-5 using the new central elements as seeds. Stop when clusters are satisfactory.

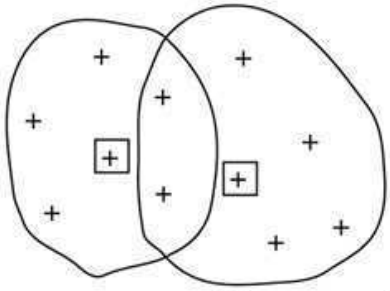
The CLUSTER/2 algorithm

7. If clusters are unsatisfactory and no improvement occurs over several iterations, select the new seeds closest to the edge of each cluster.

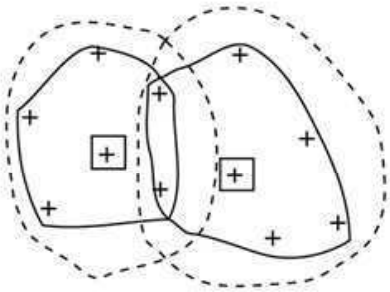
The steps of a CLUSTER/2 run



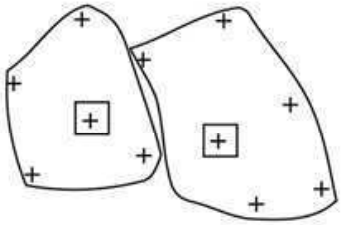
After selecting seeds (step 1).



After generating general descriptions (step 2).
Note that the categories overlap.



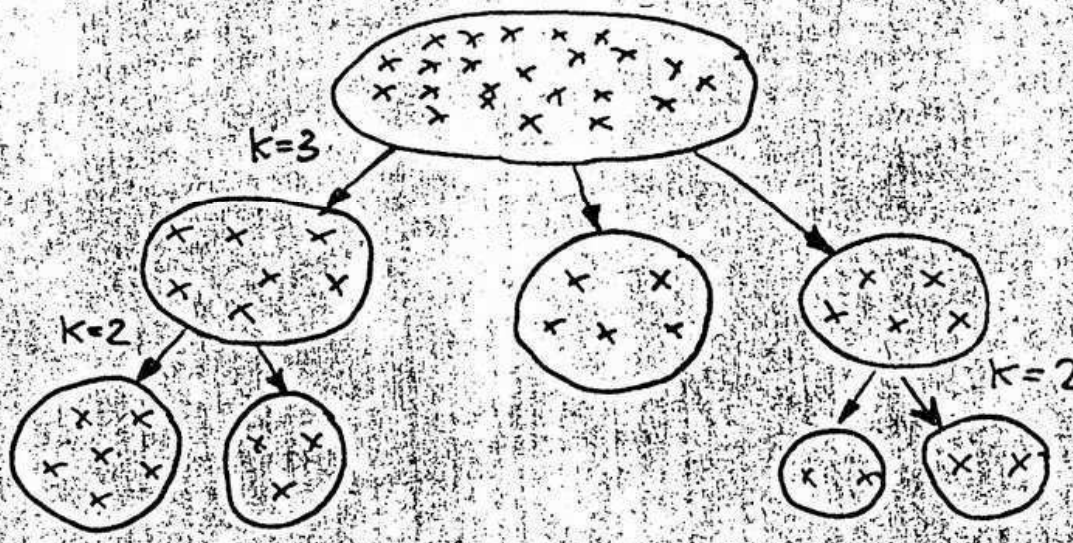
After specializing concept descriptions (step 3). There are still intersecting elements.



After eliminating duplicate elements (step 4).

ALGORYTM CLUSTER/2
Krok hierarchizacji

- stopniowy podział za pomocą podstawowego algorytmu:
  - Dla danego zbioru obserwacji iteracyjnie powtarza się algorytm CLUSTER/2 dla różnych wartości K → wybiera się najkorzystniejszy podział z uwagi na LEF
  - Po wyborze podziału, krok powyższy jest powtarzany dla skupień



- Ograniczenia powyższego podejścia
  - Duże koszty obliczeniowe → duże gwiazdy, bo tylko nie przekrywamy innych zbiorów!
  - Wiele parametrów sterujących

↓

np. dobór kryteriów do LEF

Figure 11-8: A dataset describing ten objects, using four variables

Event	x_1	x_2	x_3	x_4
e_1	0	a	0	1
e_2	0	b	0	0
e_3	0	c	1	2
e_4	1	a	0	2
e_5	1	c	1	1
e_6	2	a	1	0
e_7	2	b	0	1
e_8	2	b	1	2
e_9	2	c	0	0
e_{10}	2	c	2	2
Variable type:	L	S	L	N

(L: linear; N: nominal; S: structured)

Figure 11-9: The generalization hierarchy of the domain of variable x_2

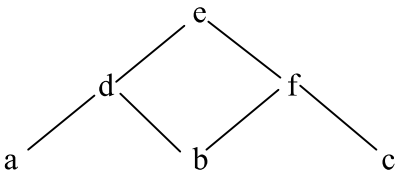


Figure 11-10: A geometrical representation of events e_1 to e_{10}. Encircled events are initial seeds.

x_1	x_2										
0	**a**		e_1								
	b	e_2									
	c						e_3				
1	**a**			e_4							
	b										
	c					e_5					
2	**a**				e_6						
	b		e_7				e_8				
	c	e_9								e_{10}	
		0	**1**	**2**	**0**	**1**	**2**	**0**	**1**	**2**	x_4
		0			**1**			**2**			x_3

Step 2. Bounded reduced stars RG($e_1|e_2$, m) and RG($e_2|e_1$, m), with m = 5, are generated by procedure Boundstar:

RG($e_1|e_2$, m) = {[x_2 = a][x_3 = 0 ∨ 1], [x_4 = 1 ∨ 2]},
RG($e_2|e_1$, m) = {[x_2 = b ∨ c], [x_4 = 0 ∨ 2]}.

These stars contain all possible complexes because m > 2. After applying the *closing the interval* and *climbing the hierarchy* generalization rules, the stars become:

RG($e_1|e_2$, m) = {[x_2 = a][x_3 ≤ 1], [x_4 = 1 ∨ 2]},
RG($e_2|e_1$, m) = {[x_2 = f], [x_4 = 0 ∨ 2]}.

Iteration 4

This iteration produces a new clustering:

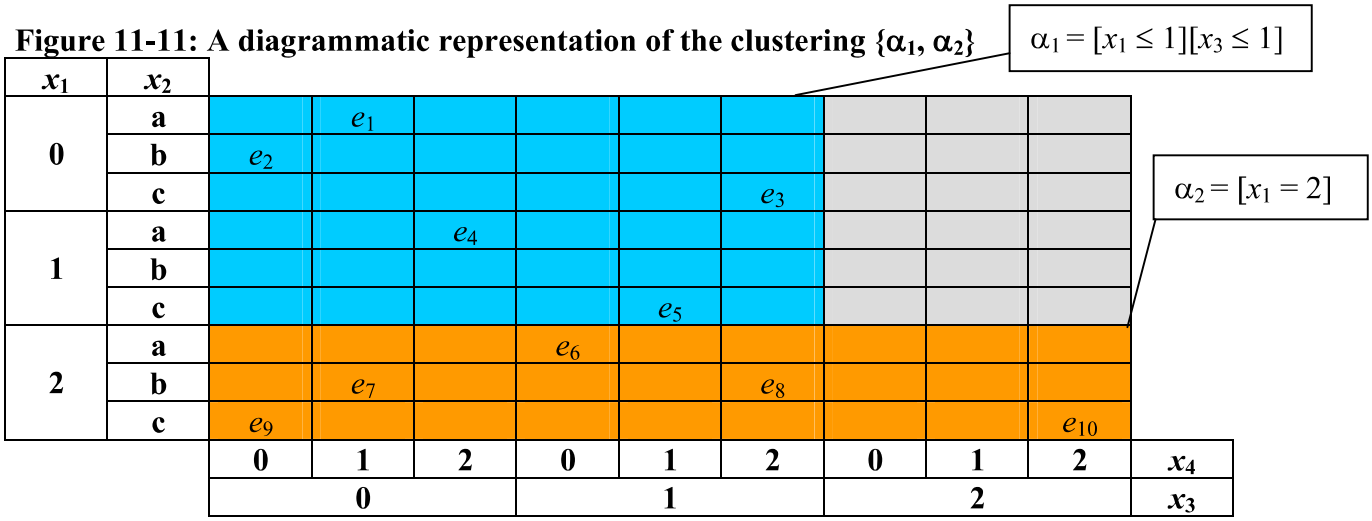
		Sparseness	Complexity
complex 1	$[x_3 \geq 1]$	49	1
complex 2	$[x_3 = 0]$	22	1
		71	**2**

It is the second "probe" iteration. If the obtained clustering was better than the previous best clustering, another probe = 2 iterations would be scheduled. Since the sparseness of the clustering obtained in this iteration (71) is not an improvement over the previous best sparseness (53), the termination criterion is satisfied. The best resulting clustering is the one produced in iteration 2:

$[x_1 \leq 1][x_3 \leq 1]$,
$[x_1 = 2]$.

Figure 11-11 shows the diagrammatic representation of this solution.

Figure 11-11: A diagrammatic representation of the clustering $\{\alpha_1, \alpha_2\}$



COBWEB – grupowanie pojęciowe

Algorytm zaproponowany przez D. Fishera w 1986r.

- Ukierunkowanie / cel grupowania → algorytm realizuje podział zbioru obiektów w ten sposób, aby znaleźć taką strukturę kategorii (klas), która prowadzi do maksymalizacji informacji, jaką można przewidzieć znając kategorię przykładu.

  Przydatność dla potencjalnej klasyfikacji nowych obiektów lub wnioskowania.

- Algorytm **przyrostowy** → działa na bieżąco, obserwując przykład tylko raz.

- Reprezentacja grupowania w formie hierarchicznego **drzewa** kategorii obiektów.

- Podstawowa wersja algorytmu COBWEB tworzy grupowania jedynie dla przykładów opisywanych wyłącznie przez **atrybuty dyskretne**.

- Opis klas jako rozkład wybranych wartości atrybutów (ważonych prawdopodobieństw).

- Wykorzystuje się probabilistyczne podejście do tworzenia kategorii.

Grupowanie pojęciowe jako przeszukiwanie

Grupowanie wykonywane przez COBWEB jest **przeszukiwaniem przestrzeni możliwych grupowań**.

Podczas opisu przeszukiwania wyróżnia się:

- heurystyczną funkcję oceny grupowania (stosowaną do kierowania przeszukiwaniem),
- reprezentację grupowania,
- operatory używane do poruszania się w przeszukiwanej przestrzeni,
- strategię przeszukiwania (sterowania).

Funkcja oceny grupowania

Stosuje się „probabilistyczną” miarę użyteczności grupowania.

Związek z badaniami psychologicznymi:

- W trakcie analizy hierarchii pojęć przez ludzi wyodrębnia się tzw. **poziom podstawowy**.
- Kategorie należące do tego poziomu dostarczają najwięcej informacji, są najbardziej zróżnicowane, mają najwięcej cech charakterystycznych i pozwalają uporządkować pozostałe informacje o świecie (Rosch 1978).
- Matematyczny opis poziomów hierarchii związany z miarami **istotności cechy** oraz **istotności kategorii**.

Więcej w E.Gatnar: „Symboliczne metody klasyfikacji danych”, PWN, Warszawa 1998.

Funkcja oceny – trochę podstaw matematycznych

Przykłady są „generowane” z pewnym rozkładem prawdopodobieństwa z dziedziny X.

Notacja:

x – przykład opisany zbiorem atrybutów $a_i \in A$; $a_i(x)$ – wartość atrybutu (cechy) dla przykładu x.

C – zbiór kategorii; $d \in C$ ozn. kategorię a c(x) to kategoria przykładu x.

Prawdopodobieństwo a podobieństwo:

Dla każdej kategorii $d \in C$ określa się:

$P_{x \in X}(a_i(x)=v_{ij}|c(x)=d)$ – interpretacja jako „stopień podobieństwa przykładów w ramach kategorii d”. (powiązanie z „**istotnością kategorii**”)

$P_{x \in X}(c(x)=d|a_i(x)=v_{ij})$ – interpretacja jako „brak podobieństwa przykładów z różnych kategorii”. (powiązanie z miarą „**istotności cechy**” – w jakim stopniu znając wartość cechy można przewidzieć kategorię)

Prawdopodobieństwa a grupowanie

Pomiar stopnia podobieństwa przykładów tej samej kategorii i zróżnicowania przykładów różnych kategorii:

$$\sum_{d \in C} \sum_{a_i} \sum_{v_{ij} \in a_i} P(a_i(x) = v_{ij}) \cdot P(c(x) = d \mid a_i(x) = v_{ij}) \cdot P(a_i(x) = v_{ij} \mid c(x) = d)$$

Interpretacja – wymiana między podobieństwem wewnątrz kategorii a brakiem podobieństwa pomiędzy kategoriami.

Na podstawie wzoru Bayesa:

$$P(a_i(x) = v_{ij}) \cdot P(c(x) = d \mid a_i(x) = v_{ij}) = P(c(x) = d) \cdot P(a_i(x) = v_{ij} \mid c(x) = d)$$

Wyrażenie można przekształcić do:

$$\sum_{d \in C} P(c(x) = d) \sum_{a_i} \sum_{v_{ij} \in a_i} \left(P(a_i(x) = v_{ij} \mid c(x) = d)\right)^2$$

<u>Miara użyteczności (funkcja oceny) grupowania (ang. category/clastering utility – CU)</u>:

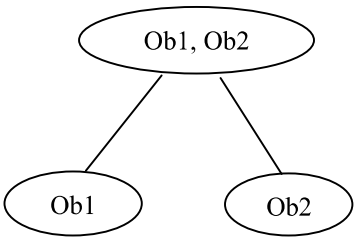
$$\frac{1}{|C|} \sum_{d \in C} P(c(x) = d) \left[\sum_{a_i} \sum_{v_{ij}} P(a_i(x) = v_{ij} \mid c(x) = d)^2 - \sum_{a_i} \sum_{v_{ij}} P(a_i(x) = v_{ij})^2 \right]$$

COBWEB - operatory grupowania

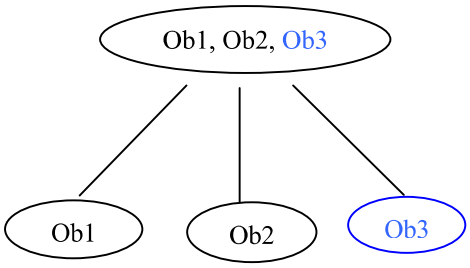
- Utworzenie nowej kategorii i dodanie do niej analizowanego obiektu.
- Dołączenie obiektu do jednej z istniejących kategorii.
- Połączenie dwóch kategorii (z zachowaniem łączonych kategorii jako węzły potomne) i umieszczenie obiektu w węźle powstałym po połączeniu.
- Podział istniejącej kategorii na jej wszystkie podkategorie (i umieszczenie obiektu w najlepszej powstałej podkategorii).

utworzenie nowej kategorii

Przed dodaniem obiektu Ob3:

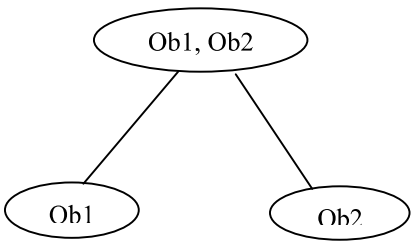


Po dodaniu obiektu Ob3:

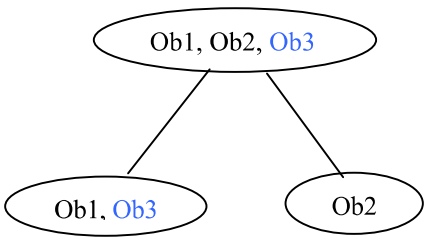


zaliczenie obiektu do istniejącej kategorii

Przed dodaniem obiektu Ob3:

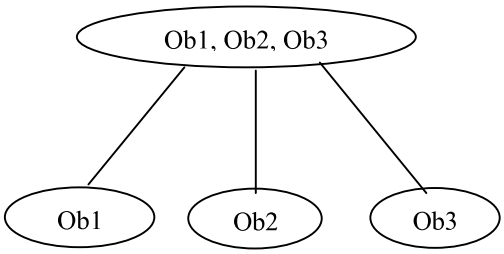


Po dodaniu obiektu Ob3:

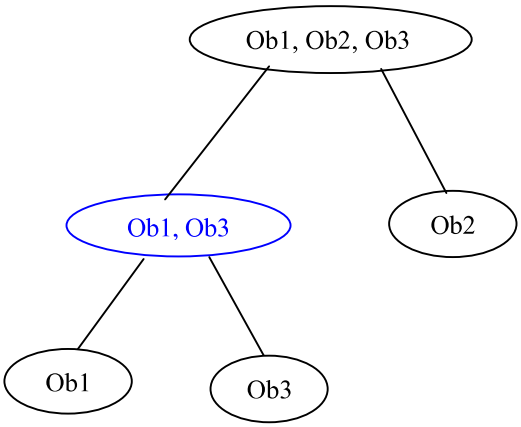


połączenie dwóch istniejących kategorii

Przed połączeniem:

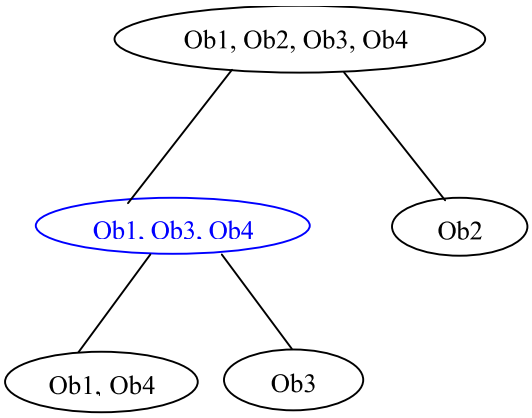


Po połączeniu pojemników z obiektami Ob1 i Ob3:

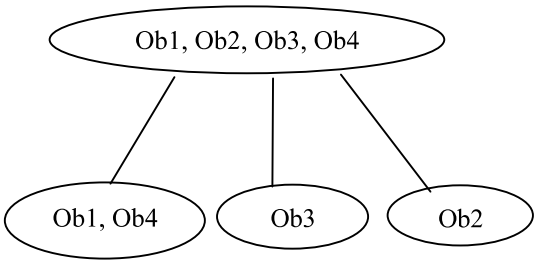


podział istniejącej kategorii

Przed podziałem:



Po podziale pojemnika z obiektami Ob1, Ob3 i Ob4:



Dwa ostatnie operatory grupowania mają za zadanie osłabienie wpływu na wyniki grupowania kolejności w jakiej są podawane przykłady.

COBWEB – strategia sterowania

Argumenty wejściowe:

- ✓ X_* - dodawany obiekt
- ✓ **n** – kategoria do której ma być dodany obiekt X_*

Function *cobweb*(X_*, **n**)

If n jest liściem **then**

1) *Dodaj_do_liścia*(**(**X_*, **n**);

2) Utwórz nowy węzeł **n'** zawierający wszystkie obiekty z **n** oraz obiekt X_* i umieść **n** jako potomka **n'**;

3) Utwórz liść dla obiektu X_* jako potomka **n'**;

Else

1) *Dodaj_do_węzła*(**(**X_*, **n**);

2) Select według jakości grupowania dla węzła **n from**

a) Utwórz nowy liść L jako potomka **n** i umieść X_* w L;
b) Umieść X_* w węźle **n'** , który jest najlepszym pojemnikiem dla X_* spośród potomków **n** i wywołaj *cobweb*(X_* , **n'**);
c) Połącz dwa najlepsze pojemniki dla X_* spośród potomków węzła **n** w nowy węzeł **n'** i wywołaj *cobweb*(X_* , **n'**);
d) Podziel najlepszy pojemnik dla X_* spośród potomków węzła **n** i wywołaj *cobweb*(X_* , **n**);

End Select

End If

End Function

COBWEB – uwagi

Przed pierwszym wywołaniem procedury należy utworzyć drzewo złożone z jednego liścia, zawierającego pierwszy przykład. Następnie dla każdego kolejnego przykładu funkcja *cobweb* powinna być wywoływana z drugim argumentem będącym węzłem znajdującym się w korzeniu drzewa.

Głównym elementem algorytmu COBWEB jest odpowiednie uwzględnienie faktu dodania przykładu do węzła „rodzica” na poziomie węzłów potomnych (strategia top-down).

Klasyfikacja nowych przykładów z wykorzystaniem wyznaczonej hierarchii kategorii

Klasyfikacja nowych przykładów za pomocą funkcji oceny użyteczności grupowania (CU) – zstępowanie po ścieżce od korzenia drzewa, wzdłuż ścieżki wyznaczonej przez kolejne najlepsze pojemniki dla klasyfikowanego przykładu, aż do osiągnięcia pożądanego poziomu.

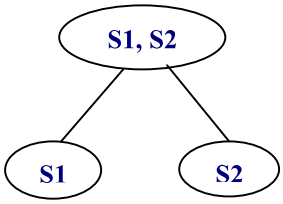
COBWEB - przykład grupowania

Dane wejściowe:

Lp.	Klasa	Cena	Osiągi	Niezawodność
S1	miejski	niska	Słabe	Mała
S2	duży	niska	Słabe	Mała
S3	kompakt	niska	Dobre	Przeciętna
S4	mały	niska	Przeciętne	Mała
S5	mały	umiarkowana	Przeciętne	Przeciętna
S6	kompakt	umiarkowana	Przeciętne	Przeciętna
S7	miejski	umiarkowana	Przeciętne	Przeciętna
S8	mały	umiarkowana	Dobre	Duża
S9	kompakt	wysoka	Dobre	Duża
S10	duży	wysoka	Przeciętne	Przeciętna
S11	duży	wysoka	Dobre	Duża

Drzewo jest inicjowane jako liść zawierający przykład S1.

Wywołanie funkcji *cobweb* dla przykładu S2 powoduje powstanie drzewa:



Dla przykładu S3 analizuje się sytuacje:
umieszczenie przykładu S3 w pojemniku z przykładem S1 lub S2
oraz utworzenie nowej kategorii dla przykładu S3.

Wynik dodania przykładu do istniejącej kategorii:

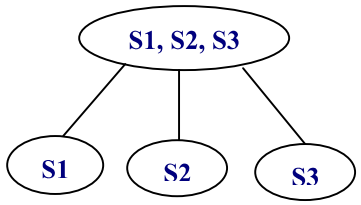
a) dodanie przykładu S3 do pojemnika z przykładem S1

$$CU=\frac{1}{2}*\begin{bmatrix}\frac{2}{3}*\left(\frac{1}{2^2}*\left(1^2+1^2+2^2+1^2+1^2+1^2+1^2\right)-\frac{1}{3^2}*\left(1^2+1^2+1^2+3^2+2^2+1^2+2^2+1^2\right)\right)+\\ \frac{1}{3}*\left(\frac{1}{1^2}*\left(1^2+1^2+1^2+1^2\right)-\frac{1}{3^2}*\left(1^2+1^2+1^2+3^2+2^2+1^2+2^2+1^2\right)\right)\end{bmatrix}=0.2(7)$$

b) dodanie przykładu S3 do pojemnika z przykładem S2: CU=0.27

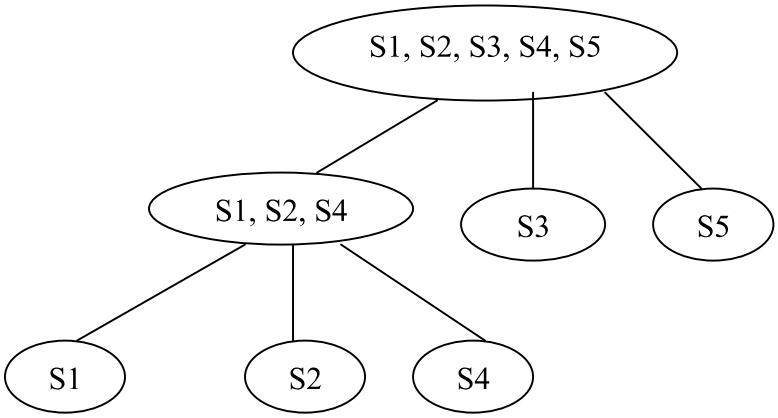
c) Utworzenia nowej kategorii dla przykładu S3: CU=0.518

Najlepszym rozwiązaniem jest sytuacja c):



Drzewo po przetworzeniu trzech przykładów

Po dodaniu przykładu S4 i S5 uzyskano strukturę przedstawioną poniżej



Analiza dodania przykładu S6:

a) Dodanie do kategorii z przykładami S1,S2,S4: CU=0.314

b) Dodanie do kategorii z przykładem S3: CU=0.463

c) Dodanie do kategorii z przykładem S5: CU=0.518

d) Utworzenie nowego pojemnika dla przykładu S6: CU=0.431

e) Połączenie dwóch najlepszych pojemników dla przykładu S6 (tj. pojedynczych kategorii z S3 i S5): CU=0.527

Najkorzystniejsza jest sytuacja e).

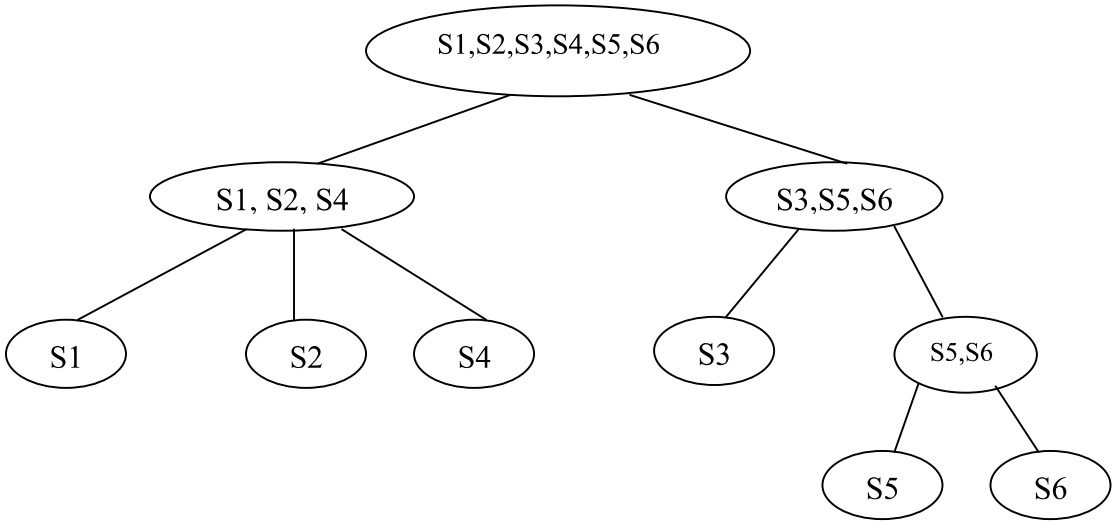
Dalszej analizie podlega poddrzewo związane z połączonym węzłem (e):

przykład S6 przydzielony do pojemnika z S3: CU=0.33

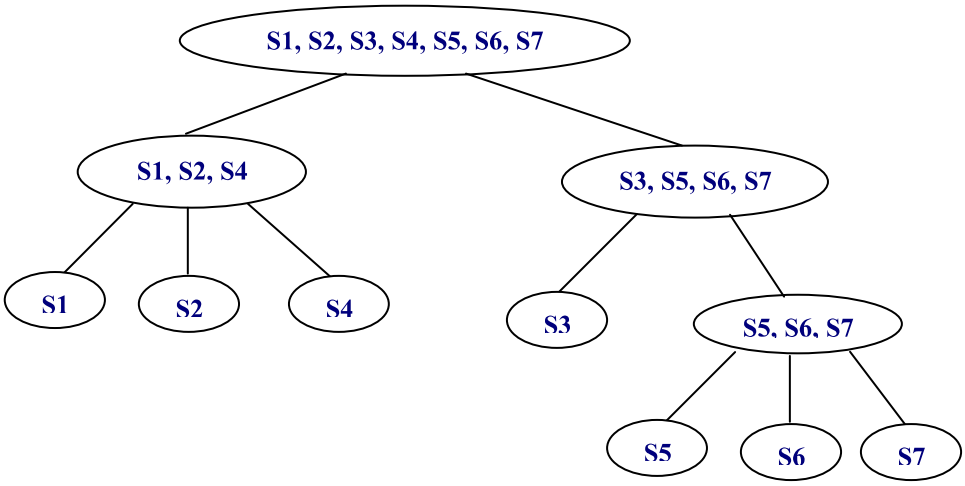
przykład S6 przydzielony do pojemnika z S5: CU=0.5

utworzenie nowej kategorii dla S6: CU=0.44

W rezultacie powstaje drzewo:



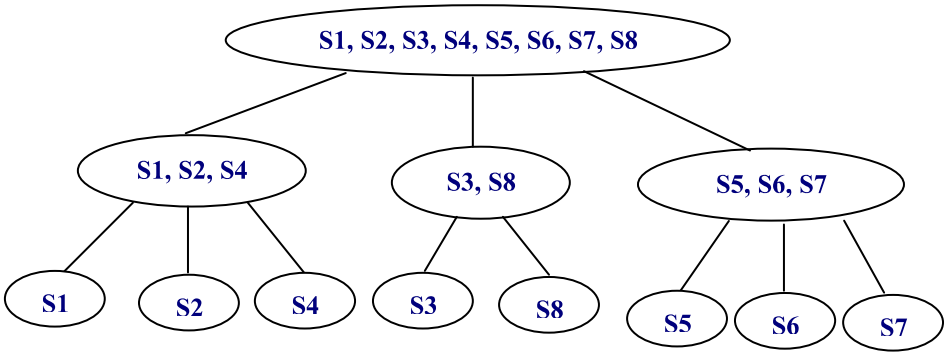
Po przetworzeniu siedmiu przykładów powstanie drzewo o poniższej strukturze:



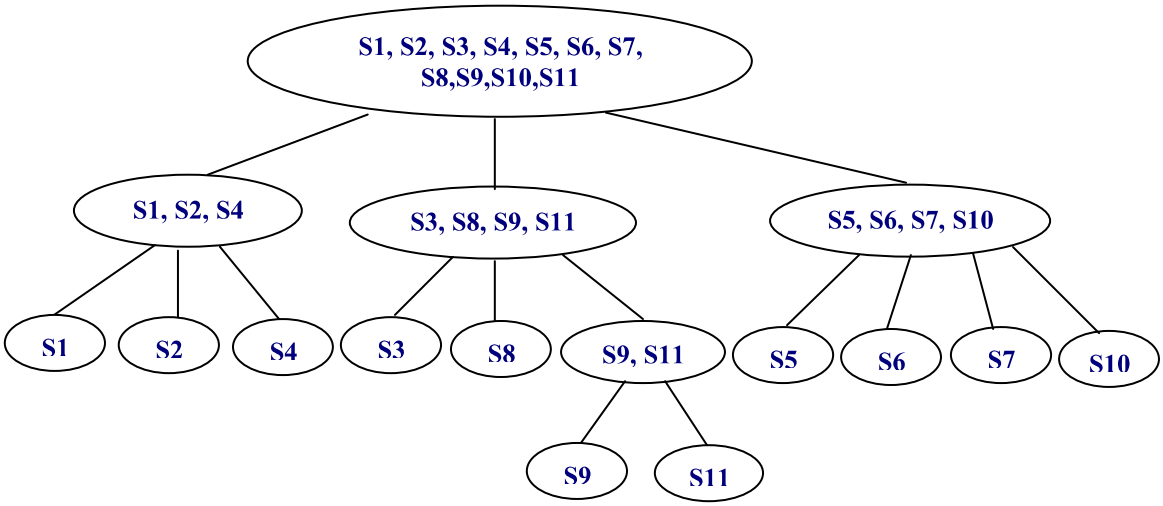
Drzewo po przetworzeniu siedmiu przykładów

Analiza dodawania przykładu S8:

Wygląd drzewa grupowania po przetworzeniu przykładu S8:



Wygląd drzewa grupowania po przetworzeniu wszystkich jedenastu przykładów:



A COBWEB clustering for four one-celled organisms (Gennari et al.,1989)