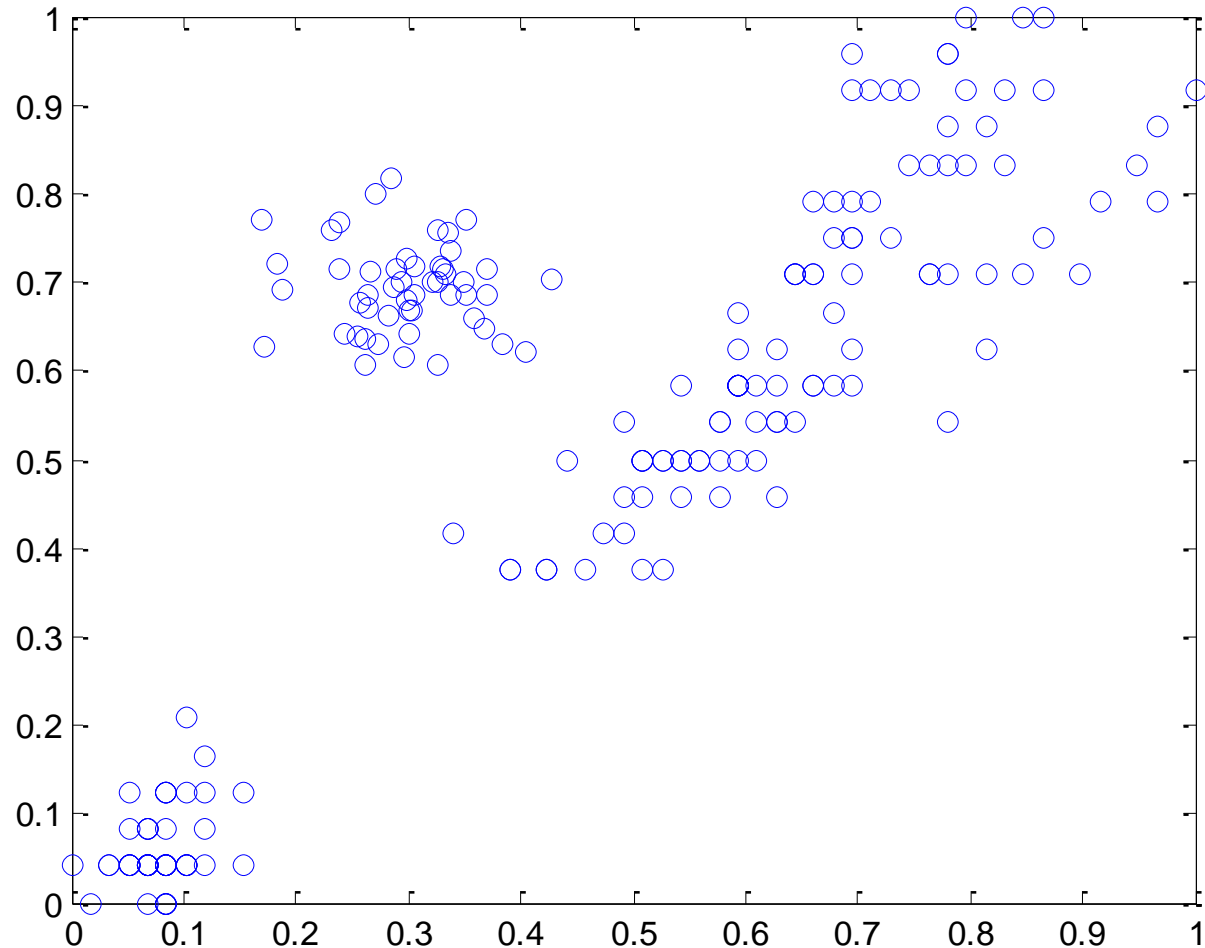


# Grupowanie danych



# Co to jest grupowanie

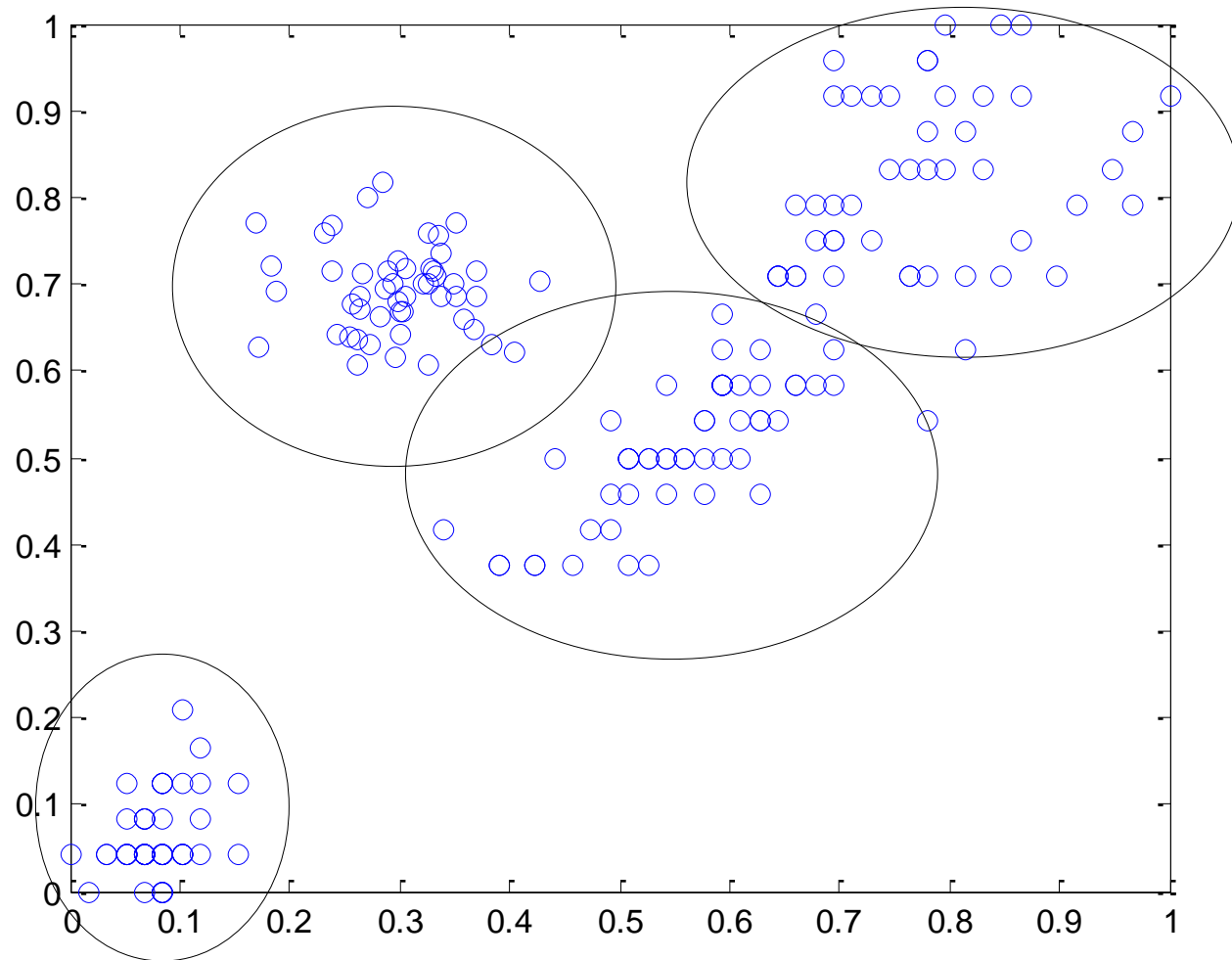
---



Szukanie grup, obszarów stanowiących lokalne gromady punktów

# Co to jest grupowanie

---



# Grupa nienad

□ analiza  
danych

■ Uczymy  
mamy

przykład

Przykład  
się n  
czego

Przykład

np. i  
dług  
dowi  
dysp



Az Ap – SDB



Captured Spirit – SDB



Llana – SDB



Child Star – BB



Very Varied – BB



Baboon Bottom – BB



Basso – IB



Blue Eyed Blond – IB



Gnu Rayz – IB



Chickee – MTB



Ozark Sky – MTB



Welch's Reward – MTB

# nie

owanie

o czego

honena,

ny co

ewno

kwiatów

ha oraz

ę

rysa

# Co to jest grupowanie - przykłady

- Analiza dokumentów np.. [www.clusty.com](http://www.clusty.com)  
-> przykład Blachnik:

The screenshot shows the Clusty search engine interface. At the top, there is a navigation bar with links for 'web', 'news', 'images', 'wikipedia', 'blogs', 'jobs', and 'more'. A search bar contains the text 'blachnik' and a 'Search' button. To the right of the search bar are links for 'advanced preferences'. Below the search bar, there are tabs for 'clusters', 'sources', and 'sites'. The 'clusters' tab is selected, and a sidebar on the left lists various clusters related to 'blachnik', such as 'Gabriele Blachnik (37)', 'Marcin Blachnik (19)', 'Rules, Prototype (13)', 'Blachnik, R (14)', 'Global (7)', 'Klaus Blachnik (6)', 'Roger Blachnik (6)', 'Structure (7)', 'Research (4)', and 'Heat, Capacity, enthalpy increments (4)'. The main content area displays 'Top 189 results of at least 518 retrieved for the query blachnik (details)'. A suggestion box says 'Did you mean: blatnik'. Below this, there are sponsored results for 'Buy Jimmy Choo Now' and 'Save Great On Parada Bags'. The search results section shows four items:

- Gabriele Blachnik** - eine Mode-Designerin, die es geschafft hat, ihren unverwechselbaren Stil seit 20 Jahren erfolgreich in der Mode zu behaupten. [www.gabriele-blachnik.de](http://www.gabriele-blachnik.de) - [cache] - Live, Ask, Gigablast
- Dissertation Barbara Blachnik** - Dissertation von Barbara Blachnik: Blachnik, Barbara: Zusatzmittel in Putzmörteln : Wirksamkeit, Dauerhaftigkeit und Auslaugung / von Barbara Blachnik - 2001. [www.ub.uni-siegen.de/epub/diss/blachnik\\_b.htm](http://www.ub.uni-siegen.de/epub/diss/blachnik_b.htm) - [cache] - Live, Ask, Gigablast
- DBLP: Marcin Blachnik** - 2008; 7: EE: Tadeusz Wieczorek, Marcin Blachnik, Krystian Maczka: Building a Model for Time Reduction of Steel Scrap Meltdown in the Electric Arc Furnace (EAF): General Strategy ... [www.informatik.uni-trier.de/~ley/db/indices/a-tree/b/Blachnik:Marcin.html](http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/b/Blachnik:Marcin.html) - [cache] - Live, Ask
- www.blachnik.com** - Die Domain "www.blachnik.com" wurde gesperrt. [www.blachnik.com](http://www.blachnik.com) - [cache] - Live, Ask

At the bottom left, there is a 'find in clusters' search bar and a 'Font size' selector with buttons for 'A', 'A', 'A', and 'A'.

# Podział metod grupowania danych

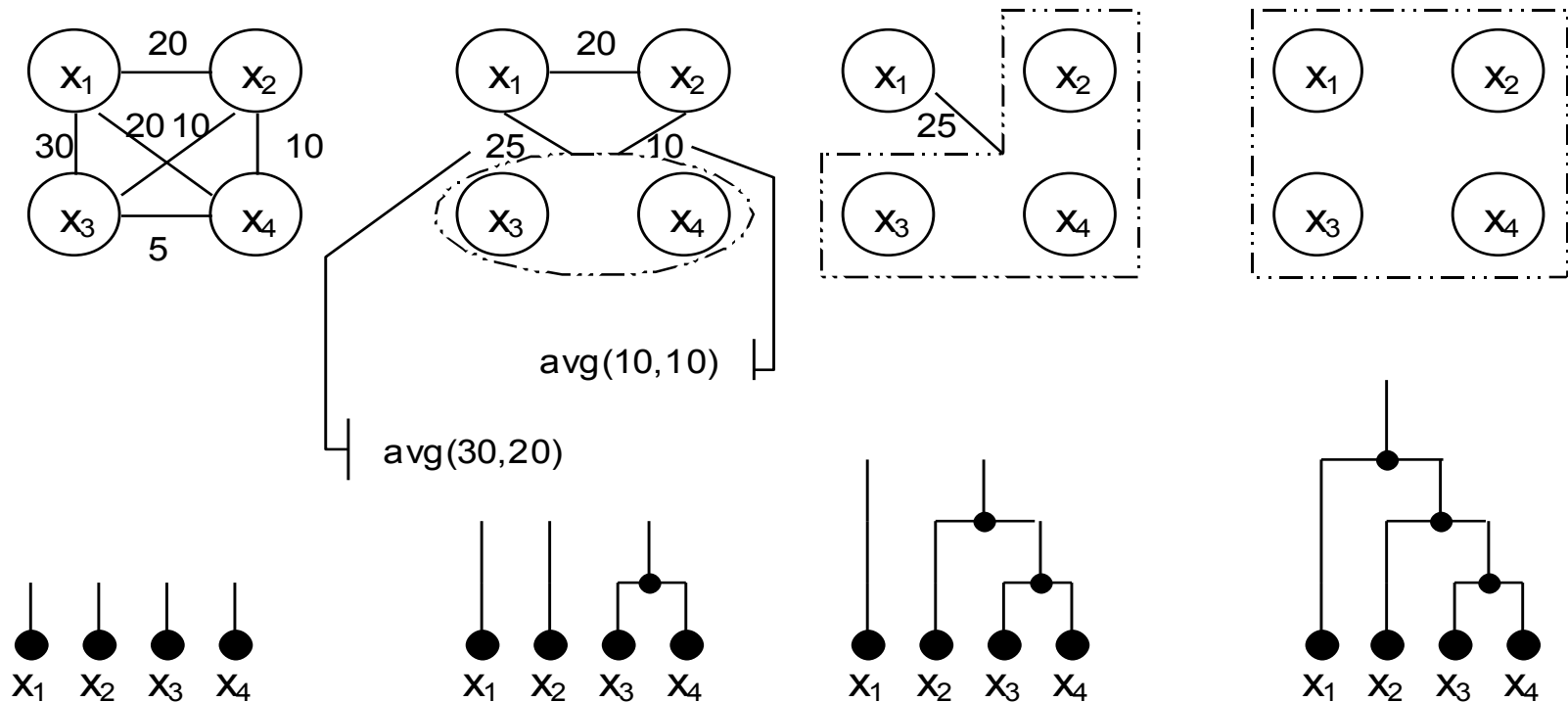
---

- Metody hierarchiczne
- Metody oparte o dekompozycję rozkładów prawdopodobieństwa
- Metody bazujące na minimalizacji skalarne współczynnika jakości
- Metody oparte na teorii grafów,

# Grupowanie hierarchiczne



# Grupowanie chierarchiczne





# Odległości pomiędzy skupiskami

- Minimum – minimalna odległość pomiędzy elementami zbiorów  $\mathbf{x}$

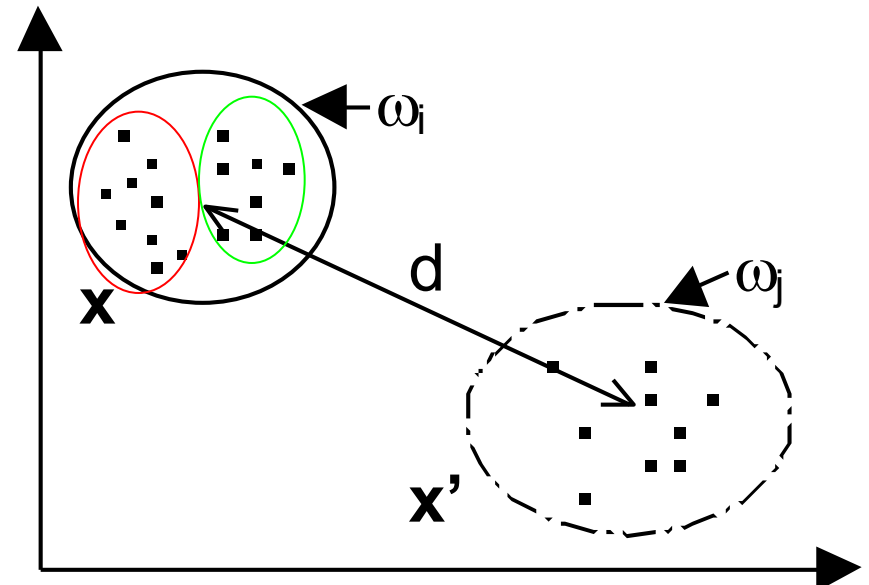
$$d_{\min}(\omega_i, \omega_j) = \min_{\mathbf{x} \in \omega_i} \|\mathbf{x} - \mathbf{x}'\|$$

- Maksimum - maksymalna odległość pomiędzy elementami zbiorów  $\mathbf{x}$  i  $\mathbf{x}'$

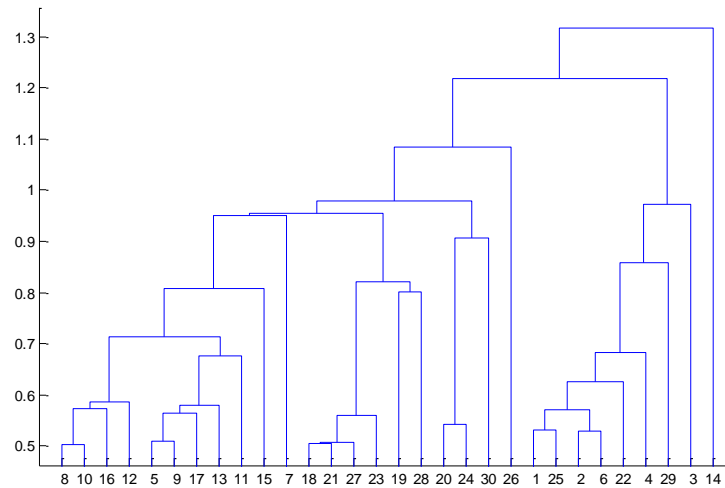
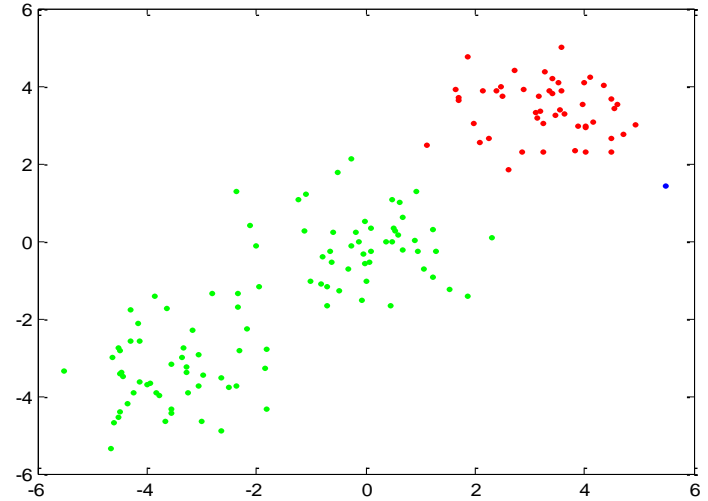
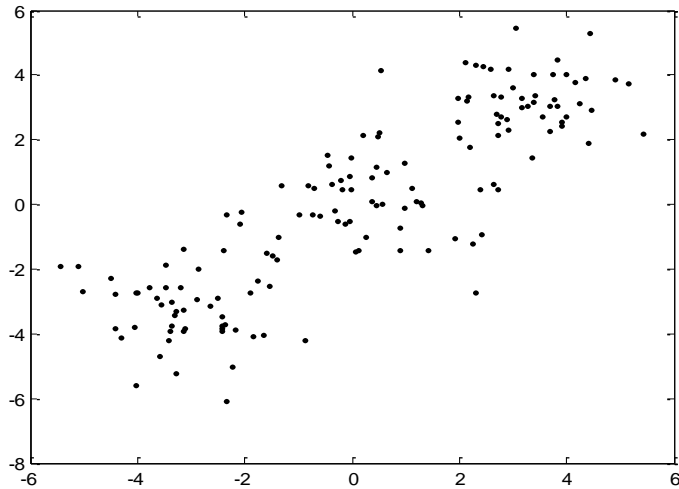
$$d_{\max}(\omega_i, \omega_j) = \max_{\mathbf{x} \in \omega_i} \|\mathbf{x} - \mathbf{x}'\|$$

- Norma różnicy wartości średnich

$$d_m(\omega_i, \omega_j) = \|m - m'\|$$



# Przykład



# Metody oceny jakości grupowania

---

## Cophenetic correlation coefficient

$$c = \frac{\sum_{i < j} (Y_{ij} - y)(Z_{ij} - z)}{\sqrt{\sum_{i < j} (Y_{ij} - y)^2 \sum_{i < j} (Z_{ij} - z)^2}}$$

Z - wektor odległości drzewa  $(m-1) \times 3$

Y - wektor odległości obiektów niezgrupowanych  
 $(m-1)m/2 \times 1$

$Y_{ij}$  - odległość pomiędzy obiektami i oraz j w Y.

$Z_{ij}$  - odległość pomiędzy obiektami i oraz j w Z.

y i z to odpowiednio wartości średnie Y i Z.

# Grupowanie oparte o minimalizację skalarnego współczynnika jakości

---

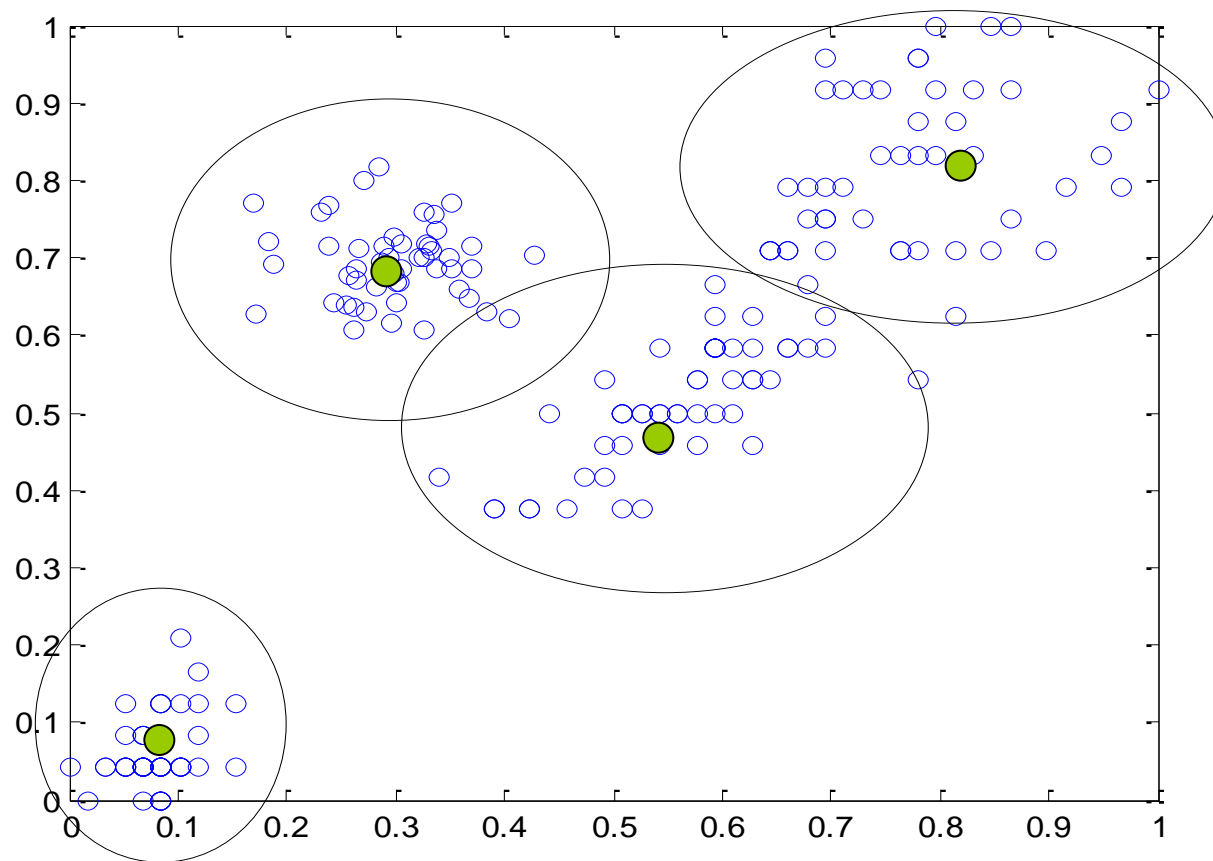
Algorytm K-średnich

# Co to jest skalarny wsp. jakości

---

- Skalarny współczynnik jakości oznacza funkcję kosztu która informuje jak „dobry” jest dany podział na grupy.
- Najprostsza postać skalarnego współczynnika jakości to suma odległości środka centrum klastra w stosunku do wszystkich wektorów w danej grupie.

# Przykład



# Oznaczenia

---

- $K$  – liczba wektorów, obiektów
- $C$  – liczba klasterów na które chcemy dokonać podziału
- $x(k)$ ;  $k=1..K$  –  $k$ -aty element z wektora obiektów  $\mathbf{X}$
- $\omega_i$  ;  $i=1..C$  –  $i$ -ty element wektora klastrów  $\omega$
- $v_i$  – centrum klastra = centrum grupy (grupy wektorów)

# Założenia grupowania

Zbudować macierz podziału

$$\mathbf{U} = [u_{ik}], \dim(\mathbf{U}) = C \times K$$

Warunki:

1° – każdy element macierzy  $u_{ik}$  należy do zbioru

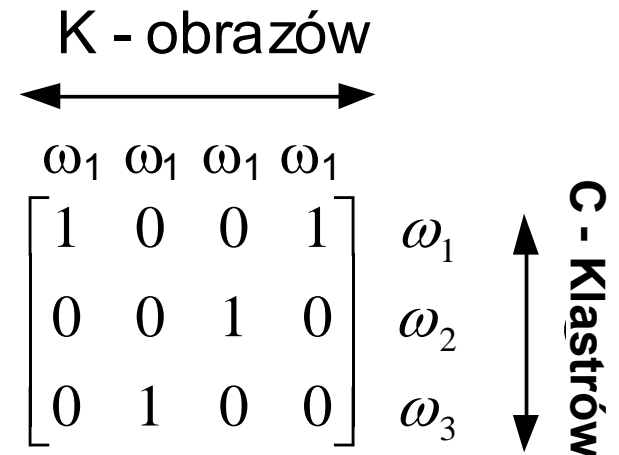
$$u_{ik} \in \{0,1\}$$

2° – w każdej kolumnie suma elementów równa jest 1

$$\sum_{i=1}^C u_{ik} = 1$$

3° – suma w wierszach należy

do przedziału  $\sum_{k=1}^K u_{ik} \in (0, K)$





# Algorytm k-średnich

---

Wskaźnik jakości:

$$J(\mathbf{U}) = \sum_{k=1}^K \sum_{i=1}^C u_{ik} d_{ik}^2$$

gdzie:

$$d_{ik} = \|x(k) - v_i\|$$

$v_i$  – zbiór (wektor) prototypów.

# Algorytm k-średnich

1. Przyjmujemy macierz podziału  $\mathbf{U}$  spełniającą trzy przedstawione uprzednio warunki

2. Wyznacza się położenie prototypów:

$$\forall_{1 \leq i \leq C} v_i = \frac{\sum_{k=1}^K u_{ik} \mathbf{x}(k)}{\sum_{k=1}^K u_{ik}}$$

3. zwiększa się licznik iteracji  $z=z+1$ ,

4. szukamy macierzy  $\mathbf{U}$  tak, by wyznaczyć dla każdego elementu wektora danych  $\mathbf{x}$  minimalną odległość od wzorców

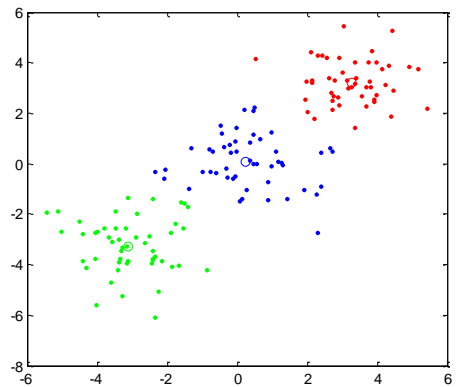
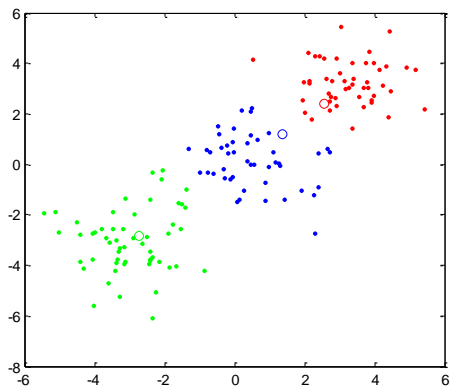
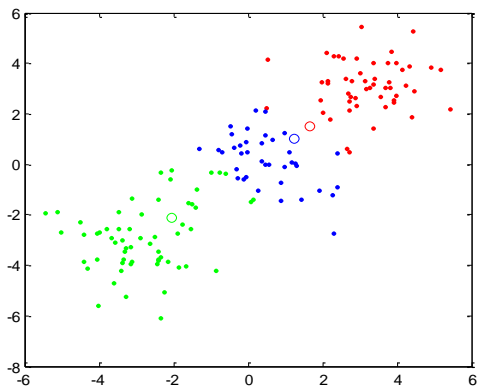
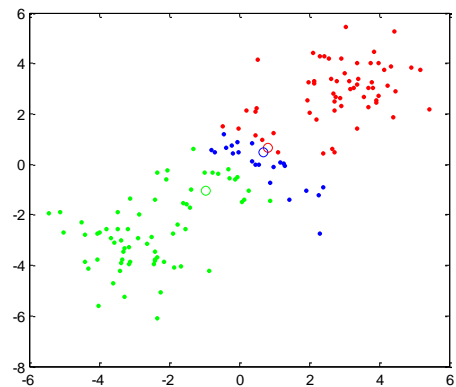
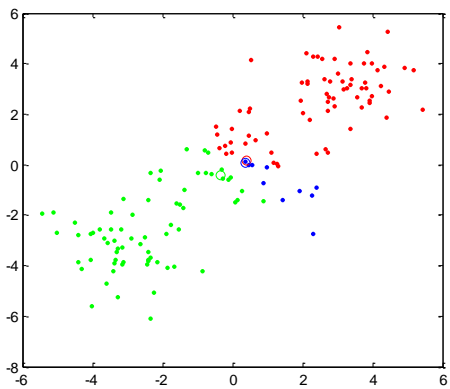
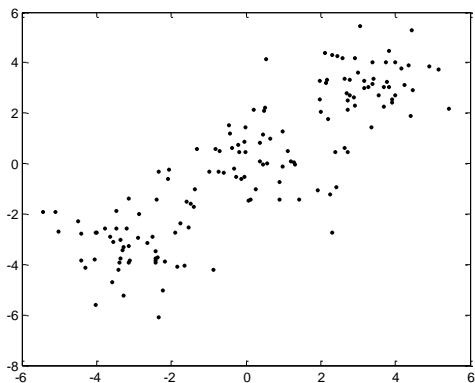
$$\forall_{1 \leq i \leq C} \forall_{1 \leq k \leq K} u_{ik} = \begin{cases} 1 & \min_j \|x(k) - v_j\| = \|x(k) - v_i\| \\ 0 & \text{w pozostałych przypadkach} \end{cases}$$

5. Sprawdzamy czy spełniony jest warunek

$$\|\mathbf{U}^{(z)} - \mathbf{U}^{(z-1)}\| < \varepsilon$$

6. Jeśli różnica pomiędzy macierzami  $\mathbf{U}$  w kolejnych iteracjach jest mniejsza od założonego  $\varepsilon$  to kończymy proces iteracji, jeśli nie to idź do 2

# Przykład



# Algorytm VQ



# Algorytm VQ

---

- Algorytm kwantyzacji wektorów (VQ = ang. vector quantization)
- Podobna zasada działania do k-Średnich ale aktualizacji położenia wektora odbywa się po każdej prezentacji wektora uczącego
- W sieciach VQ centrum klastra nazywane jest również wektorem kodującym, które w terminologii sieci neuronowych nazywane są również neuronami.
- W sieciach VQ wagi neuronów określają położenie neuronu w przestrzeni wejściowej (podobnie jak w algorytmie k-Średnich)

# Formuła aktualizacji

---

- Formuła aktualizacji położenia wektorów kodujących:

$$\mathbf{v}_i = \mathbf{v}_i + \alpha (\mathbf{x}_j - \mathbf{v}_i)$$

Gdzie:

$\mathbf{v}_i$  wektor kodujący podlegający aktualizacji (wektor kodujący nażący najbliższej wektora  $\mathbf{x}_j$ )

$\alpha$  Współczynnik uczenia – maleje z każdą iteracją programu

$\mathbf{x}_j$  j-ty wektory uczący

# Algorytm VQ

---

1. Zainicjuj położenie wektorów kodujących
2. Iteracyjnie  $l$ -razy
  1. Dla każdego wektora treningowego
    1. Znajdź najbliższy wektor kodujący (dla danej metryki)
    2. Dokonaj aktualizacji położenia (wag) neuronu zgodnie z zależnością (1)
  2. Dokonaj aktualizacji wsp.  $\alpha$  wg. zależności (2)

# Algorytm VQ

---

## Inicjalizacja wektorów kodujących:

1. Przez wybór wektorów kodujących spośród wektorów uczących –
  - w tym celu najlepiej jest wykorzystać istniejące już wektory danych i losowo wybrać ze zbioru danych wektory których położenie będzie inicjowało położenie neuronów
2. Poprzez w pełni losową inicjalizację – zbadaj przestrzeń zmienności zmiennych wejściowych i wylosuj położenia wektorów kodujących
3. Poprzez PCA – dokonaj redukcji wymiarowości korzystając z PCA, wyznacz w zredukowanej przestrzeni położenia wektorów kodujących i przetransformuj je do przestrzeni oryginalnej



# Algorytm VQ a k-Średnich

---

- Który lepszy?
- Algorytm k-Średnich – większa złożoność pamięciowa – konieczność przechowywania macierzy przynależności.  
Dla dużych danych i dużej liczby klasterów jest to istotne ograniczenie
- Algorytm VQ – mniejsza złożoność pamięciowa ale większa złożoność obliczeniowa –
  - każdy wektor treningowy powoduje konieczność aktualizacji położenia prototypu
  - Aktualizacja położenia prototypu powoduje konieczność aktualizacji – ponownego liczenia odległości.
  - Czuły na kolejność prezentacji wzorców – dlatego zalecane jest na wstępie losowe poukładanie wektorów.
- Z doświadczenia – algorytm VQ jest lepszy od k-Średnich dzięki pozytywnemu wpływowi chaosu!!!

# Sieci SOM



# Sieci SOM

---

- SOM – samoorganizujące mapy Kochonena
- Idea podobna do VQ z pewnymi dodatkami
- Dodatki to narzucenie wymogów sąsiedztwa na wektory kodujące
- Strategia typu WTM – Winner Take Most, w sieciach VQ strategia WTA – Winner Take All

# Strategia WTM

---