w wykładzie wykorzystano:

1. *Internetowy Podręcznik Statystyki,* http://www.statsoft.pl/textbook/stathome.html
2. Dr Hab. Hung Son Nguyen, *Clustering - Efektywne metody grupowania danych,* wykład
3. Dr inż. Agata Kołakowska, *Systemy uczące się a inne dziedziny nauki*

Analiza Skupień 2

Cluster analysis

Krzysztof Regulski, WIMiIP, KISiM,
regulski@agh.edu.pl
B5, pok. 408

Rodzaje modeli:

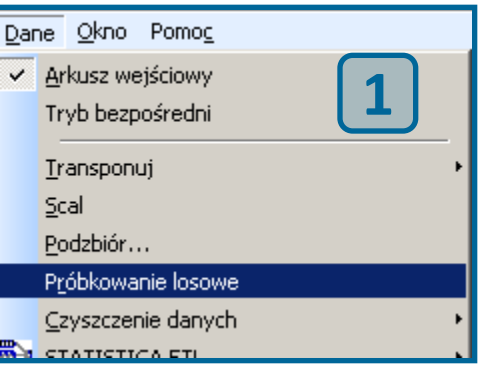
- metoda k-średnich,
- metody hierarchiczne,
- grupowanie probabilistyczne - algorytm EM, COWEB
- algorytm BIRCH, ROCK
- grupowanie oparte na gęstości
- sieci Kohonena

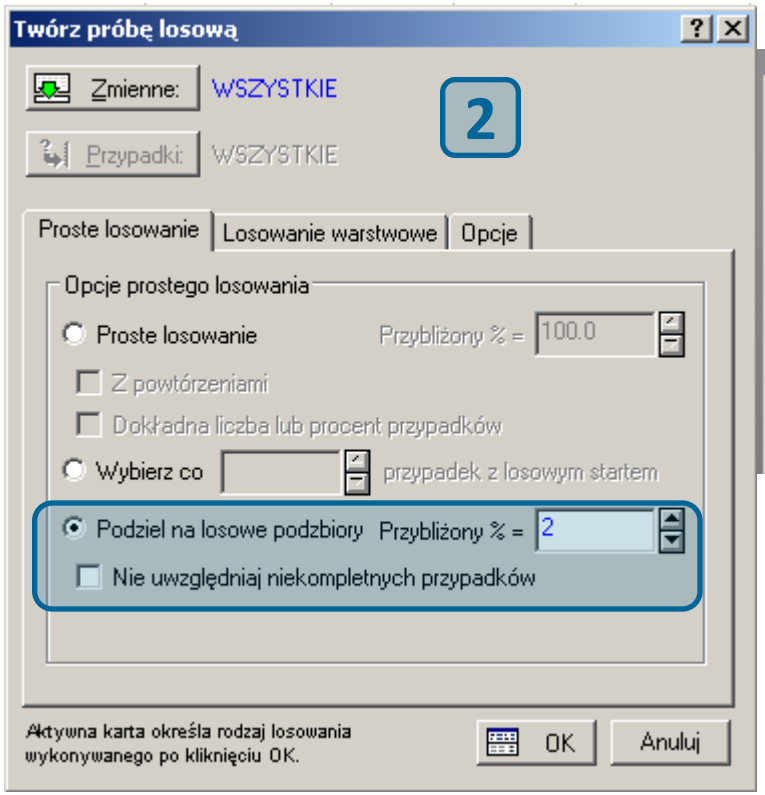
Cele grupowania

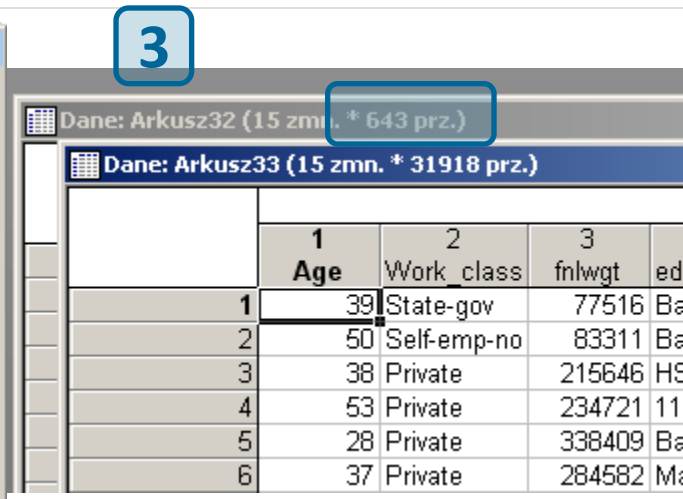
- Poznanie rozkładu przykładów (danych).
- Wyróżnienie przypadków, które można uznać za typowe lub za wyjątki.
- Znajdowanie naturalnego podziału danych na istotne podgrupy.
- Dekompozycja danych na części, które są łatwiejsze do opisania – bardziej jednolite.

STATISTICA –
przykład metody k-średnich

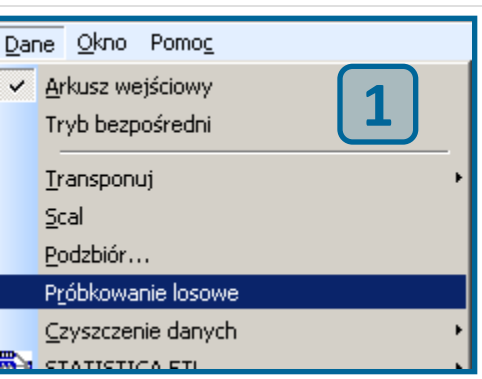
Przykład w STATISTICA

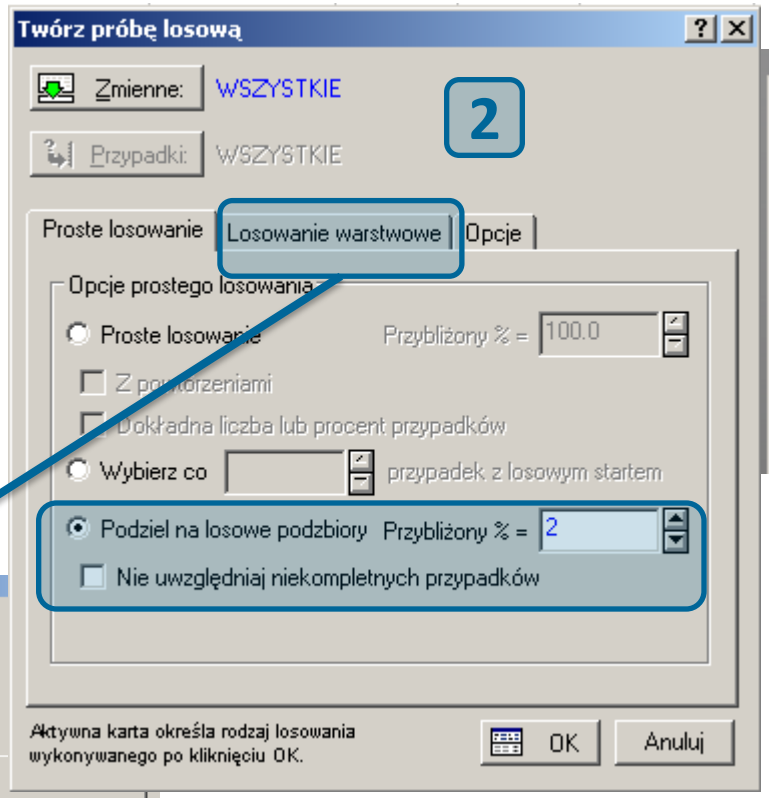


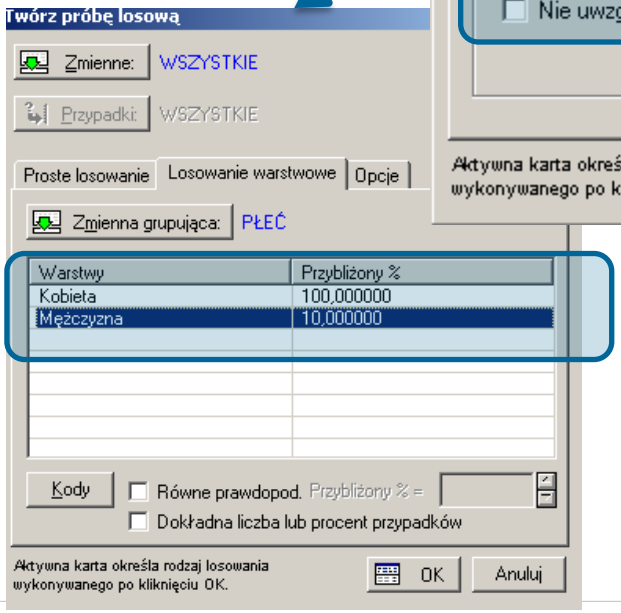


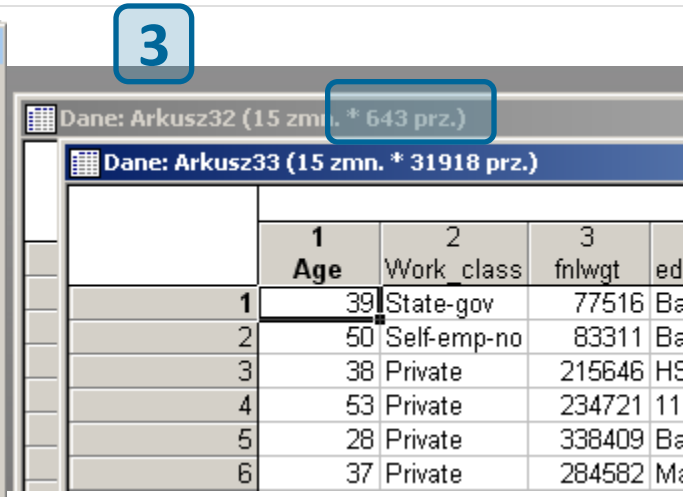


Przykład w STATISTICA

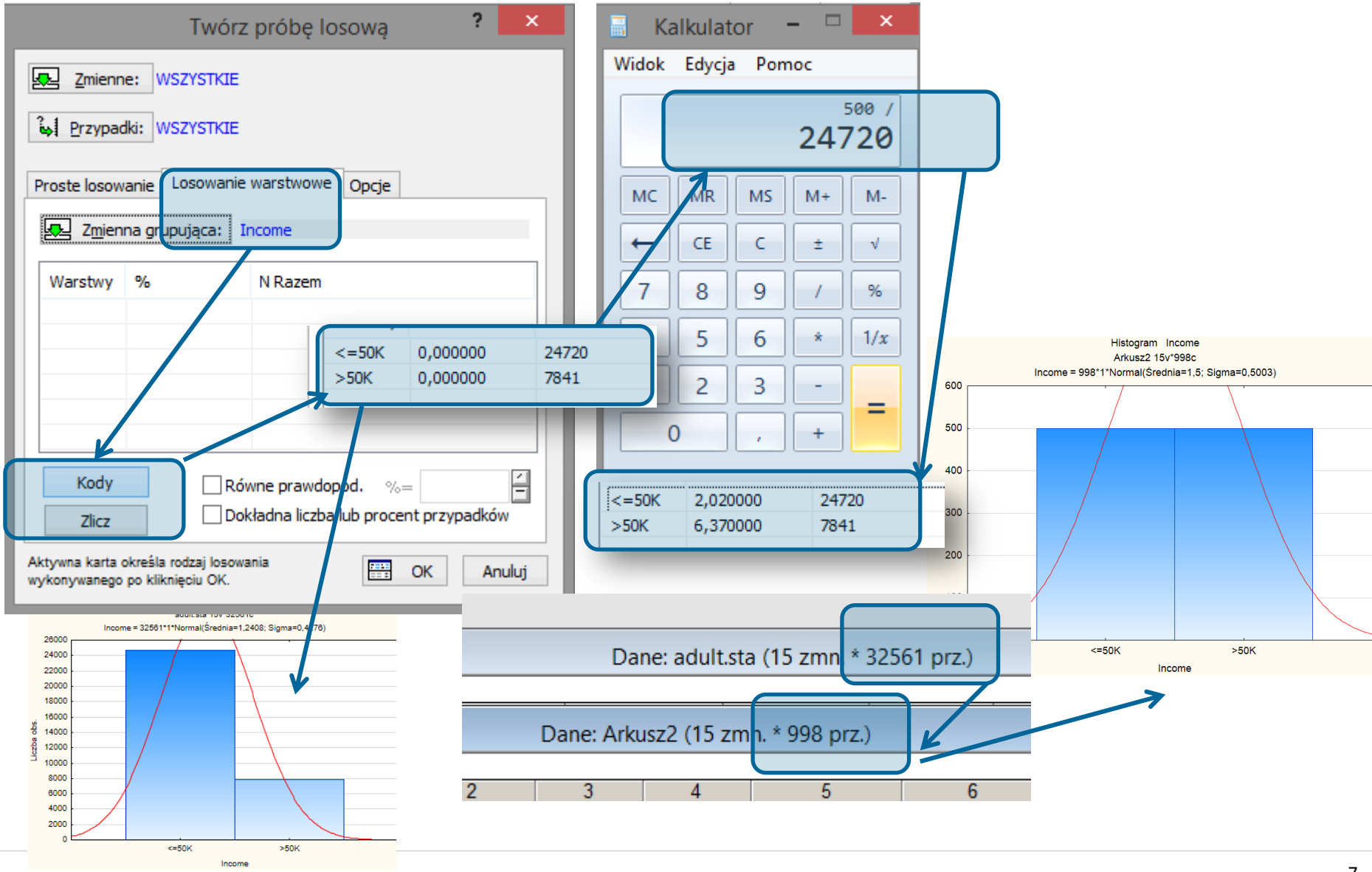


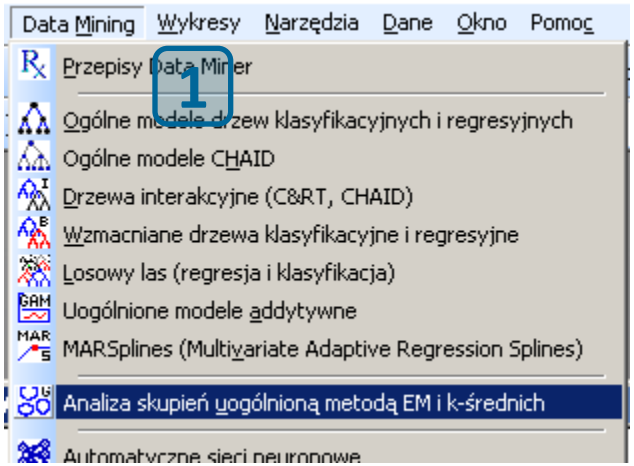


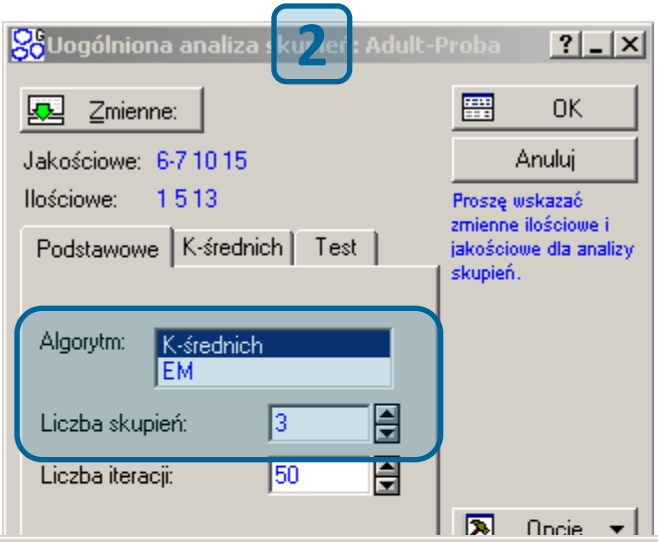


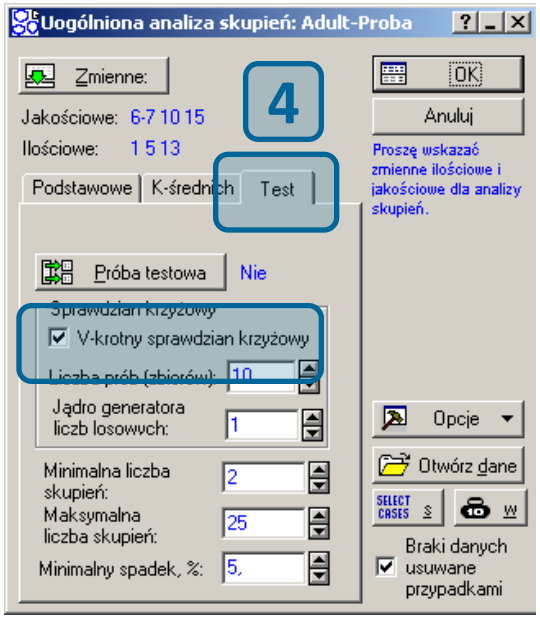


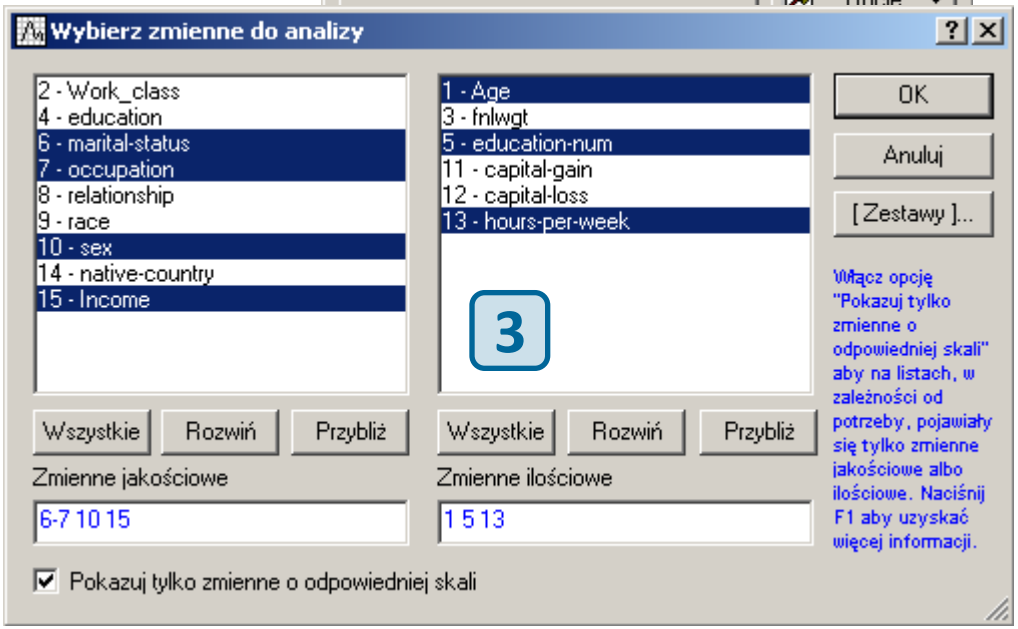
Losowanie warstwowe

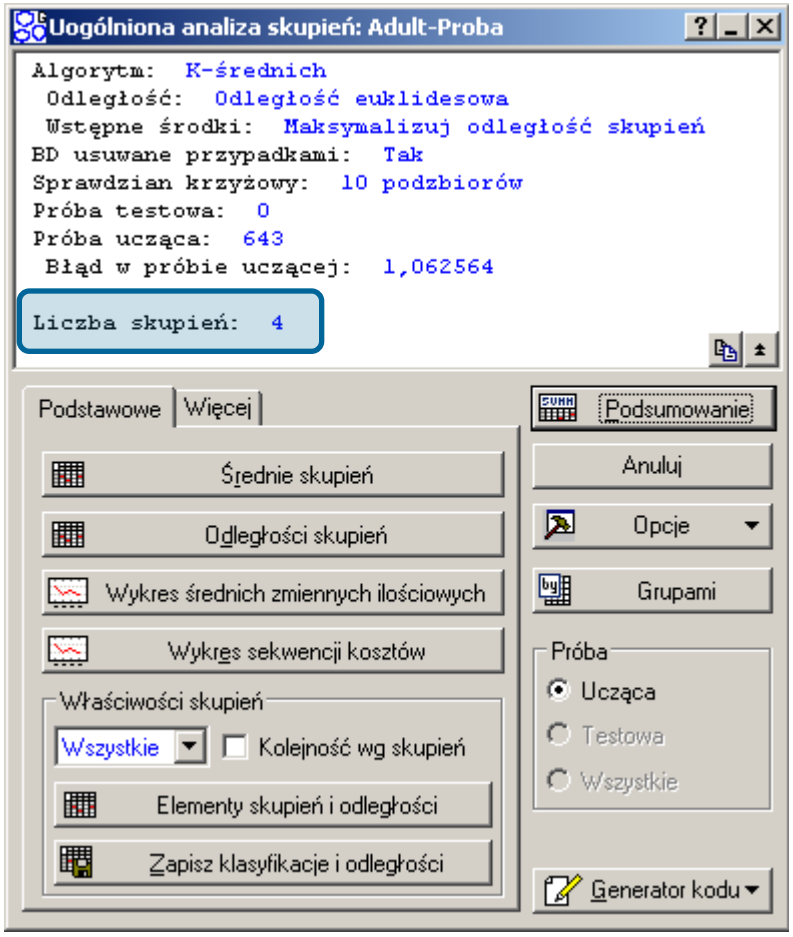


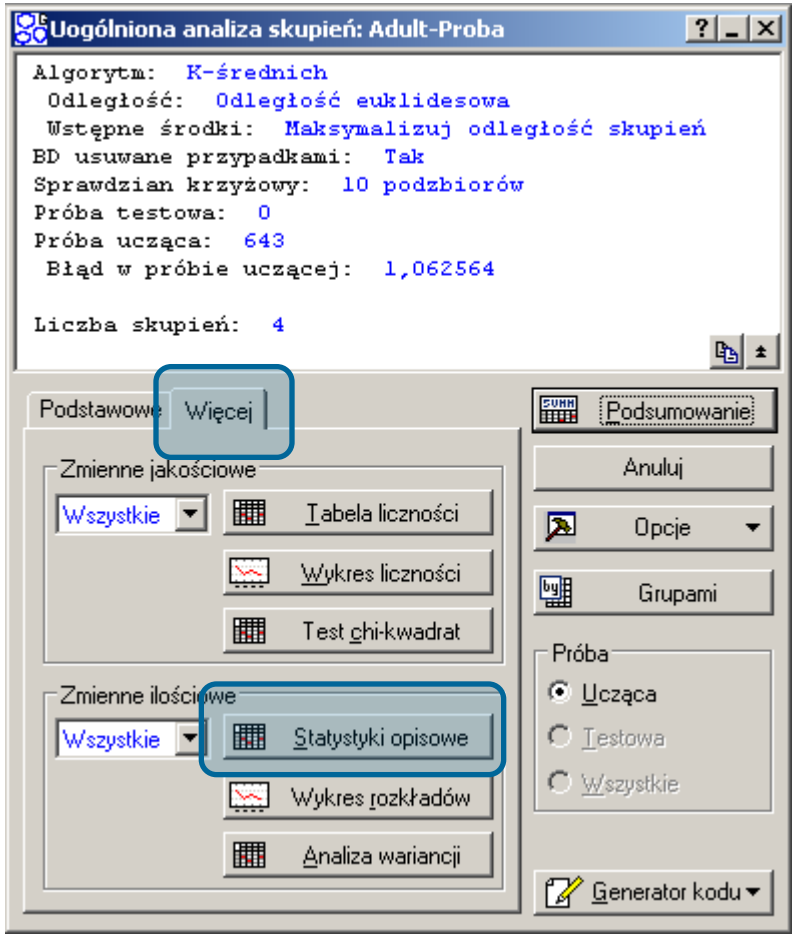

Data Mining
Wykresy
Narzędzia
Dane
Okno
Pomoc
Przepisy Data Miner
Ogólne modele drzew klasyfikacyjnych i regresyjnych
Ogólne modele CHAID
Drzewa interakcyjne (C&RT, CHAID)
Wzmacniane drzewa klasyfikacyjne i regresyjne
Losowy las (regresja i klasyfikacja)
Uogólnione modele addytywne
MARSplines (Multivariate Adaptive Regression Splines)
Analiza skupień uogólnioną metodą EM i k-średnich
Automatyczne sieci neuronowe
1


Uogólniona analiza skupień: Adult-Proba
Zmienne:
OK
Anuluj
Jakościowe: 6-7 10 15
Ilościowe: 1 5 13
Podstawowe
K-średnich
Test
Algorytm:
K-średnich
EM
Liczba skupień: 3
Liczba iteracji: 50
Proszę wskazać zmienne ilościowe i jakościowe dla analizy skupień.
Opcje
2


Uogólniona analiza skupień: Adult-Proba
Zmienne:
OK
Anuluj
Jakościowe: 6-7 10 15
Ilościowe: 1 5 13
Podstawowe
K-średnich
Test
Próba testowa
Nie
V-krotny sprawdzian krzyżowy
Liczba prób (zbiorów): 10
Jądro generatora liczb losowych: 1
Minimalna liczba skupień: 2
Maksymalna liczba skupień: 25
Minimalny spadek, %: 5,
Opcje
Otwórz dane
Braki danych usuwane przypadkami
Proszę wskazać zmienne ilościowe i jakościowe dla analizy skupień.
4


Wybierz zmienne do analizy
2 - Work_class
4 - education
6 - marital-status
7 - occupation
8 - relationship
9 - race
10 - sex
14 - native-country
15 - Income
1 - Age
3 - fnlwgt
5 - education-num
11 - capital-gain
12 - capital-loss
13 - hours-per-week
OK
Anuluj
[Zestawy]...
Wszystkie
Rozwiń
Przybliż
Zmienne jakościowe
6-7 10 15
Zmienne ilościowe
1 5 13
Pokazuj tylko zmienne o odpowiedniej skali
Włącz opcję "Pokazuj tylko zmienne o odpowiedniej skali" aby na listach, w zależności od potrzeby, pojawiały się tylko zmienne jakościowe albo ilościowe. Naciśnij F1 aby uzyskać więcej informacji.
3


Uogólniona analiza skupień: Adult-Proba
Algorytm: K-średnich
Odległość: Odległość euklidesowa
Wstępne środki: Maksymalizuj odległość skupień
BD usuwane przypadkami: Tak
Sprawdzian krzyżowy: 10 podzbiorów
Próba testowa: 0
Próba ucząca: 643
Błąd w próbie uczącej: 1,062564
Liczba skupień: 4
Podstawowe
Więcej
Podsumowanie
Średnie skupień
Odległości skupień
Wykres średnich zmiennych ilościowych
Wykres sekwencji kosztów
Właściwości skupień
Wszystkie
Kolejność wg skupień
Elementy skupień i odległości
Zapisz klasyfikacje i odległości
Anuluj
Opcje
Grupami
Próba
Ucząca
Testowa
Wszystkie
Generator kodu


Uogólniona analiza skupień: Adult-Proba
Algorytm: K-średnich
Odległość: Odległość euklidesowa
Wstępne środki: Maksymalizuj odległość skupień
BD usuwane przypadkami: Tak
Sprawdzian krzyżowy: 10 podzbiorów
Próba testowa: 0
Próba ucząca: 643
Błąd w próbie uczącej: 1,062564
Liczba skupień: 4
Podstawowe
Więcej
Podsumowanie
Zmienne jakościowe
Wszystkie
Tabela liczności
Wykres liczności
Test chi-kwadrat
Zmienne ilościowe
Wszystkie
Statystyki opisowe
Wykres rozkładów
Analiza wariancji
Anuluj
Opcje
Grupami
Próba
Ucząca
Testowa
Wszystkie
Generator kodu

Statystyki opisowe skupień

Statystyki zmiennej ilościowej Age (Adult-Proba)
Liczba skupień: 4
Całkowita liczba przypadków uczących: 643

	Skupienie 1	Skupienie 2	Skupienie 3	Skupienie 4	Ogółem
Minimum	17,00000	22,00000	18,00000	21,00000	17,00000
Maksimum	73,00000	90,00000	80,00000	90,00000	90,00000
Średnia	30,30736	45,48299	40,13253	45,07071	38,58631
Odchylenie std.	11,92972	11,76502	13,12113	12,85417	13,94951

– Wniosek 1: skupienie 1. to ludzie młodsi (30lat), skupienia 2 i 4 to starsi (45 lat), skupienie 3 (40lat)

Statystyki zmiennej ilościowej education-num (Adult-Proba)
Liczba skupień: 4
Całkowita liczba przypadków uczących: 643

	Skupienie 1	**Skupienie 2**	Skupienie 3	Skupienie 4	Ogółem
Minimum	1,00000	6,00000	2,00000	3,00000	1,00000
Maksimum	15,00000	16,00000	14,00000	14,00000	16,00000
Średnia	9,74459	11,87755	9,19880	9,35354	10,03110
Odchylenie std.	2,26690	2,23269	2,26494	2,33567	2,48716

– Wniosek 2: skupienie 2. to ludzie lepiej wykształceni niż skupienia 1., 3. i 4.

Statystyki zmiennej ilościowej hours-per-week (Adult-Proba)
Liczba skupień: 4
Całkowita liczba przypadków uczących: 643

	Skupienie 1	Skupienie 2	Skupienie 3	Skupienie 4	Ogółem
Minimum	4,00000	10,00000	5,00000	4,00000	4,00000
Maksimum	99,00000	84,00000	99,00000	99,00000	99,00000
Średnia	35,60606	45,14286	43,72892	41,85859	40,84603
Odchylenie std.	12,51350	11,40115	13,48732	11,09974	12,94728

– Wniosek 3: najmniej pracują osoby ze skupienia 1, najciężej ze skupienia 2.; skupienia 3 i 4 pracują normalnie – ok. 40h/tydzień

Co już wiemy?

Podstawowe | Wi

Wykres średnich zmiennych ilościowych

Skupienie 1:

- młodzi (30lat);
- średnio wykształceni;
- pracujący stosunkowo mało;

Skupienie 2:

- w średnim wieku (ok. 45lat);
- bardzo dobrze wykształceni;
- pracujący dużo;

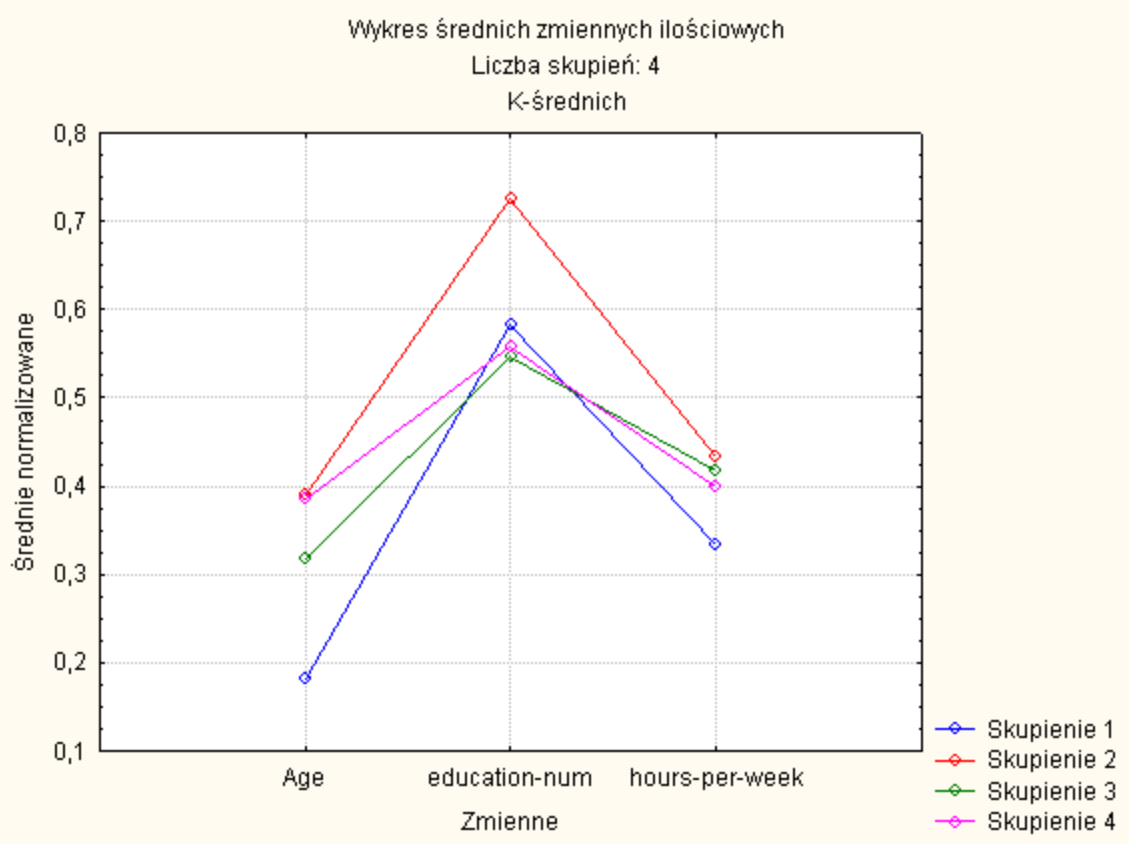
Skupienie 3:

- w wieku ok. 40 lat;
- słabo wykształceni;
- pracujący stosunkowo dużo;

Skupienie 4:

- w średnim wieku (ok. 45lat);
- raczej słabo wykształceni;
- pracujący ok. 41h/tyg

kto jest najlepszym klientem?



Podstawowe | Więcej

Średnie skupień

Średnie skupień (metoda k-średnich) (Adult-Proba)
Liczba skupień: 4
Całkowita liczba przypadków uczących: 643

Skupienie	marital-status	occupation	sex	Income	Age	education-num	hours-per-week	Liczba przypadków	Procent (%)
1	Never-married	Other-service	Female	<=50K	30,30736	9,74459	35,60606	231	35,92535
2	Married-civ-spouse	Prof-specialty	Male	>50K	45,48299	11,87755	45,14286	147	22,86159
3	Married-civ-spouse	Sales	Male	<=50K	40,13253	9,19880	43,72892	166	25,81649
4	Divorced	Craft-repair	Male	<=50K	45,07071	9,35354	41,85859	99	15,39658

Tabela liczności dla zmiennej jakościowej: occupation (Adult-Prob
Liczba skupień: 4
Całkowita liczba przypadków uczących: 643

	Skupienie 1	**Skupienie 2**	Skupienie 3	Skupienie 4	Razem
Other-service	52	1	8	0	61
Sales	15	9	45	2	71
Craft-repair	3	17	15	46	81
Machine-op-inspct	9	2	15	4	30
Farming-fishing	2	3	14	2	21
Prof-specialty	22	47	2	8	79
Tech-support	7	9	2	2	20
Adm-clerical	39	9	11	11	70
Handlers-cleaners	16	5	8	2	31
?	23	6	10	6	45
Transport-moving	13	6	17	5	41
Armed-Forces	0	0	1	0	1
Exec-managerial	26	25	18	7	76
Protective-serv	2	8	0	4	14
Priv-house-serv	2	0	0	0	2

Najlepiej zarabia skupienie 2 – kim ONI są?

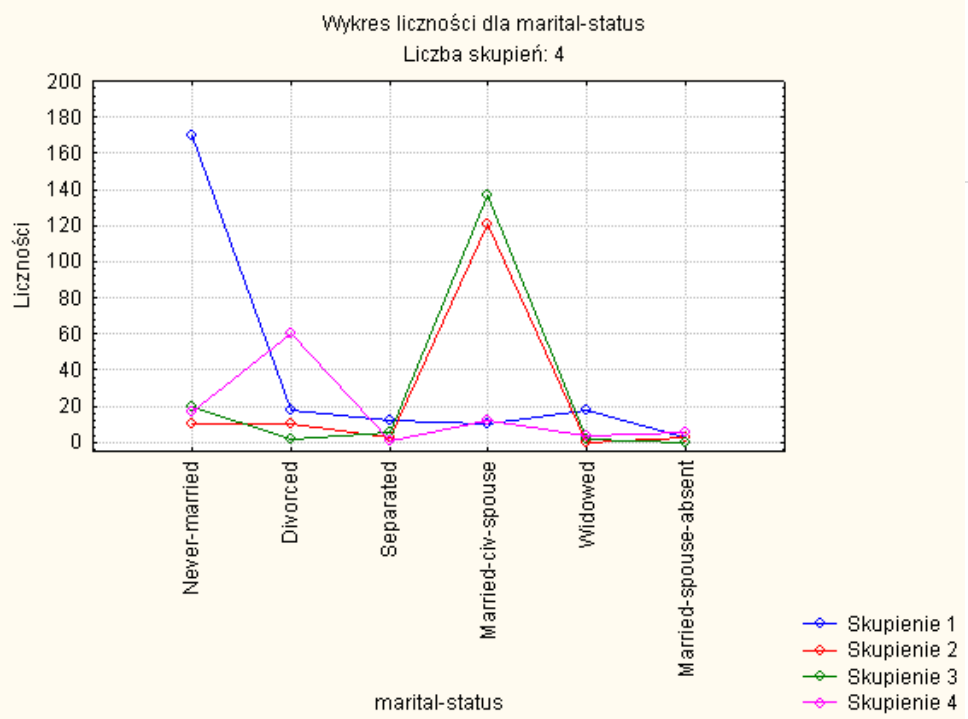
Skupienie 2:

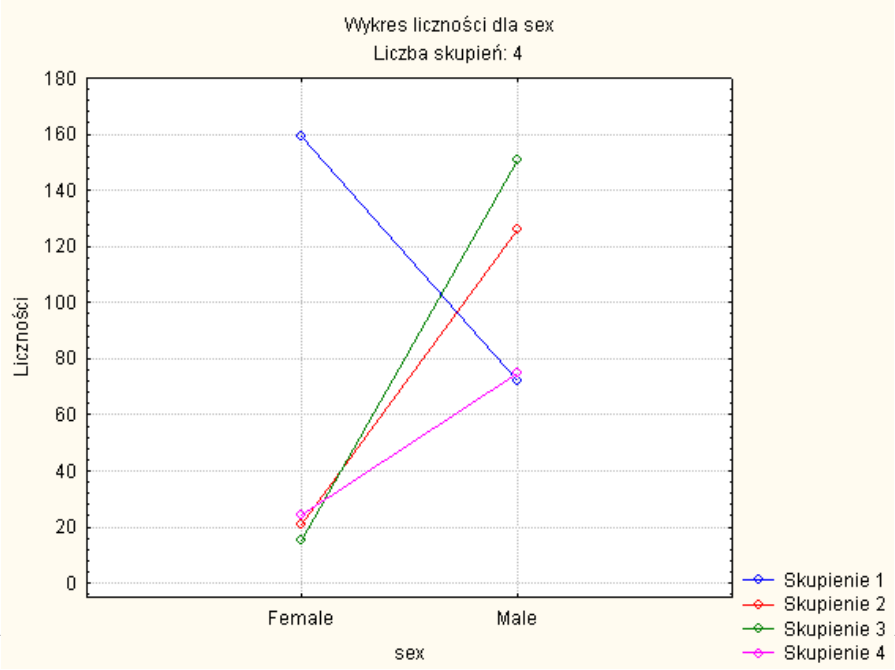
- w średnim wieku (ok. 45lat);
- bardzo dobrze wykształceni;
- pracujący dużo;

Co nowego o skupieniu 2?

Skupienie 2 to:

- ludzie średnim wieku (ok. 45lat);
- bardzo dobrze wykształceni;
- pracujący dużo;
- w małżeństwie;
- zawód: specjalista/kierownik
- mężczyzna

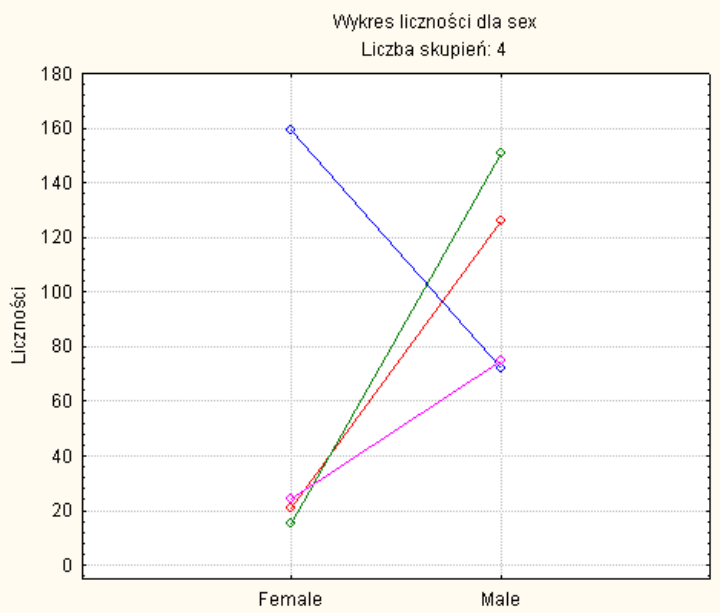


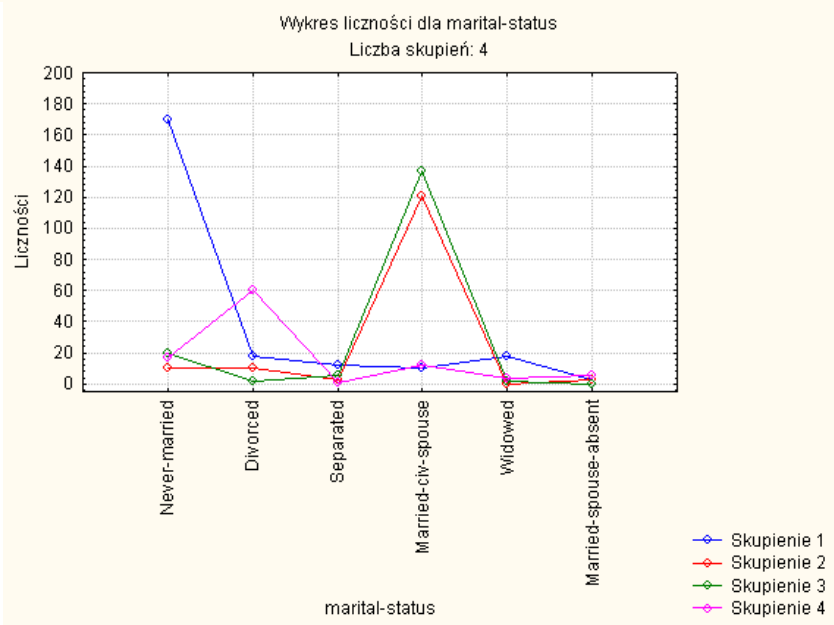


Co z kobietami?

Kobiety w naszej próbie zdominowały skupienie 1:

- są młode (30lat);
- średnio wykształcone;
- pracujące stosunkowo mało (dzieci?);
- niezamężne;
- pracują w usługach;
- prawie wszystkie zarabiają poniżej 50 000$;

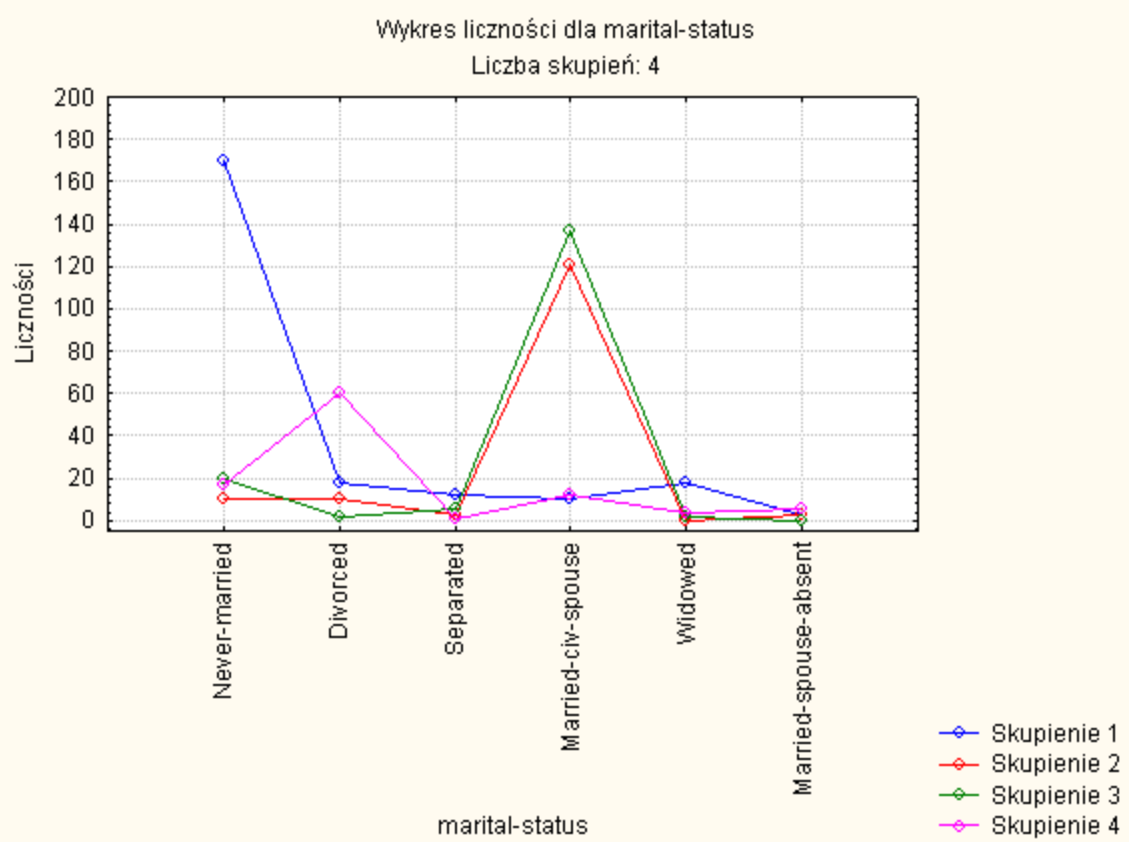




Kim są pozostali?

Skupienie 3:

- w wieku ok. 40 lat;
- słabo wykształceni;
- pracujący stosunkowo dużo;
- mężczyźni;
- żonaci;
- głównie sprzedawcy;
- zarabiają poniżej 50 000$



Skupienie 4:

- w średnim wieku (ok. 45lat); raczej słabo wykształceni; pracujący ok. 41h/tyg;
- kobiety i mężczyźni po rozwodzie,
- rzemieślnicy, fachowcy…
- zarabiający poniżej 50 000$

Kto się najbardziej różni?

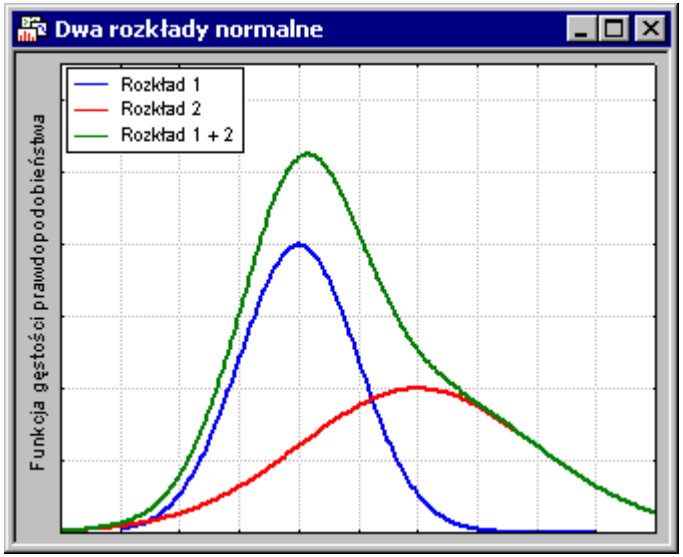
- Najbardziej rozróżnialne są skupienia 1. i 2. – czyli młodzi specjaliści i panny na wydaniu
- Widać to również w zarobkach.
- Po ślubie różnice się wyrównują…

	Standaryzowana odległość między centroidami k-średnich Liczba skupień: 4			
	Skupienie 1	Skupienie 2	Skupienie 3	Skupienie 4
Skupienie 1	0,000000	2,018295	1,739756	1,745254
Skupienie 2	2,018295	0,000000	1,427405	1,740558
Skupienie 3	1,739756	1,427405	0,000000	1,416005
Skupienie 4	1,745254	1,740558	1,416005	0,000000

Probabilistyczny algorytm EM
Expectation Maximisation

Probabilistyczny algorytm EM
Expectation Maximisation

- Przykład: obserwujemy dużą próbę pomiarów jednej zmiennej ilościowej.
- Zamiast patrzyć tylko na odległość, uwzględniamy dodatkowo informację o rozkładzie przykładów.



- Zamiast przypisywać definitywnie przykład do grupy estymuje prawdopodobieństwo takiego przynależenia.
- różne rozkłady, jak np. rozkład normalny , logarytmiczno-normalny czy Poissona . Możemy także wybrać różne rozkłady dla różnych zmiennych i stąd, wyznaczać grupy z mieszanin różnych typów rozkładów.

Algorytm EM

- **Zmienne jakościowe.** Implementacja algorytmu *EM* potrafi korzystać ze zmiennych jakościowych. Najpierw losowo przydziela prawdopodobieństwa (wagi) każdej z klas (kategorii), w każdym ze skupień. W kolejnych iteracjach prawdopodobieństwa są poprawiane tak, by zmaksymalizować wiarygodność danych przy podanej ilości skupień.
- **Prawdopodobieństwa klasyfikacyjne zamiast klasyfikacji.** Wyniki analizy skupień metodą *EM* są inne niż obliczone metodą *k*-średnich. Ta ostatnia wyznacza skupienia. Algorytm *EM* nie wyznacza przyporządkowania obserwacji do klas lecz prawdopodobieństwa klasyfikacyjne.

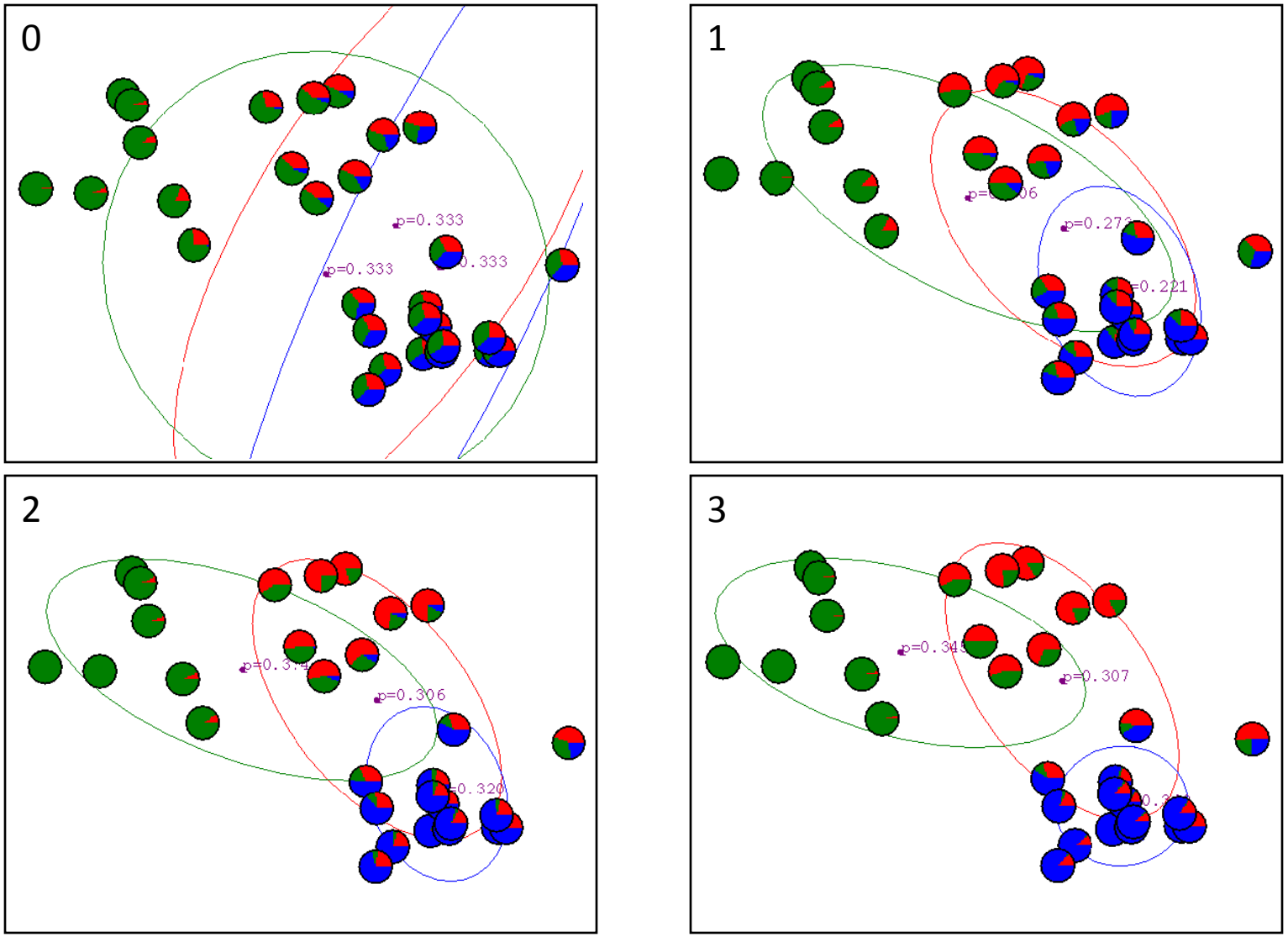
Algorytm EM

– Metoda składa się z dwu kroków wykonywanych na przemian tak długo, aż pomiędzy kolejnymi przebiegami nie dochodzi do zauważalnej poprawy.

1. **Estymacja** (*expectation*). Dla aktualnego, estymowanego układu parametrów rozkładu przykładów dokonaj przypisania przykładom prawdopodobieństwa przynależenia do grup.

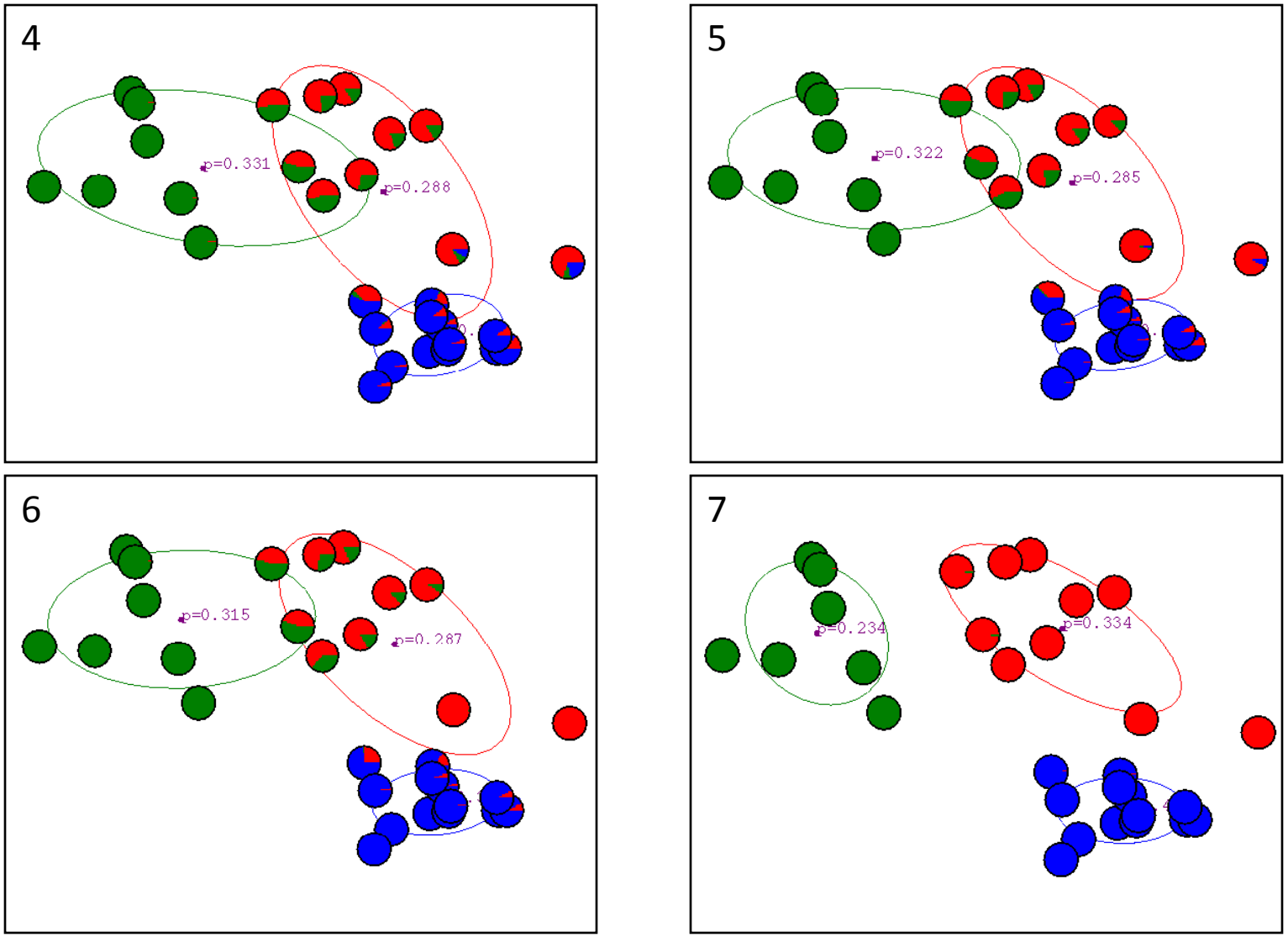
2. **Maksymalizuj** - Zamień aktualne parametry rozkładu na takie, które prowadzą do modelu bardzie zgodnego z danymi (rozkładem przykładów). W tym celu wykorzystaj prawdopodobieństwa przynależenia do grup uzyskane w kroku 1.

Gaussian mixture model



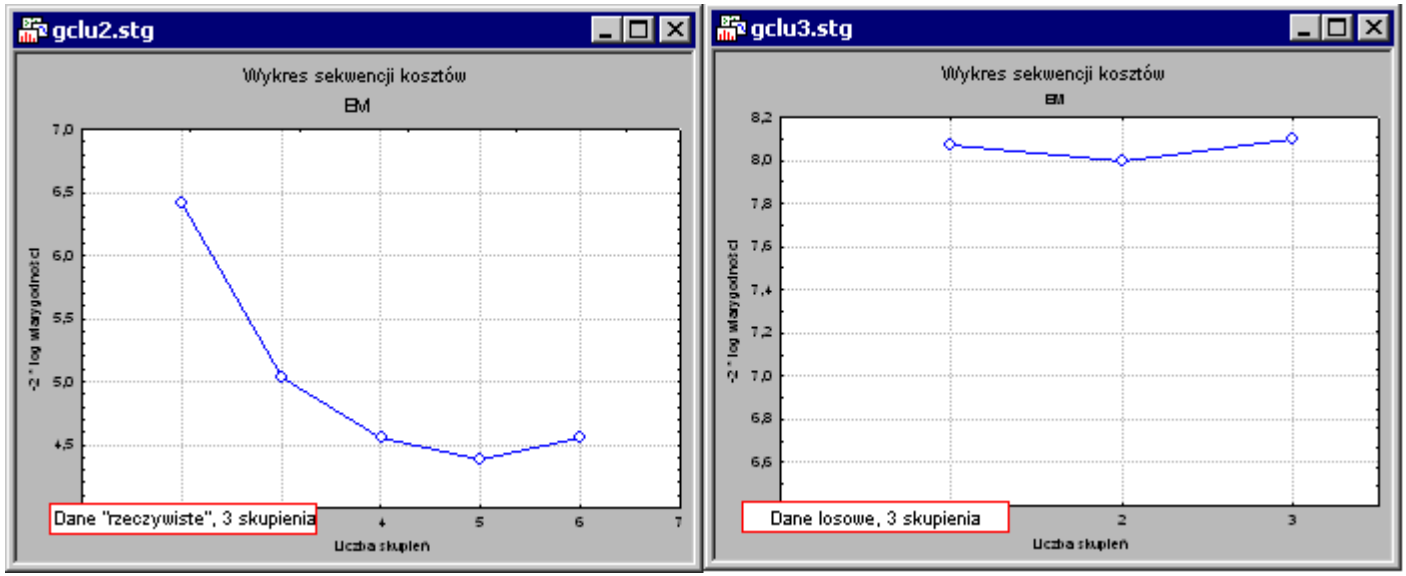
http://www.autonlab.org/tutorials/gmm.html

Gaussian mixture model



Jak dobrać *k*?

- Podobnie jak w przypadku metody k-średnich, problem dotyczy liczby skupień
- W praktyce analityk nie ma zazwyczaj pojęcia ile skupień jest w próbie. Algorytm *v-krotnego sprawdzianu krzyżowego* – automatycznie wyznacza liczbę skupień danych.



AGH

Fuzzy c-means

Fuzzy c-means

- Założenie: każdy przykład może należeć do więcej niż jednej grup.

- Macierz *U* opisująca stopień przynależności poszczególnych przykładów do grup

$$U = \begin{bmatrix} u_{11} & \mathrm{K} & u_{1c} \\ \mathrm{M} & \mathrm{O} & \mathrm{M} \\ u_{N1} & \Lambda & u_{Nc} \end{bmatrix} \qquad 0 \le u_{ij} \le 1 \qquad \sum_{j=1}^{L} u_{ij} = 1$$

N – liczba przykładów
c – liczba skupisk
u_{ij} – stopień przynależności przykładu *i*-tego do grupy *j*-tej

Fuzzy c-means

Algorytm znajduje parametry minimalizujące następującą funkcję:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{c} u_{ij}^{m} \left\| x_i - \mu_j \right\|^2$$

N – liczba przykładów
c – liczba skupisk
x_i – i-ty przykład
μ_j – środek i-tego skupiska
u_{ij} – stopień przynależności przykładu x_i do grupy j-tej
$m \geq 1$

Fuzzy c-means

1. Inicjalizuj macierz $U = [u_{ij}]$
2. Oblicz centra skupisk $C = [\mu_j]$

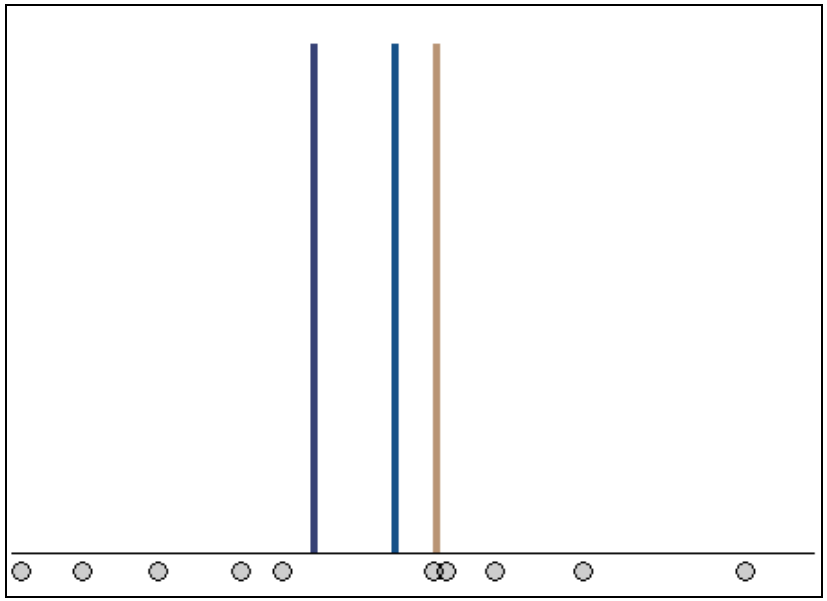
$$\mu_j = \frac{\sum_{i=1}^{N} u_{ij}^m x_i}{\sum_{i=1}^{N} u_{ij}^m}$$

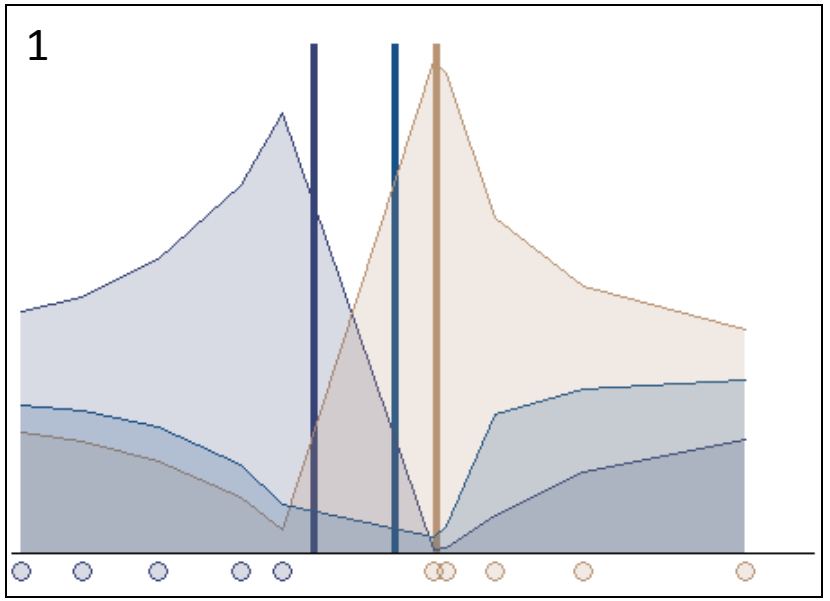
3. Aktualizuj macierz U

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|x_i - \mu_j\|}{\|x_i - \mu_k\|} \right)^{\frac{2}{m-1}}}$$

4. Jeżeli $\max_{ij}\{|u_{ij}^{(k+1)} - u_{ij}^{(k)}|\} > \varepsilon$, to wróć do kroku 2

Fuzzy c-means

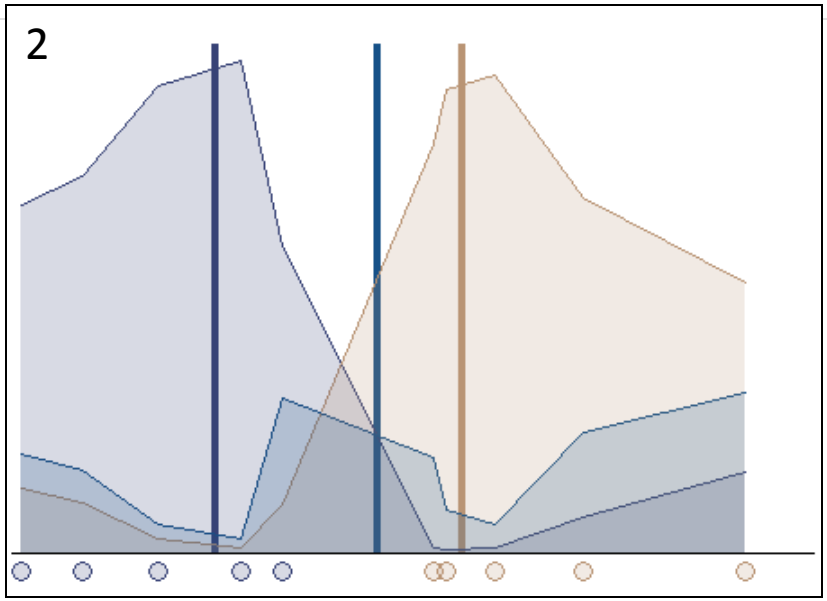


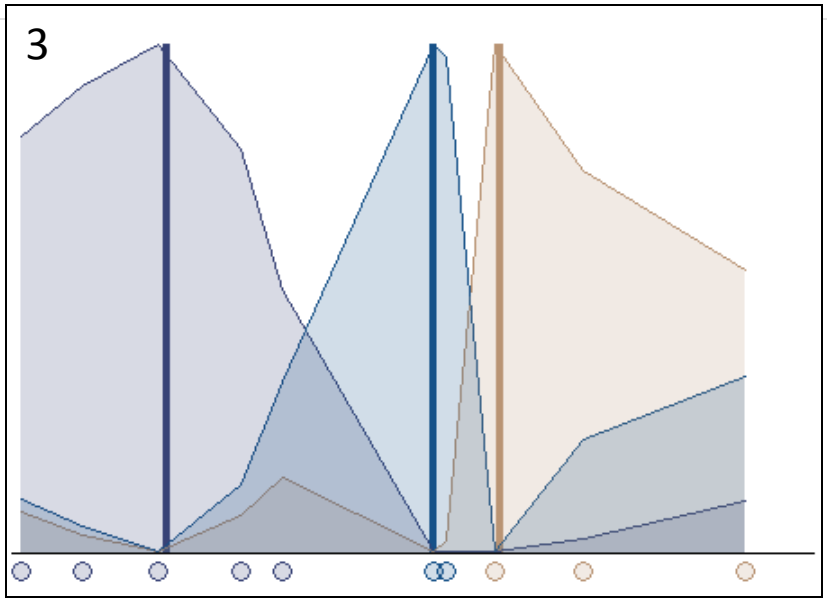


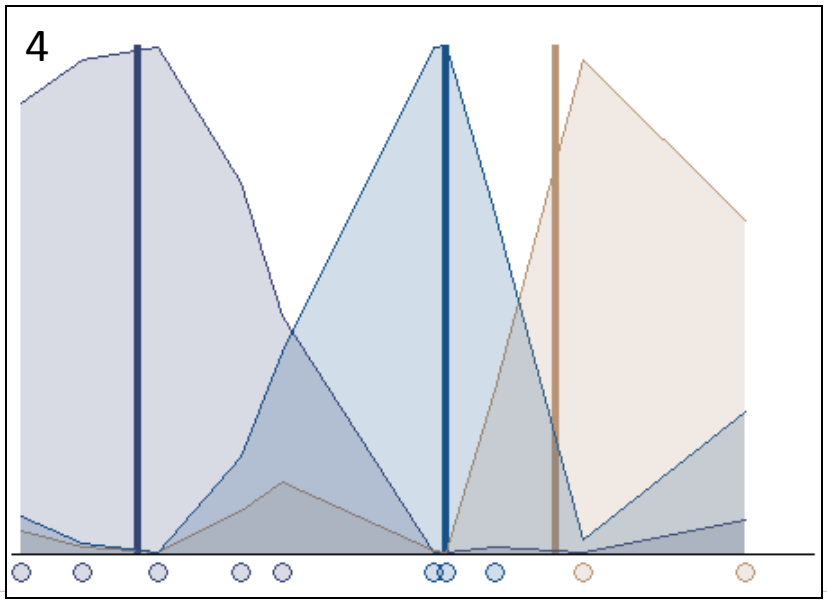
m = 2

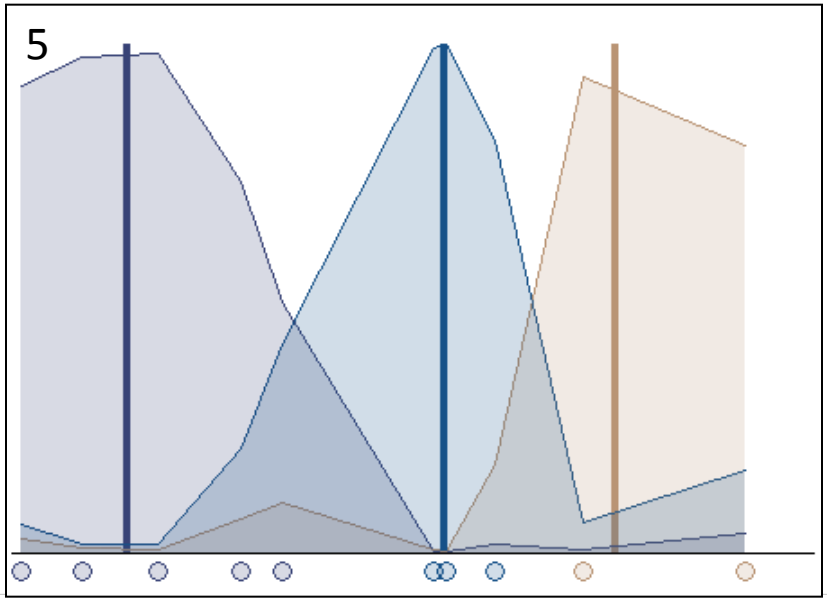
ε = 0,2

Fuzzy c-means









COBWEB – grupowanie probabilistyczne

COBWEB – grupowanie probabilistyczne

Algorytm COBWEB jako przykład przeszukiwania przestrzeni rozwiązań:

- elementy przestrzeni – różne grupowania
- funkcja oceny grupowania
- operatory do poruszania się w przestrzeni
- strategia przeszukiwania

Algorytm COBWEB/CLASSIT:

- Na początku hierarchia składa się z pojedynczego pustego węzła.
- Kolejno dodajemy przykłady i dokonujemy uaktualnienia drzewa hierarchii gdy jest to potrzebne.
- Uaktualnianie polega na przypisywaniu przykładu do właściwego węzła (liścia drzewa) hierarchii i może prowadzić do zmiany drzewa przez utworzenie nowych węzłów lub scalenie już istniejących.
- Decyzje o zmianie struktury drzewa są oparte na obserwacji zmian charakterystyki liczbowej (miary) zwanej funkcją oceny (*cathegory utility*)

COBWEB

Funkcja oceny grupowania:

$$J = \frac{1}{L}\sum_{i=1}^{L} P(c_i)[\sum_{j=1}^{N}\sum_{k} P(a_{jk} \mid c_i)^2 - \sum_{j=1}^{N}\sum_{k} P(a_{jk})^2]$$

L – liczba grup

N – liczba atrybutów

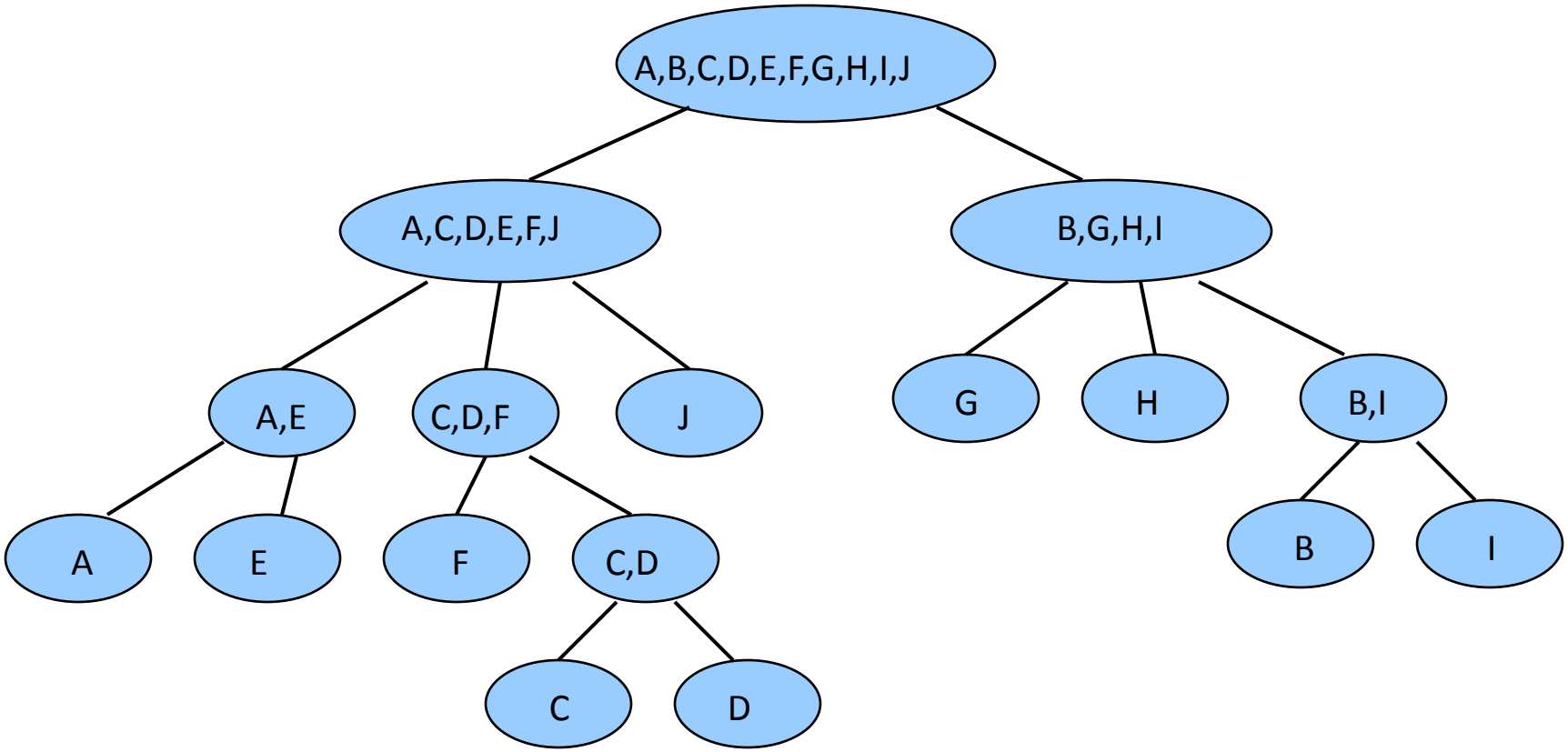
$P(a_{jk}|c_i)$ – prawdopodobieństwo tego, że dla losowo wybranego przykładu atrybut j-ty przyjmuje wartość k-tą, zakładając, że przykład należy do grupy c_i

- Funkcja ocenia przyrost oczekiwanej liczby dających się poprawnie przewidzieć wartości atrybutów przy założeniu znajomości grupowania, w stosunku do oczekiwanej liczby odgadnięć bez znajomości grupowania.
- Im większa wartość funkcji, tym lepsze grupowanie.

COBWEB

Przykłady dzielone są stopniowo na grupy.

Reprezentacja grupowania – drzewo grupowania oraz dla każdego węzła wyznaczone odpowiednie prawopodobieństwa



COBWEB

Wartość funkcji oceny jakości grupowania wyznaczana jest lokalnie.

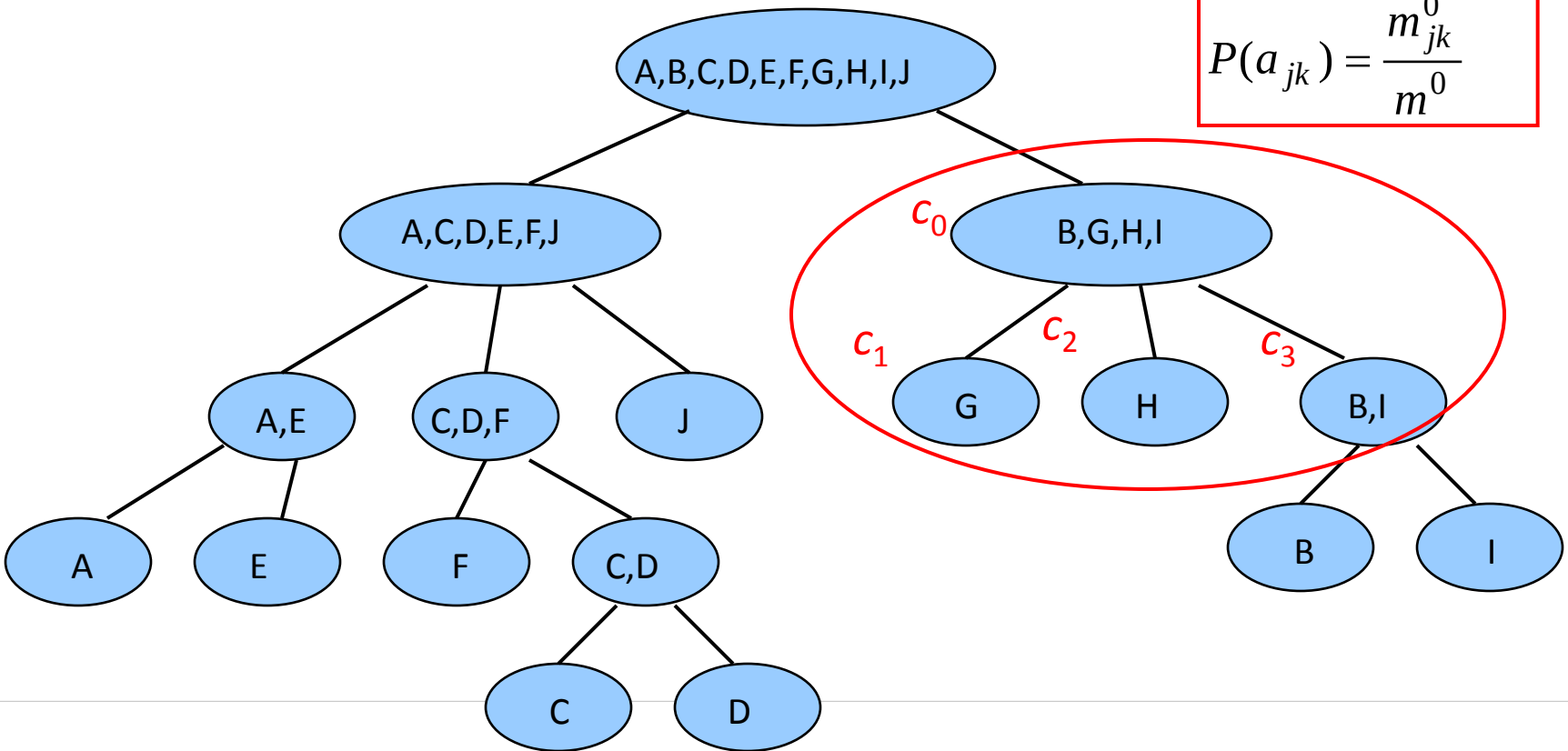
Szacowanie prawdopodobieństw:

$$P(c_i) = \frac{m^i}{m^0}$$

$$P(a_{jk} \mid c_i) = \frac{m^i_{jk}}{m^i}$$

$$P(a_{jk}) = \frac{m^0_{jk}}{m^0}$$

grupa c_0 dzielona na grupy c_1, c_2,..., c_L
m^i – liczba przykładów w grupie (węźle) i-tej
m_{jk}^i - liczba przykładów w grupie (węźle) i-tej,
dla których atrybut j-ty przyjmuje wartość k-tą

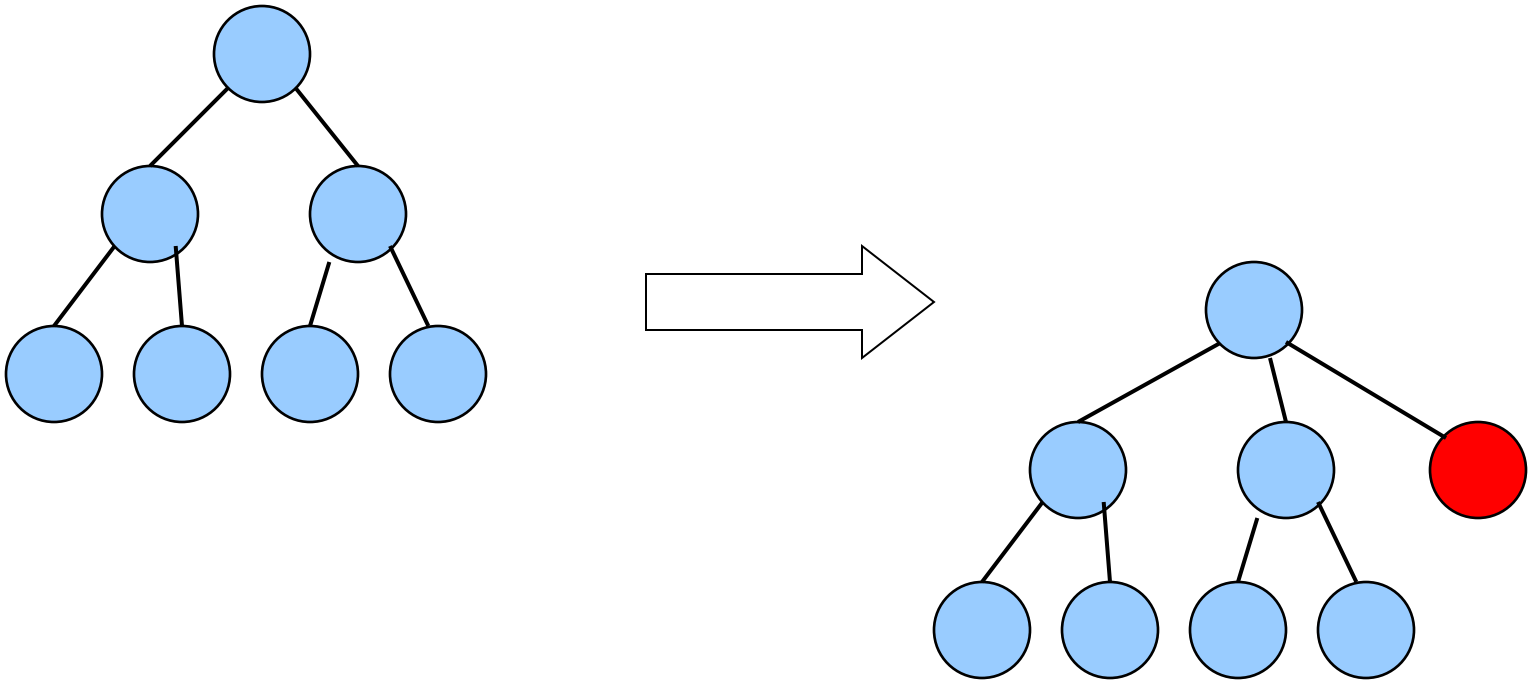


COBWEB

- Drzewo grupowania modyfikowane jest po zaobserwowaniu każdego przykładu uczącego (uczenie inkrementacyjne).
- Operatory do poruszania się w przestrzeni rozwiązań (do konstrukcji drzewa):
  - zaliczenie przykładu do jednej z istniejących grup
  - utworzenie nowej grupy dla przykładu
  - połączenie dwóch grup i umieszczenie przykładu w powstałej grupie
  - podzielenie grupy na pewną liczbę oddzielnych grup i umieszczenie przykładu w jednej z nich

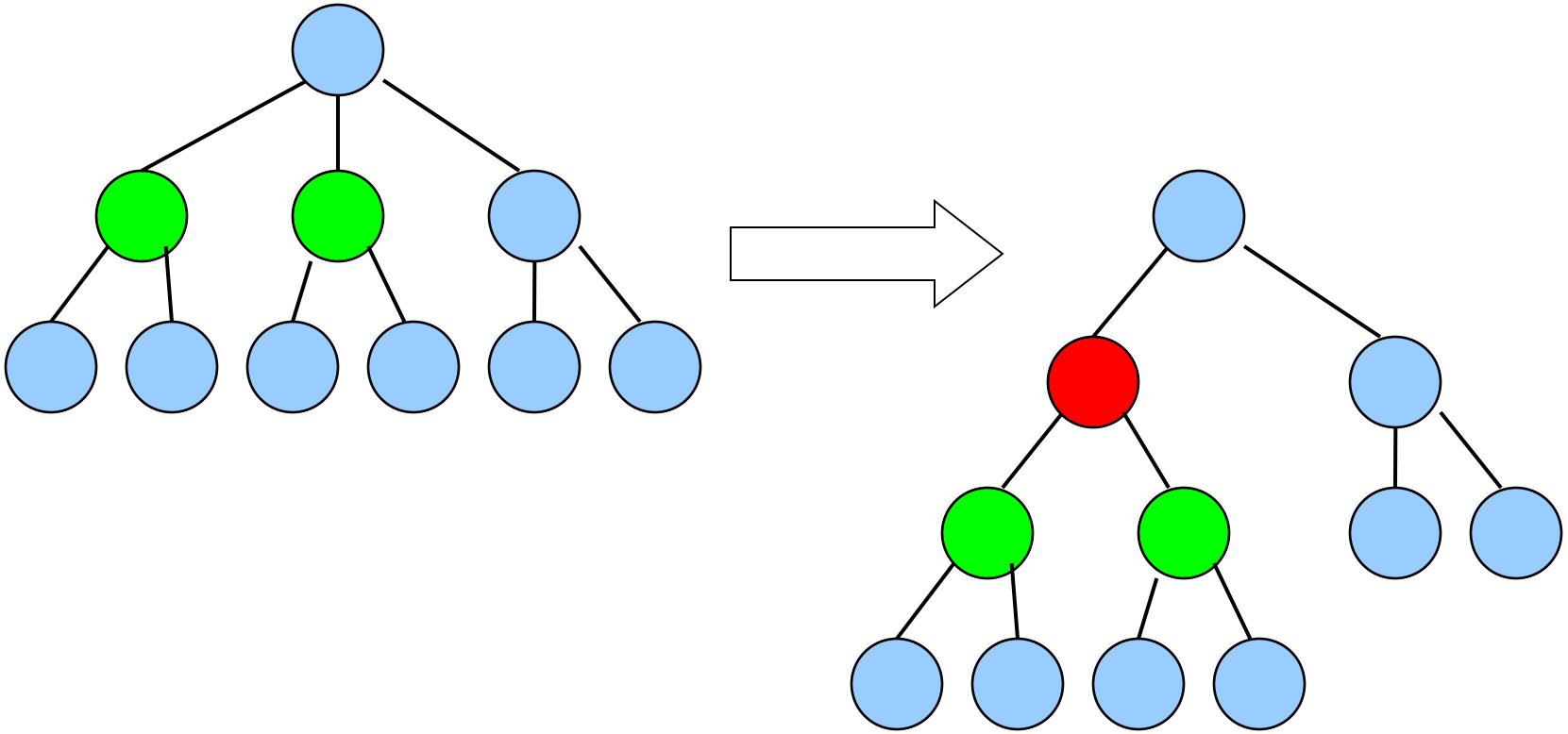
COBWEB

Utworzenie nowej grupy



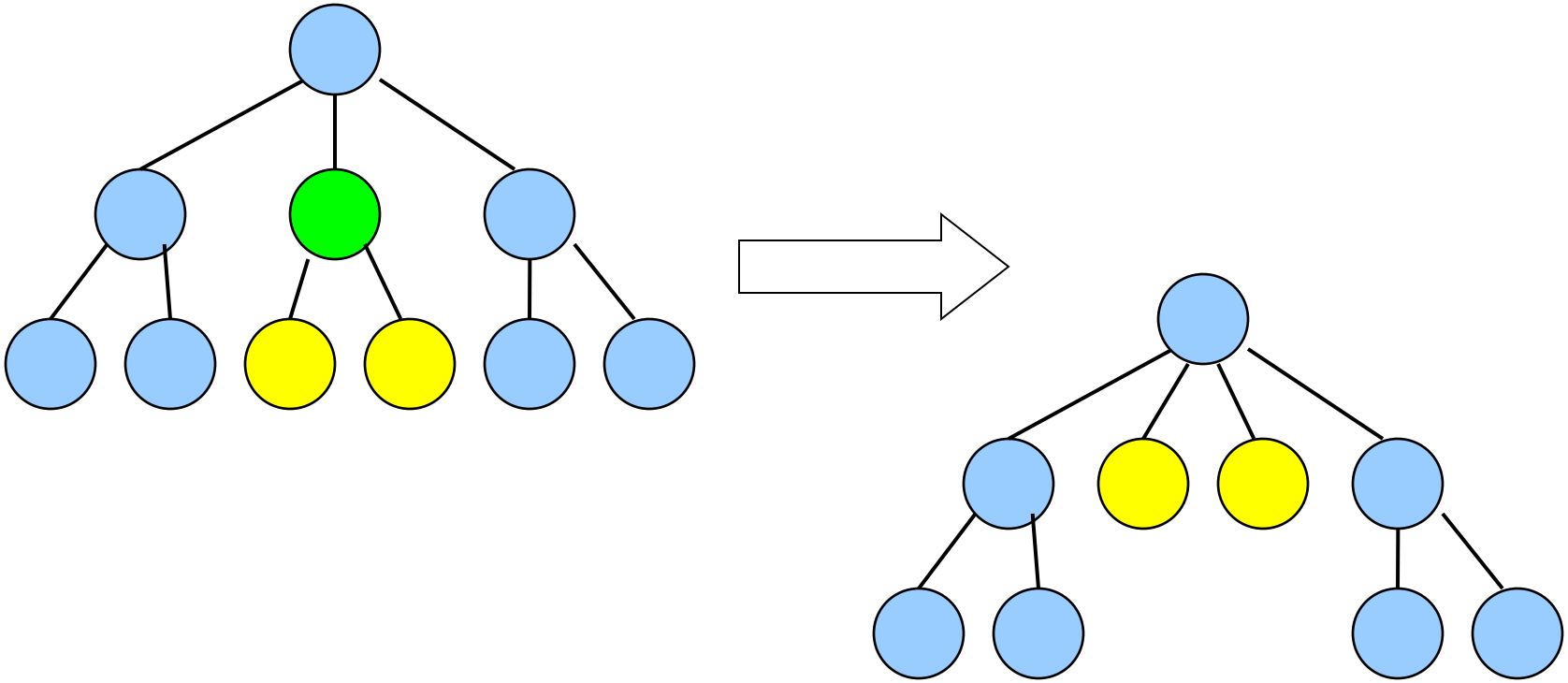
COBWEB

Łączenie dwóch grup



COBWEB

Podział grupy

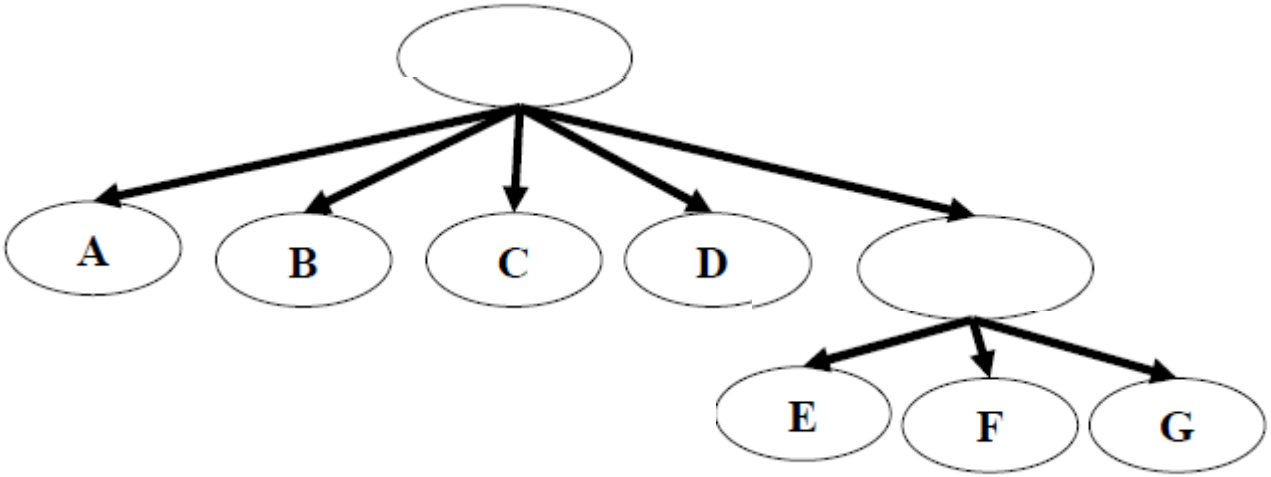


COBWEB

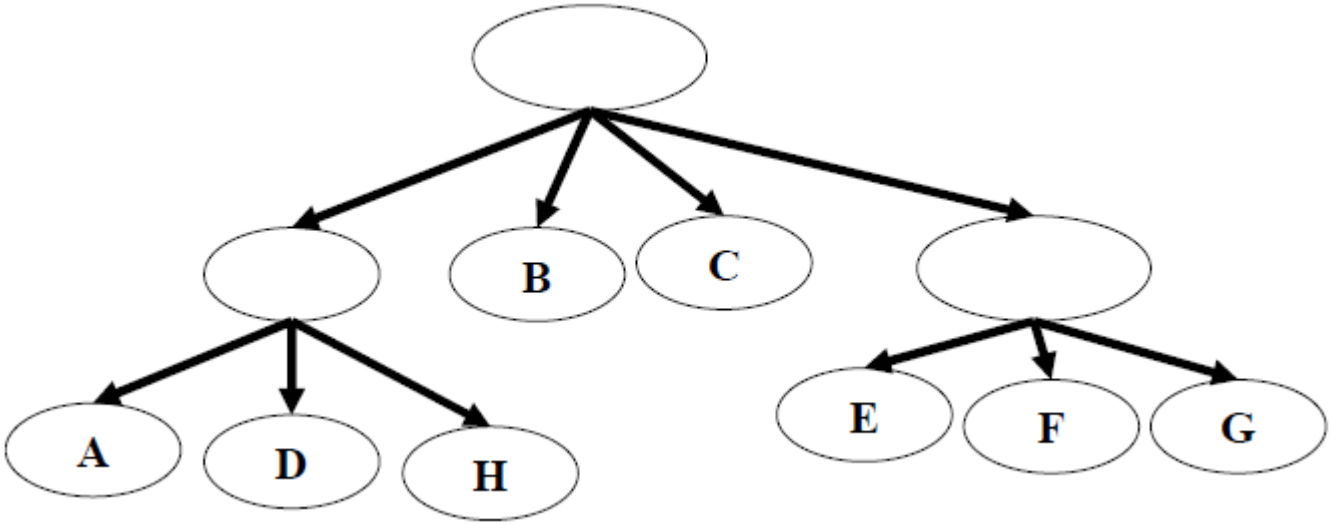
```
function COBWEB (x – przykład uczący, n – węzeł)
  begin
    if n nie jest liściem then
      {dodaj x do węzła n}
      wybierz wariant najlepszy ze względu na jakość grupowania
        1.utwórz nowy liść jako potomka n i umieść w nim x
        2.umieść x w n' - najlepszym potomku n i wywołaj COBWEB(x,n')
        3.połącz dwa najlepsze węzły potomne n tworząc n' i wywołaj
                                                    COBWEB(x,n')
        4.podziel najlepszego potomka n i wywołaj COBWEB(x,n)
      end wybierz
    else
      {dodaj x do liścia n}
      utwórz n' zawierający przykłady z n oraz x
      umieść n jako potomka n'
      utwórz liść z przykładem x jako potomka węzła n'
    endif
  end function
```

Funkcja wywoływana jest dla wszystkich przykładów uczących.

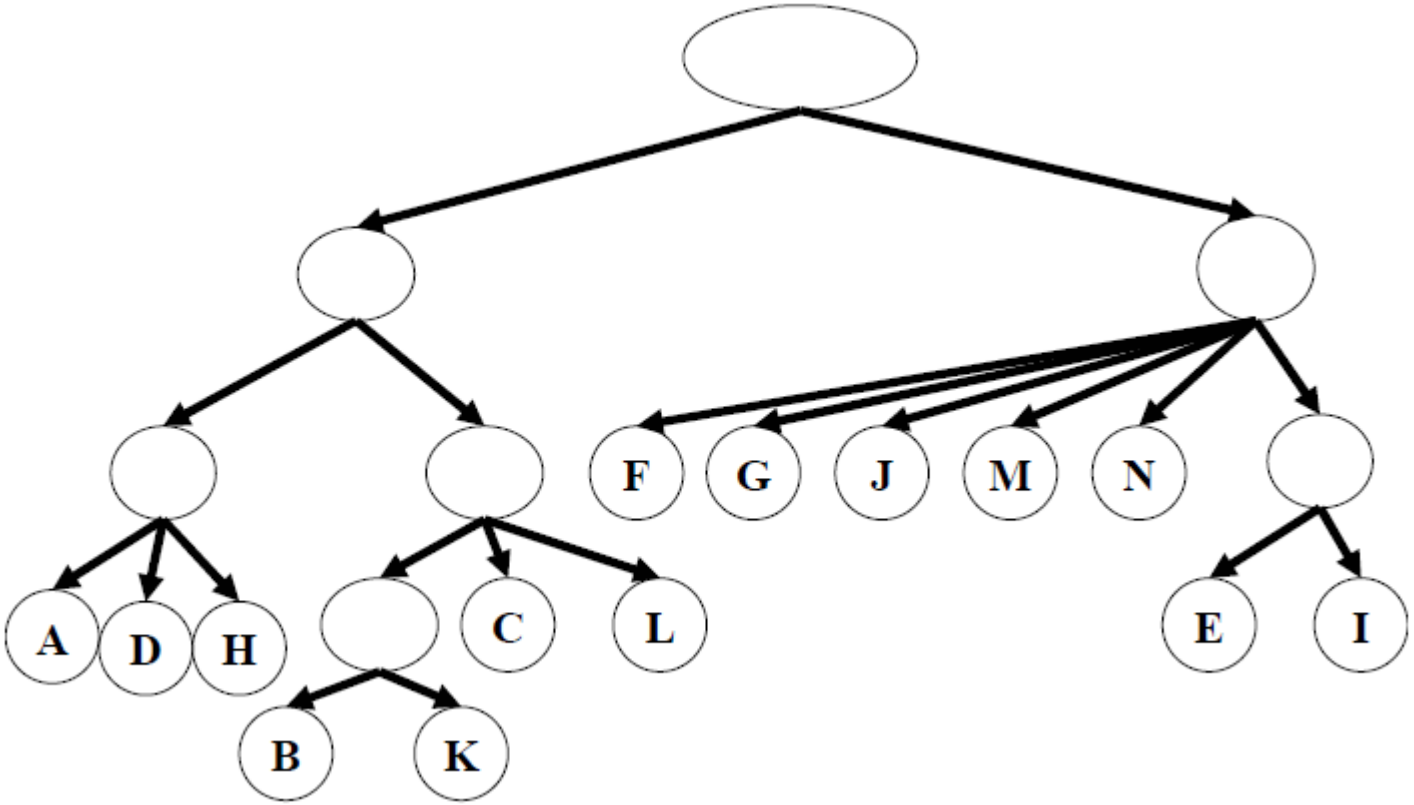
COBWEB



COBWEB



COBWEB



Ulepszony algorytm hierarchiczny- *BIRCH*

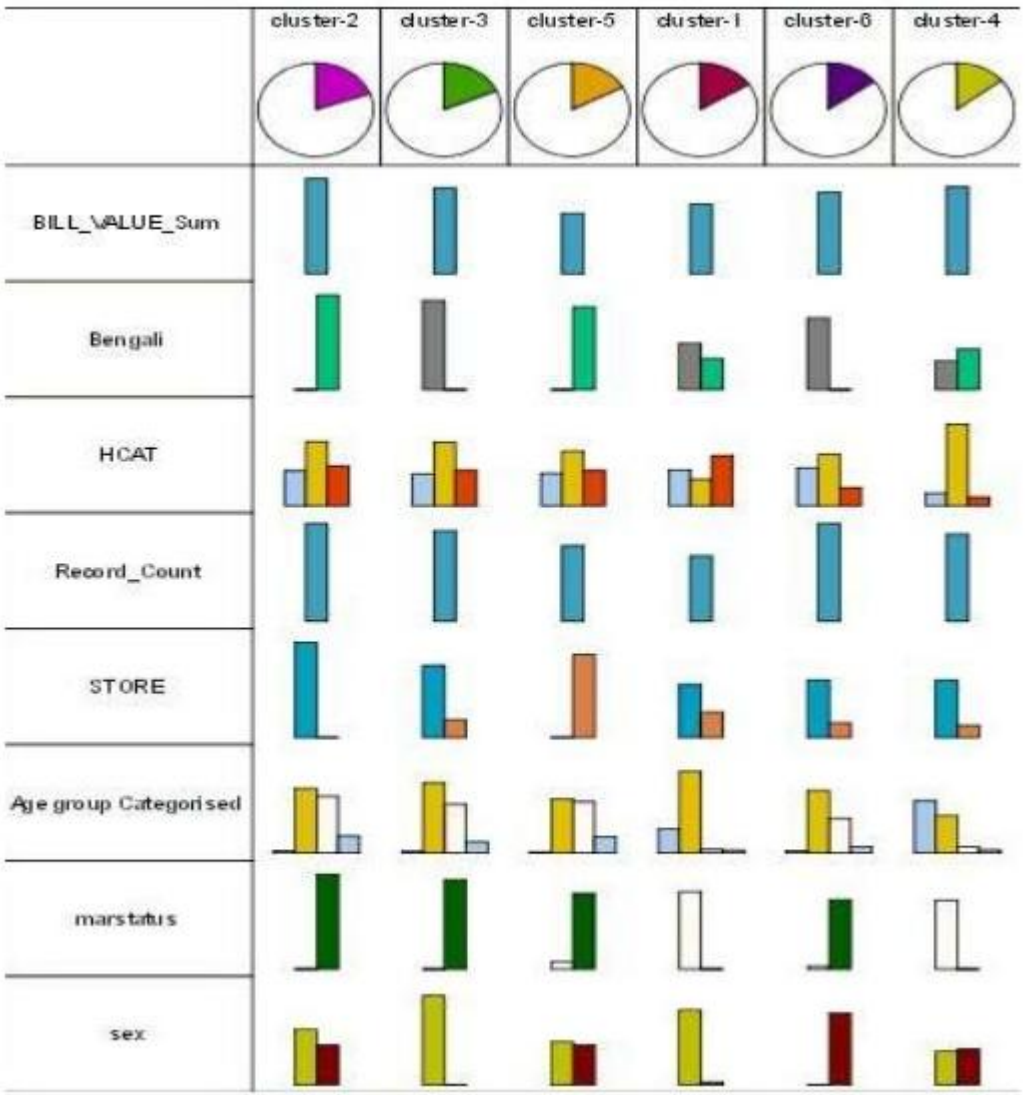
Ulepszony algorytm hierarchiczny- BIRCH

BIRCH *(m.in. SPSS, Clementine)*

(*Balanced Iterative Reducing and Clustering using Hierarchies*):

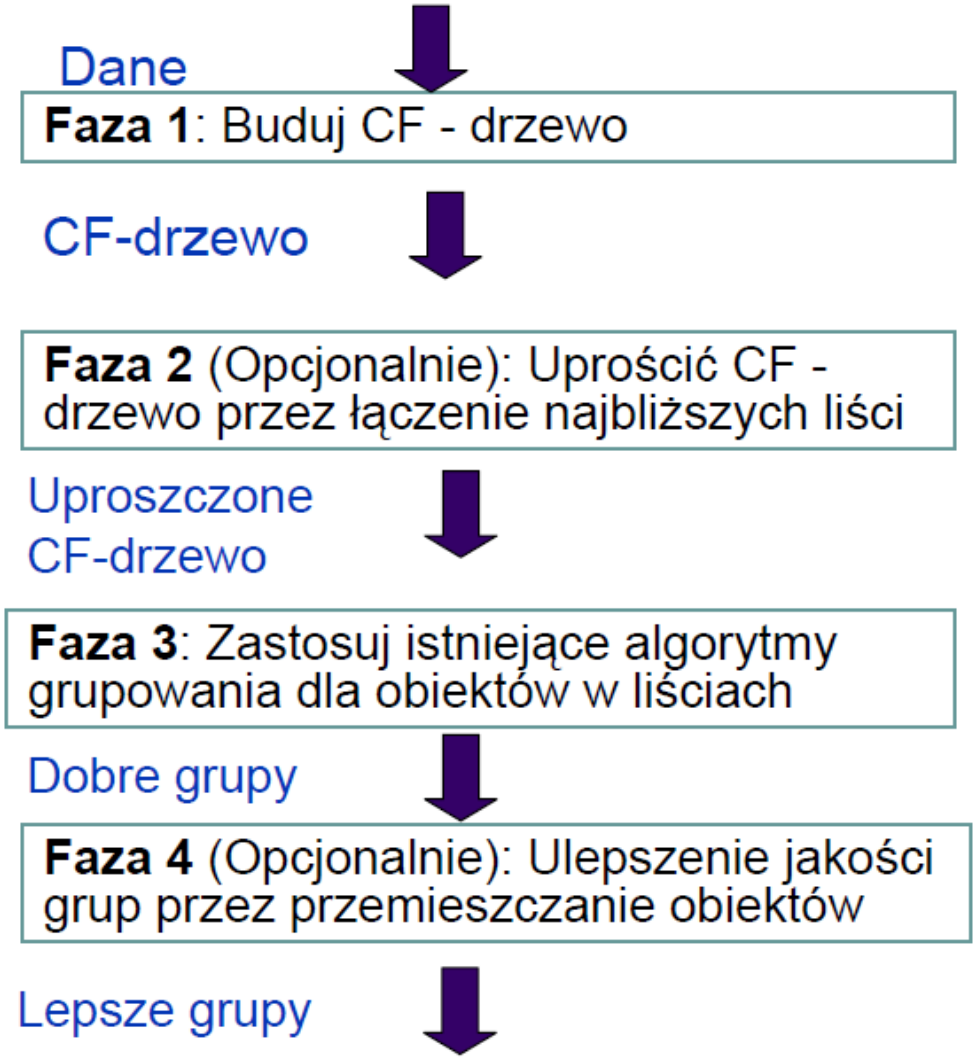
- Działa efektywnie: decyzja dla jednej grupy (dzielenie czy połączenie z inną grupą) nie wymaga przeglądania całego zbioru danych,
- koszt jest liniowy względem rozmiaru danych, jednokrotne przeglądanie zbioru danych
- Algorytm działa dla danych dynamicznie zmienionych
- Wykrywa szumy w danych
- *Two-step clustering* : podział i łączenie

wizualizacja



- W kolumnach klastry ułożone względem malejącej liczebności
- W wierszach zmienne kategoryczne

Algorytm BIRCH – Schemat blokowy



Algorytm BIRCH – Faza 3,4

- **Faza 3**:
  - » Każda grupa w liściu jest reprezentowana przez środek ciężkości. Zastosuj dowolny algorytm grupowania dla zbioru środków
  - » Zastosuj dowolny algorytm grupowania bezpośrednio na obiektach w grupie.
- **Faza 4** (ulepszenie jakości grup):
  - » Wyznaczaj środki grup generowanych przez fazę 3
  - » Dla każdego obiektu *o*: przemieszczaj go do grupy, której środek jest najbliżej *o*.

BIRCH - Ocena

Zalety:

- Wyznacza grupy przez jedno przeglądanie zbioru danych.
- Proces wstępny dla wielu algorytmów grupowania

Wady:

- Działa tylko dla danych numerycznych
- Wrażliwy na kolejność obiektów

BIRCH – Struktura CF drzewa

CF *(Clustering Feature)* – drzewo:

- Zrównoważone drzewo
- Ma trzy parametry:
  - *B* – maksymalna liczba rozgałęzień (współczynnik rozgałęzienia),
  - *L* – maksymalna liczba obiektów w liściach
  - *T* – maksymalny promień (grup w liściach) - próg
- Węzeł wewnętrzny: $[CF_i, child_i]$ $i= 1,2,...,B$
- Węzeł zewnętrzny (liść): $[CF_i]$ $i =1,2,...,L$

Opis grupy

– Niech grupa ***CF*** zawiera *n* punktów $\vec{x}_i \in R^d$

– **Środek**, **promień** (*R*) i **średnica** (*D*) grupy są zdefiniowane:

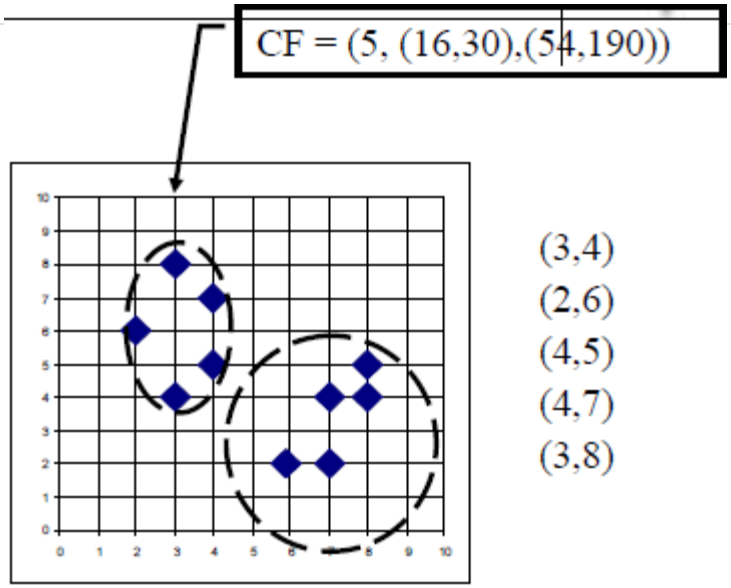
$$\vec{x}_0 = \frac{\sum_{i=1}^{N} \vec{x}_i}{N}$$

$$R = \left(\frac{\sum_{i=1}^{N} (\vec{x}_i - \vec{x}_0)^2}{N}\right)^{\frac{1}{2}}$$

$$D = \left(\frac{\sum_{i=1}^{N} \sum_{j=1}^{N} (\vec{x}_i - \vec{x}_j)^2}{N(N-1)}\right)^{\frac{1}{2}}$$

– Parametry $\vec{x}_0$, *R* i *D* opisują grupę obiektów *CF*.

Opis grupy – Wektor CF

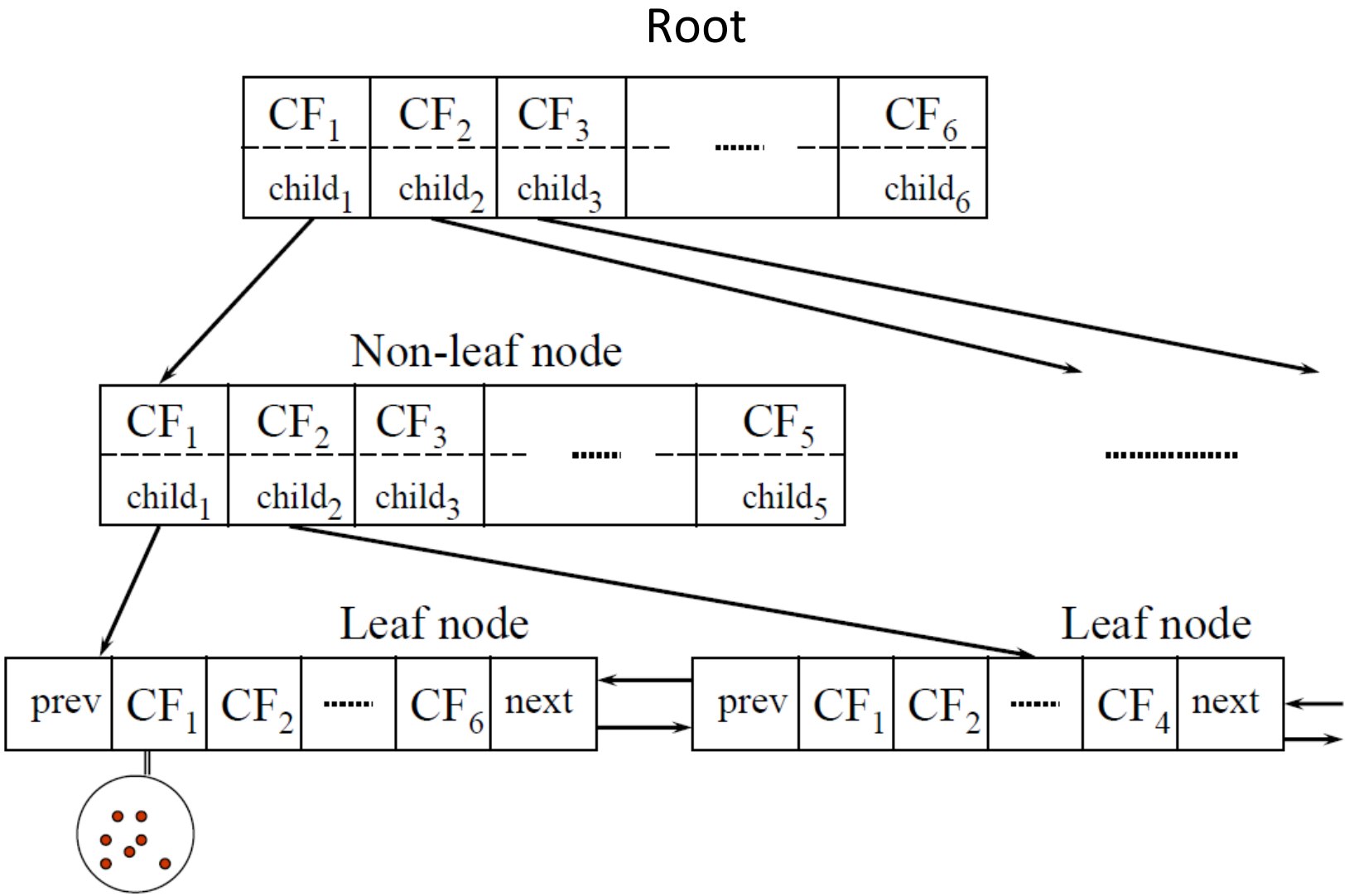


- **Opis grupy : *CF = (n, $\overrightarrow{LS}$, SS)***
- *n*: liczba punktów w grupie

  $\overrightarrow{LS}$: $\sum_{i=1}^{N} = \overrightarrow{X_i}$

  SS: $\sum_{i=1}^{N} = X_i^2$
- **Twierdzenie:**
- Niech $CF_1 = (n_1, \overrightarrow{LS_1}, SS_1)$ i $CF_2 = (n_2, \overrightarrow{LS_2}, SS_2)$ będą opisami dwóch grup G_1 i G_2, to $CF = (n_1+n_2, \overrightarrow{LS_1}+\overrightarrow{LS_2}, SS_1+SS_2)$ będzie opisem grupy, która jest połączeniem G_1 i G_2

CF - drzewo

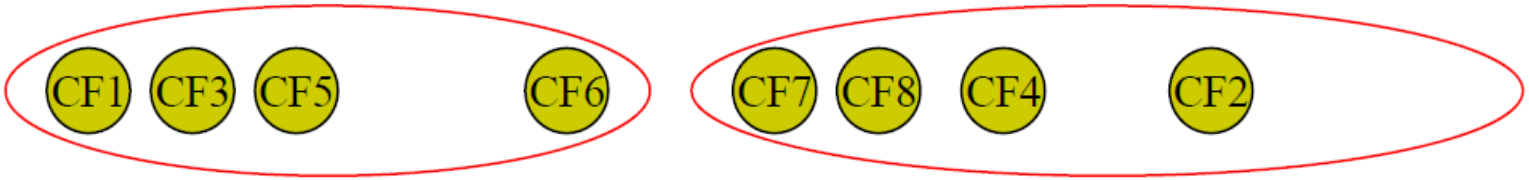

Root
CF1 CF2 CF3 ..... CF6
child1 child2 child3 child6
Non-leaf node
CF1 CF2 CF3 ..... CF5
child1 child2 child3 child5
Leaf node
Leaf node
prev CF1 CF2 ..... CF6 next
prev CF1 CF2 ..... CF4 next

Wstawianie obiektu do drzewa

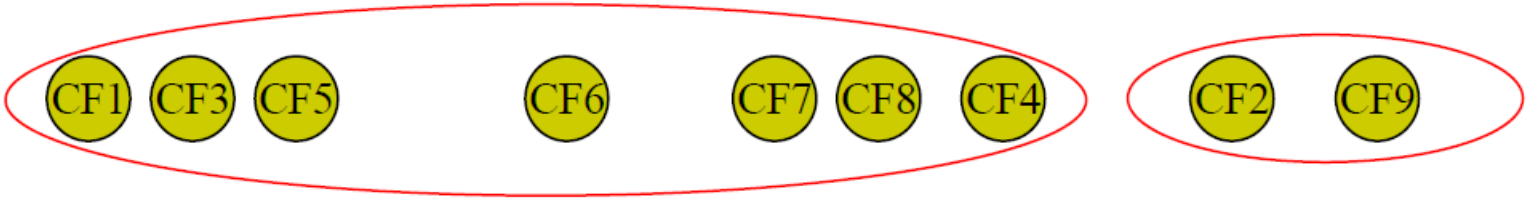
- **Krok 1**. Wybierz liść *CFx* do wstawiania. Użyj jednej z funkcji odległości *D* do wyznaczenia najbliższej grupy do badanego punktu
- **Krok 2**. Jeśli w *CFx* jest miejsce to wstaw *x*, jeśli nie: Podziel liść *CFx* na dwa liście i przelicz ścieżkę od *CFx* do korzenia.
- **Krok 3**. Rekonstruuj drzewo przez połączenie dwóch najbliższych węzłów lub podziel na dwa (w razie potrzeby): *merge i resplite*

Efekt *splite, merge i resplite*

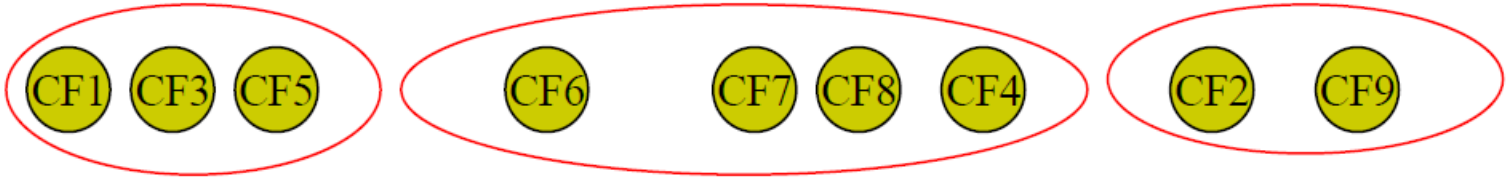
Splite



Merge



Resplite



Grupowanie oparte na gęstości

Ograniczenie algorytmów grupowania opartych na odległości

- Każda grupa jest reprezentowana przez jeden obiekt lub środek ciężkości

- Grupy są wypukłymi figurami.

Grupowanie oparte na gęstości

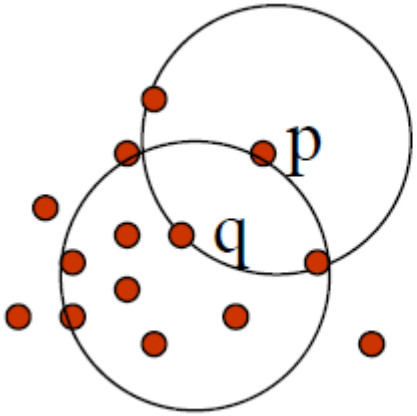
- Grupa składa się z punktów sąsiednich o wysokiej gęstości w otoczeniu
- Regiony pokrywające grupy mają wyższą gęstość niż regiony na zewnątrz

Grupowanie oparte na gęstości

- Główne **zalety**:
  - Odkrywa grupy o dowolnym kształcie
  - Odkrywa szumy
  - Jednokrotne przeglądanie zbioru danych
- Interesujące algorytmy:
  - DBSCAN: Ester, et al. (KDD'96)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98)

Pojęcia podstawowe

- Dwa parametry:
  - » ε : promień definiujący otoczenie obiektu
  - » ***MinPts***: minimalna liczba punktów w ε -otoczeniu
- **Rdzeń**: obiekt, który ma co najmniej *MinPts* w *ε* - otoczeniu
- **Brzegowy obiekt**: obiekt, który ma mniej niż *MinPts* w ε - otoczeniu.



$MinPts = 5$

$\varepsilon = 1$ cm

Pojęcia podstawowe

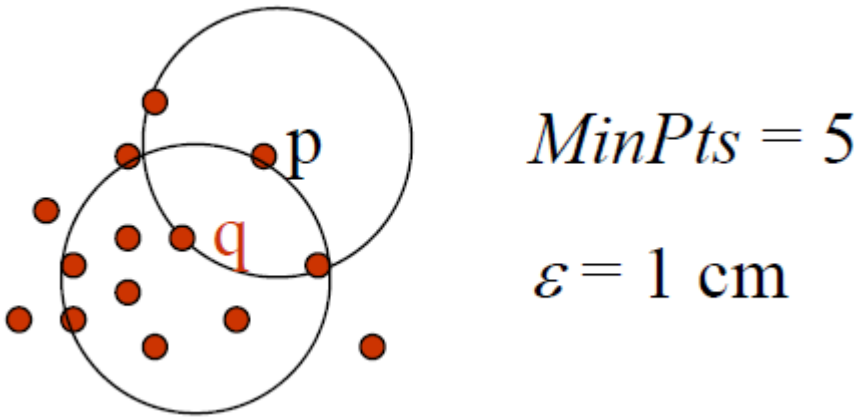
– ε -otoczenie:

$N_\varepsilon(p)$: $\{q \in D \mid dist(p,q) \leq \varepsilon\}$

– Dane są parametry ε i *MinPts*. Punkt ***p*** jest bezpośrednio wyprowadzony z punktu ***q*** jeśli

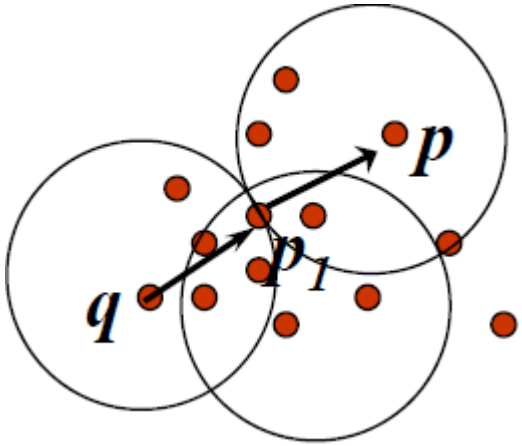
1) $\boldsymbol{p} \in N_\varepsilon(q)$

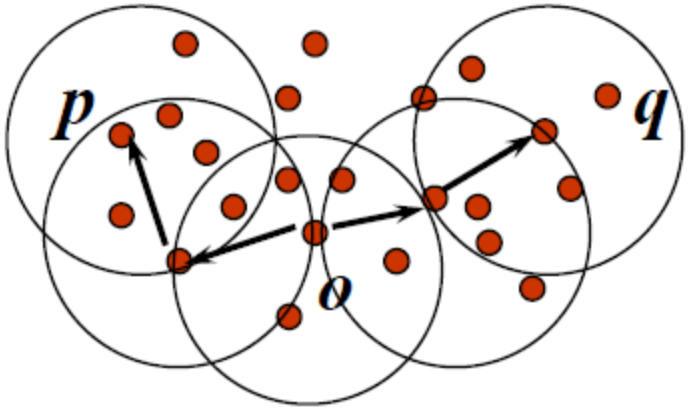
2) $|N_\varepsilon(q)| \geq \boldsymbol{MinPts}$



Density-Based Clustering

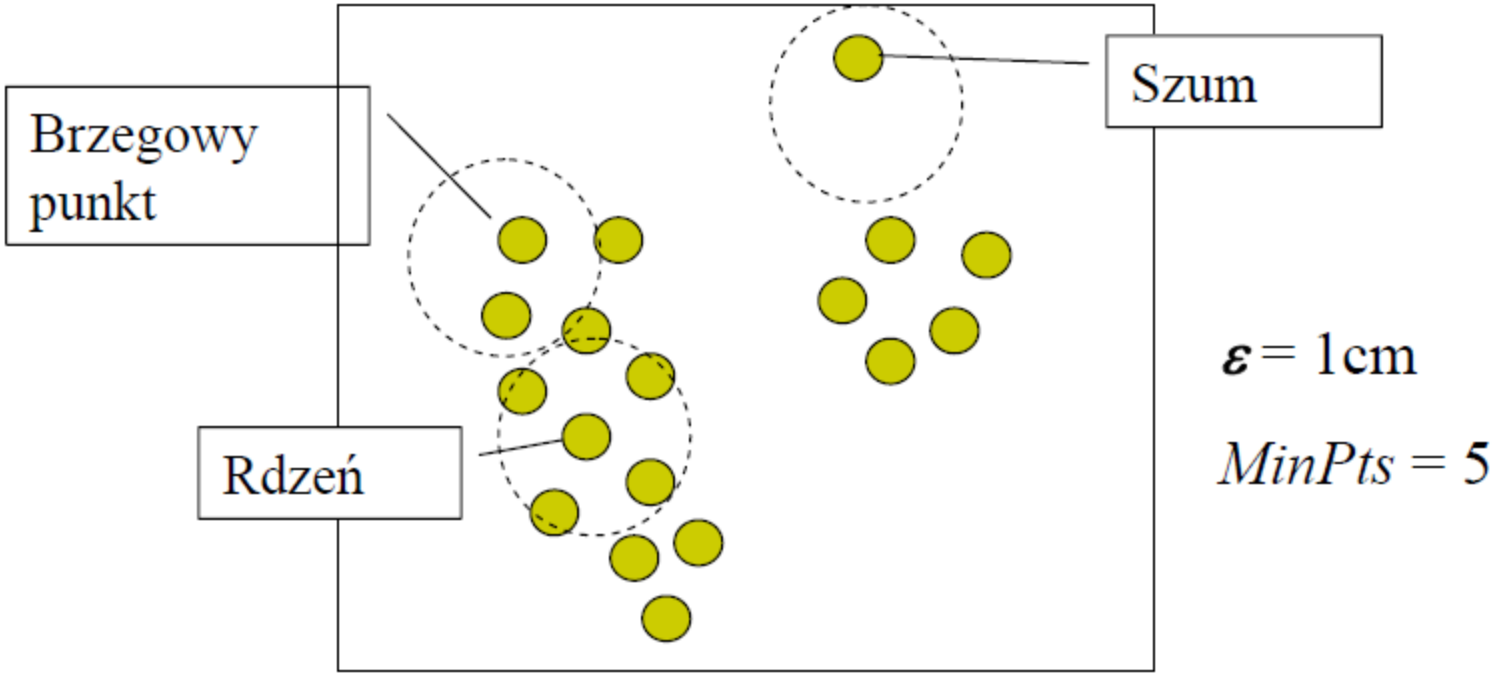
- Punkt *p* jest wyprowadzony z punktu *q* jeśli istnieje ciąg punktów p_1, …, p_n *taki, że* $p_1 = q$, $p_n = p$ i p_{i+1} jest bezpośrednio osiągalny z p_i
- Punkt *p* i *q* są połączone jeśli istnieje punkt *o* taki, że *p* i *q* są wyprowadzone z *o*





DBSCAN: Density Based Spatial Clustering of Applications with Noise

– Grupa: Maksymalny zbiór punktów połączonych



Algorytm DBSCAN

- **Krok 1**. Wybierz dowolny punkt ***p***
- **Krok 2**. Wyszukaj zbiór *G* wszystkich punktów osiągalnych z punktu *p* w sensie ε i *MinPts.*
- **Krok 3**. Jeśli ***p*** jest rdzeniem, **return** *G* (grupa była utworzona).
- **Krok 4**. jeśli ***p*** jest punktem brzegowym (żaden punkt nie jest osiągalny z *p*) to sprawdź następny nieodwiedzony punkt
- **Krok 5**. Kontynuuj **until** wszystkie punkty odwiedzone

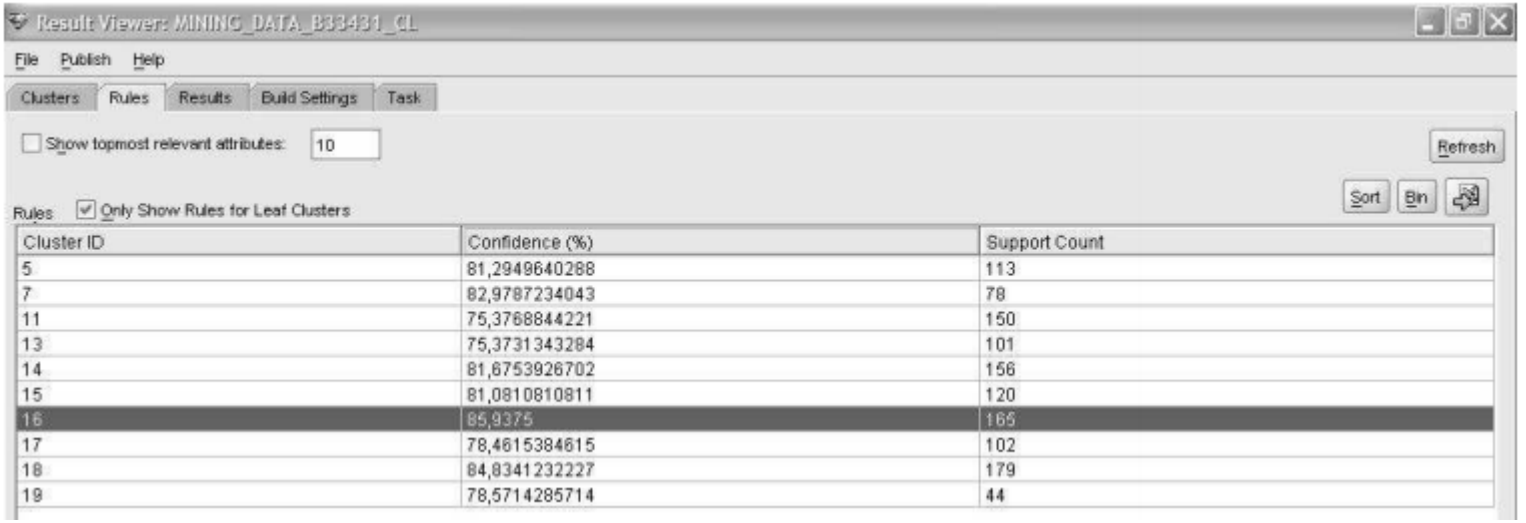
AGH

Algorytmy O-Cluster, WaveCluster, oraz ROCK

Algorytm O-Cluster
algorytm ortogonalnego partycjonowania

- Algorytm ten dokonuje rzutowania wszystkich obiektów na ortogonalne osie odpowiadające atrybutom wejściowym.
- Dla każdego wymiaru wyznaczane są histogramy, które następnie są analizowane w poszukiwaniu obszarów mniejszej gęstości.
- Dane są partycjonowane za pomocą hiperpłaszczyzn przecinających osie atrybutów w punktach mniejszej gęstości.
- Docelowa liczba grup wyznaczana jest automatycznie na podstawie charakterystyki danych.
- W przeciwieństwie do algorytmu k-średnich, algorytm O-Cluster nie tworzy sztucznych grup w obszarach o jednostajnej gęstości.
- Wrażliwy na szumy
- Zaimplementowany w **Oracle Data Mining**

OracleDataMiner O'cluster



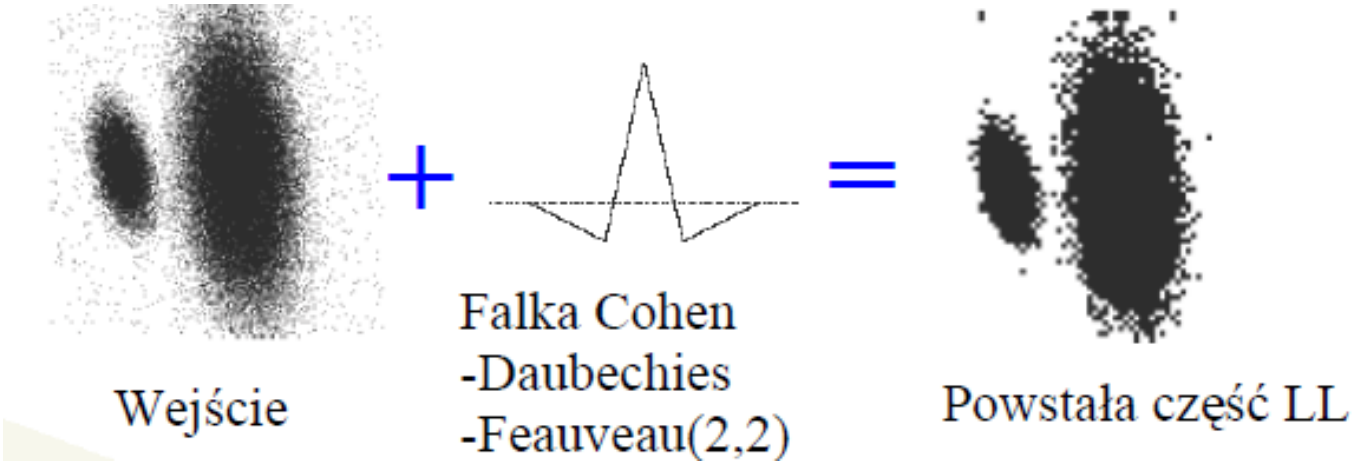
Cluster ID	Confidence (%)	Support Count
5	81,2949640288	113
7	82,9787234043	78
11	75,3768844221	150
13	75,3731343284	101
14	81,6753926702	156
15	81,0810810811	120
16	85,9375	165
17	78,4615384615	102
18	84,8341232227	179
19	78,5714285714	44

WaveCluster

- WaveCluster wykorzystuje dyskretną tranformatę falkową (*discreete wavelet transform*), ktora:
- Dzieli 1-wymiarowy sygnał wejściowy na 2 pasma (zmniejszając dwukrotnie rozdzielczość):
  - Wysokiej częstotliwości – odpowiada brzegom grup
  - Niskiej częstotliwości – odpowiada wnętrzom grup
- Sygnał 2-wymiarowy dzielimy stosując 2 razy transformatę 1-wymiarową. Otrzymujemy 4 pasma częstotliwości:
  - LL – niska-niska
  - LH – niska-wysoka
  - HL – wysoka-niska
  - HH – wysoka-wysoka

Przykład zastosowania falki

– Podziału sygnału dokonujemy stosując odpowiedni filtr-falkę:

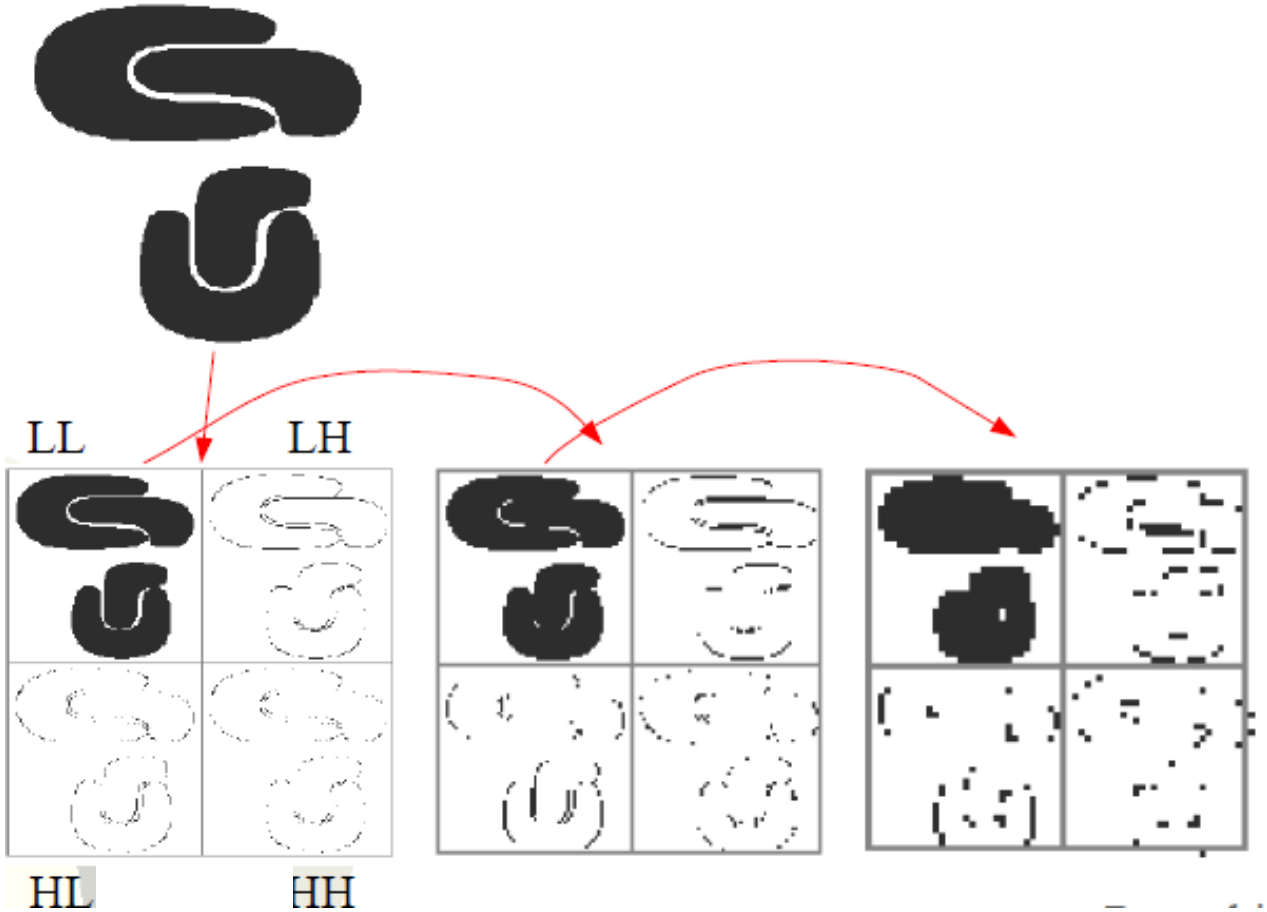


– Wyostrzyliśmy kształty i wyeliminowaliśmy szum

Źródło: Gholamhosein Sheikoholeslami, Surojit Chatterjee, Aidong Zhang, "WaveCluster: A Multi-resolution clustering approach for very large spatial databases", Proceedings of the 24th VLDB Conference, NY, USA, 1998

Działanie algorytmu – przykład

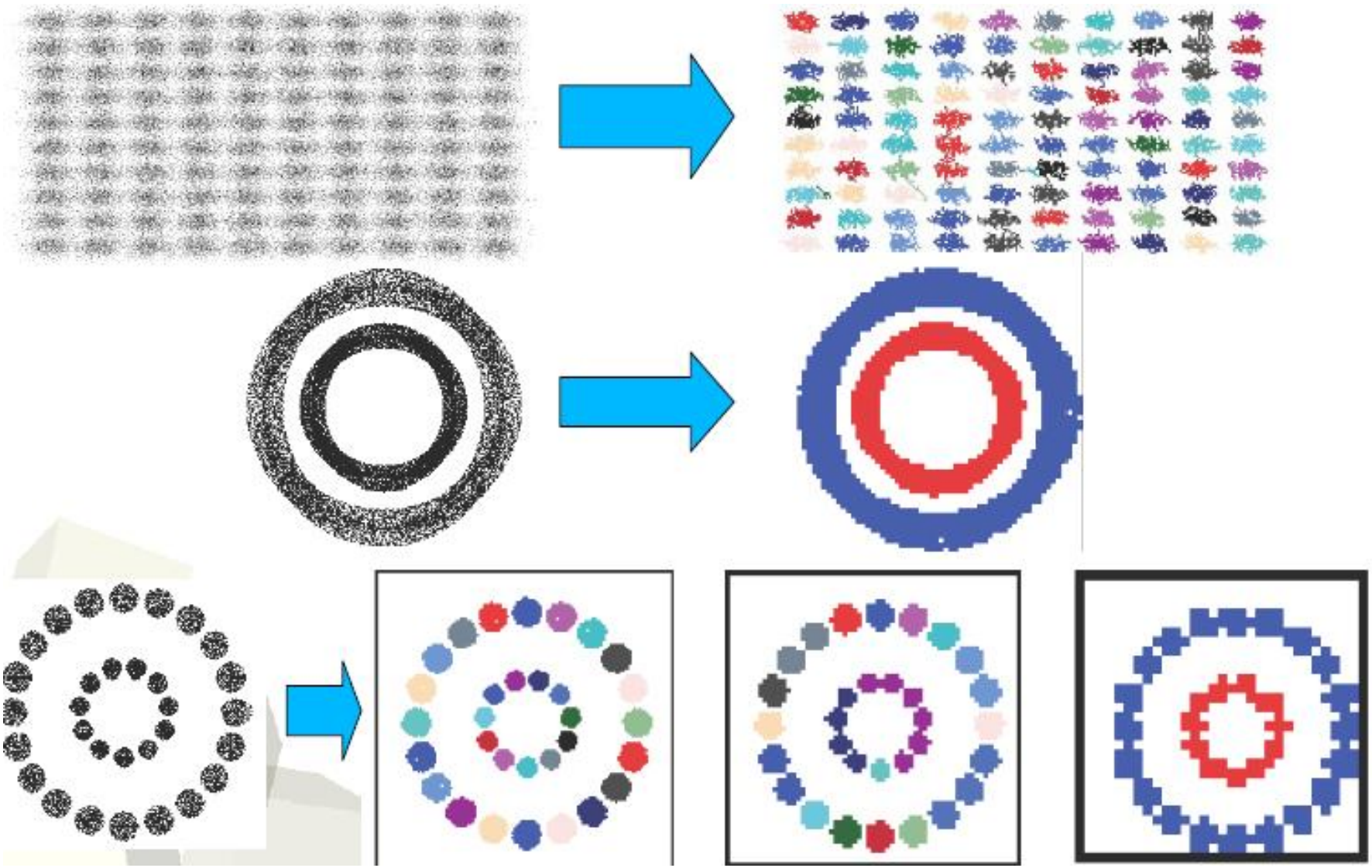
– Przykład wykonywania kolejnych transformacji



WaveCluster

- Algorytm (wejście: zbior wielowymiarowych punktow(obiektow), wyjście: pogrupowane punkty)
  1. Podziel przestrzeń na jednostki (każda z jednostek sumuje informację punktów w niej zawartych)
  2. Zastosuj transformatę falkową na przestrzeni
  3. Znajdź połączone jednostki w przekształconej przestrzeni (określamy grupy)
  4. Przypisz przekształconym jednostkom etykiety grup
  5. Przejdź do zwykłej przestrzeni - dokonaj mapowania: jednostka przekształcona→ zwykłe jednostki
  6. Przypisz punkty do klastrow
- Operację powtarzamy aż do uzyskania zadowalającej rozdzielczości (a raczej zadowalającego rozmycia)

Przykłady znalezionych grup



Zalety i wady algorytmu

- Nie trzeba podawać trudnych do określenia parametrow (jak np. w k-means, k-medoids), tylko:
  - » Wymiar jednostki (hiperprostokąta), za pomocą której dzielimy przestrzeń
  - » Ilość zastosowań transformaty falkowej (szukana rozdzielczość)
- Znajduje grupy dowolnych kształtów
- Wydajny (złożoność O(n)), można zaimplementować równolegle
- Odporny na szumy
- Mamy dostępne wiele poziomów dokładności (wada i zaleta)
- Wada: Dobrze radzi sobie tylko z danymi niskowymiarowymi (do 20 wymiarów)

Algorytm ROCK

- Dla danych nienumerycznych
- nie używa reprezentantów do grupowania, tylko wprowadza pojecie połączenia (*link*)
- Sąsiedztwo punktu *p* - taki zbiór punktów, który jest do *p* podobny.
- *sim*(p_i, p_j) - *funkcja podobieństwa*, znormalizowana, mówi o bliskości punktów *pi* i *pj* i przyjmuje wartości od 0 do 1
- Dla danego progu Θ [0, 1] punkty p_i i p_j są sąsiadami wtedy i tylko wtedy, gdy:

$$sim(p_i, p_j) > \theta$$

ROCK

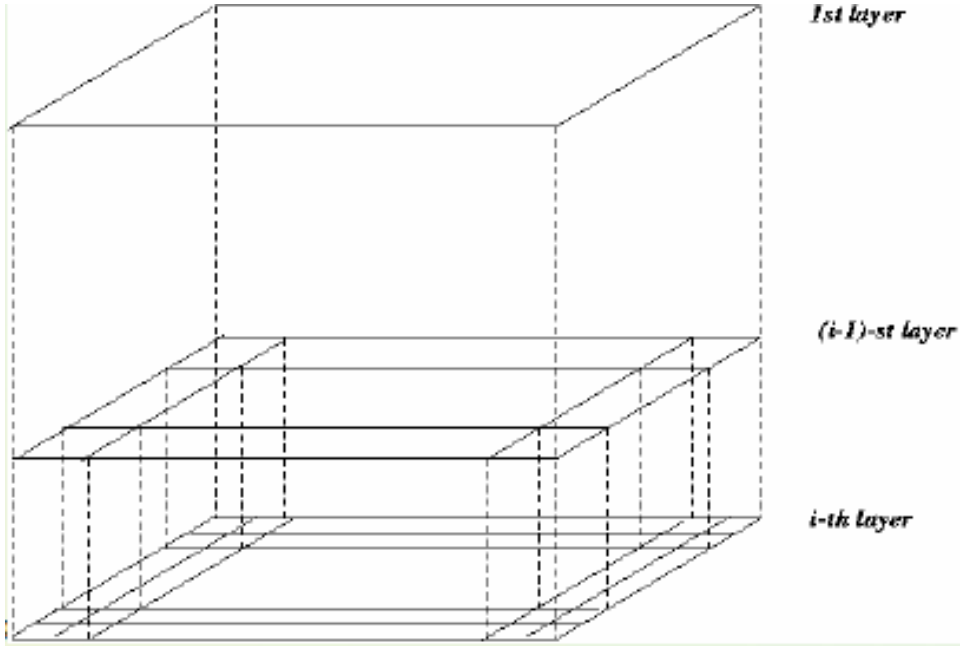
- dwa punkty mogą być do siebie podobne, jednak należeć do różnych klas w naturalnie stworzonych grupach
- w takiej sytuacji, pomimo podobieństwa pary punktów jest mało prawdopodobnym, żeby punkty te miały duża liczbę wspólnych sąsiadów
- Połączeniem (*link* (p_i, p_j)) pomiędzy punktami p_i i p_j jest liczba mówiąca o ilości wspólnych sąsiadów jakie maja punkty p_i i p_j
- Funkcja celu E_l ma za zadanie zmaksymalizować liczbę połączeń w jednej grupie jednocześnie nie dopuszczając rozwiązania, które będzie nadmiernie łączyć grupy:

$$E_l = \sum_{i=1}^{k} n_i * \sum_{p_q, p_r \in C_i} \frac{link(p_q, p_r)}{n_i^{1+2f(\theta)}}$$

Algorytmy gridowe

Algorytmy gridowe
STING: A Statistical Information Grid Approach

- Wang, Yang and Muntz (VLDB'97)
- The spatial area area is divided into rectangular cells
- There are several levels of cells corresponding to different levels of resolution

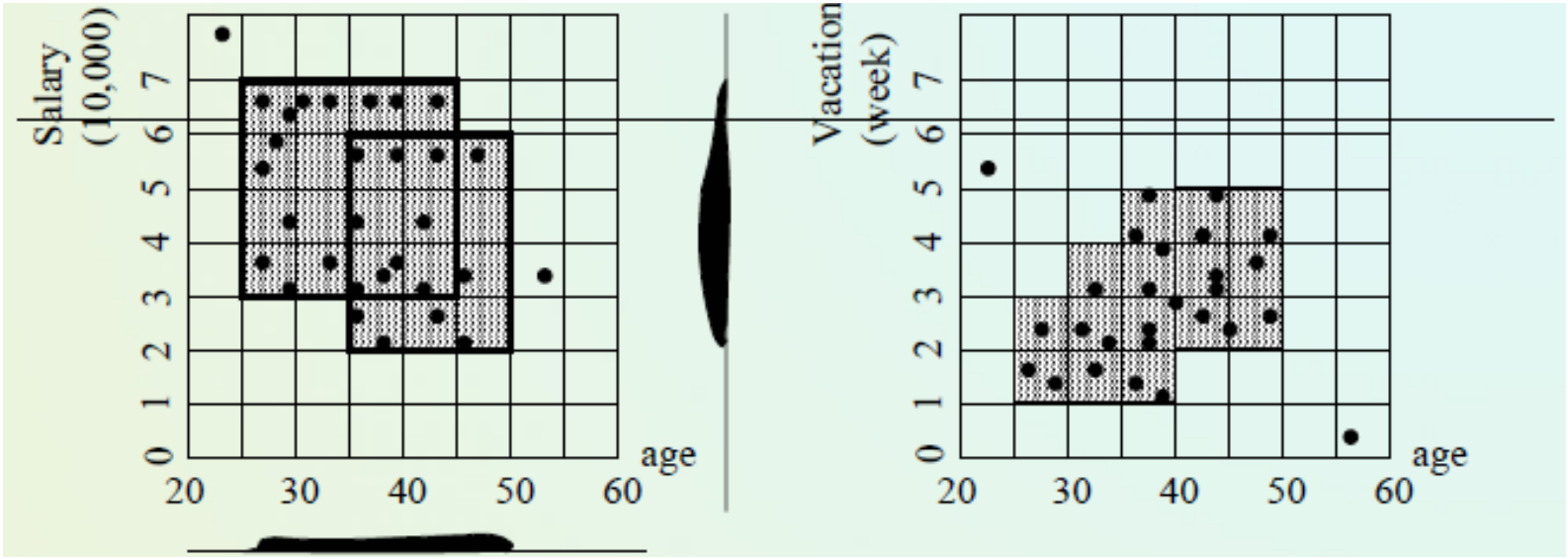


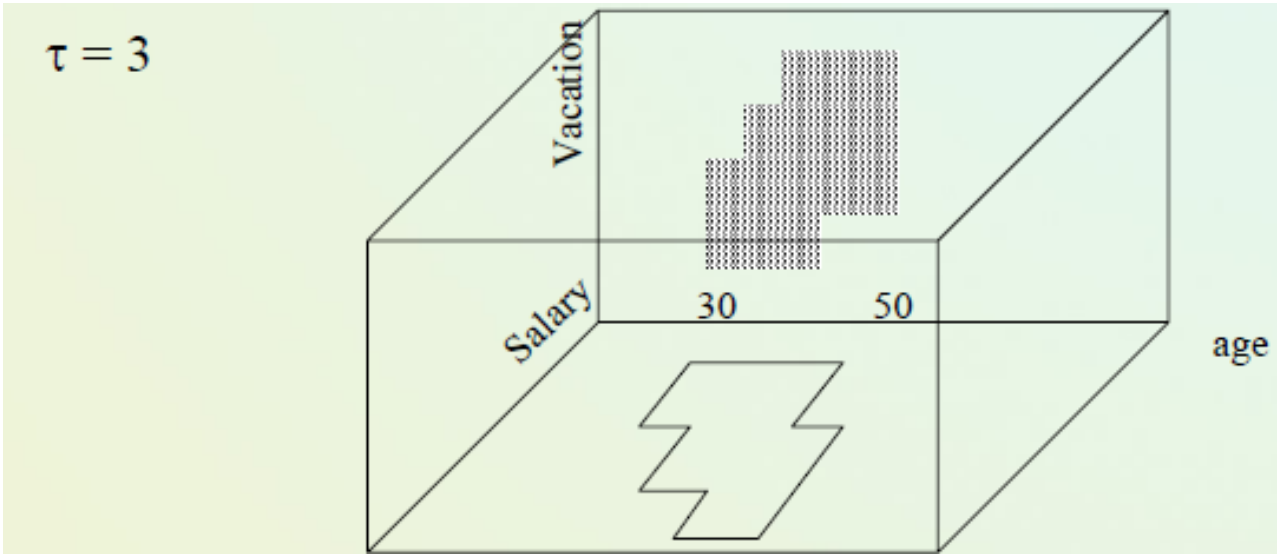
CLIQUE (Clustering In QUEst)

- Agrawal, Gehrke, Gunopulos, Raghavan (SIGMOD'98).
- Automatically identifying subspaces of a high dimensional data space that allow better clustering than original space
- CLIQUE can be considered as both density-based and gridbased
  - » It partitions each dimension into the same number of equal length interval
  - » It partitions an m-dimensional data space into non-overlapping rectangular units
  - » A unit is dense if the fraction of total data points contained in the unit exceeds the input model parameter
  - » A cluster is a maximal set of connected dense units within a subspace

CLIQUE: The Major Steps

- Partition the data space and find the number of points that lie inside each cell of the partition.
- Identify the subspaces that contain clusters using the Apriori principle
- Identify clusters:
  - » Determine dense units in all subspaces of interests
  - » Determine connected dense units in all subspaces of interests.
- Generate minimal description for the clusters
  - » Determine maximal regions that cover a cluster of connected dense units for each cluster
  - » Determination of minimal cover for each cluster





Źródło: prof. J. Stefanowski, wykłady, http://www.cs.put.poznan.pl/jstefanowski/

Strength and Weakness of *CLIQUE*

Strength

- It *automatically* finds subspaces of the highest dimensionality such that high density clusters exist in those subspaces
- It is *insensitive* to the order of records in input and does not presume some canonical data distribution
- It scales *linearly* with the size of input and has good scalability as the number of dimensions in the data increases

Weakness

- The accuracy of the clustering result may be degraded at the expense of simplicity of the method