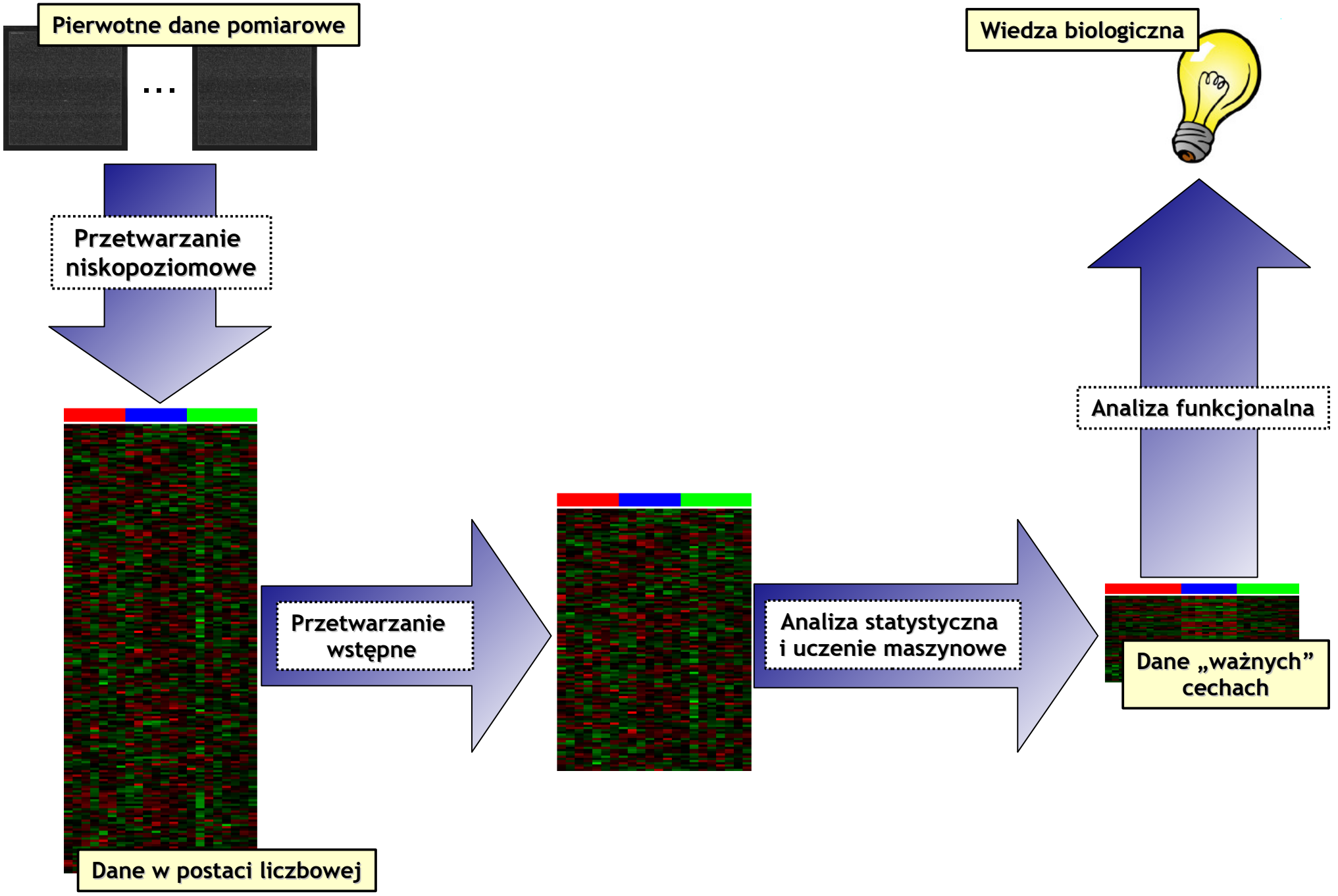
Uczenie maszynowe w bioinformatyce

Wykład 6: klasteryzacja

Tymon Rubel

Zakład Elektroniki Jądrowej i Medycznej
Instytut Radioelektroniki i Technik Multimedialnych PW

Etapy przetwarzania i analizy danych



Etapy przetwarzania i analizy danych

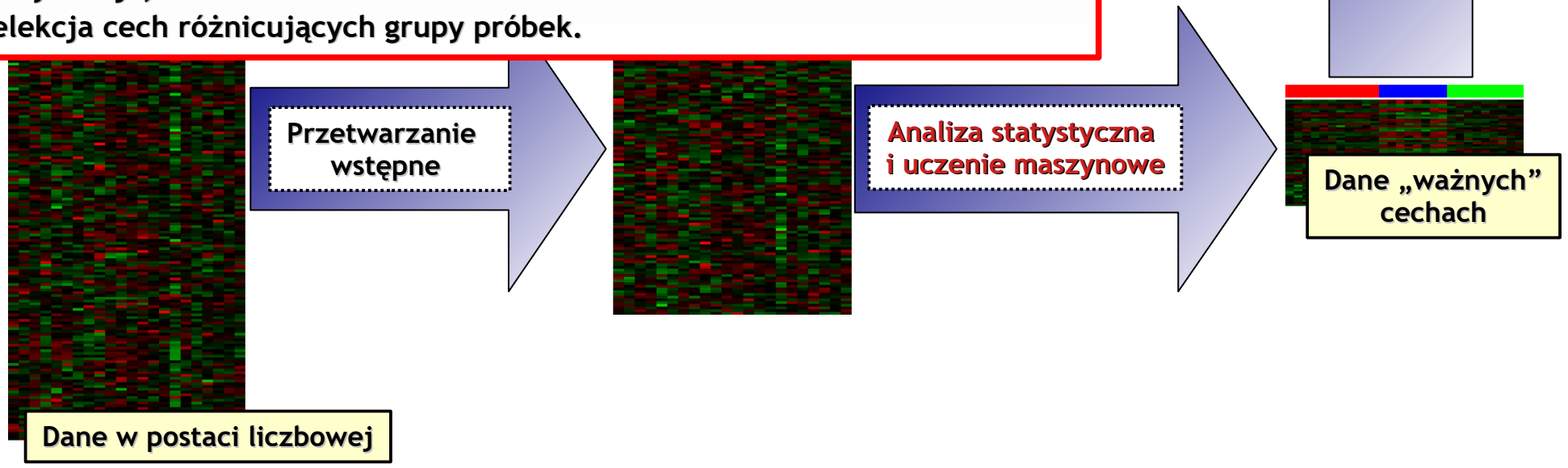
Wybór **metod statystycznych i uczenia się maszyn** używanych na etapie analizy danych jest ściśle uzależniony od celu badań. Stosowane tutaj techniki można w ogólności podzielić na **działające z nadzorem** lub **nienadzorowane**.

Metody działające bez nadzoru stosuje się do wykrywania zależności pomiędzy próbkami i/lub cechami jedynie w oparciu o dane, bez używania dodatkowych informacji (w tym też nie uwzględniając przynależności badanych obiektów do zadanych z góry klas). Do metod nienadzorowanych zaliczają się m. in.:

- klasteryzacja;
- techniki redukcji wymiarowości.

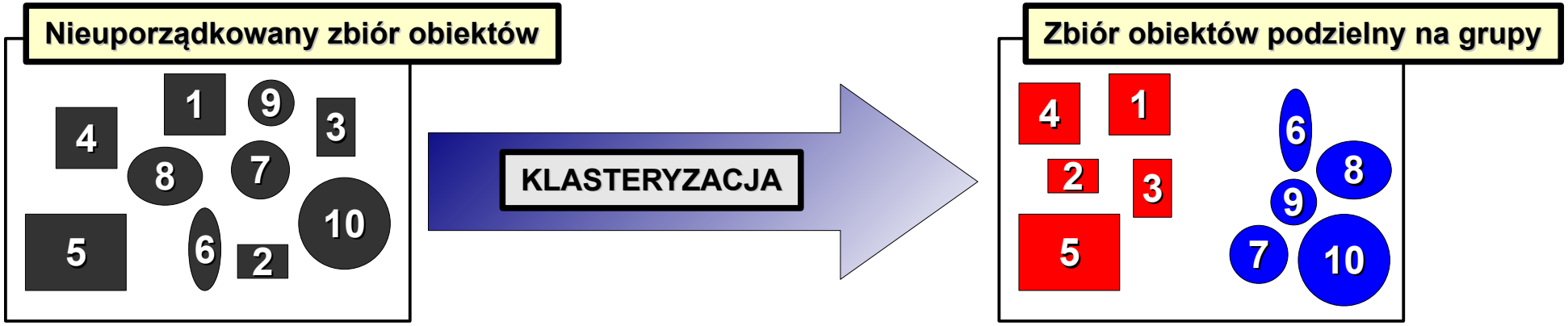
Metody nadzorowane korzystają z danych i dodatkowej wiedzy, dotyczącej np. oczekiwanych wartości na wyjściu lub sposobu przypisania próbek do znanych klas. Przykładami zagadnień, w których używa się takich metod są:

- klasyfikacja;
- selekcja cech różnicujących grupy próbek.



Klasteryzacja

Klasteryzacja (analiza skupień, grupowanie) oznacza podzielenie zbioru obiektów na pewną liczbę rozłącznych **klastrów (skupisk, grup)**, w taki sposób, aby każdy klaster zawierał obiekty wzajemnie do siebie podobne (wedle ustalonego kryterium), przy zachowaniu możliwie dużego niepodobieństwa do obiektów z pozostałych grup.



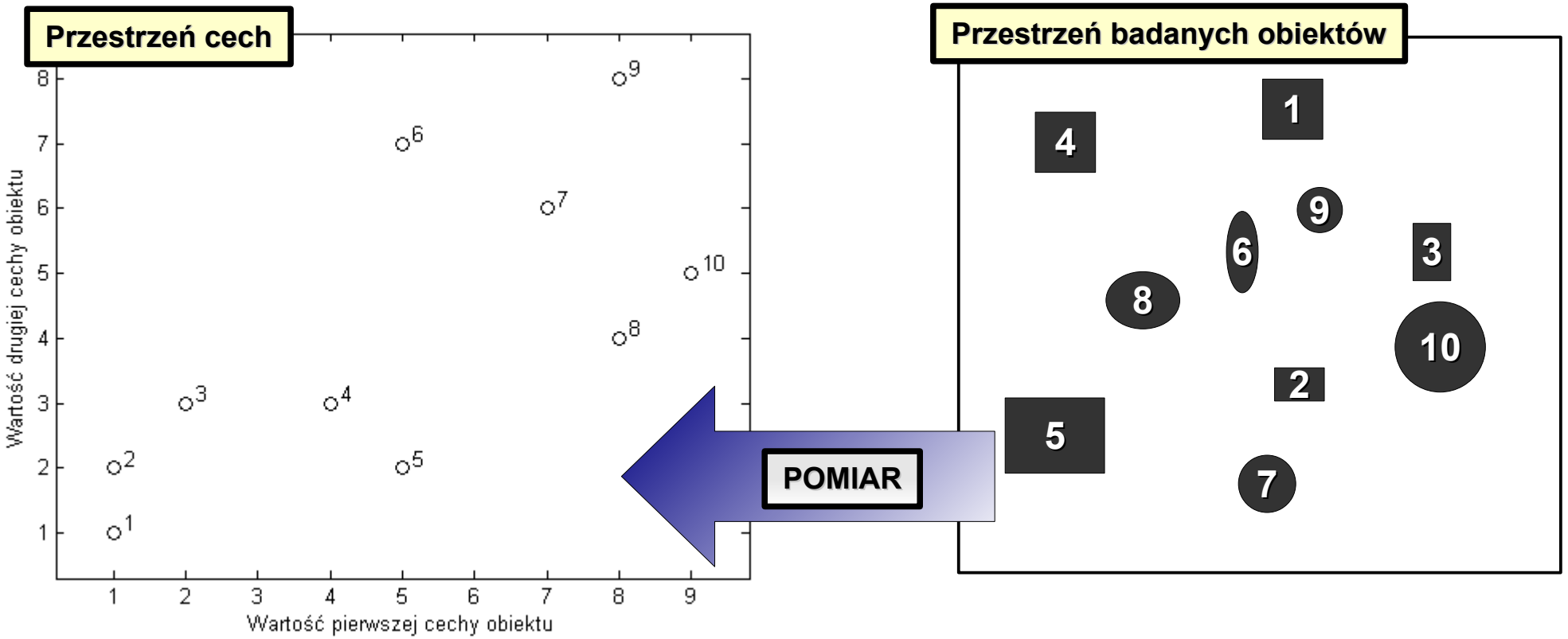
Klasteryzacja odbywa się w sposób **nienadzorowany**, co oznacza, że obiekty łączone są w grupy jedynie w oparciu o zmierzone wartości opisujących je cech. Tym samym jej wykonanie nie wymaga wstępnej wiedzy o przynależności obiektów do znanych z góry klas (jeżeli jednak taka wiedza istnieje, to może być wykorzystana do nadania interpretacji wynikom klasteryzacji).

Z powyższego stwierdzenia wynika, że klasteryzacja jest techniką o charakterze **odkrywczym (eksploracyjnym)**, czyli może posłużyć do wykrywania na podstawie danych nieznanych wcześniej zależności pomiędzy badanymi obiektami.

Klasteryzacja

Przebieg wnioskowania przy wykorzystaniu klasteryzacji:

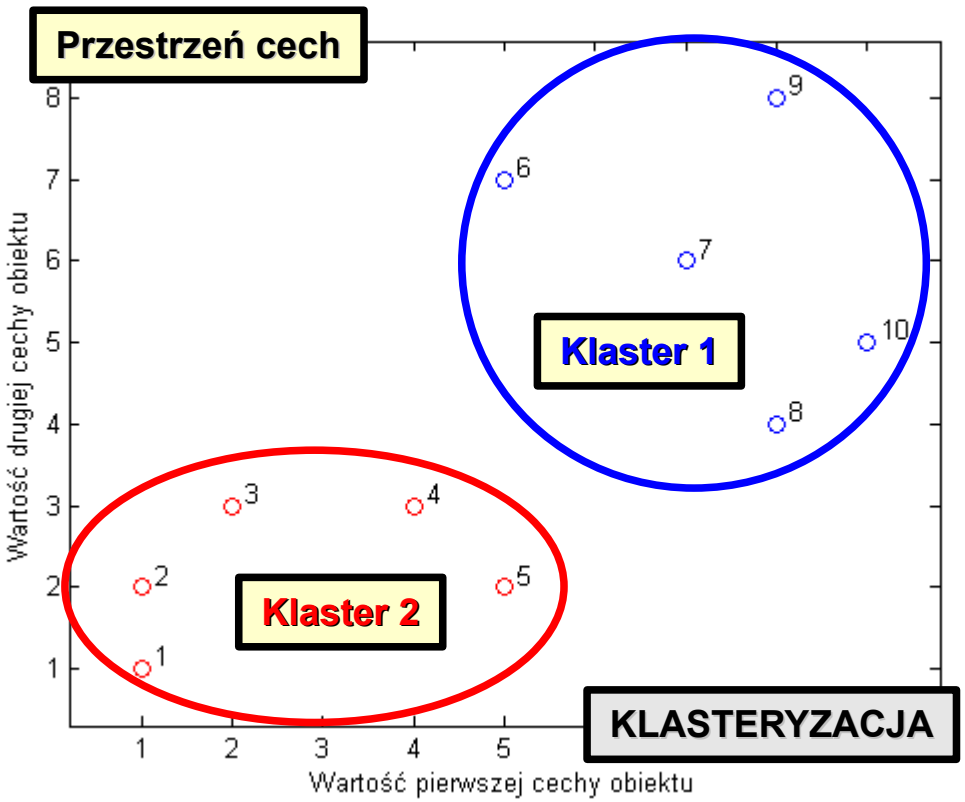
1. **Pomiar wartości cech opisujących obiekty (przeniesienie problemu z pierwotnej przestrzeni obiektów do przestrzeni cech).**
2. **Wykonanie klasteryzacji wektorów cech opisujących obiekty.**
3. **Przeniesienie wyników klasteryzacji do przestrzeni obiektów i próba nadania im interpretacji w oparciu o dodatkową wiedzę. Zakłada się przy tym, że podział na klastry w przestrzeni cech odzwierciedla rzeczywiste relacje pomiędzy obiektami.**

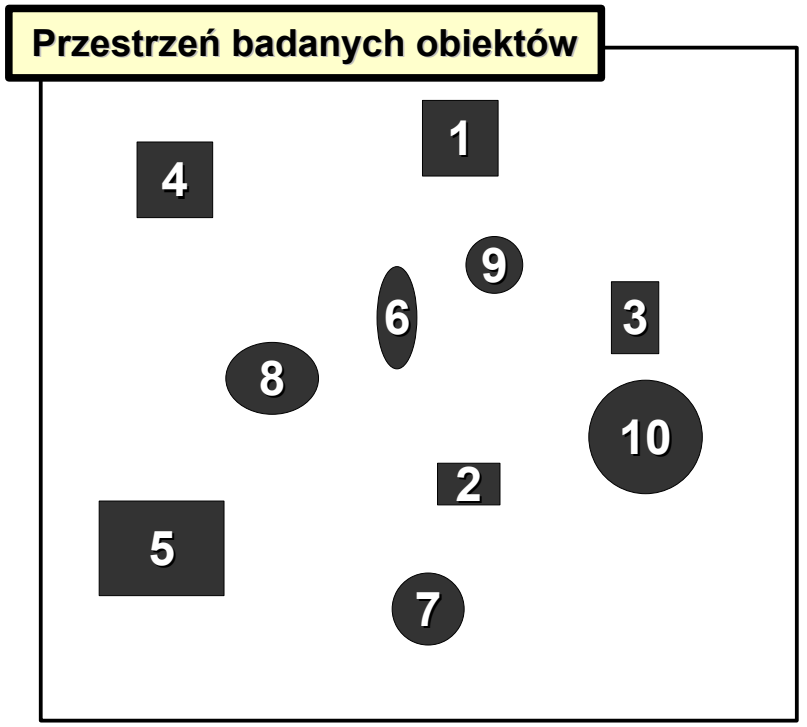


Klasteryzacja

Przebieg wnioskowania przy wykorzystaniu klasteryzacji:

1. **Pomiar wartości cech opisujących obiekty (przeniesienie problemu z pierwotnej przestrzeni obiektów do przestrzeni cech).**
2. **Wykonanie klasteryzacji wektorów cech opisujących obiekty.**
3. **Przeniesienie wyników klasteryzacji do przestrzeni obiektów i próba nadania im interpretacji w oparciu o dodatkową wiedzę. Zakłada się przy tym, że podział na klastry w przestrzeni cech odzwierciedla rzeczywiste relacje pomiędzy obiektami.**

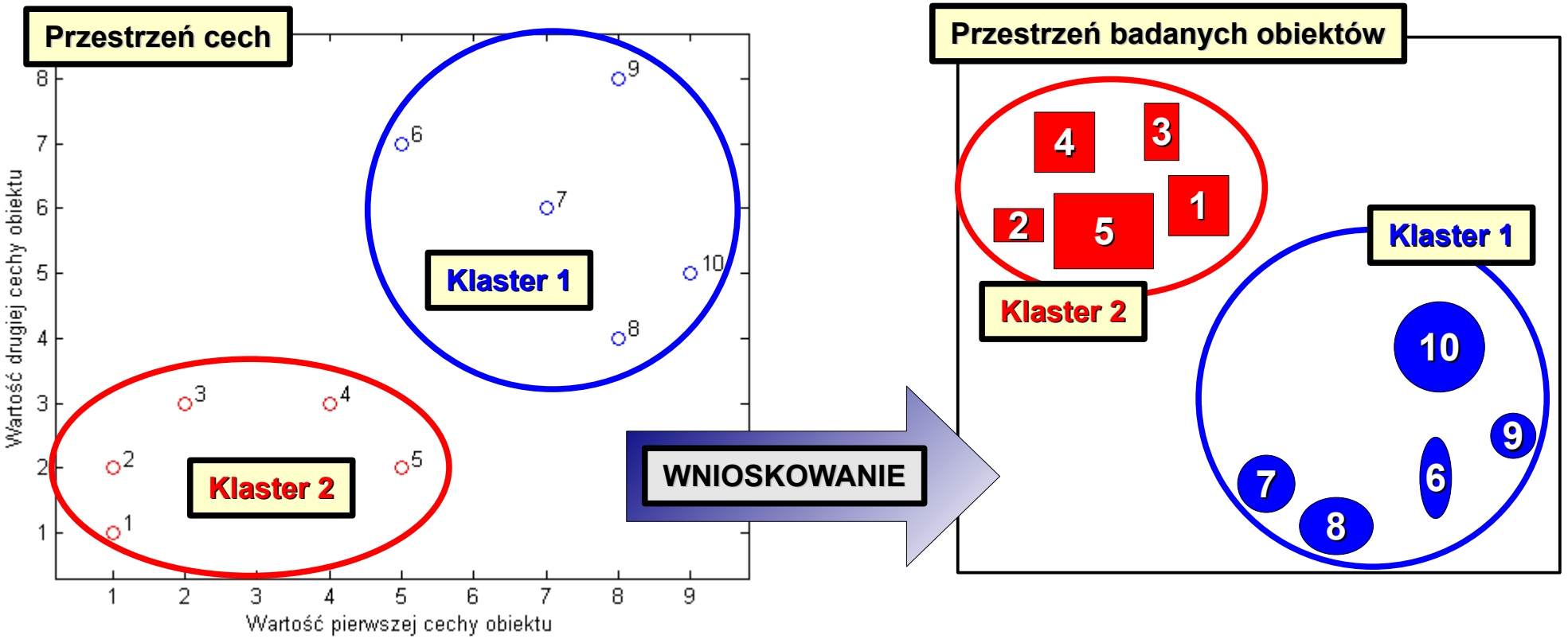




Klasteryzacja

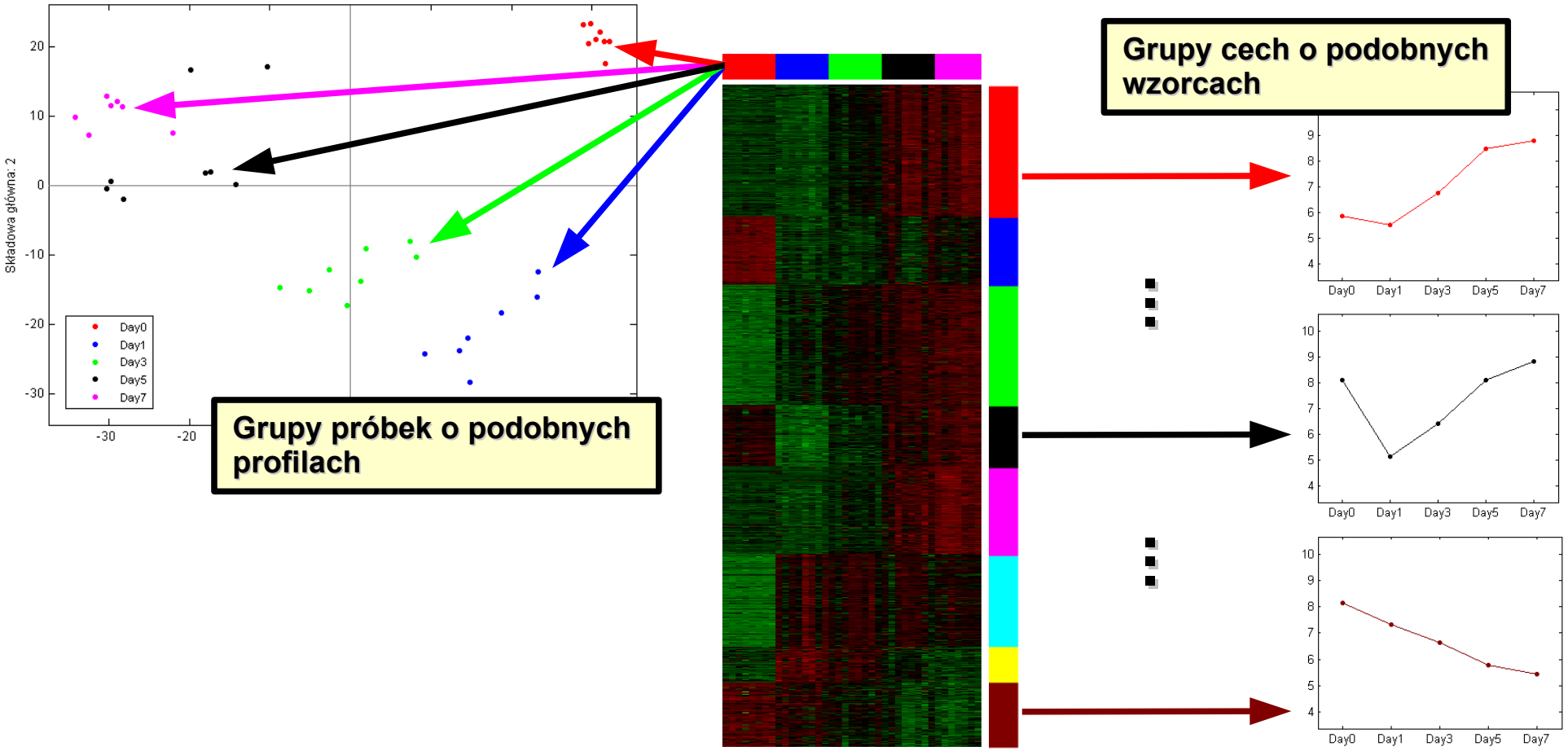
Przebieg wnioskowania przy wykorzystaniu klasteryzacji:

1. **Pomiar wartości cech opisujących obiekty (przeniesienie problemu z pierwotnej przestrzeni obiektów do przestrzeni cech).**
2. **Wykonanie klasteryzacji wektorów cech opisujących obiekty.**
3. **Przeniesienie wyników klasteryzacji do przestrzeni obiektów i próba nadania im interpretacji w oparciu o dodatkową wiedzę. Zakłada się przy tym, że podział na klastry w przestrzeni cech odzwierciedla rzeczywiste relacje pomiędzy obiektami.**



Klasteryzacja

Klasteryzacji poddawane mogą być zarówno próbki, jak i cechy (np. geny, białka). Oba rodzaje obiektów reprezentowane są przez wektory liczb rzeczywistych, będące kolumnami (próbki) lub wierszami (cechy) macierzy danych X.



Wykonując klasteryzację mamy nadzieję, że możliwe będzie nadanie interpretacji biologicznej wykrytym grupom, np. klastry będą zawierać geny uczestniczące w tych samych procesach biologicznych lub próbki od pacjentów o jednakowej diagnozie.

Klasteryzacja

Pomimo pozornej prostoty, zadanie klasteryzacji jest trudne w praktycznej realizacji, szczególnie dla dużych zbiorów danych.

Nawet przy znajomości właściwego kryterium grupowania (co wcale nie jest sytuacją oczywistą) problemem pozostaje fakt, że zwykle nie istnieje możliwość sprawdzenia wszystkich podziałów na klastry i wybrania tego, który optymalizuje dane kryterium.

Liczba możliwych podziałów zbioru N wektorów cech na K klastrów wynosi:

$$L(N,K)=\frac{1}{K!}\sum_{i=1}^{K}(-1)^{K-i}\binom{K}{i}i^{N}$$

$$L(10,3)=9330$$

$$L(50,4)\approx 5.3\cdot 10^{28}$$

$$L(100,5)\approx 6.6\cdot 10^{67}$$

W efekcie problem klasteryzacji – z wyjątkiem trywialnych przypadków – nie może być rozwiązany w sposób dokładny (optymalny w sensie globalnym). Zamiast tego stosuje się różnego rodzaju algorytmy rozwiązywania go w sposób przybliżony.

Spotykane algorytmy znacząco różnią się pomiędzy sobą w wielu elementach, w tym m.in.: definicją pojęcia „klaster”, funkcją celu (przyjętym kryterium grupowania) oraz stosowanymi miarami podobieństwa obiektów.

Uczenie maszynowe w bioinformatyce

Wykład 8: klasteryzacja (miary niepodobieństwa)

Tymon Rubel

Zakład Elektroniki Jądrowej i Medycznej
Instytut Radioelektroniki i Technik Multimedialnych PW

Klasteryzacja: miary niepodobieństwa

W przypadku znacznej części algorytmów grupowanie odbywa się w oparciu o miarę niepodobieństwa pomiędzy obiektami zbioru A. Jest to funkcja $d: A \rightarrow [0,\infty)$, która dla dowolnych elementów a, b zbioru A spełnia warunki:

$$d(\boldsymbol{a},\boldsymbol{b}) = 0 \quad \Leftrightarrow \quad \boldsymbol{a} = \boldsymbol{b}$$

$$d(\boldsymbol{a},\boldsymbol{b}) > 0 \quad \Leftrightarrow \quad \boldsymbol{a} \neq \boldsymbol{b}$$

$$d(\boldsymbol{a},\boldsymbol{b}) = d(\boldsymbol{b},\boldsymbol{a})$$

Miara niepodobieństwa jest jednocześnie metryką (funkcją odległości) jeżeli dodatkowo dla każdej trójki elementów a, b, c zbioru A spełnia nierówność trójkąta:

$$d(\boldsymbol{a},\boldsymbol{c}) \leq d(\boldsymbol{a},\boldsymbol{b}) + d(\boldsymbol{b},\boldsymbol{c})$$

Klasteryzacja: macierz niepodobieństwa

Znaczna część algorytmów klasteryzacji korzysta z macierzy niepodobieństwa, której każdy element d_{ij} reprezentuje wartość miary niepodobieństwa $d(i, j)$ pomiędzy i-tym a j-tym obiektem:

$$\boldsymbol{D} = \begin{bmatrix} d_{11} & d_{12} & d_{13} & \cdots & d_{1N} \\ d_{21} & d_{22} & d_{23} & \cdots & d_{2N} \\ d_{31} & d_{32} & d_{32} & \cdots & d_{3N} \\ \vdots & \vdots & \vdots & & \vdots \\ d_{N1} & d_{N2} & d_{N3} & \cdots & d_{NN} \end{bmatrix} = \begin{bmatrix} 0 & d_{21} & d_{31} & \cdots & d_{N1} \\ d_{21} & 0 & d_{32} & \cdots & d_{N2} \\ d_{31} & d_{32} & 0 & \cdots & d_{N3} \\ \vdots & \vdots & \vdots & & \vdots \\ d_{N1} & d_{N2} & \cdots & d_{N(N-1)} & 0 \end{bmatrix}$$

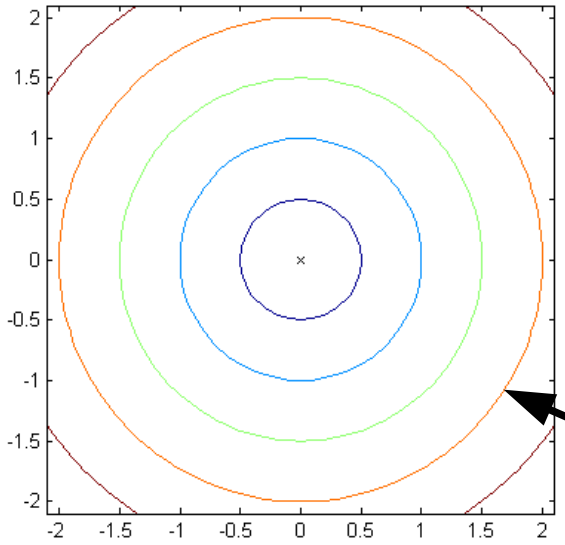
Z właściwości miary niepodobieństwa wynika, że jest to macierz symetryczna ($d(i, j)=d(j, i)$), a wszystkie elementy na jej głównej przekątnej są równe 0 ($d(i, j)=0$). Oznacza to, że można ograniczyć się do przechowywania jej $N(N - 1)/2$ elementów leżących poniżej (lub powyżej) głównej przekątnej.

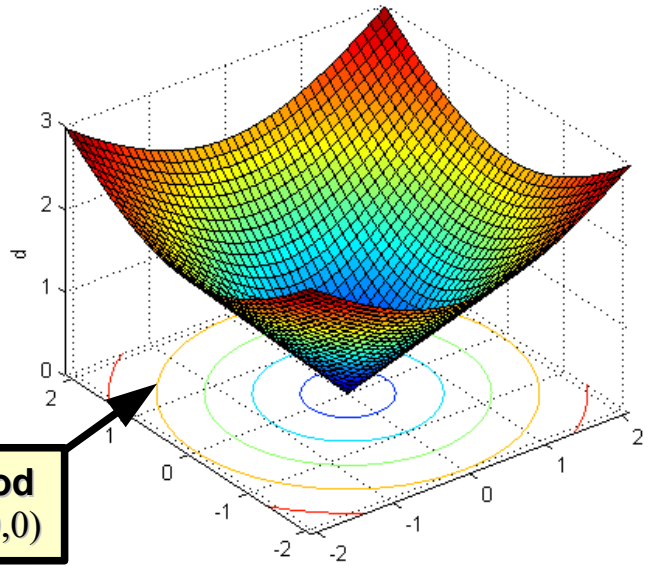
	MATLAB	R
Macierz niepodobieństwa:	`pdist`	`dist`

Klasteryzacja: miary niepodobieństwa

Odległość Euklidesa

$$d(\boldsymbol{a},\boldsymbol{b}) = \left((\boldsymbol{a}-\boldsymbol{b})^T(\boldsymbol{a}-\boldsymbol{b})\right)^{1/2} = \sqrt{\sum_{i=1}^{P}(a_i-b_i)^2}$$



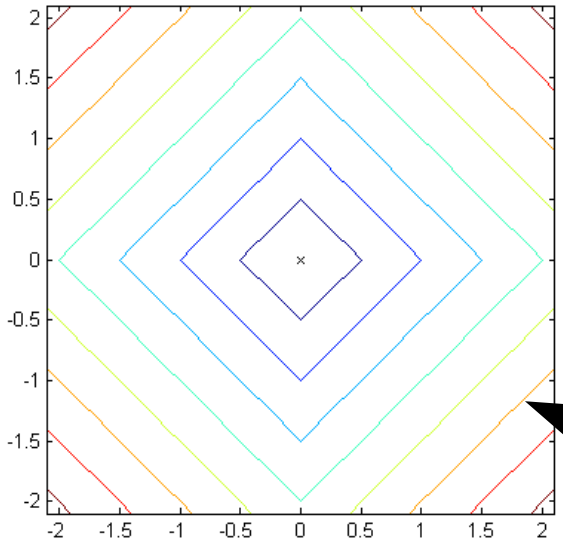


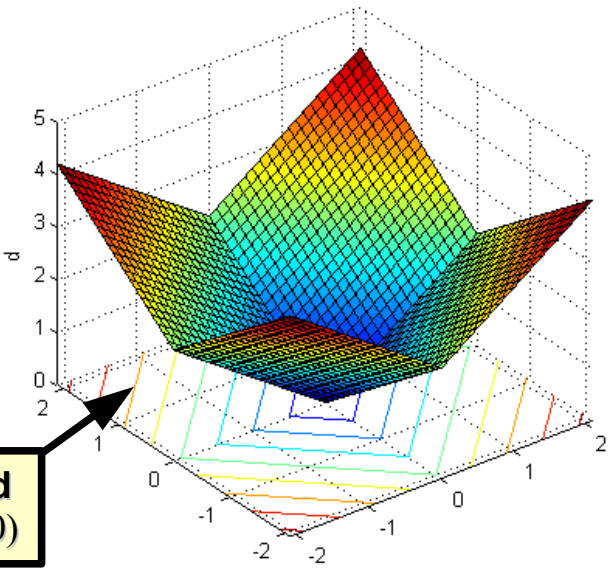
Krzywe równej odległości od punktu o współrzędnych $(0,0)$

Klasteryzacja: miary niepodobieństwa

Odległość miejska (*manhattan, city-block*)

$$d(\boldsymbol{a},\boldsymbol{b}) = \sum_{i=1}^{P} |a_i - b_i|$$



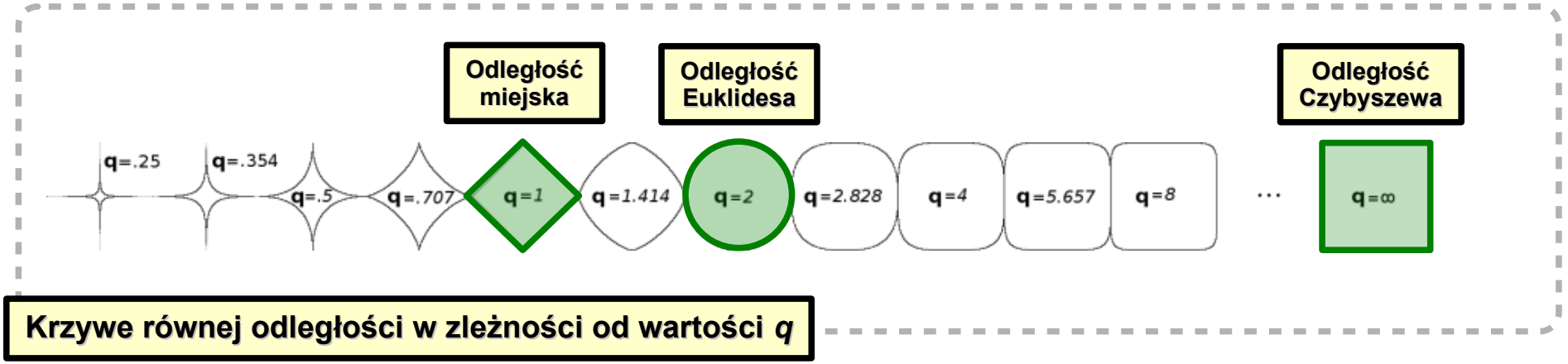


Krzywe równej odległości od punktu o współrzędnych $(0,0)$

Klasteryzacja: miary niepodobieństwa

Obie wymienione metryki są szczególnymi przypadkami odległości Minkowskiego.

$$d(\boldsymbol{a},\boldsymbol{b}) = \left(\sum_{i=1}^{P} |a_i - b_i|^q \right)^{1/q}$$



Krzywe równej odległości w zleżności od wartości *q*

Wszystkie miary odległości należące do tej rodziny zależą jedynie od współrzędnych punktów. Jednak istnieją też miary odległości uwzględniające charakterystykę zbioru danych (przykładami są ważona odleglość Euklidesa oraz odległość Mahalanobisa).

Klasteryzacja: miary niepodobieństwa

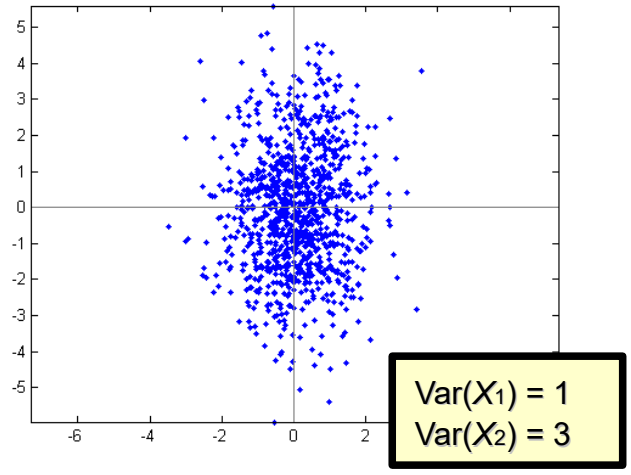
Ważona odległość Euklidesa

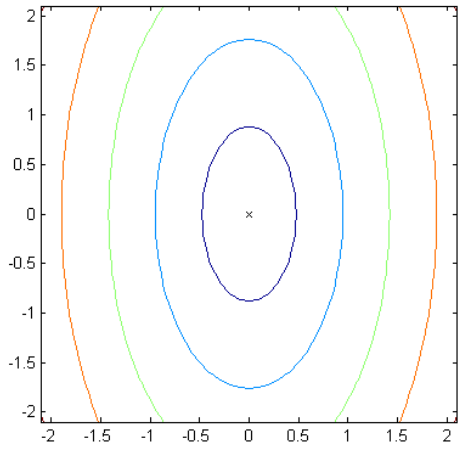
$$d(\boldsymbol{a},\boldsymbol{b}) = \left((\boldsymbol{a}-\boldsymbol{b})^T \boldsymbol{W}^{-1}(\boldsymbol{a}-\boldsymbol{b})\right)^{1/2} = \sqrt{\sum_{i=1}^{P} \frac{1}{w_i}(a_i - b_i)^2}$$

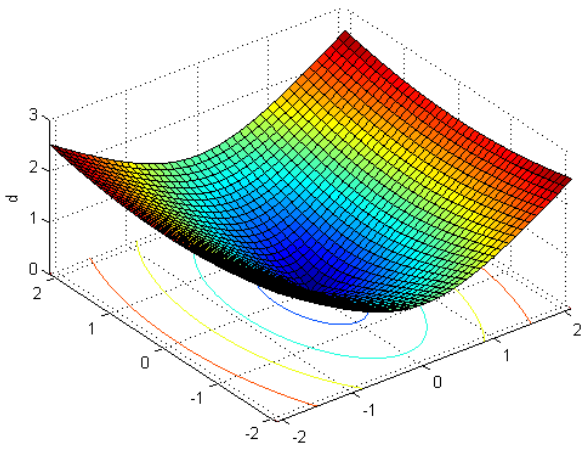
W – macierz diagonalna, której element w_i na głównej przekątnej jest równy estymowanej wariancji i-tej cechy (i-tego wiersza macierzy danych X):

$$w_i = s_i^2 = \hat{\sigma}_i^{\,2} = \frac{1}{N-1}\sum_{j=1}^{N}(x_{ij} - \bar{x}_i)^2$$

Dzięki wagom uwzględniane są różnice skali (rozrzutu wartości) pomiędzy cechami.







Jeżeli wariancja każdej cechy równa się 1 (np. w wyniku standaryzacji), to odległość ta jest równoważna zwykłej odległości Euklidesa.

Klasteryzacja: miary niepodobieństwa

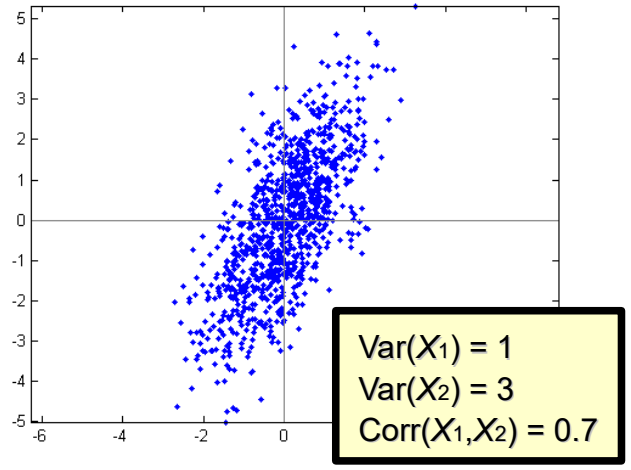
Odległość Mahalanobisa

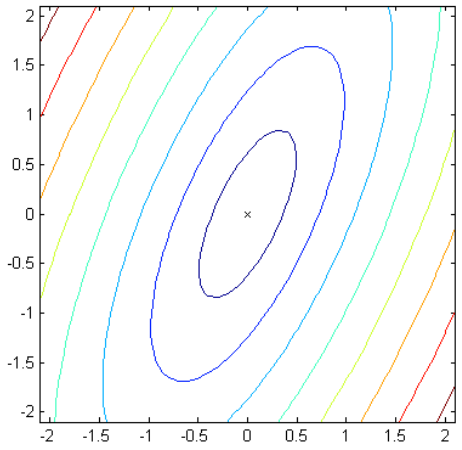
$$d(\boldsymbol{a},\boldsymbol{b}) = \left((\boldsymbol{a}-\boldsymbol{b})^T \boldsymbol{W}^{-1} (\boldsymbol{a}-\boldsymbol{b})\right)^{1/2}$$

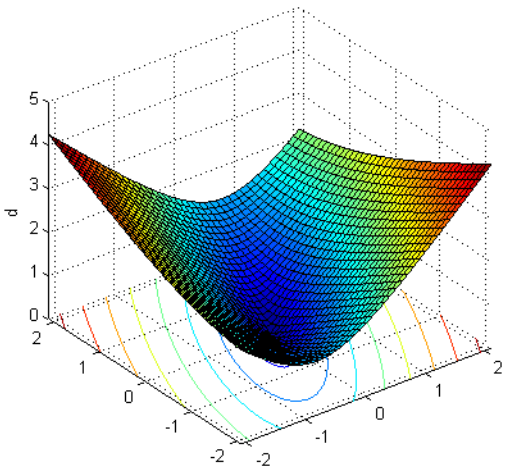
W – symetryczna macierz równa estymowanej macierzy kowariancji cech:

$$\boldsymbol{W} = \boldsymbol{S} = \hat{\boldsymbol{\Sigma}} = \frac{1}{N-1}\sum_{i=1}^{N}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T = \frac{1}{N-1}(\boldsymbol{X} - \bar{\boldsymbol{x}})(\boldsymbol{X} - \bar{\boldsymbol{x}})^T$$

Odległość ta uwzględnia nie tylko różnice w rozrzutach wartości cech, ale również korelacje występujące pomiędzy cechami.







W przypadku braku korelacji jest toższama z ważoną odległością Euklidesa.

Klasteryzacja: miary niepodobieństwa

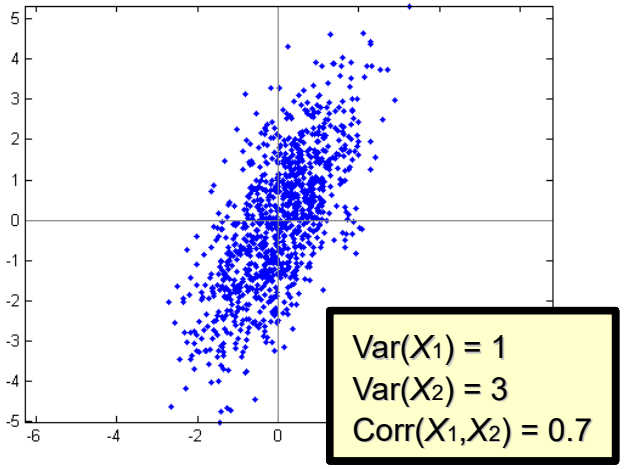
Odległość Mahalanobisa

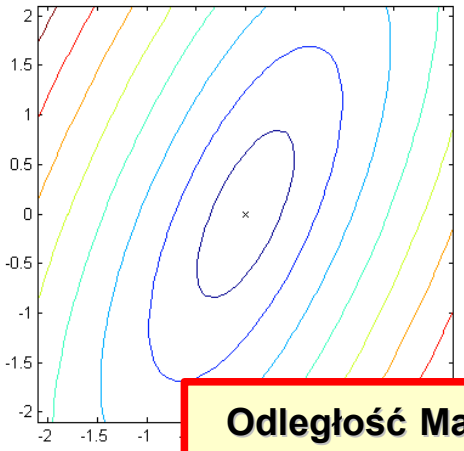
$$d(\boldsymbol{a},\boldsymbol{b}) = \left((\boldsymbol{a}-\boldsymbol{b})^T \boldsymbol{W}^{-1} (\boldsymbol{a}-\boldsymbol{b})\right)^{1/2}$$

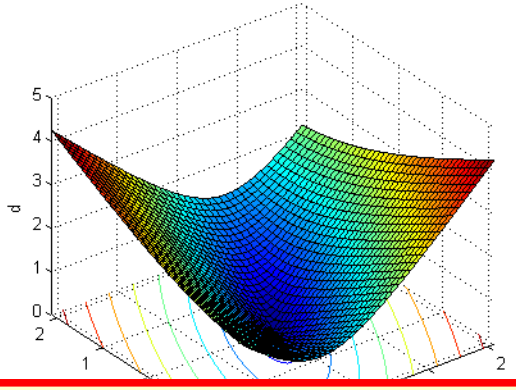
W – symetryczna macierz równa estymowanej macierzy kowariancji cech:

$$\boldsymbol{W} = \boldsymbol{S} = \hat{\boldsymbol{\Sigma}} = \frac{1}{N-1}\sum_{i=1}^{N}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T = \frac{1}{N-1}(\boldsymbol{X} - \bar{\boldsymbol{x}})(\boldsymbol{X} - \bar{\boldsymbol{x}})^T$$

Odległość ta uwzględnia nie tylko różnice w rozrzutach wartości cech, ale również korelacje występujące pomiędzy cechami.







W przypadku braku korelacji jest tożsa

Odległość Mahalanobisa *d*(*μ*, *b*) można interpretować jako odległość między wektorem *b*, a *P*-wymiarowym rozkładem normalnym *N*(*μ*, *Σ*). Wówczas linie równej odległości od wektora wartości średnich *μ* oznaczają linie jednakowego prawdopodobieństwa.

Klasteryzacja: miary niepodobieństwa

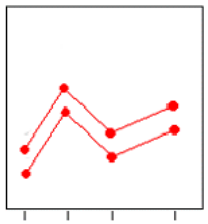
Miary niepodobieństwa wynikające ze współczynnika korelacji

$$d(\boldsymbol{a}, \boldsymbol{b}) = 1 - r(\boldsymbol{a}, \boldsymbol{b})$$

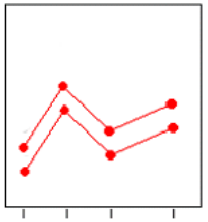
$$d(\boldsymbol{a}, \boldsymbol{b}) = 1 - r(\boldsymbol{a}, \boldsymbol{b})^2$$

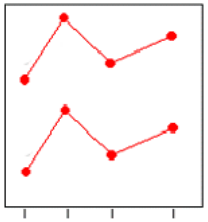
gdzie $r(a, b)$ jest współczynnikiem korelacji (liniowej lub Spearmana) pomiędzy a i b.

Obie są niezależne od wariancji i wartości średnich cech, dzięki czemu pozwalają grupować cechy o podobnych wzorcach nawet w przypadku braku normalizacji.

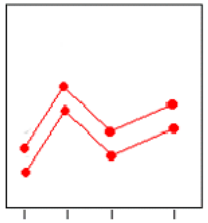


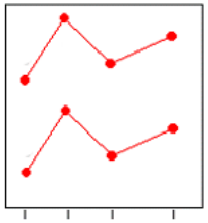
Wzorce podobne w sensie odległości Euklidesa

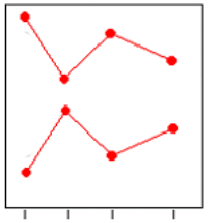




Wzorce podobne w sensie współczynnika korelacji







Wzorce podobne w sensie kwadratu współczynnika korelacji

Uczenie maszynowe w bioinformatyce

Wykład 6: klasteryzacja (sposoby definiowania klastrów)

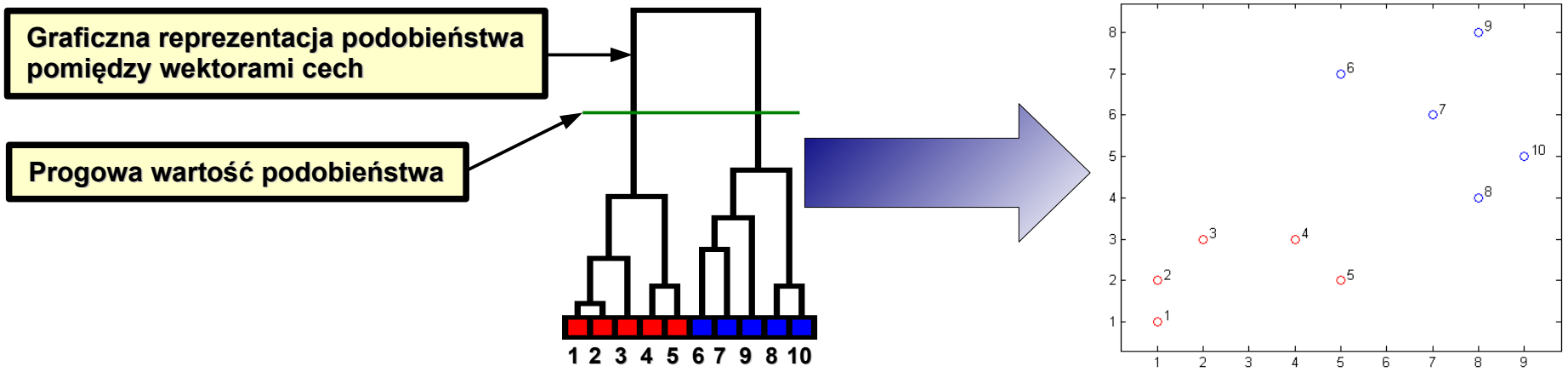
Tymon Rubel

Zakład Elektroniki Jądrowej i Medycznej
Instytut Radioelektroniki i Technik Multimedialnych PW

Klasteryzacja: różne definicje klastra

Najistotniejszym elementem każdego algorytmu jest sposób w jaki zdefiniowane są klastry. Najczęściej spotykanymi podejściami są:

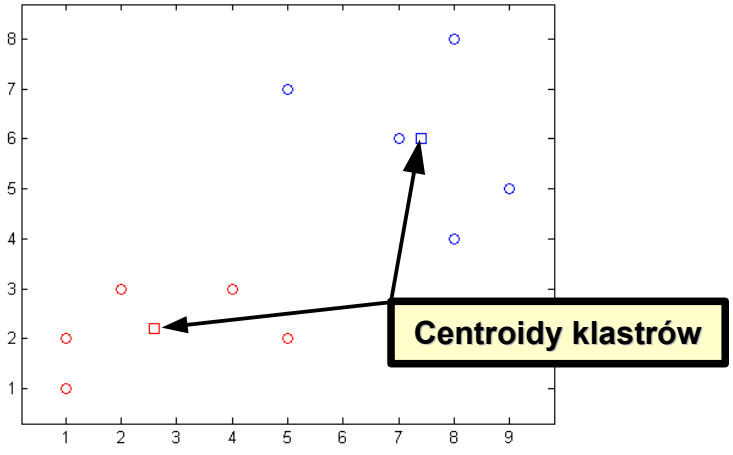
- **brak formalnej definicji klastrów:** algorytm tworzy jedynie hierarchiczną strukturę odwzorowującą relacje podobieństwa pomiędzy wektorami cech. Przykładami tej kategorii są **algorytmy aglomeracyjnej klasteryzacji hierarchicznej**;



Klasteryzacja: różne definicje klastra

Najistotniejszym elementem każdego algorytmu jest sposób w jaki zdefiniowane są klastry. Najczęściej spotykanymi podejściami są:

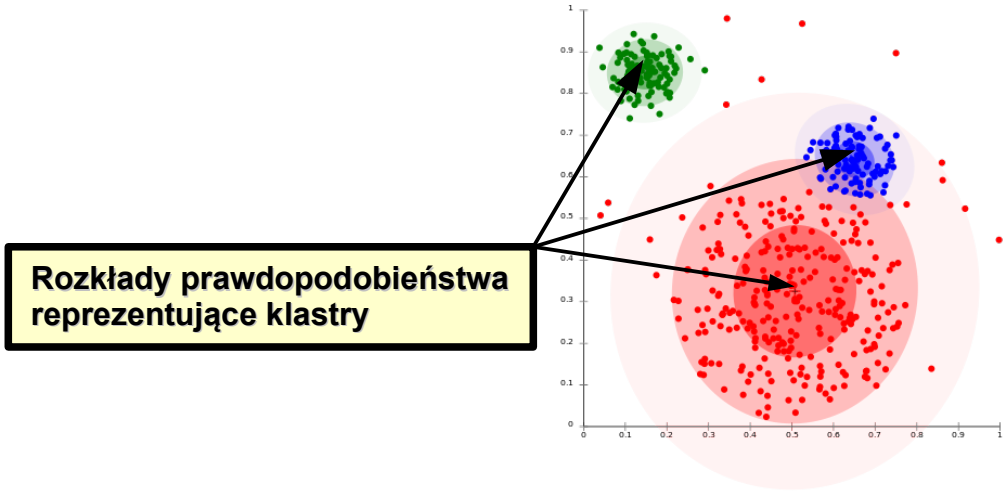
- **klastry reprezentowane poprzez centroidy, czyli „centralne” wektory, wokół których grupowane są elementy zbioru danych. Liczba centroidów może być narzucona z góry lub wynikać działania algorytmu. Przykład: algorytm *K*-średnich;**



Klasteryzacja: różne definicje klastra

Najistotniejszym elementem każdego algorytmu jest sposób w jaki zdefiniowane są klastry. Najczęściej spotykanymi podejściami są:

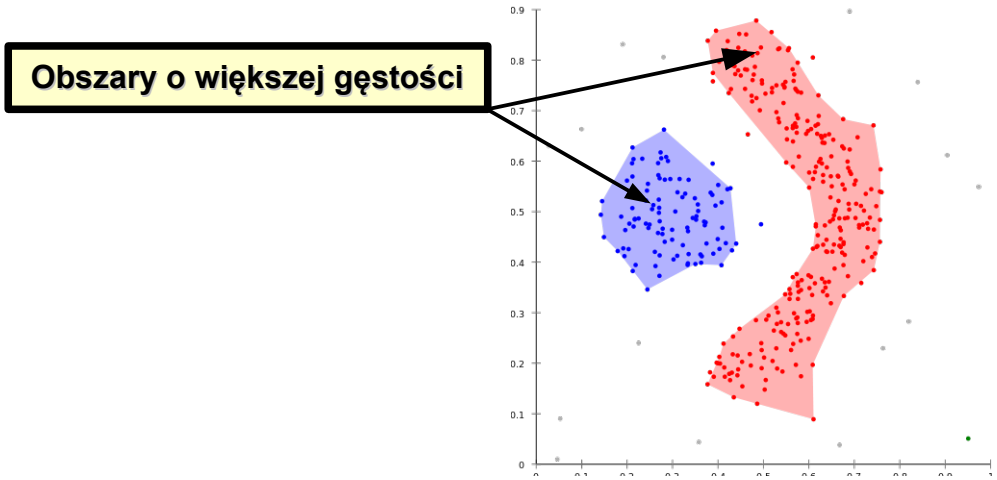
- **klastry reprezentowane przez wielowymiarowe rozkłady prawdopodobieństwa: wektory przypisywane są do danego klastra na podstawie prawdopodobieństwa, że pochodzą one ze związanego z tym klastrem rozkładu. Typowym przykładem tego podejścia jest klasteryzacja za pomocą algorytmu E-M (*Expectation-Maximization*);**



Klasteryzacja: różne definicje klastra

Najistotniejszym elementem każdego algorytmu jest sposób w jaki zdefiniowane są klastry. Najczęściej spotykanymi podejściami są:

- **klastry jako obszary o większej gęstości niż resztą przestrzeni cech:** wektory tworzą wspólny klaster jeżeli wzajemnie znajdują się na swoich listach sąsiadów. Klasycznymi przykładami są algorytmy **DBSCAN** i **Jarvisa-Patricka**.



Uczenie maszynowe w bioinformatyce

Wykład 6: klasteryzacja (algorytm *K*-średnich)

Tymon Rubel

Zakład Elektroniki Jądrowej i Medycznej
Instytut Radioelektroniki i Technik Multimedialnych PW

Klasteryzacja: algorytm *K*-średnich

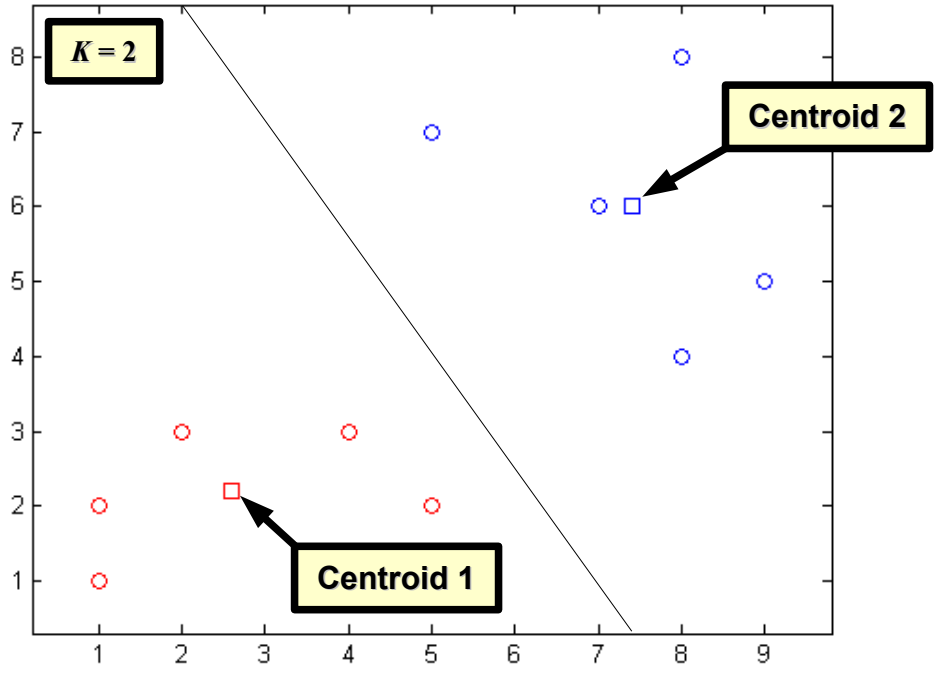
Jedną z najpopularniejszych metod dzielenia zbioru danych na z góry zadaną liczbę klastrów jest algorytm *K*-średnich (*K-means*).

W algorytmie tym każdy klaster C_j reprezentowany jest przez swój centroid c_j, będący wektorem wartości średnich wektorów x_j wchodzących w skład klastra:

$$c_j = \frac{1}{N_j} \sum_{i=1}^{N_j} x_i$$

gdzie $N_j = |C_j|$ jest liczbą wektorów w klastrze C_j.

Przynależność wektora x_j do klastra C_j ustalana jest na podstawie wartości miary niepodobieństwa do centoridu c_j.



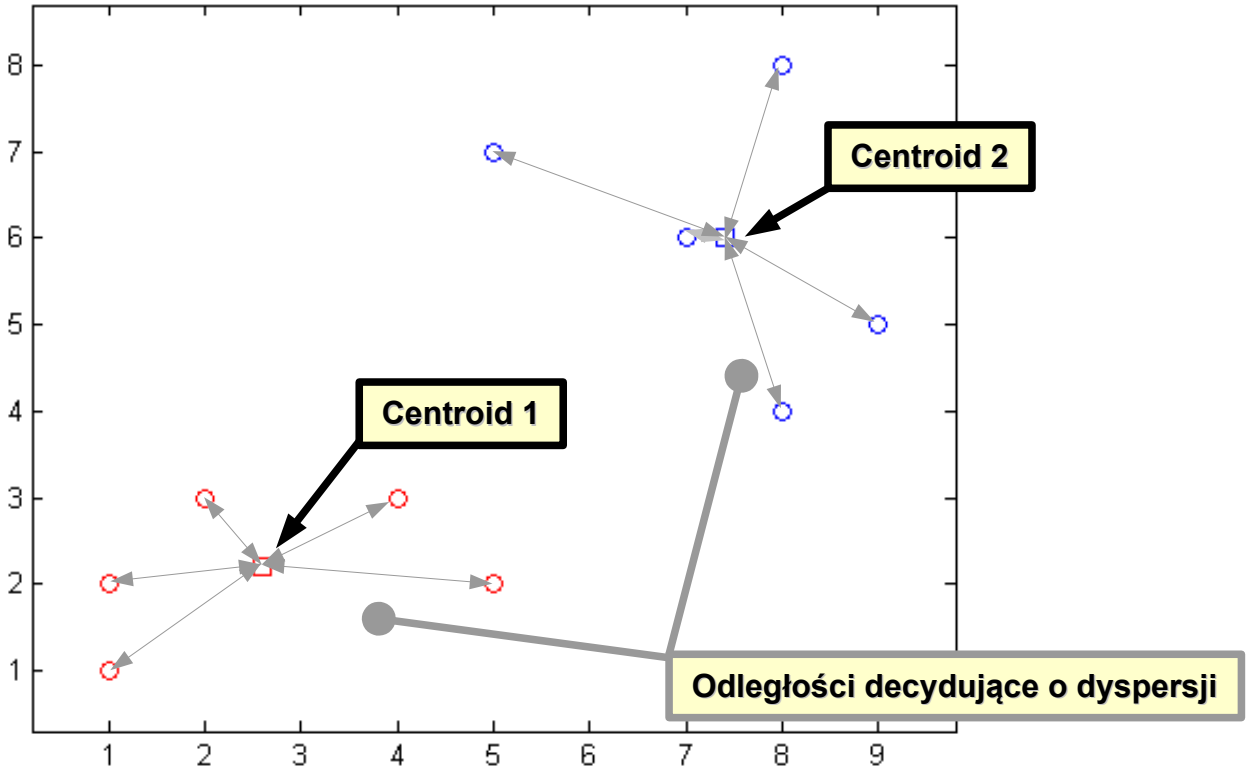
	MATLAB	R
***K*-średnich:**	`kmeans`	`kmeans`

Klasteryzacja: algorytm *K*-średnich

Celem algorytmu jest minimalizacja sumarycznej dyspersji podziału $\zeta_K = \{C_1, .., C_K\}$, równej sumie odległości wektorów od centroidów klastrów, do których należą:

$$D(\zeta_K) = \sum_{j=1}^{K} \sum_{x_i \in C_j} d(\boldsymbol{x}_i, \boldsymbol{c}_j)$$

Przy użyciu kwadratu odległości Euklidesa jako miary niepodobieństwa, dyspersja $D(\zeta_K)$ podzielona przez liczbę wektorów N jest równa błędowi średniokwadratowemu reprezentacji danych, w której wektory zastępowane są centroidami swoich klastrów.



Klasteryzacja: algorytm *K*-średnich

Klasteryzacja rozpoczyna się od utworzenia K wstępnych centoidów (np. poprzez losowy wybór wektorów ze zbioru danych). Wokół początkowych położeń centroidów tworzone są klastry (wektory przypisywane są do najbliższych im centroidów), a następnie w sposób iteracyjny powtarzane są dwa kroki:

- **określane są prawdziwe położenia cetroidów (jako wektory wartości średnich wektorów wchodzących w skład poszczególnych klastrów);**
- **każdy wektor zbioru danych przypisywany jest do klastra C_j, którego centroid c_j znajduje się najbliżej.**

Algorytm przerywany jest w momencie gdy klastry otrzymane w danej iteracji nie różnią się w znaczący sposób od tych otrzymanych w poprzedniej iteracji, co jest równoważne spełnieniu warunku:

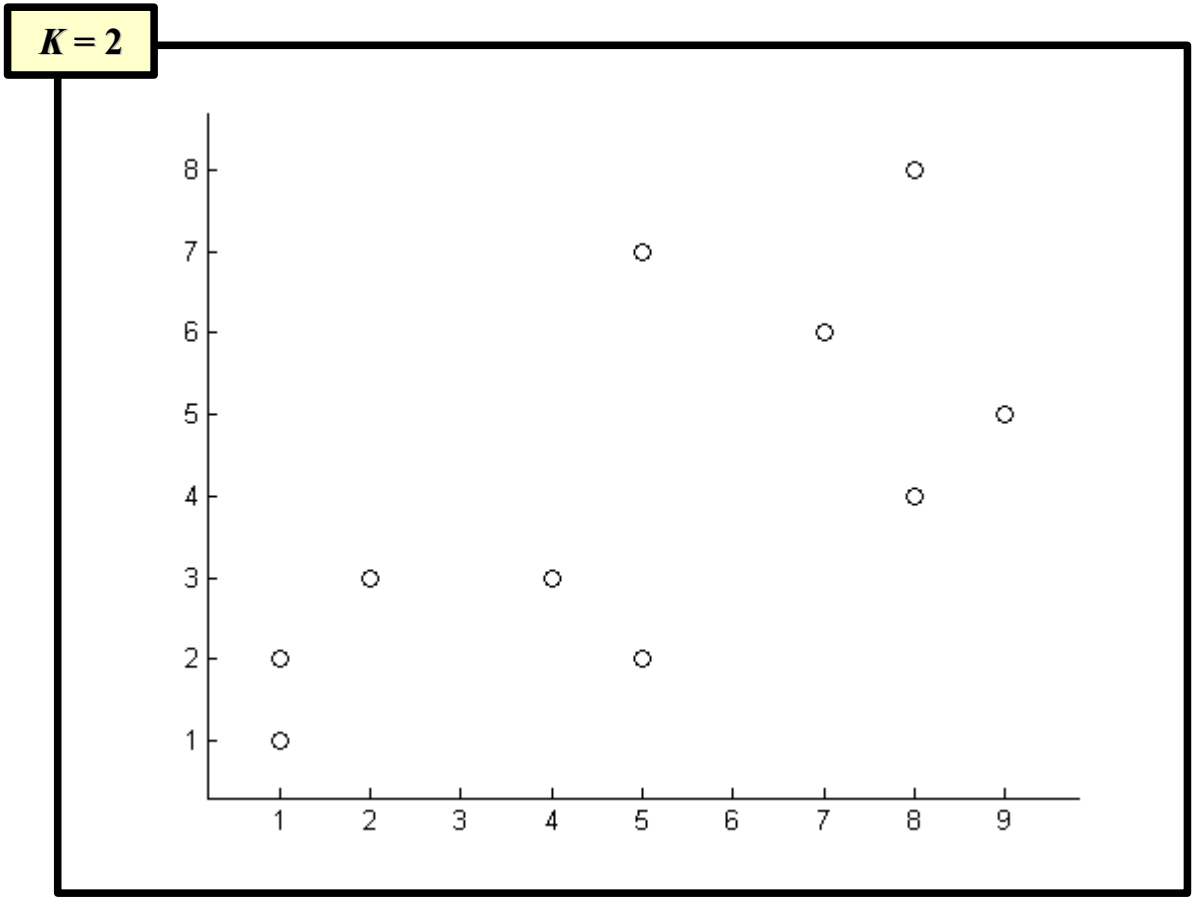
$$\Delta D = \frac{D_{i-1} - D_i}{D_{i-1}} < T$$

gdzie: D_i **– dyspersja podziału zbioru danych na klastry w** i**-tej iteracji;**
T **– zadany próg (jakaś mała wartość).**

Klasteryzacja: algorytm *K*-średnich

Zbiór danych:

$$X = \begin{bmatrix} 1.0 & 1.0 & 2.0 & 4.0 & 5.0 & 5.0 & 7.0 & 8.0 & 8.0 & 9.0 \\ 1.0 & 2.0 & 3.0 & 3.0 & 2.0 & 7.0 & 6.0 & 4.0 & 8.0 & 5.0 \end{bmatrix}$$

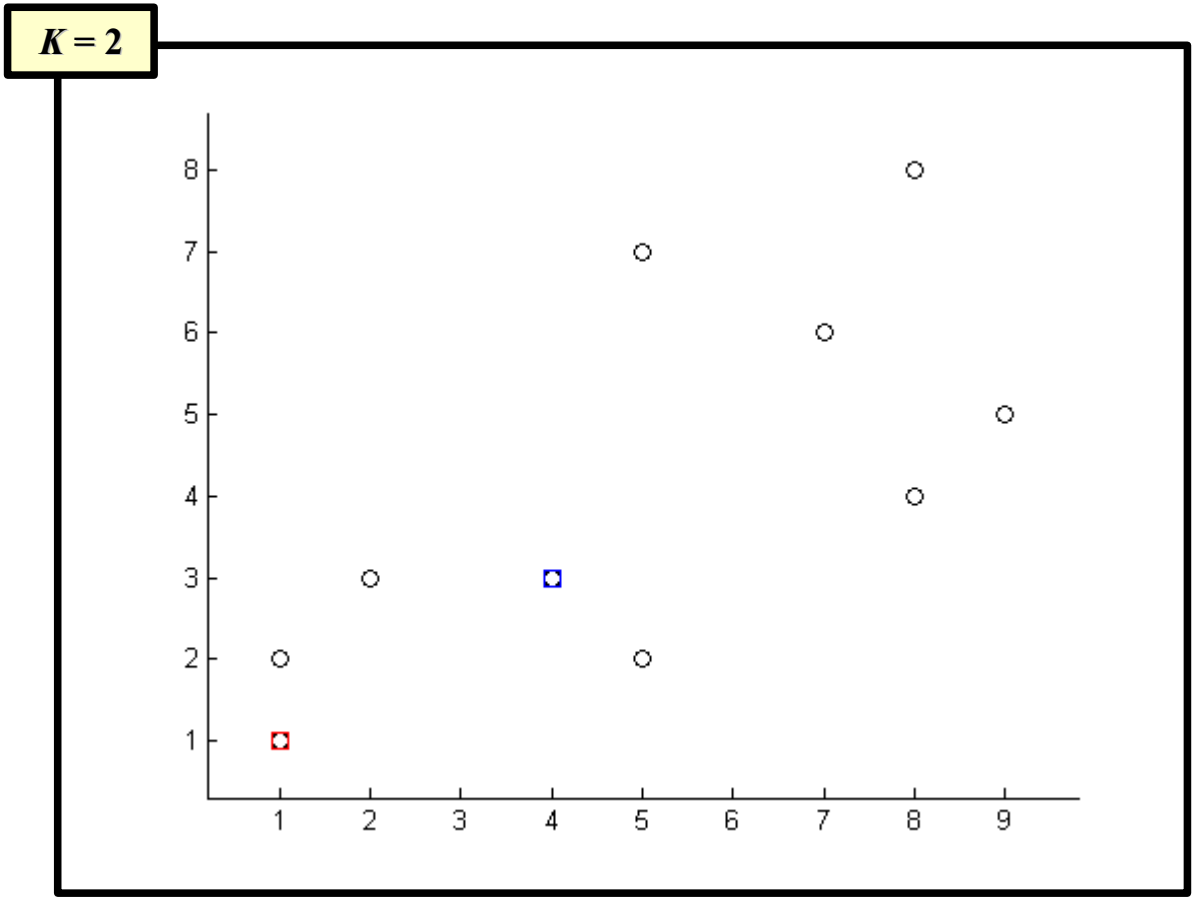


Klasteryzacja: algorytm *K*-średnich

Zbiór danych:

$$X = \begin{bmatrix} 1.0 & 1.0 & 2.0 & 4.0 & 5.0 & 5.0 & 7.0 & 8.0 & 8.0 & 9.0 \\ 1.0 & 2.0 & 3.0 & 3.0 & 2.0 & 7.0 & 6.0 & 4.0 & 8.0 & 5.0 \end{bmatrix}$$

Inicjalizacja: losowy wybór początkowych centroidów

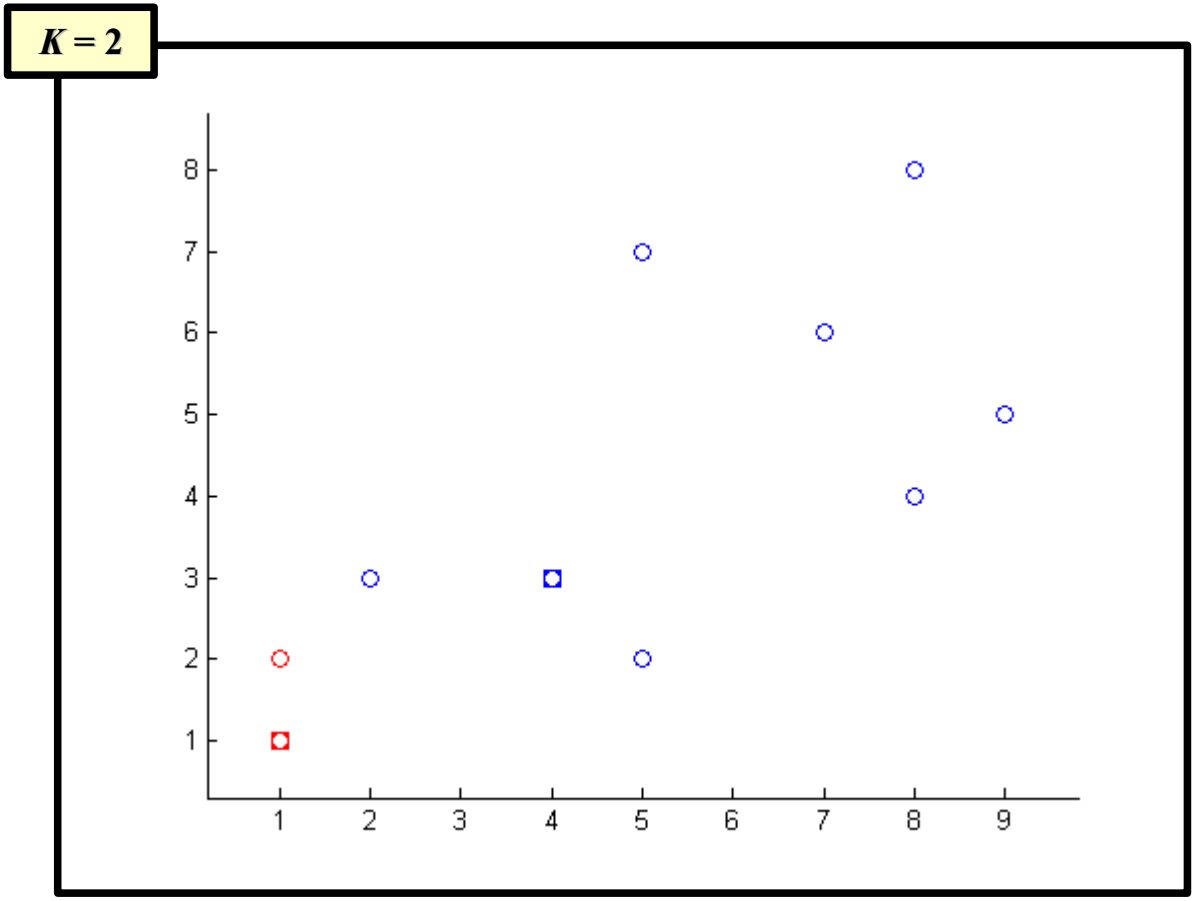


Klasteryzacja: algorytm *K*-średnich

Zbiór danych:

$$X = \begin{bmatrix} 1.0 & 1.0 & 2.0 & 4.0 & 5.0 & 5.0 & 7.0 & 8.0 & 8.0 & 9.0 \\ 1.0 & 2.0 & 3.0 & 3.0 & 2.0 & 7.0 & 6.0 & 4.0 & 8.0 & 5.0 \end{bmatrix}$$

Inicjalizacja: wyznaczenie klastrów wokół początkowych centroidów

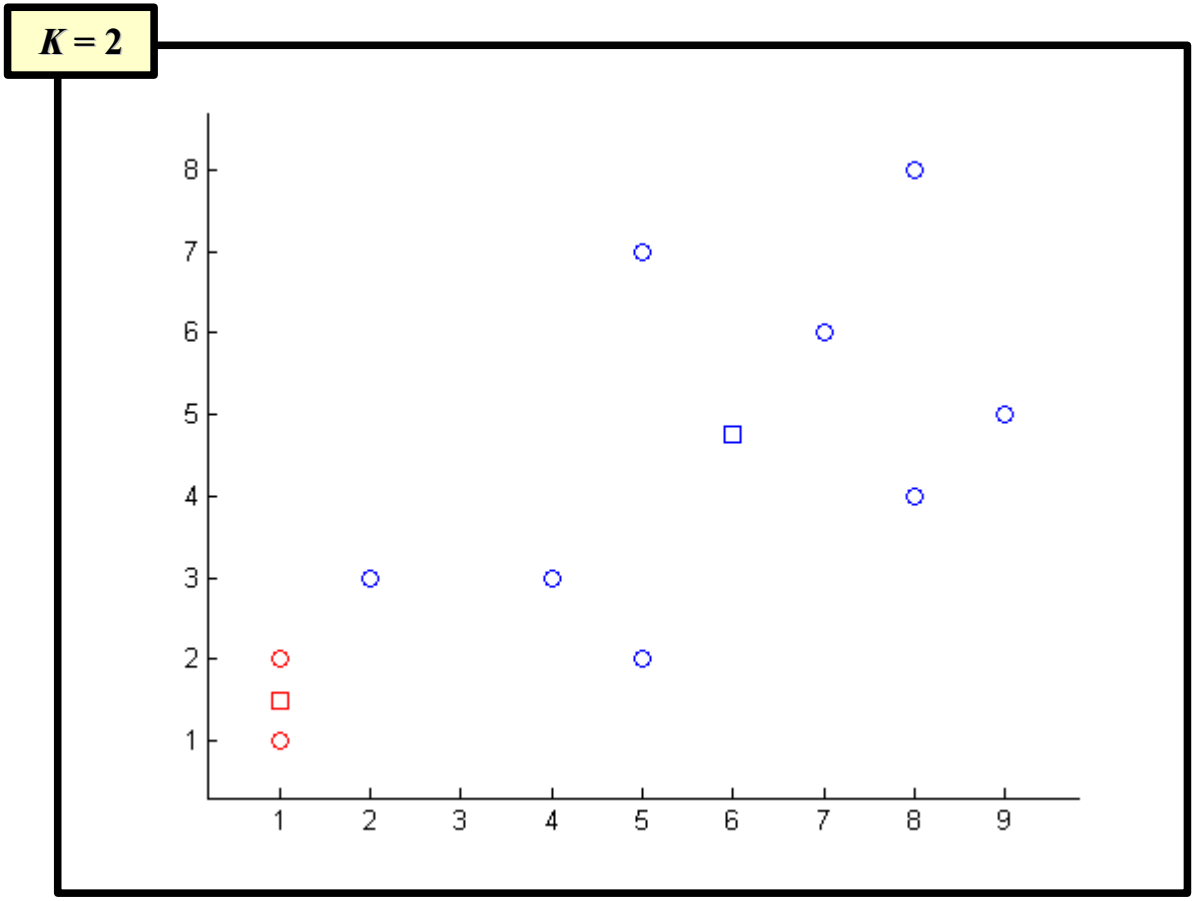


Klasteryzacja: algorytm *K*-średnich

Zbiór danych:

$$X = \begin{bmatrix} 1.0 & 1.0 & 2.0 & 4.0 & 5.0 & 5.0 & 7.0 & 8.0 & 8.0 & 9.0 \\ 1.0 & 2.0 & 3.0 & 3.0 & 2.0 & 7.0 & 6.0 & 4.0 & 8.0 & 5.0 \end{bmatrix}$$

Iteracja 1: **określenie położenia centroidów klastrów**

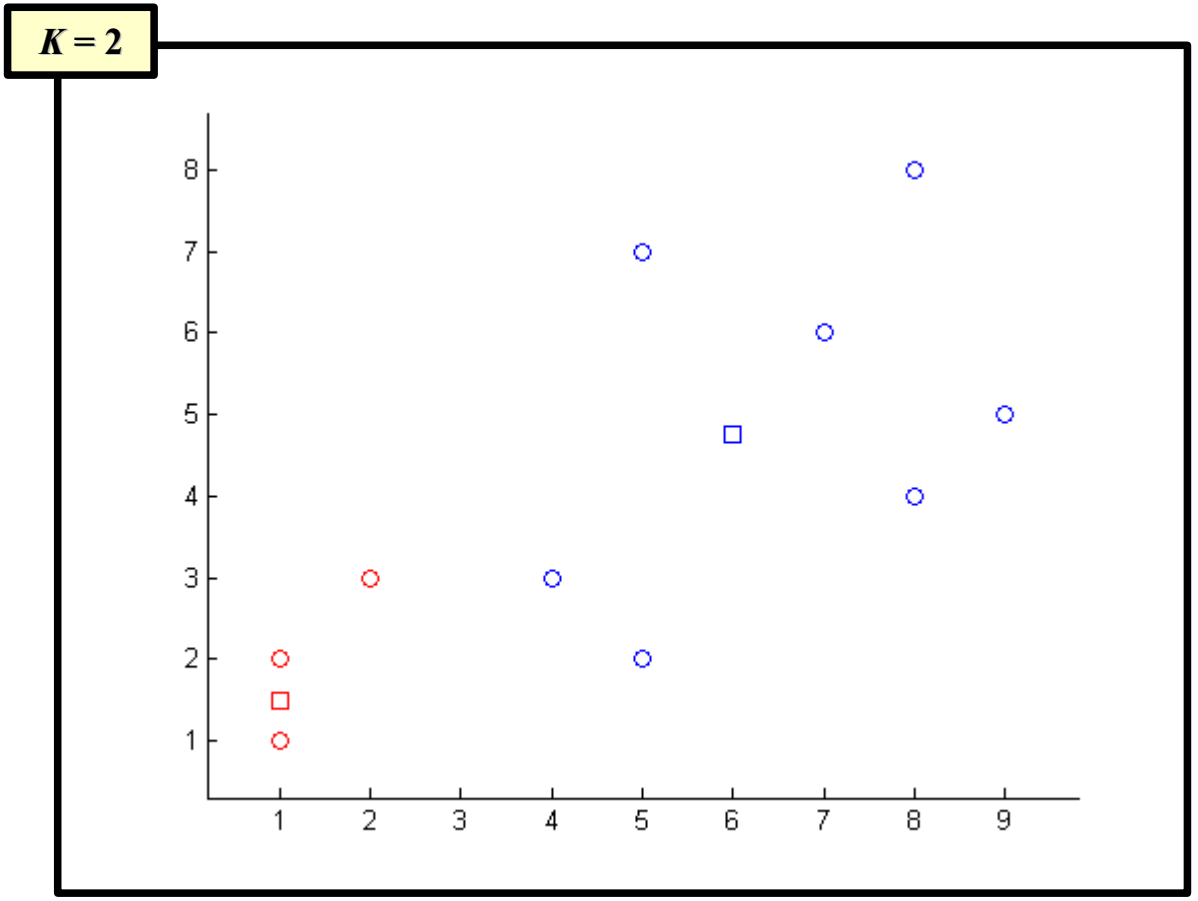


Klasteryzacja: algorytm *K*-średnich

Zbiór danych:

$$X = \begin{bmatrix} 1.0 & 1.0 & 2.0 & 4.0 & 5.0 & 5.0 & 7.0 & 8.0 & 8.0 & 9.0 \\ 1.0 & 2.0 & 3.0 & 3.0 & 2.0 & 7.0 & 6.0 & 4.0 & 8.0 & 5.0 \end{bmatrix}$$

Iteracja 1: **ponowne przypisanie wektorów do klastrów**

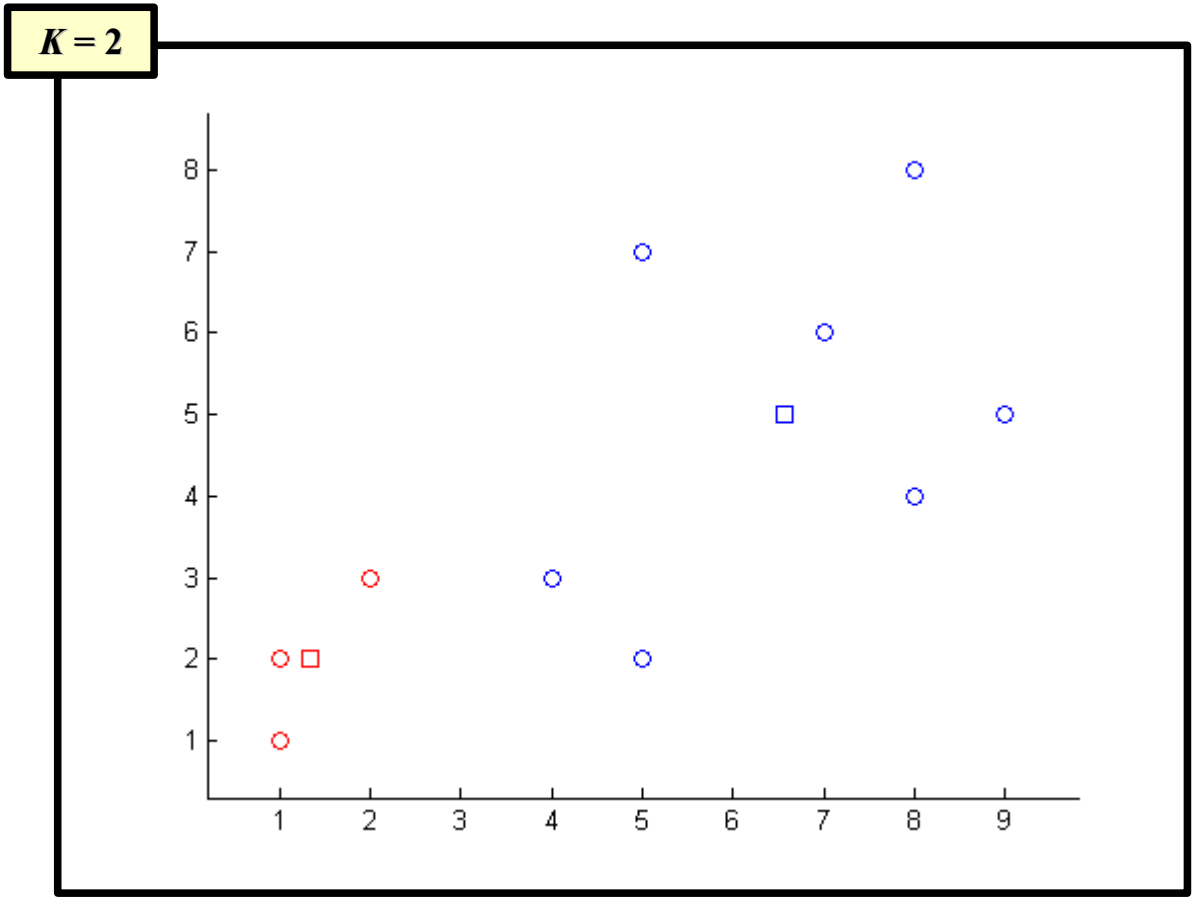


Klasteryzacja: algorytm *K*-średnich

Zbiór danych:

$$X = \begin{bmatrix} 1.0 & 1.0 & 2.0 & 4.0 & 5.0 & 5.0 & 7.0 & 8.0 & 8.0 & 9.0 \\ 1.0 & 2.0 & 3.0 & 3.0 & 2.0 & 7.0 & 6.0 & 4.0 & 8.0 & 5.0 \end{bmatrix}$$

Iteracja 2: **określenie położenia centroidów klastrów**

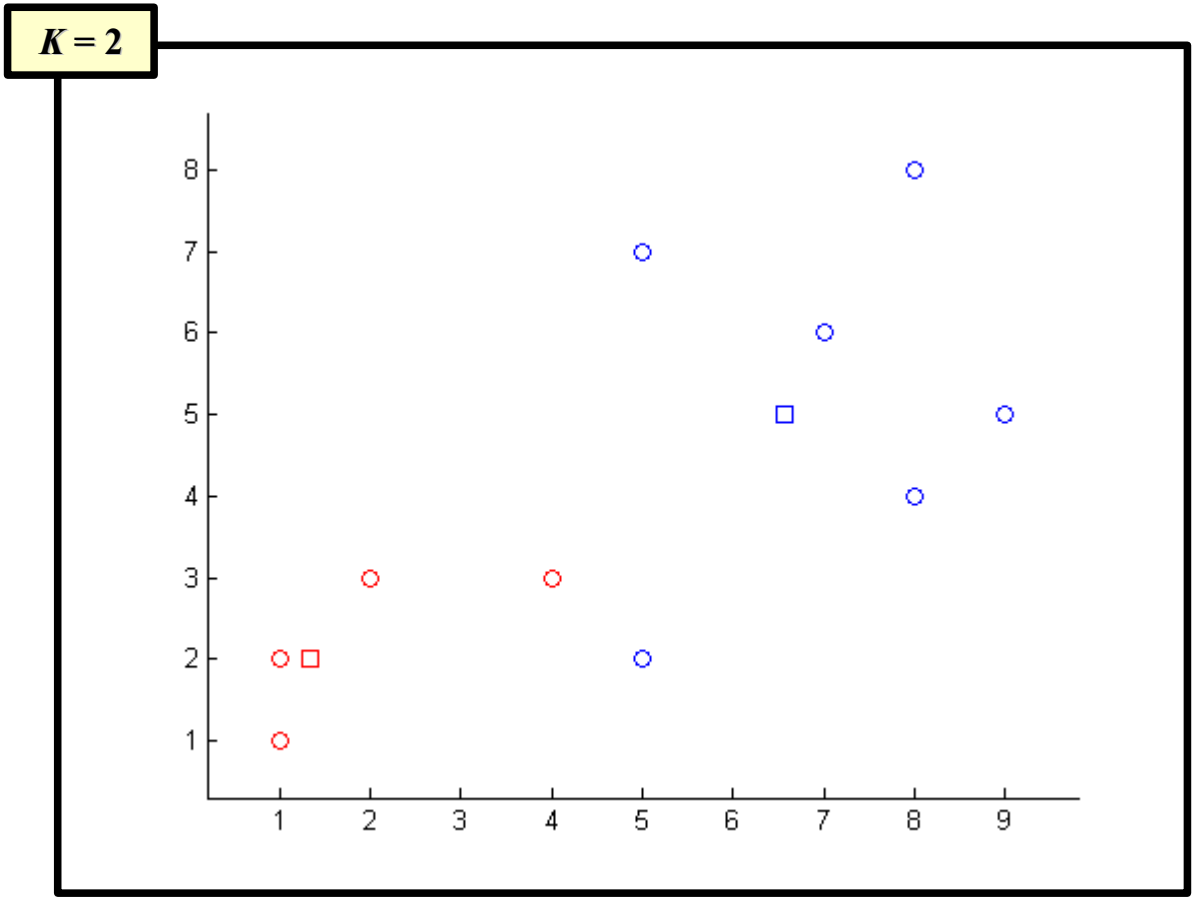


Klasteryzacja: algorytm *K*-średnich

Zbiór danych:

$$X = \begin{bmatrix} 1.0 & 1.0 & 2.0 & 4.0 & 5.0 & 5.0 & 7.0 & 8.0 & 8.0 & 9.0 \\ 1.0 & 2.0 & 3.0 & 3.0 & 2.0 & 7.0 & 6.0 & 4.0 & 8.0 & 5.0 \end{bmatrix}$$

Iteracja 2: **ponowne przypisanie wektorów do klastrów**

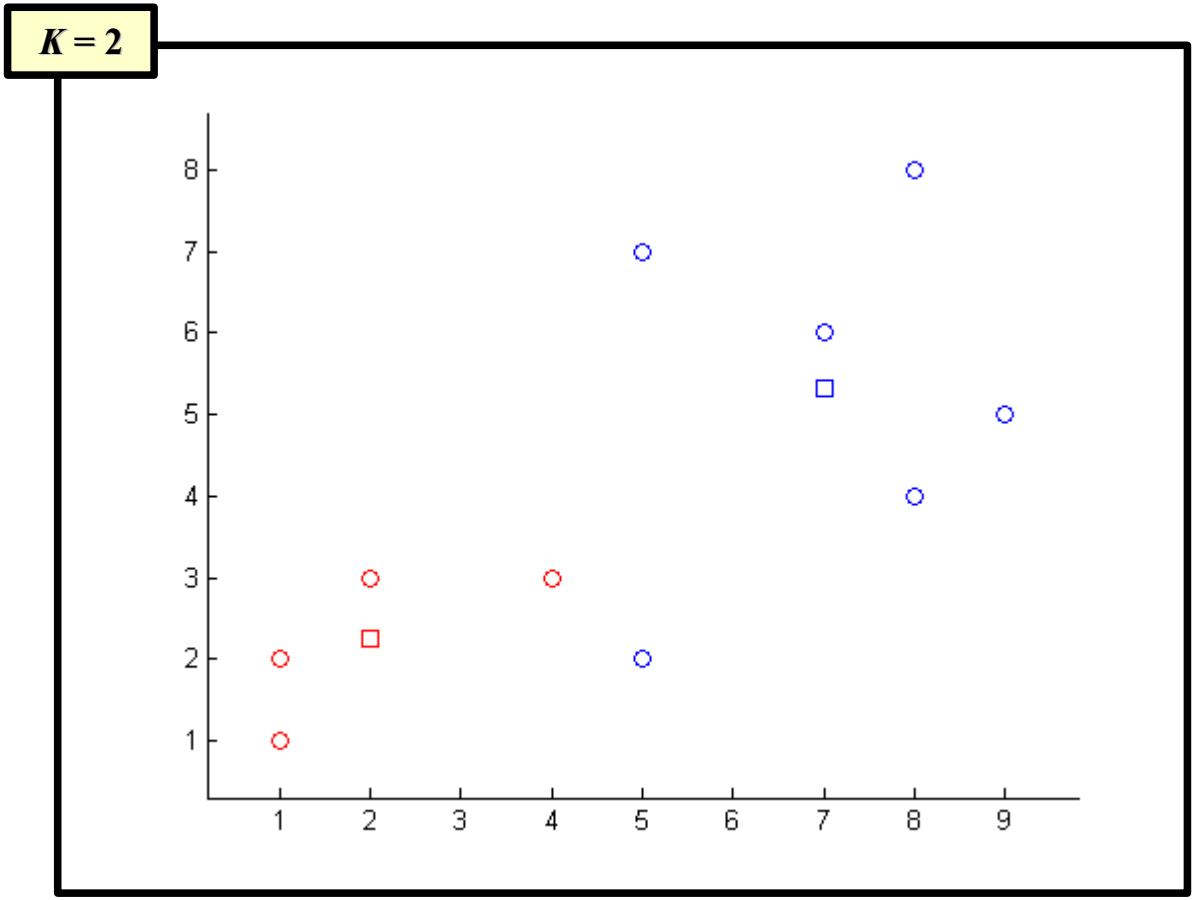


Klasteryzacja: algorytm *K*-średnich

Zbiór danych:

$$X = \begin{bmatrix} 1.0 & 1.0 & 2.0 & 4.0 & 5.0 & 5.0 & 7.0 & 8.0 & 8.0 & 9.0 \\ 1.0 & 2.0 & 3.0 & 3.0 & 2.0 & 7.0 & 6.0 & 4.0 & 8.0 & 5.0 \end{bmatrix}$$

Iteracja 3: **określenie położenia centroidów klastrów**

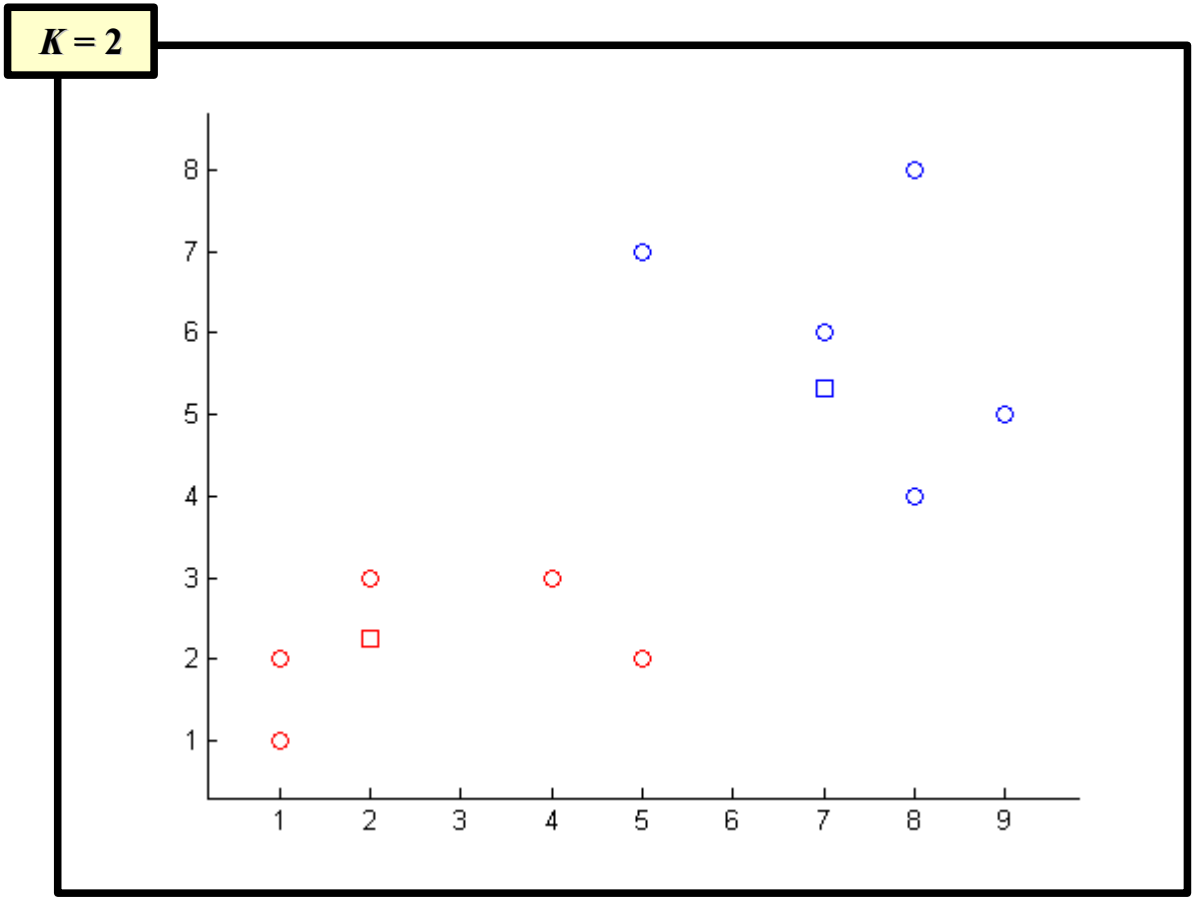


Klasteryzacja: algorytm *K*-średnich

Zbiór danych:

$$X = \begin{bmatrix} 1.0 & 1.0 & 2.0 & 4.0 & 5.0 & 5.0 & 7.0 & 8.0 & 8.0 & 9.0 \\ 1.0 & 2.0 & 3.0 & 3.0 & 2.0 & 7.0 & 6.0 & 4.0 & 8.0 & 5.0 \end{bmatrix}$$

Iteracja 3: **ponowne przypisanie wektorów do klastrów**

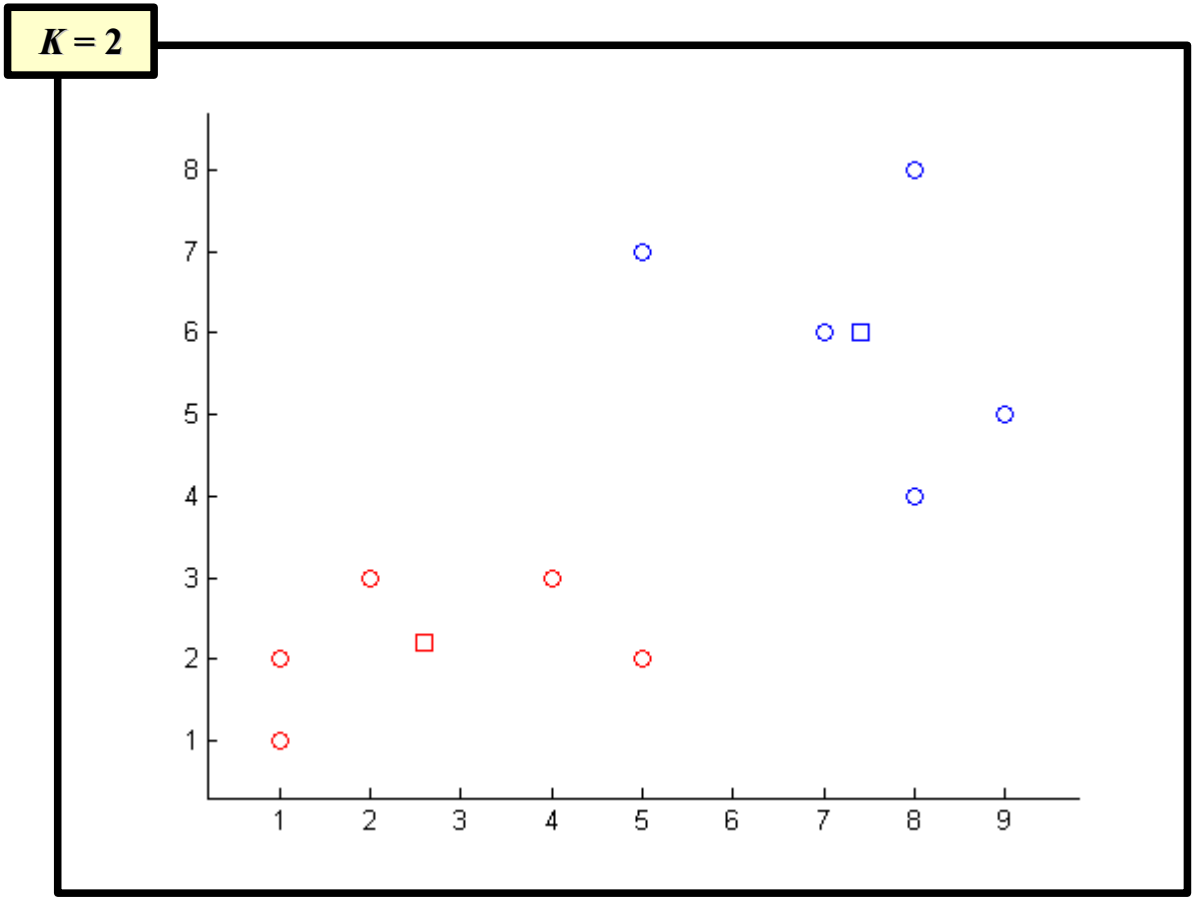


Klasteryzacja: algorytm *K*-średnich

Zbiór danych:

$$X = \begin{bmatrix} 1.0 & 1.0 & 2.0 & 4.0 & 5.0 & 5.0 & 7.0 & 8.0 & 8.0 & 9.0 \\ 1.0 & 2.0 & 3.0 & 3.0 & 2.0 & 7.0 & 6.0 & 4.0 & 8.0 & 5.0 \end{bmatrix}$$

Iteracja 4: **określenie położenia centroidów klastrów**

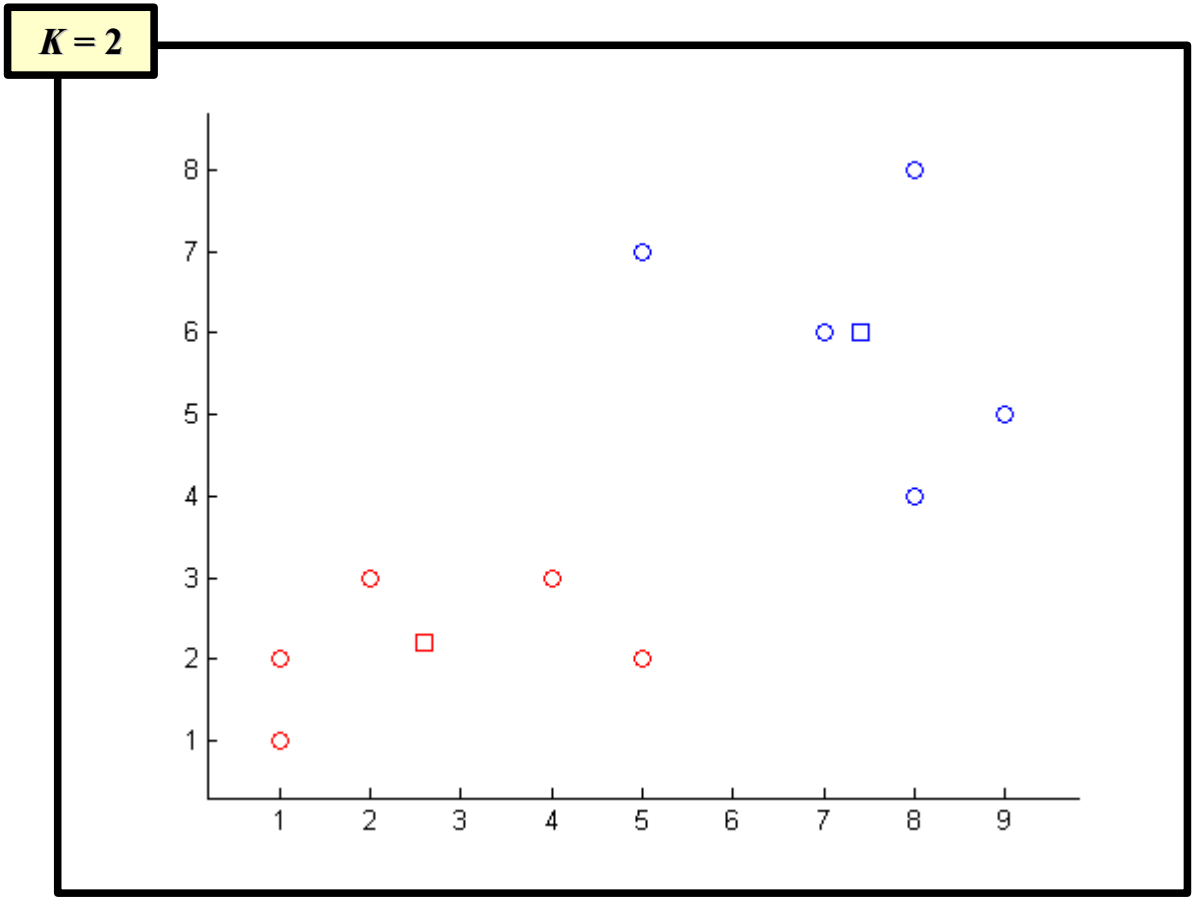


Klasteryzacja: algorytm *K*-średnich

Zbiór danych:

$$X = \begin{bmatrix} 1.0 & 1.0 & 2.0 & 4.0 & 5.0 & 5.0 & 7.0 & 8.0 & 8.0 & 9.0 \\ 1.0 & 2.0 & 3.0 & 3.0 & 2.0 & 7.0 & 6.0 & 4.0 & 8.0 & 5.0 \end{bmatrix}$$

Iteracja 4: **ponowne przypisanie wektorów do klastrów**

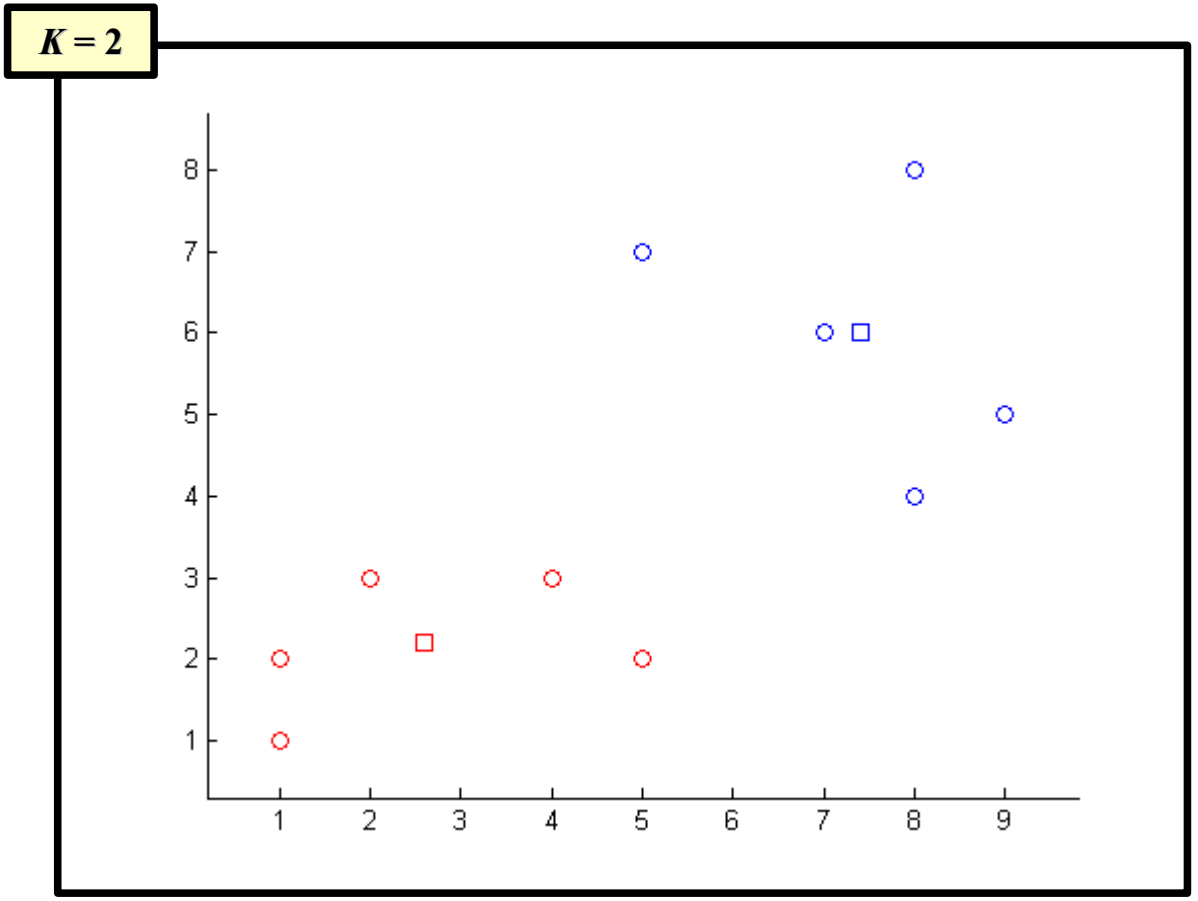


Klasteryzacja: algorytm *K*-średnich

Zbiór danych:

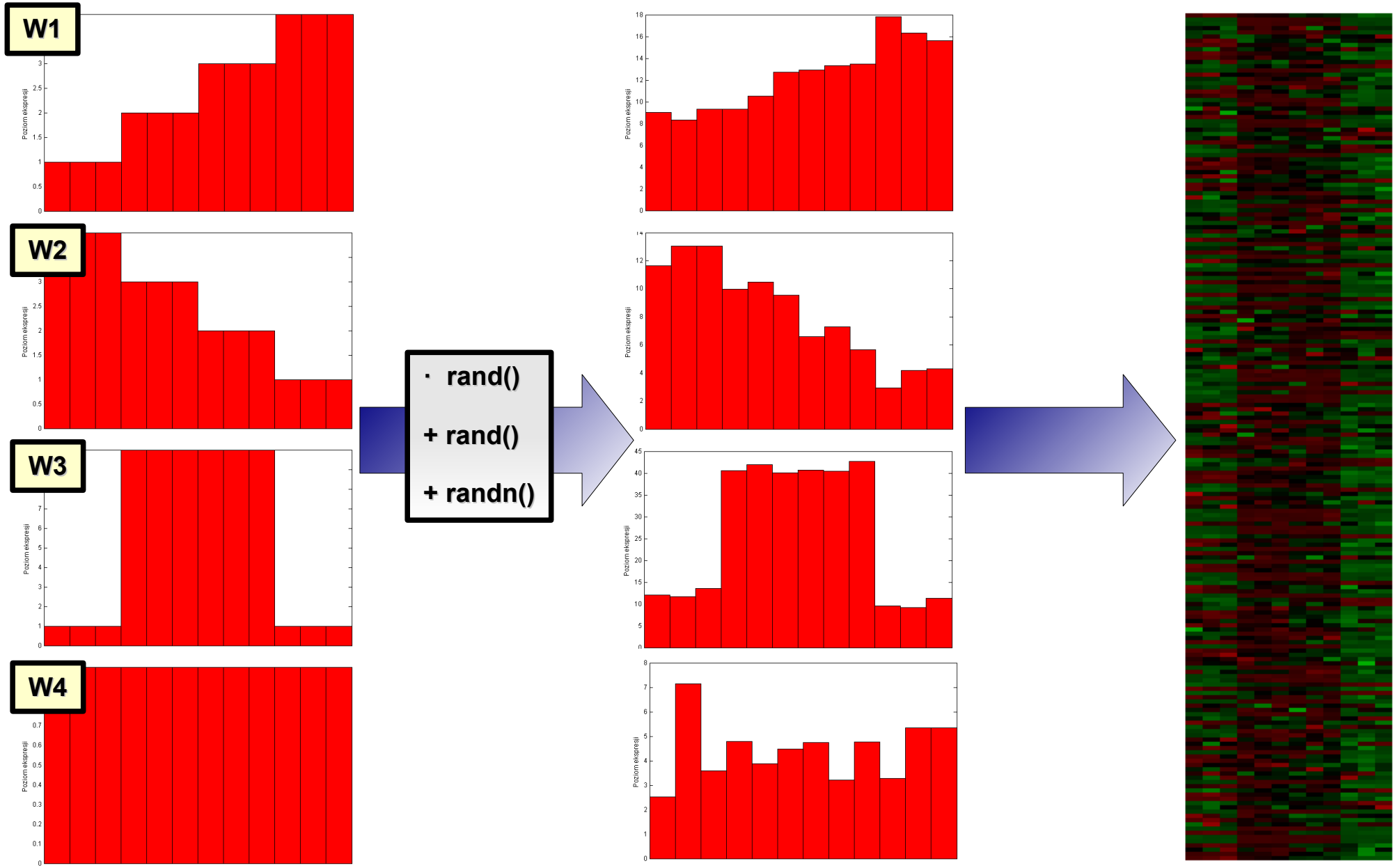
$$X = \begin{bmatrix} 1.0 & 1.0 & 2.0 & 4.0 & 5.0 & 5.0 & 7.0 & 8.0 & 8.0 & 9.0 \\ 1.0 & 2.0 & 3.0 & 3.0 & 2.0 & 7.0 & 6.0 & 4.0 & 8.0 & 5.0 \end{bmatrix}$$

Iteracja 5: **określenie położenia centroidów klastrów (brak zmian → koniec)**



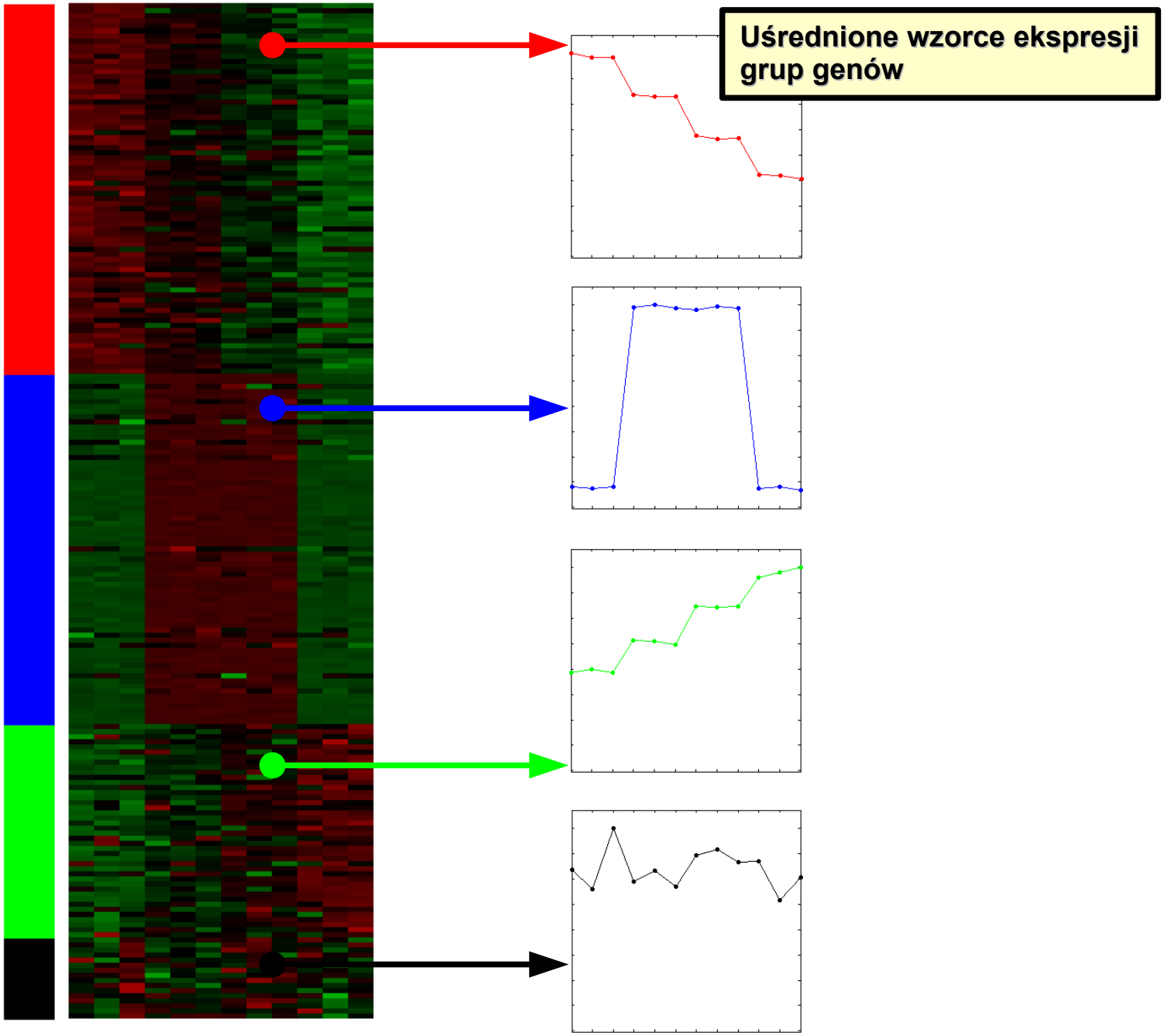
Klasteryzacja: algorytm *K*-średnich (przykład działania)

Przykład klasteryzacji *K*-średnich syntetycznego zbioru danych złożonego z 200 „genów” o wartościach ekspresji wygenerowanych przez losowe przeskalowanie, dodanie składowej stałej oraz szumu Gaussowskiego do czterech wzorców:



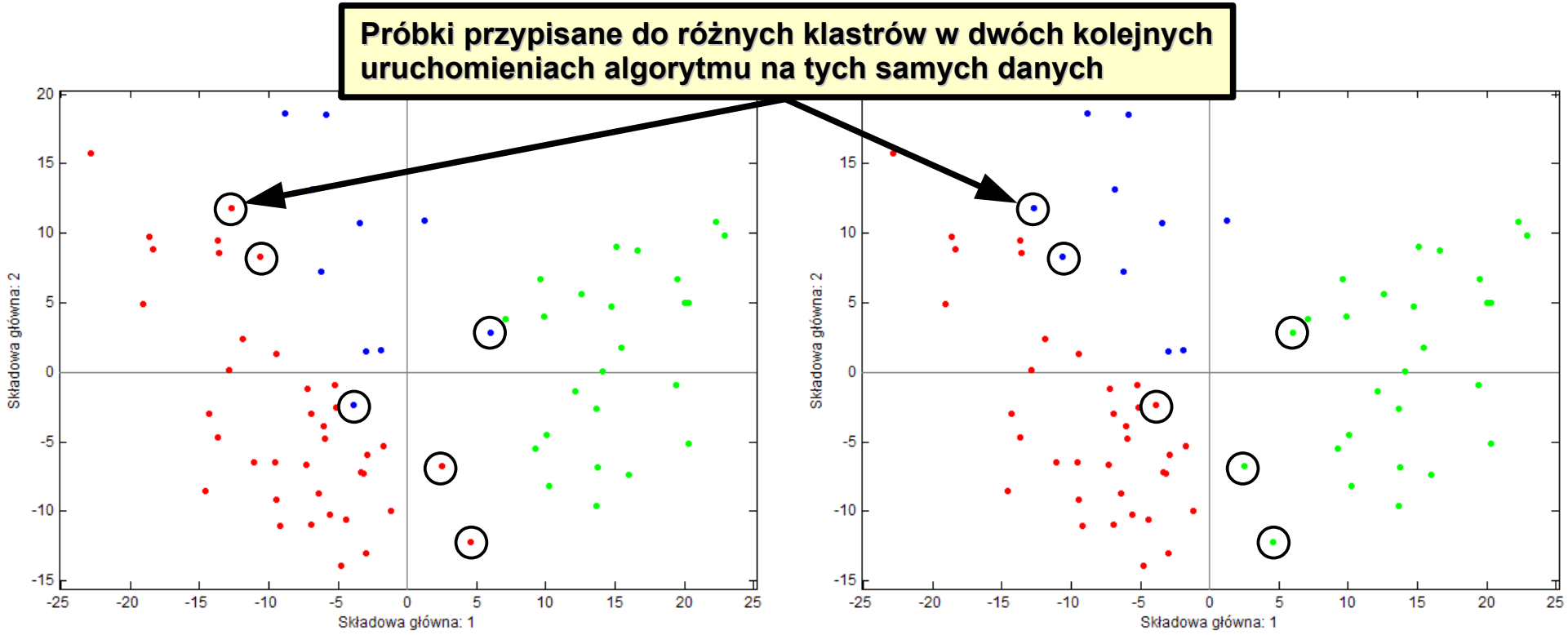
Klasteryzacja: algorytm *K*-średnich (przykład działania)

Wynik klasteryzacji dla $K = 4$ i korelacyjnej miary niepodobieństwa:



Klasteryzacja: algorytm *K*-średnich (wielostartowość)

Algorytm *K*-średnich gwarantuje zbieżność (dyspersja $D(\zeta_K)$ w danej iteracji nigdy nie jest większa niż w poprzedniej), co nie oznacza jednak, że otrzymany podział zbioru będzie optymalny w sensie globalnym. Ponadto istnieje zależność uzyskanego podziału od losowego wyboru początkowych położeń centroidów, tak więc kolejne uruchomienia algorytmu mogą prowadzić do różnych klastrów:



Prostym (ale bardzo skutecznym) rozwiązaniem tego problemu jest implementacja wielostartowa: algorytm uruchamiany jest wielokrotnie, za każdym razem z innymi (losowymi) położeniami początkowych centroidów, a następnie wybierany jest podział o najmniejszej wartości dyspersji.

Klasteryzacja: algorytm *K*-średnich (ustalanie początkowych centroidów)

Alternatywą dla implementacji wielostartowej jest użycie jednej z metod ustalania początkowych położeń centroidów. Przykładami tego typu metod mogą być:

- **wykonanie wstępnej klasteryzacji dla losowo wybranego podzbioru wektorów (przykładowo dla 10% próbek). Uzyskane w ten sposób centroidy stają się punktami startowymi dla właściwej klasteryzacji całego zbioru danych;**

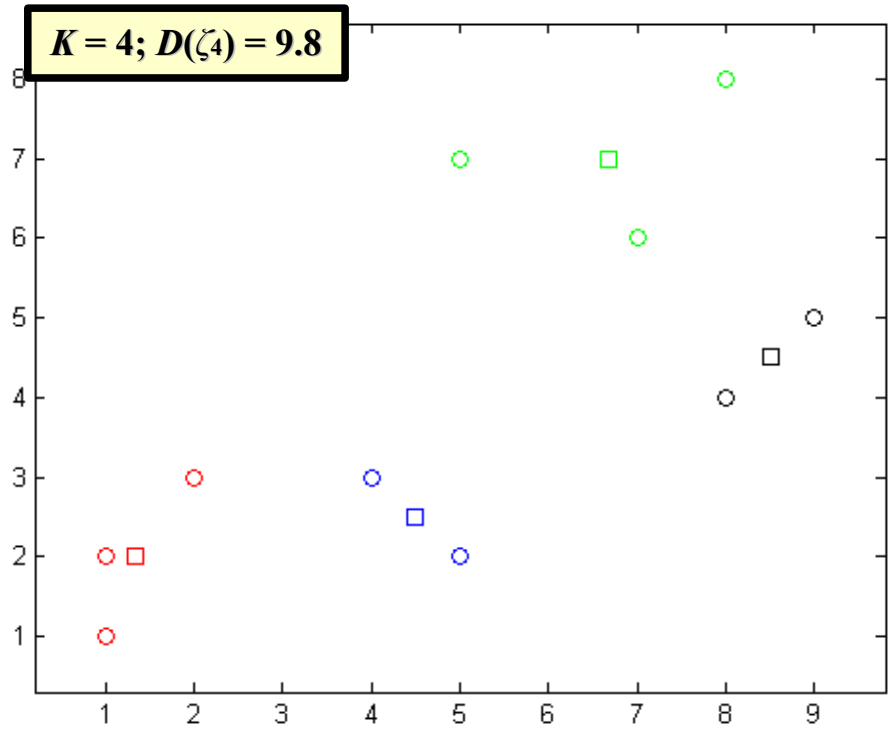
- **algorytm *K*-means++: pierwszy centroid wybierany jest losowo (z rozkładem jednostajnym) ze wszystkich wektorów zbioru danych, każdy kolejny wybierany jest spośród pozostałych wektorów również w sposób losowy, ale z prawdopodobieństwem proporcjonalnym do kwadratu odległości od najbliższego już wybranego centroidu;**

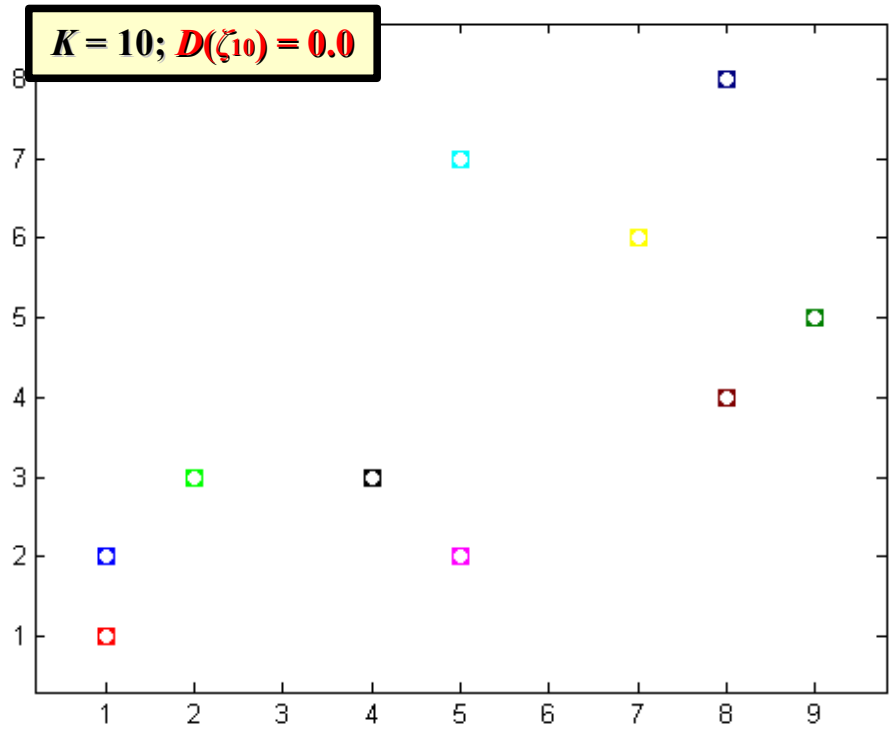
- **połączenie algorytmu *K*-średnich z metodą optymalizacji globalnej (np. algorytmem genetycznym lub symulowanym wyżarzaniem), co zmniejsza ryzyko przedwczesnej zbieżności procesu klasteryzacji po osiągnięciu lokalnego minimum funkcji celu. Takie połączenie jest „mądrzejszą" wersją wielostartowości, w której punkty startowe dla kolejnych uruchomień algorytmu *K*-średnich nie są wybierane losowo tylko są optymalizowane przez algorytm globalny.**

Klasteryzacja: algorytm *K*-średnich (estymacja liczby klastrów)

Poważnym ograniczeniem algorytmu *K*-średnich jest to, że zawsze dzieli on zbiór danych na zadaną liczbę klastrów, niezależnie od faktycznej struktury tego zbioru.

Optymalizowane kryterium nie chroni przed narzuceniem nadmiernej liczby klastrów, bo dyspersja $D(\zeta_K)$ jest tym mniejsza, im więcej jest klastrów (gdy każdy wektor jest osobnym klastrem wynosi ona 0).

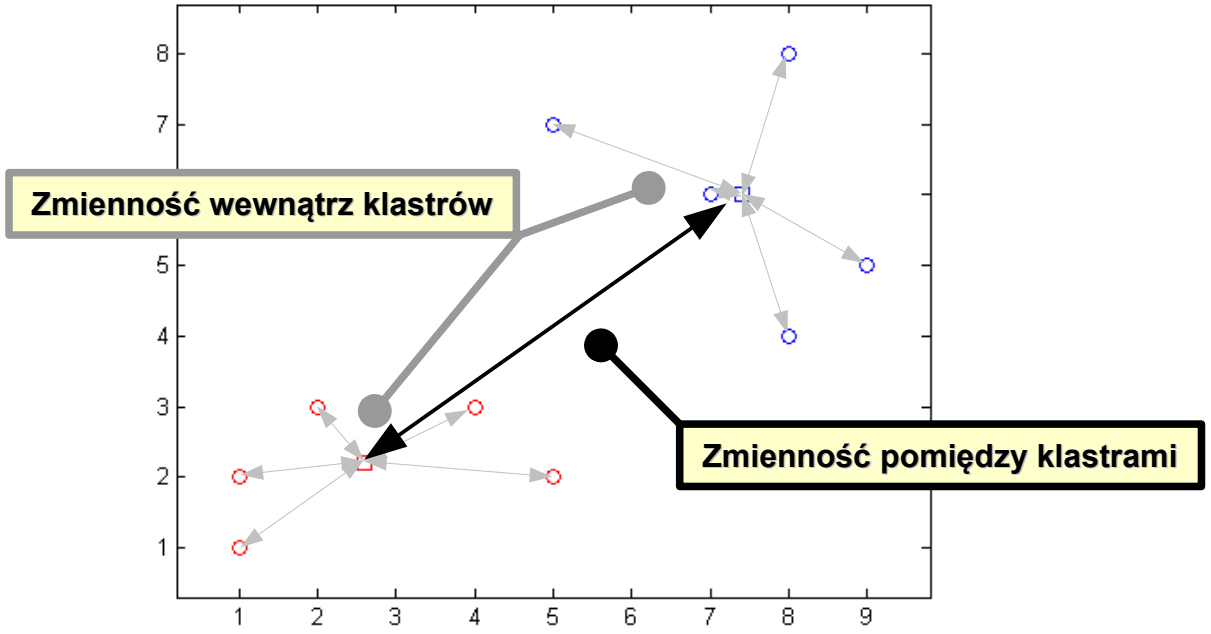




Klasteryzacja: algorytm *K*-średnich (estymacja liczby klastrów)

Konieczne jest ustalanie prawidłowej liczby klastrów „na oko” (np. na podstawie pewniej wiedzy *a priori* na temat zbioru danych) lub korzystając z dodatkowych algorytmów estymacji liczby klastrów.

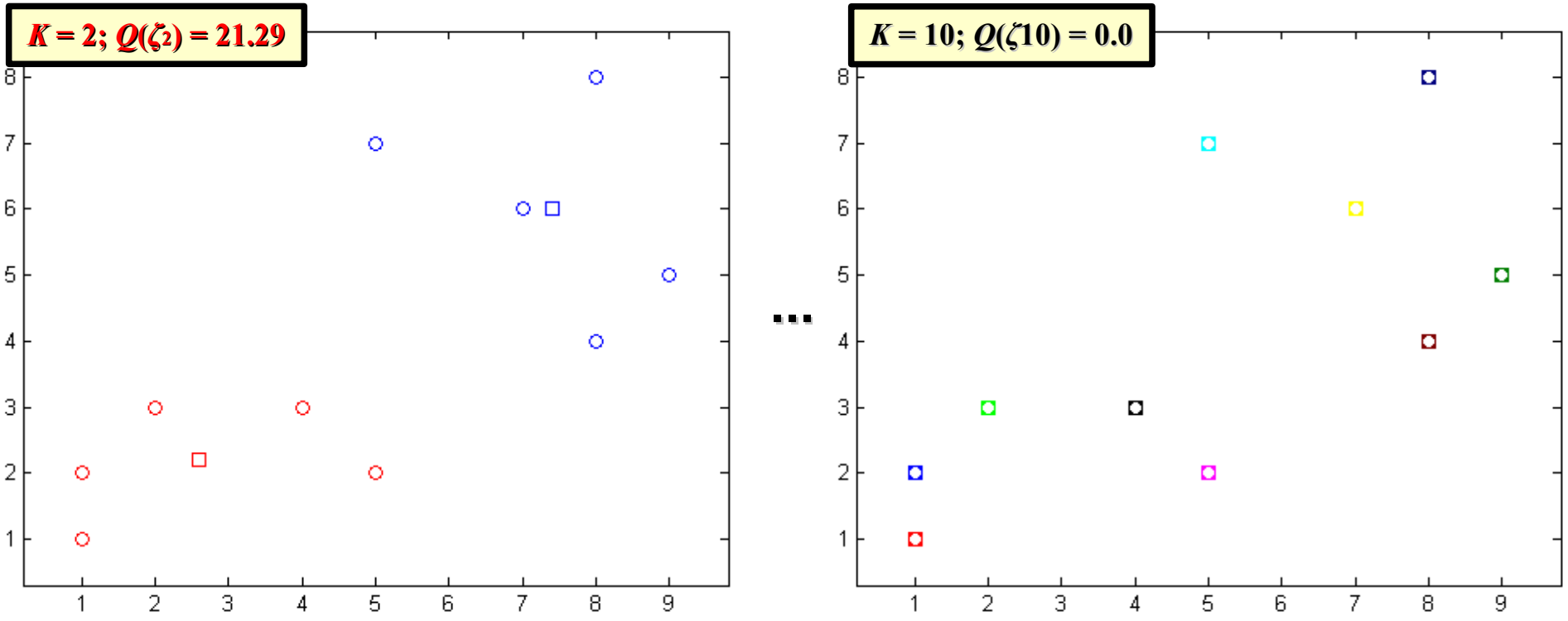
Podstawą działania metod estymacji liczby klastrów jest założenie, że „dobry” podział charakteryzuje się nie tylko małym rozproszeniem wektorów w klastrach (małą zmiennością wewnątrz klastrów, czyli dyspersją), ale także możliwie dużym oddaleniu od siebie środków klastrów (dużą zmiennością pomiędzy klastrami).



Klasteryzacja: algorytm *K*-średnich (estymacja liczby klastrów)

Metody estymacji liczby klastrów najczęściej działają we współpracy z algorytmem klasteryzacji. Ogólny schemat ich działania można przedstawić w trzech krokach:

- **wykonanie klasteryzacji dla różnej liczby klastrów ($K = 2, 3, \ldots$);**
- **wyznaczenie dla każdego K wartości liczbowego wskaźnika jakości podziału Q;**
- **wybranie liczby K, dla której wartość wskaźnika jest maksymalna.**



Poszczególne metody różnią się sposobem zdefiniowania liczbowego wskaźnika, będącego miarą stosunku rozrzutów pomiędzy klastrami i wewnątrz nich.

Klasteryzacja: algorytm *K*-średnich (estymacja liczby klastrów)

Przykładem liczbowego wskaźnika pozwalającego ocenić jakość podziału na klastry jest indeks Calińskiego-Harabasza, który dla zbioru N wektorów przyjmuje postać:

$$CH(\zeta_K) = \frac{\mathrm{tr}\left(\boldsymbol{B}(\zeta_K)\right)/(K-1)}{\mathrm{tr}\left(\boldsymbol{W}(\zeta_K)\right)/(N-K)}$$

gdzie $\mathrm{tr}(X)$ oznacza ślad macierzy X, czyli sumę elementów na głównej przekątnej.

$W(\zeta_K)$ jest symetryczną macierzą o wymiarze $P \times P$ (P to liczba cech), zwaną macierzą zmienności wewnątrz klastrów (WSS – *Within Sum of Squares*):

$$\boldsymbol{W}(\zeta_K) = \sum_{j=1}^{K} \sum_{x_i \in C_j} (\boldsymbol{x}_i - \boldsymbol{c}_j)(\boldsymbol{x}_i - \boldsymbol{c}_j)^T \qquad \boldsymbol{c}_j = \frac{1}{N_j} \sum_{i=1}^{N_j} \boldsymbol{x}_i$$

gdzie c_j to centroid j-tego klastra, a Nj jest liczebnością tego klastra.

$B(\zeta_K)$ to symetryczna macierz o wymiarze $P \times P$, określana macierzą zmienności pomiędzy klastrami (BSS – *Between Sum of Squares*):

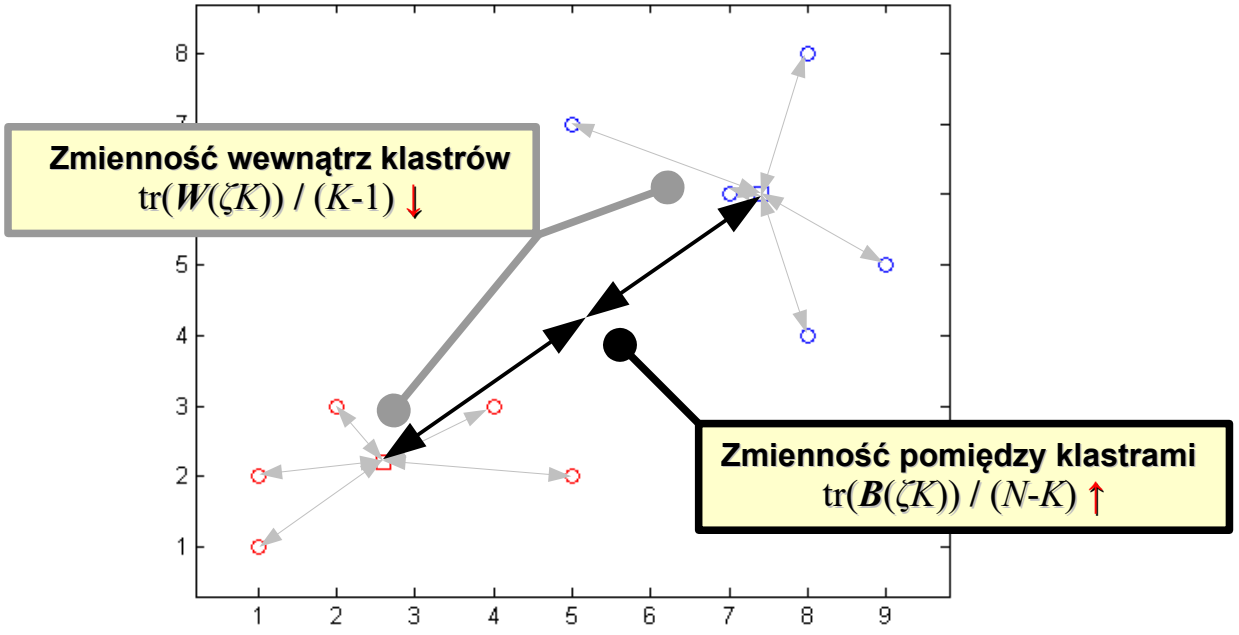
$$\boldsymbol{B}(\zeta_K) = \sum_{j=1}^{K} N_j (\boldsymbol{c}_j - \bar{\boldsymbol{x}})(\boldsymbol{c}_j - \bar{\boldsymbol{x}})^T \qquad \bar{\boldsymbol{x}} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_i$$

gdzie $\bar{x}$ to wektor wartości średnich całego zbioru danych (wszystkich N wektorów).

Klasteryzacja: algorytm *K*-średnich (estymacja liczby klastrów)

Wielkość $\mathrm{tr}(W(\zeta_K)) / (K\text{-}1)$ jest miarą rozproszenia wektorów wokół centroidów, do których zostały one przypisane, tym samym odpowiada dyspersji $D(\zeta_K)$, która jest minimalizowana podczas działania algorytmu *K*-średnich. Dla dobrego podziału na klastry powinna być możliwie mała.

Wielkość $\mathrm{tr}(B(\zeta_K)) / (N\text{-}K)$ jest miarą rozproszenia centroidów wokół ogólnej średniej całego zbioru danych. Dla dobrego podziału na klastry powinna być jak największa.

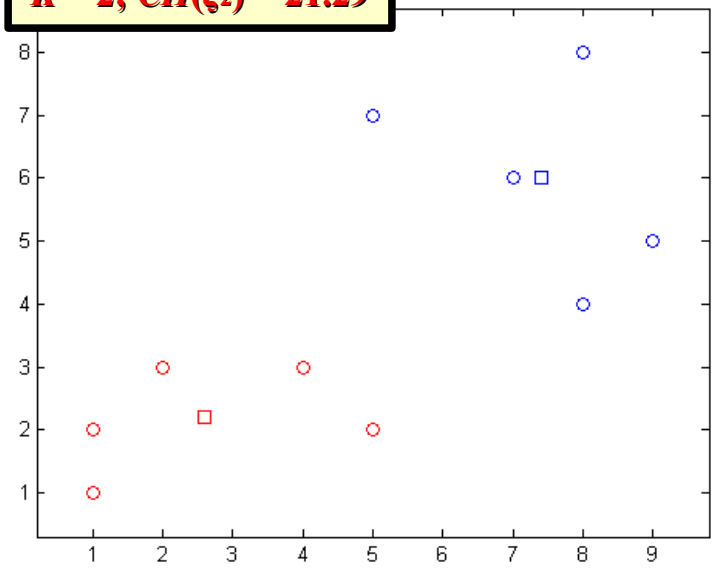


Oznacza to, że dobre podziały będą charakteryzować się dużą wartością indeksu CH.

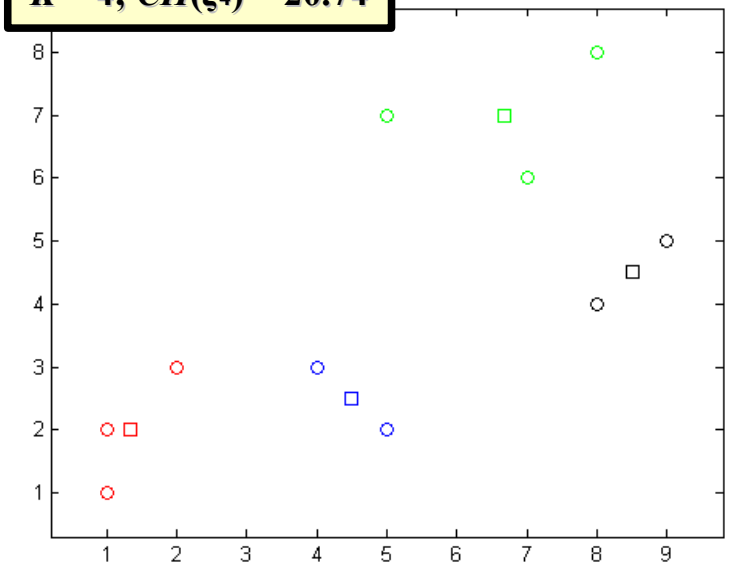
Klasteryzacja: algorytm *K*-średnich (estymacja liczby klastrów)

Wybór optymalnej liczby klastrów polega na wykonaniu klasteryzacji dla pewnego zakresu wartości K i przyjęciu tej, dla której indeks CH jest największy.

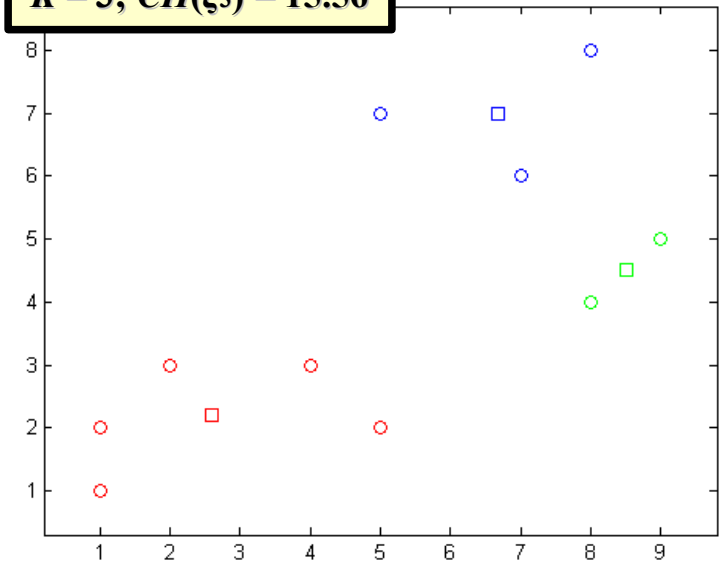
$K = 2$; $CH(\zeta_2) = 21.29$



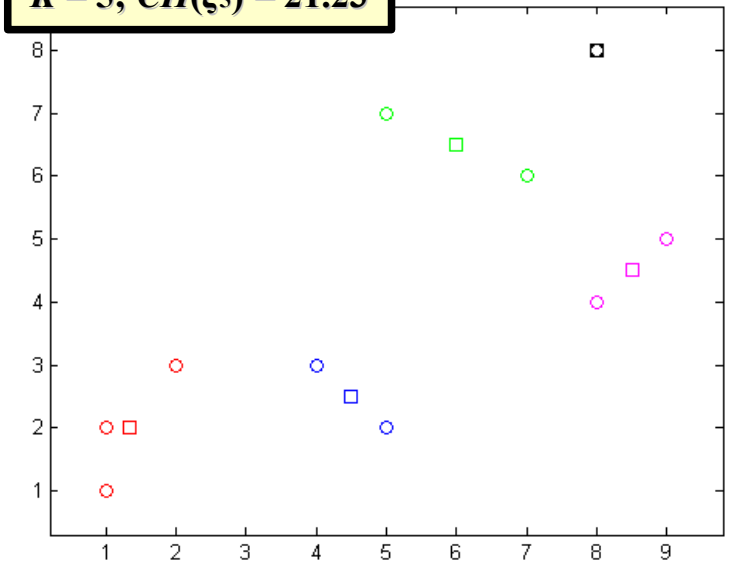
$K = 4$; $CH(\zeta_4) = 20.74$



$K = 3$; $CH(\zeta_3) = 15.56$



$K = 5$; $CH(\zeta_5) = 21.23$



Klasteryzacja: algorytm *K*-średnich (estymacja liczby klastrów)

Innym przykładem wskaźnika używanego do estymacji liczby klastrów jest miara silhouette, która dla każdego wektora w zbiorze danych definiowana jest jako:

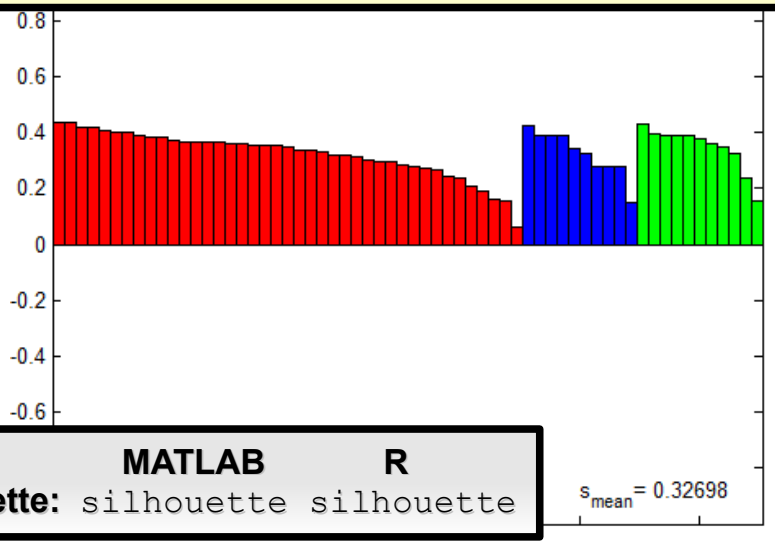
$$sil_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

gdzie: a_i **– średnia odległość między** i**-tym wektorem a pozostałymi wektorami z tego samego klastra (jest miarą rozproszenia wewnątrz klastra);**

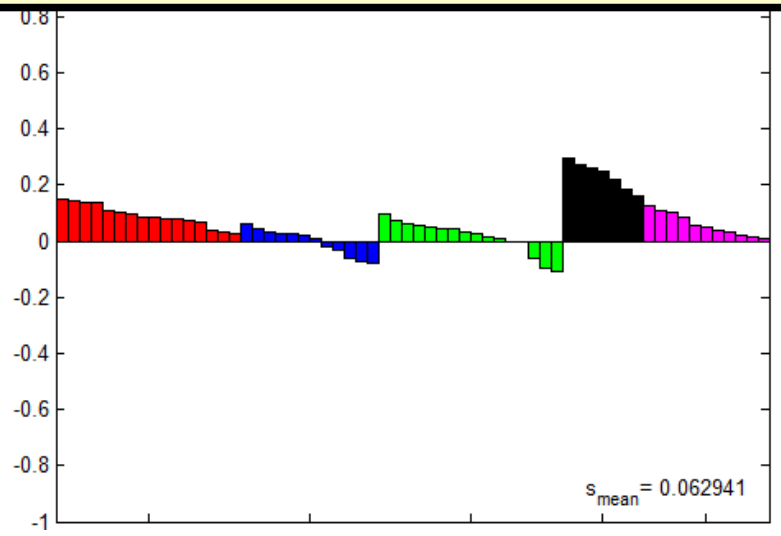
b_i **– najmniejsza średnia odległość pomiędzy** i**-tym wektorem a wektorami z pozostałych klastrów (jest miarą rozproszenia pomiędzy klastrami).**

Miara sil_i **przyjmuje wartości z zakresu** $[-1;1]$**, przy czym im większa jest jej wartość, tym** i**-ty wektor mniej różni się do innych wektorów z danego klastra, i jednocześnie tym bardziej różni się od wektorów z pozostałych klastrów (czyli wysoka wartość** sil_i **świadczy o prawidłowym przypisaniu** i**-tego wektora do klastra).**

Wykres wartości *sil* dla „dobrego" podziału na klastry



Wykres wartości *sil* dla „słabego" podziału na klastry



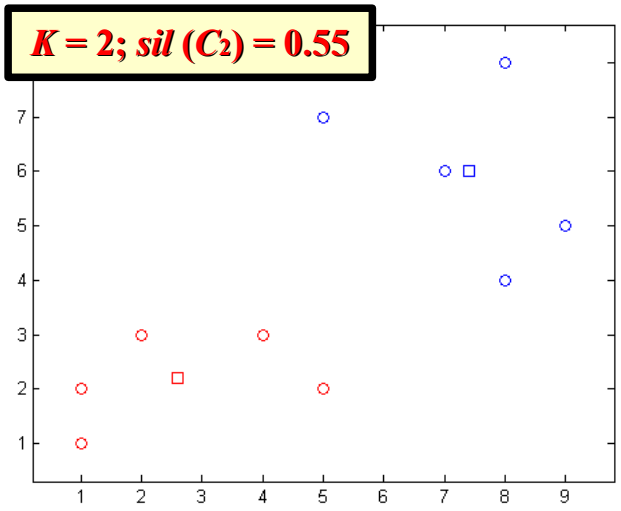
	MATLAB	R
Silhouette:	`silhouette`	`silhouette`

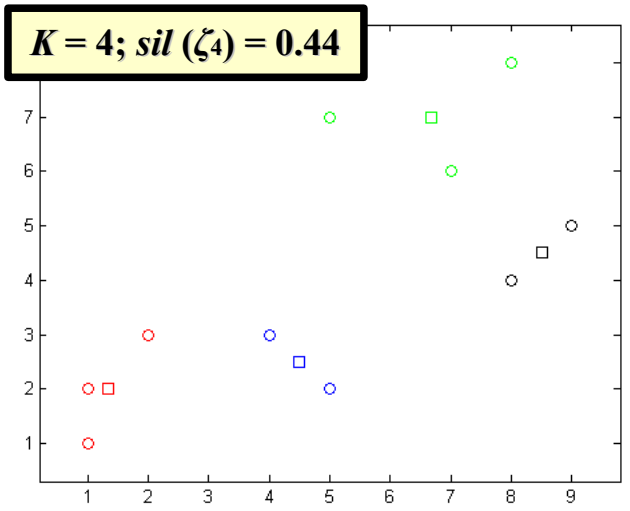
Klasteryzacja: algorytm *K*-średnich (estymacja liczby klastrów)

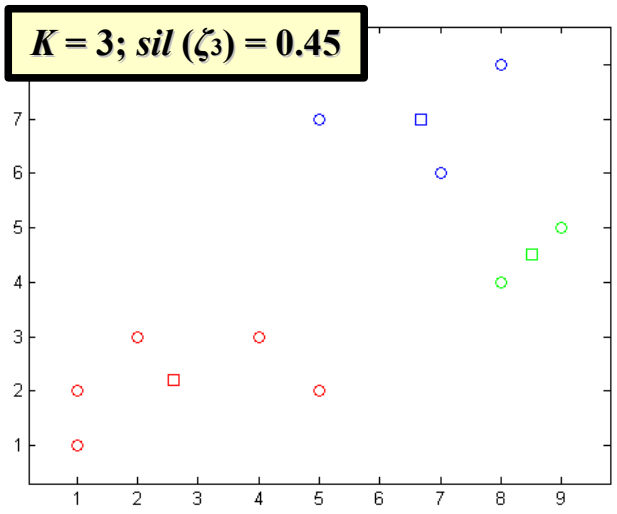
Jako wskaźnik jakości podziału zbioru danych na klastry można przyjąć uśrednioną po wszystkich N wektorach wartość miary sil_i:

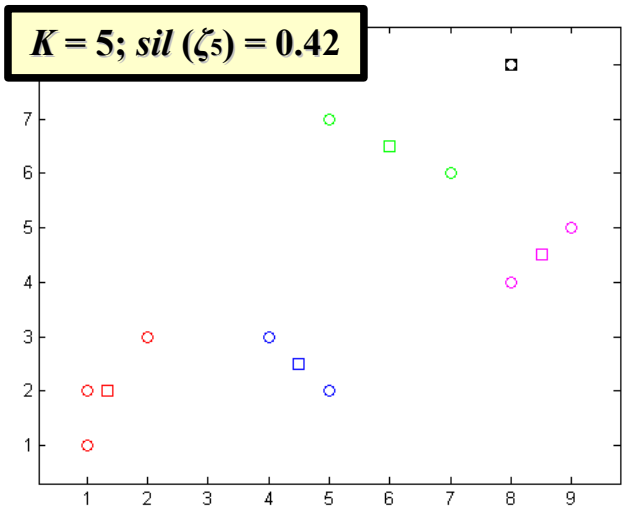
$$\overline{sil}(\zeta_K) = \frac{1}{N}\sum_{i=1}^{N} sil_i$$

Optymalnym K jest to, dla którego wartość uśrednionej miary sil jest maksymalna.









Uczenie maszynowe w bioinformatyce

Wykład 6: klasteryzacja (metody *density-based*)

Tymon Rubel

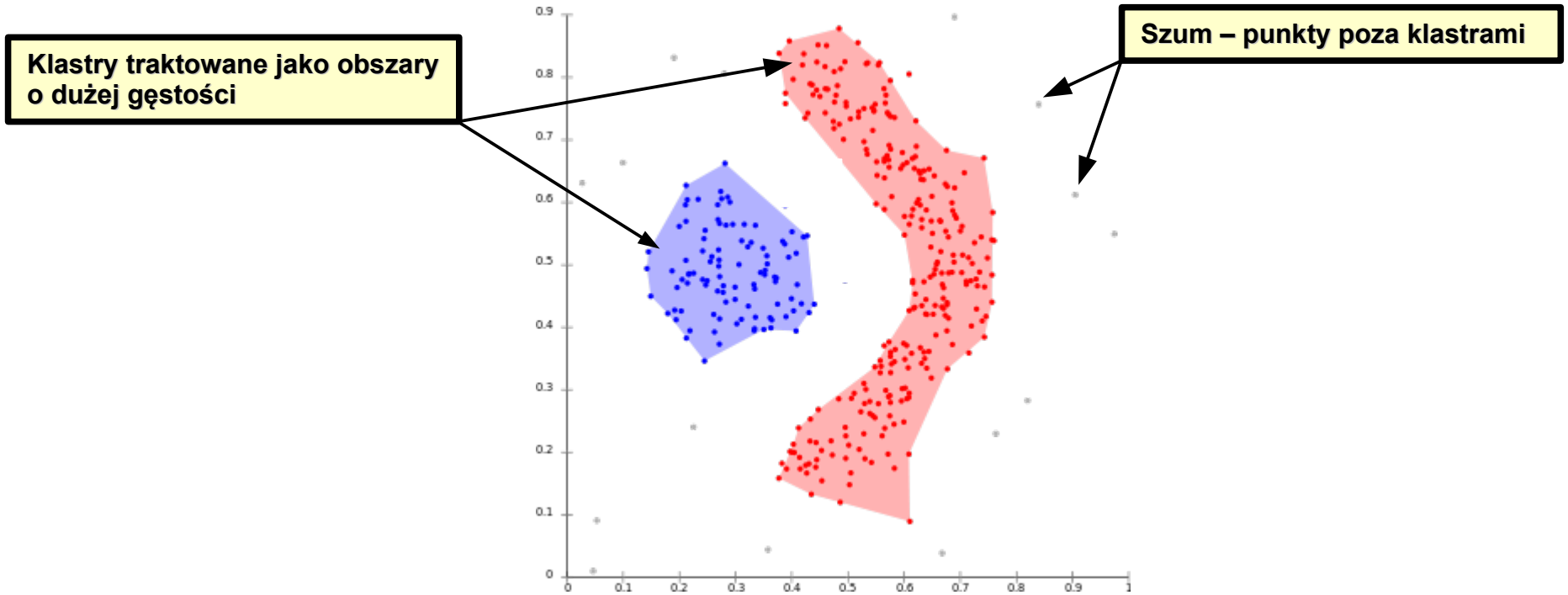
Zakład Elektroniki Jądrowej i Medycznej
Instytut Radioelektroniki i Technik Multimedialnych PW

Klasteryzacja: density-based clustering

Algorytmy „gęstościowe” (*density-based*) definiują klastry jako obszary przestrzeni o dużej gęstości, w ramach których wektory cech wzajemnie znajdują się na swoich listach najbardziej podobnych sąsiadów.

Podstawowymi zaletami takiego podejścia są:

- **brak konieczności określania z góry liczby klastrów;**
- **niezależność od kształtów klastrów;**
- **odporność wobec punktów odstających, które traktowane są jako szum.**

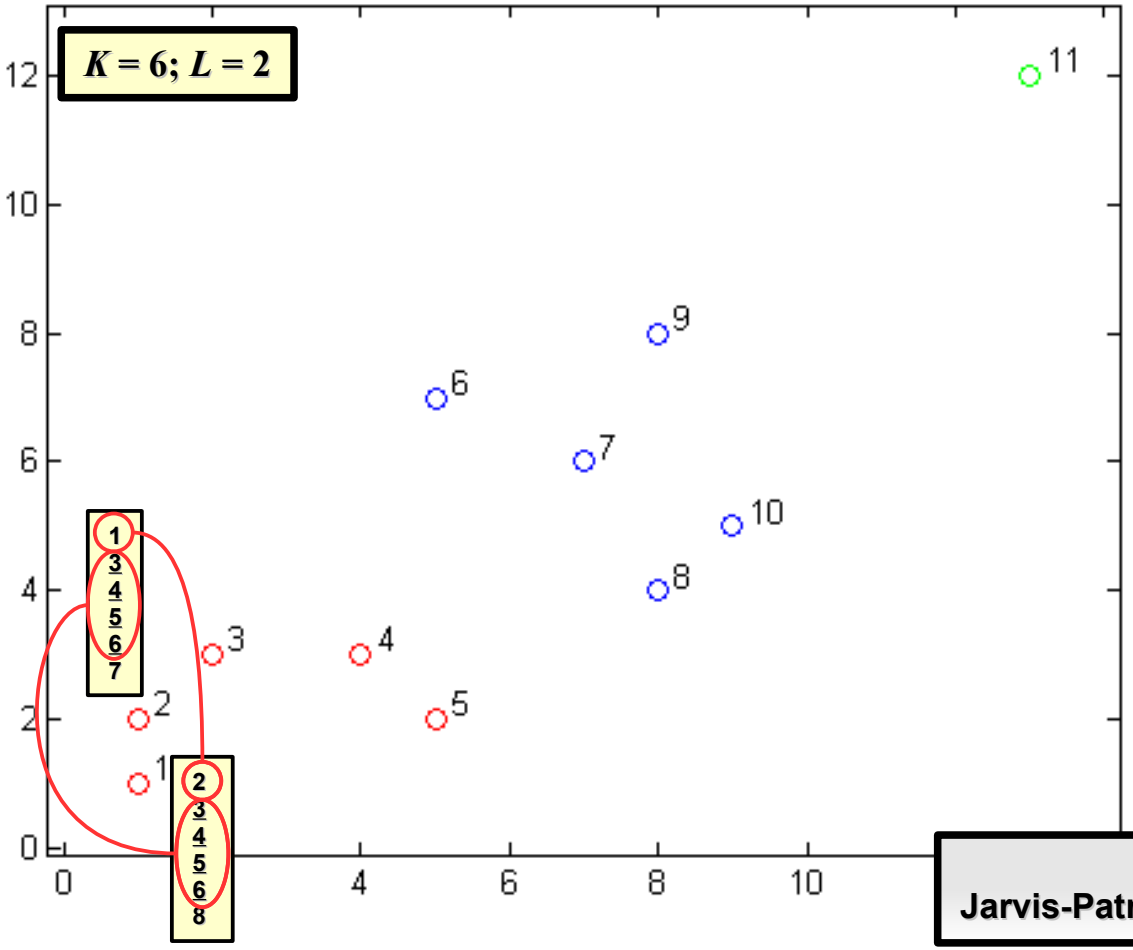


Klasteryzacja: density-based clustering (algorytm Jarvisa-Patricka)

W algorytmie Jarvisa-Patricka dwa wektory są przypisywane do tego samego klastra jeżeli spełnione są dwa warunki:

- **znajdują się one nawzajem na swoich listach K najbliższych sąsiadów;**
- **mają co najmniej L wspólnych sąsiadów w rozpatrywanym otoczeniu o wielkości K.**

Pierwszy z parametrów decyduje przede wszystkim o wielkości uzyskanych klastrów, podczas gdy drugi odpowiada za ich „gęstość”.

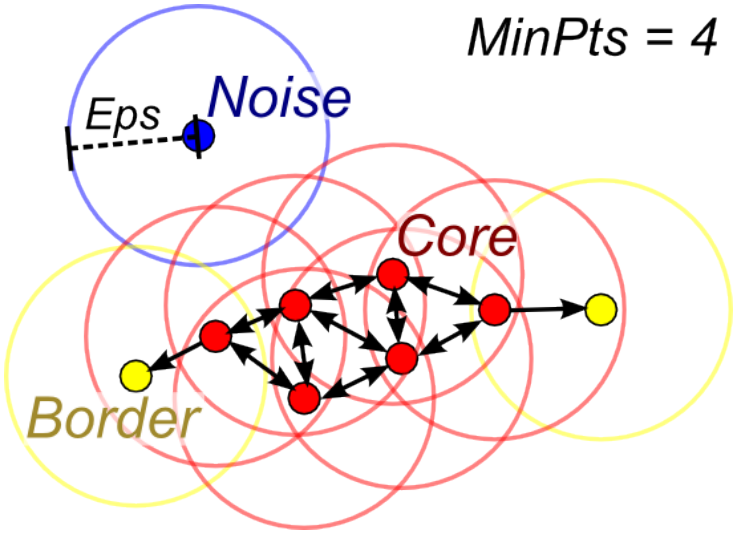


	MATLAB	R
Jarvis-Patrick:	-	`jpclust`

Klasteryzacja: density-based clustering (algorytm DBSCAN)

Algorytm DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) opiera się na koncepcji łączności gęstościowej (*density-connectedness*) i tworzy klastry zgodnie z następującymi założeniami:

- **punkt P, którego otoczenie o promieniu ε zawiera co najmniej $minPts$ punktów (wliczając w to P) jest traktowany jako ziarno (*core point*) i każdy jego sąsiad jest włączany do klastra;**
- **jeżeli któryś z sąsiadów stanowi jednocześnie *core point*, to proces rozszerzania klastra jest kontynuowany również w jego otoczeniu o promieniu ε;**
- **pozostali sąsiedzi stanowią tzw. punkty brzegowe (*border point*);**
- **punkty nie znajdujące się w otoczeniu żadnego *core point* stanowią szum (*noise*).**

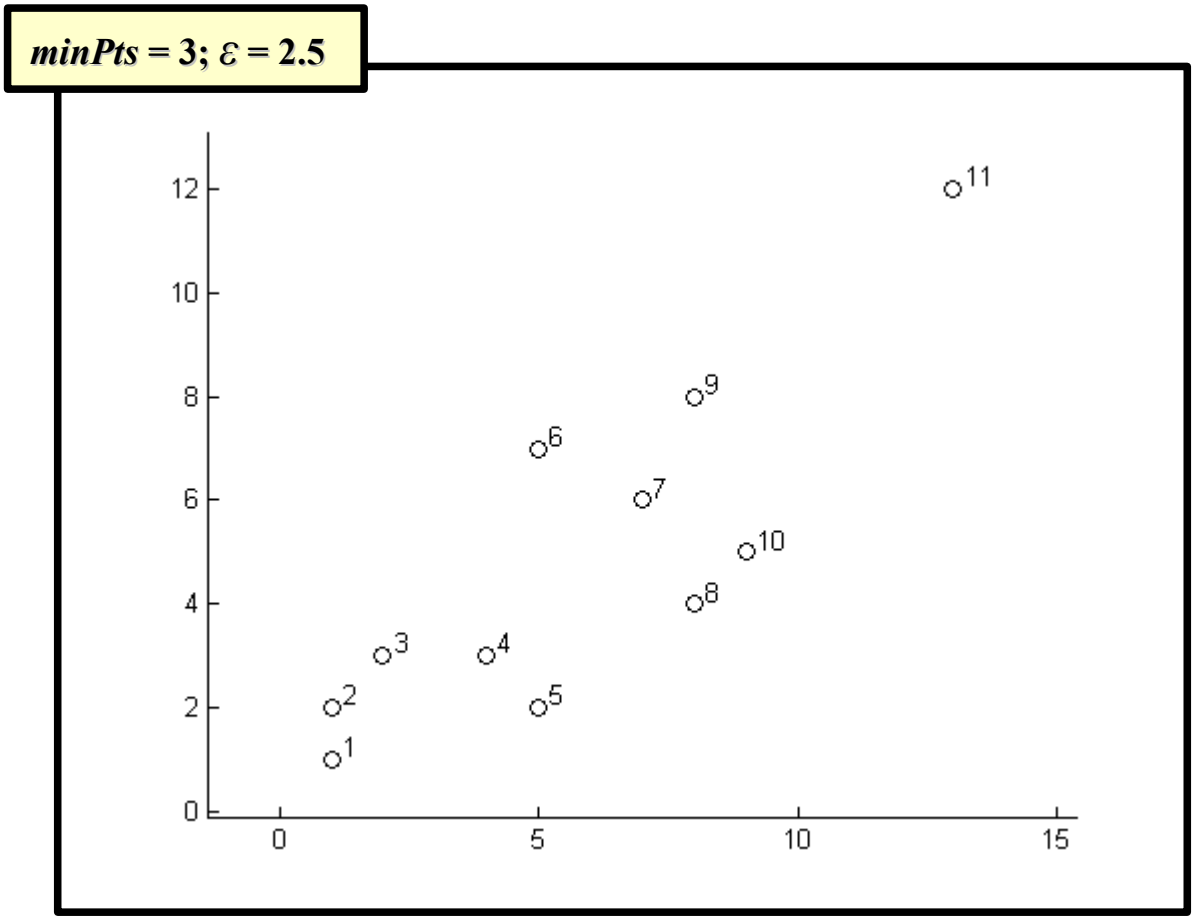


	MATLAB	R
DBSCAN:	`dbscan`	`dbscan`

Klasteryzacja: density-based clustering (algorytm DBSCAN)

Zbiór danych:

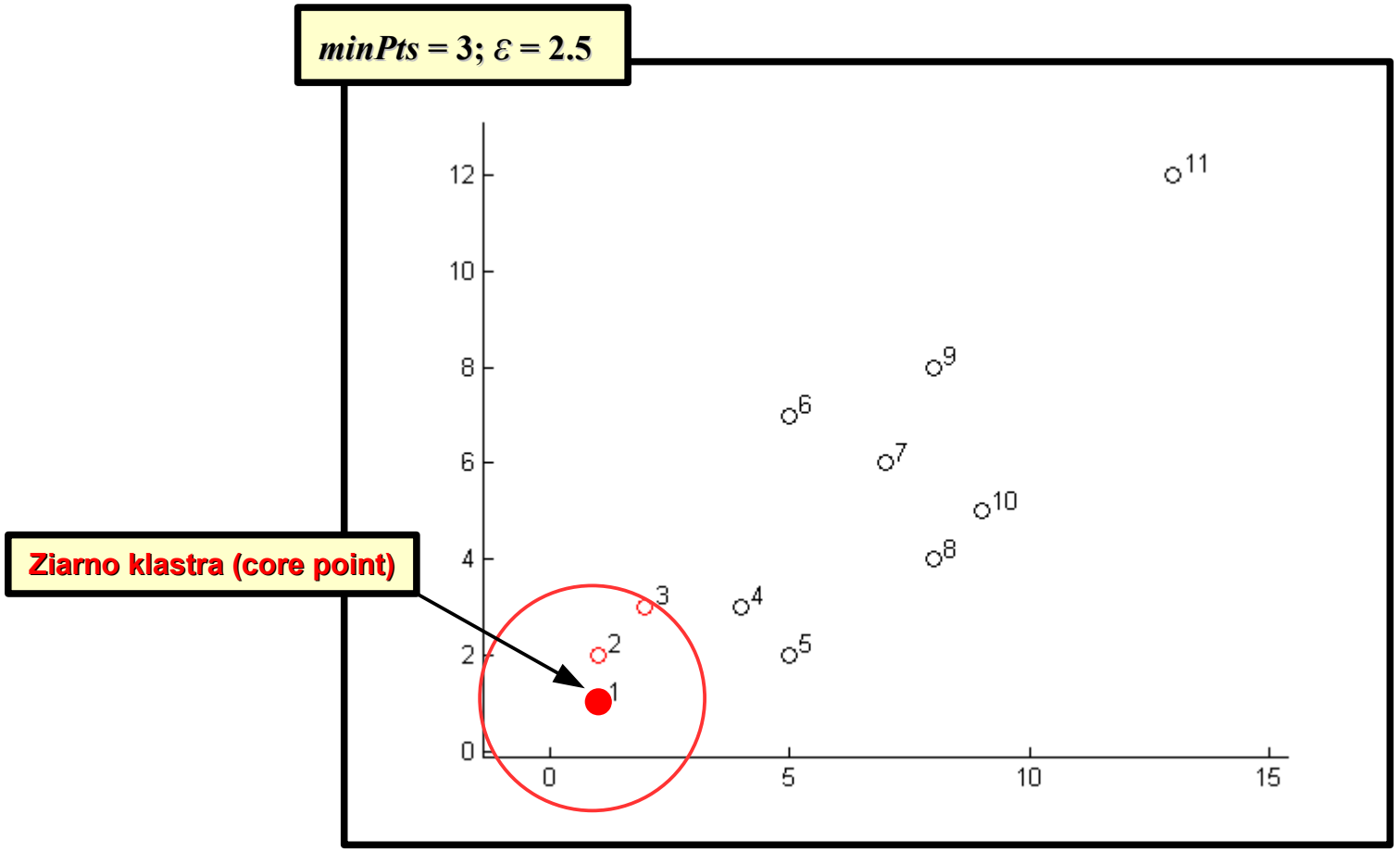
$$X = \begin{bmatrix} 1.0 & 1.0 & 2.0 & 4.0 & 5.0 & 5.0 & 7.0 & 8.0 & 8.0 & 9.0 & 13 \\ 1.0 & 2.0 & 3.0 & 3.0 & 2.0 & 7.0 & 6.0 & 4.0 & 8.0 & 5.0 & 12 \end{bmatrix}$$



Klasteryzacja: density-based clustering (algorytm DBSCAN)

Zbiór danych:

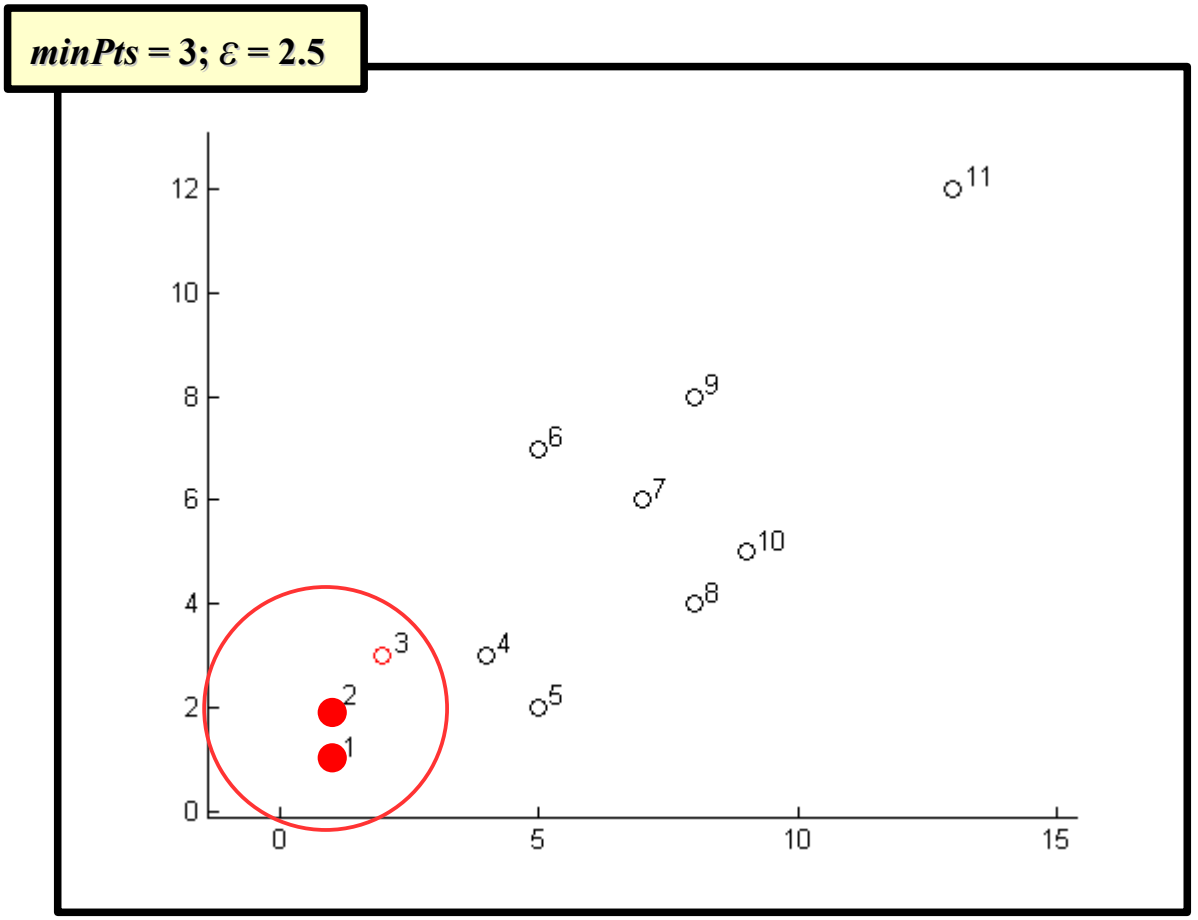
$$X = \begin{bmatrix} 1.0 & 1.0 & 2.0 & 4.0 & 5.0 & 5.0 & 7.0 & 8.0 & 8.0 & 9.0 & 13 \\ 1.0 & 2.0 & 3.0 & 3.0 & 2.0 & 7.0 & 6.0 & 4.0 & 8.0 & 5.0 & 12 \end{bmatrix}$$



Klasteryzacja: density-based clustering (algorytm DBSCAN)

Zbiór danych:

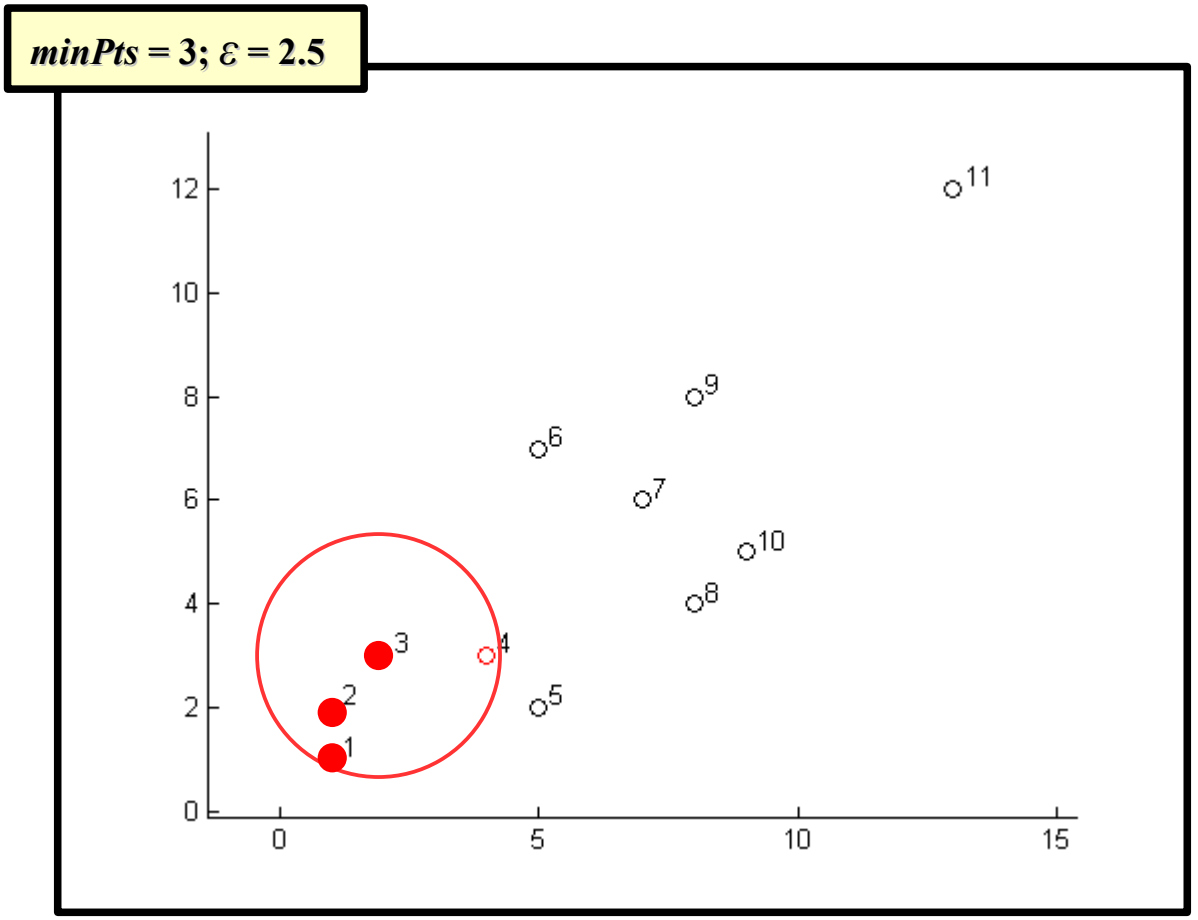
$$X = \begin{bmatrix} 1.0 & 1.0 & 2.0 & 4.0 & 5.0 & 5.0 & 7.0 & 8.0 & 8.0 & 9.0 & 13 \\ 1.0 & 2.0 & 3.0 & 3.0 & 2.0 & 7.0 & 6.0 & 4.0 & 8.0 & 5.0 & 12 \end{bmatrix}$$



Klasteryzacja: density-based clustering (algorytm DBSCAN)

Zbiór danych:

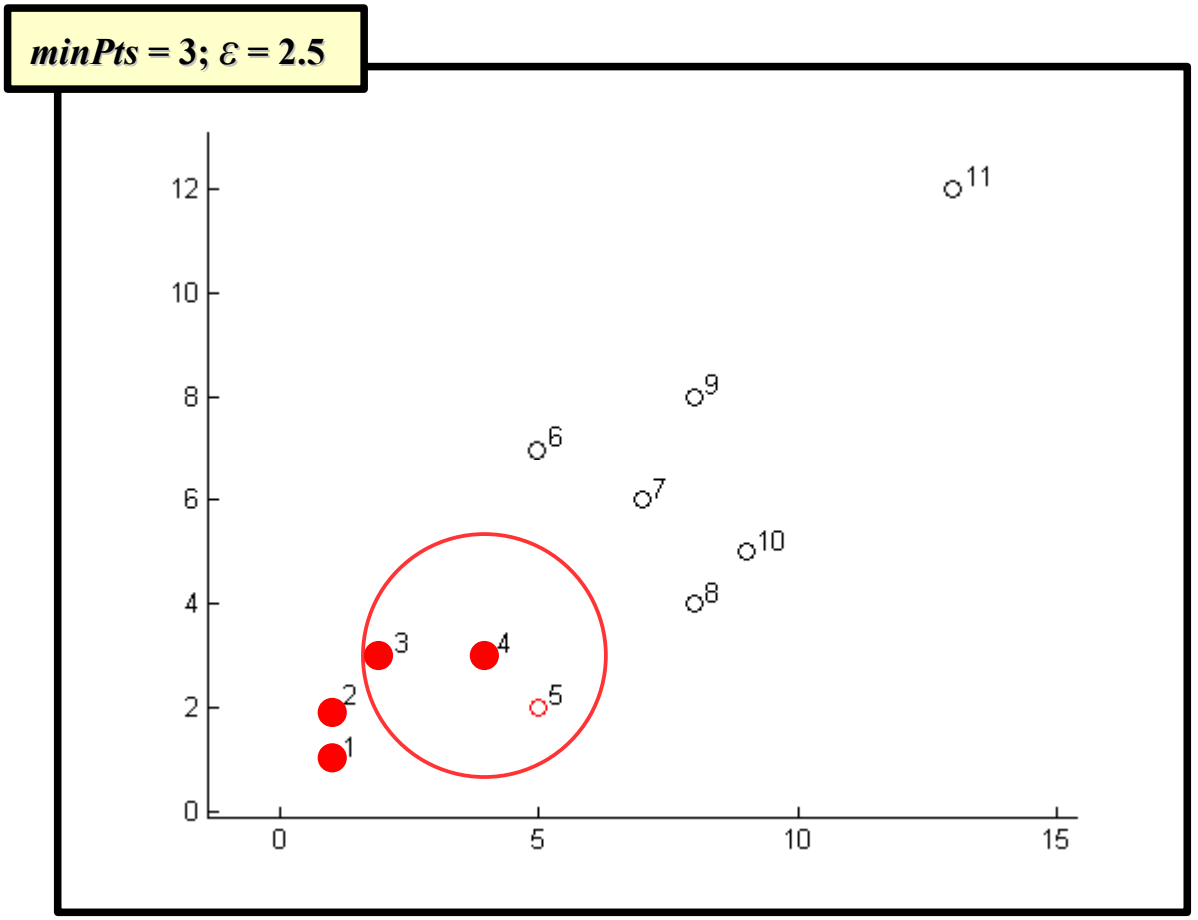
$$X = \begin{bmatrix} 1.0 & 1.0 & 2.0 & 4.0 & 5.0 & 5.0 & 7.0 & 8.0 & 8.0 & 9.0 & 13 \\ 1.0 & 2.0 & 3.0 & 3.0 & 2.0 & 7.0 & 6.0 & 4.0 & 8.0 & 5.0 & 12 \end{bmatrix}$$



Klasteryzacja: density-based clustering (algorytm DBSCAN)

Zbiór danych:

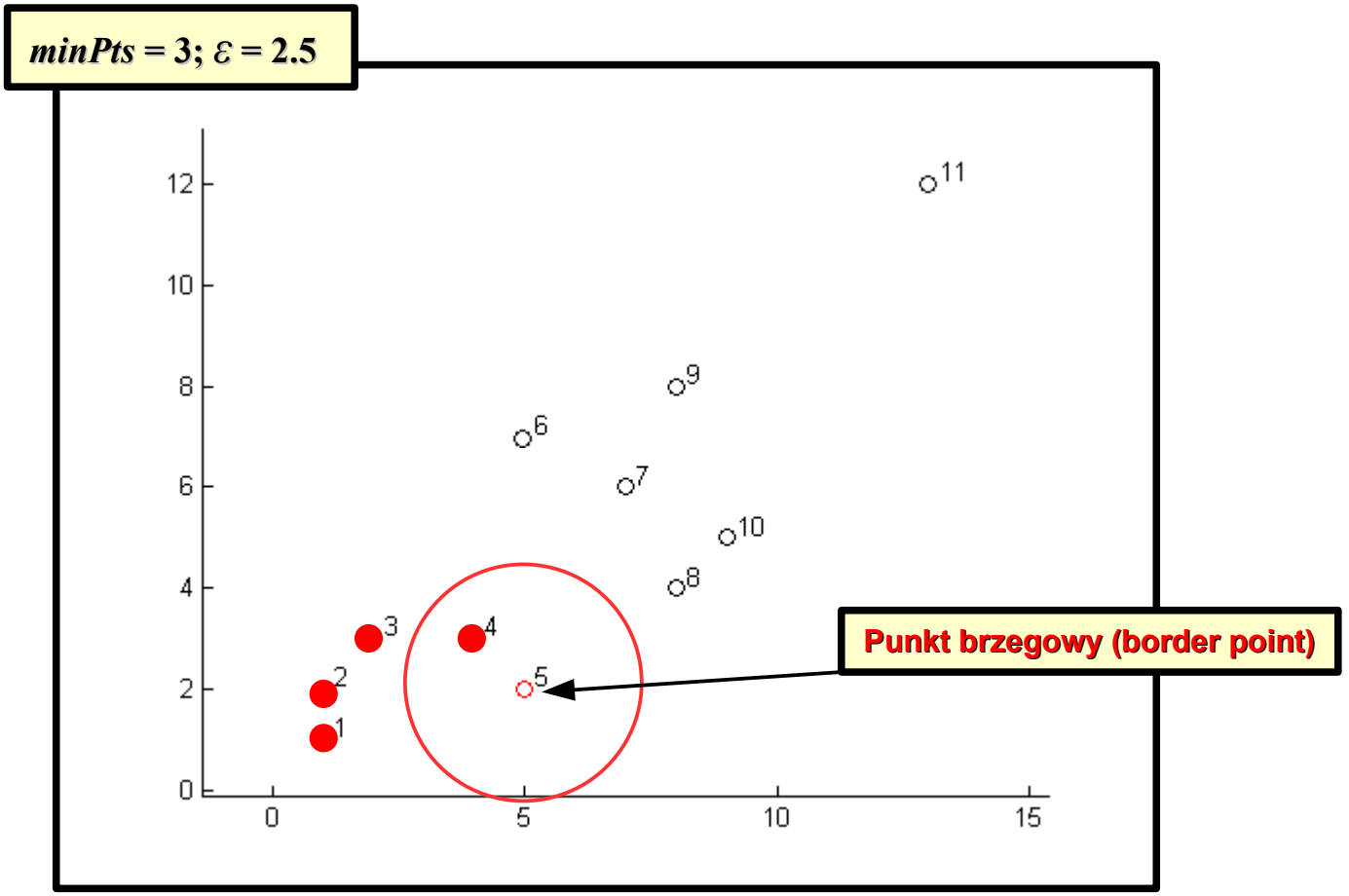
$$X = \begin{bmatrix} 1.0 & 1.0 & 2.0 & 4.0 & 5.0 & 5.0 & 7.0 & 8.0 & 8.0 & 9.0 & 13 \\ 1.0 & 2.0 & 3.0 & 3.0 & 2.0 & 7.0 & 6.0 & 4.0 & 8.0 & 5.0 & 12 \end{bmatrix}$$



Klasteryzacja: density-based clustering (algorytm DBSCAN)

Zbiór danych:

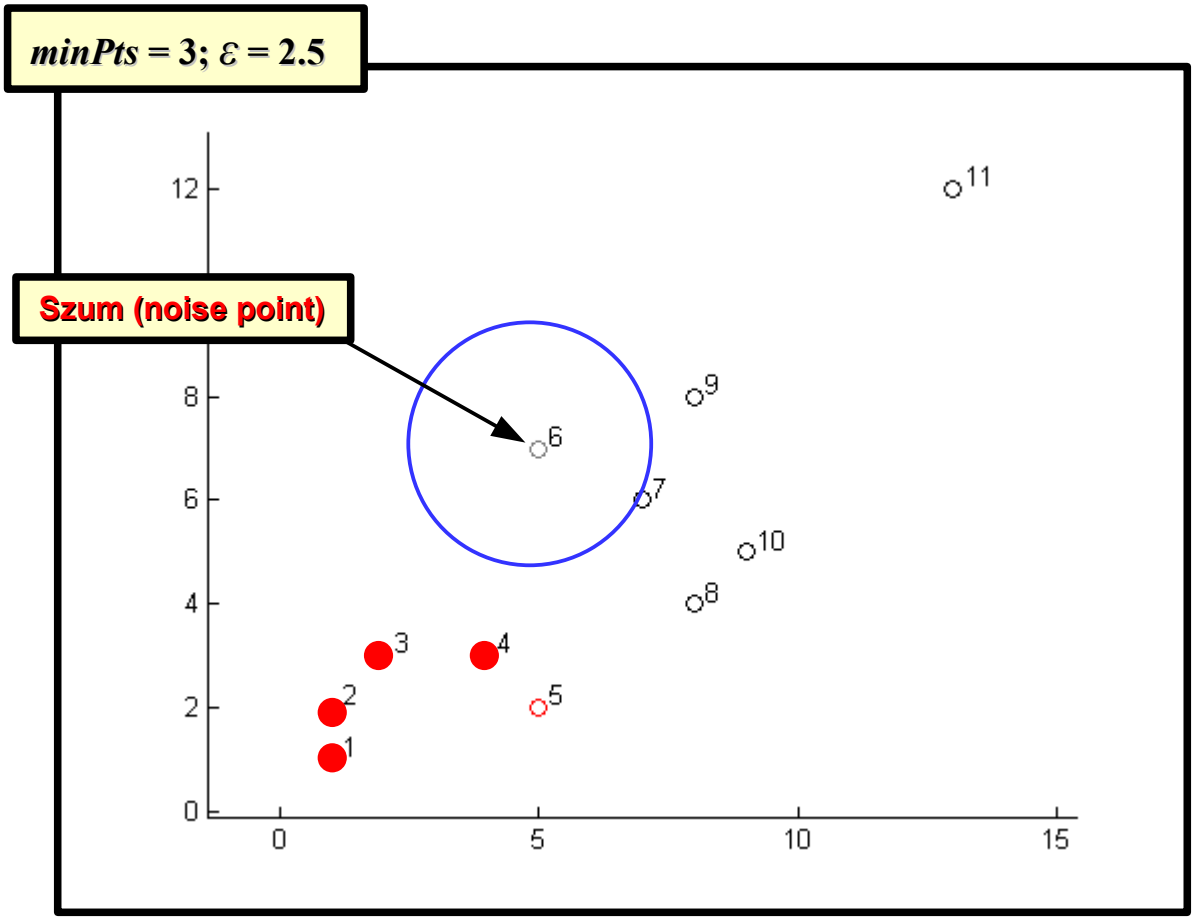
$$X = \begin{bmatrix} 1.0 & 1.0 & 2.0 & 4.0 & 5.0 & 5.0 & 7.0 & 8.0 & 8.0 & 9.0 & 13 \\ 1.0 & 2.0 & 3.0 & 3.0 & 2.0 & 7.0 & 6.0 & 4.0 & 8.0 & 5.0 & 12 \end{bmatrix}$$



Klasteryzacja: density-based clustering (algorytm DBSCAN)

Zbiór danych:

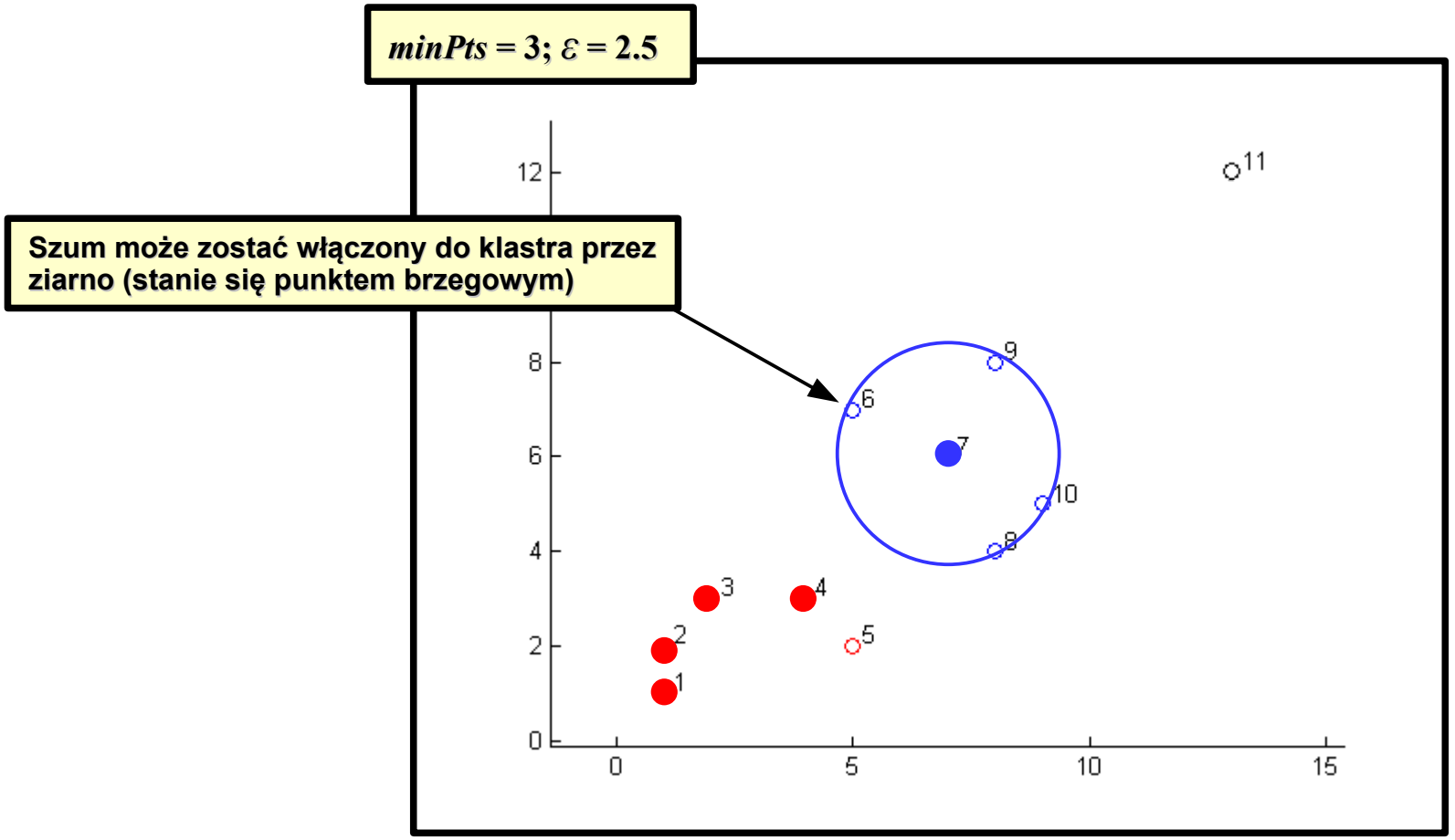
$$X = \begin{bmatrix} 1.0 & 1.0 & 2.0 & 4.0 & 5.0 & 5.0 & 7.0 & 8.0 & 8.0 & 9.0 & 13 \\ 1.0 & 2.0 & 3.0 & 3.0 & 2.0 & 7.0 & 6.0 & 4.0 & 8.0 & 5.0 & 12 \end{bmatrix}$$



Klasteryzacja: density-based clustering (algorytm DBSCAN)

Zbiór danych:

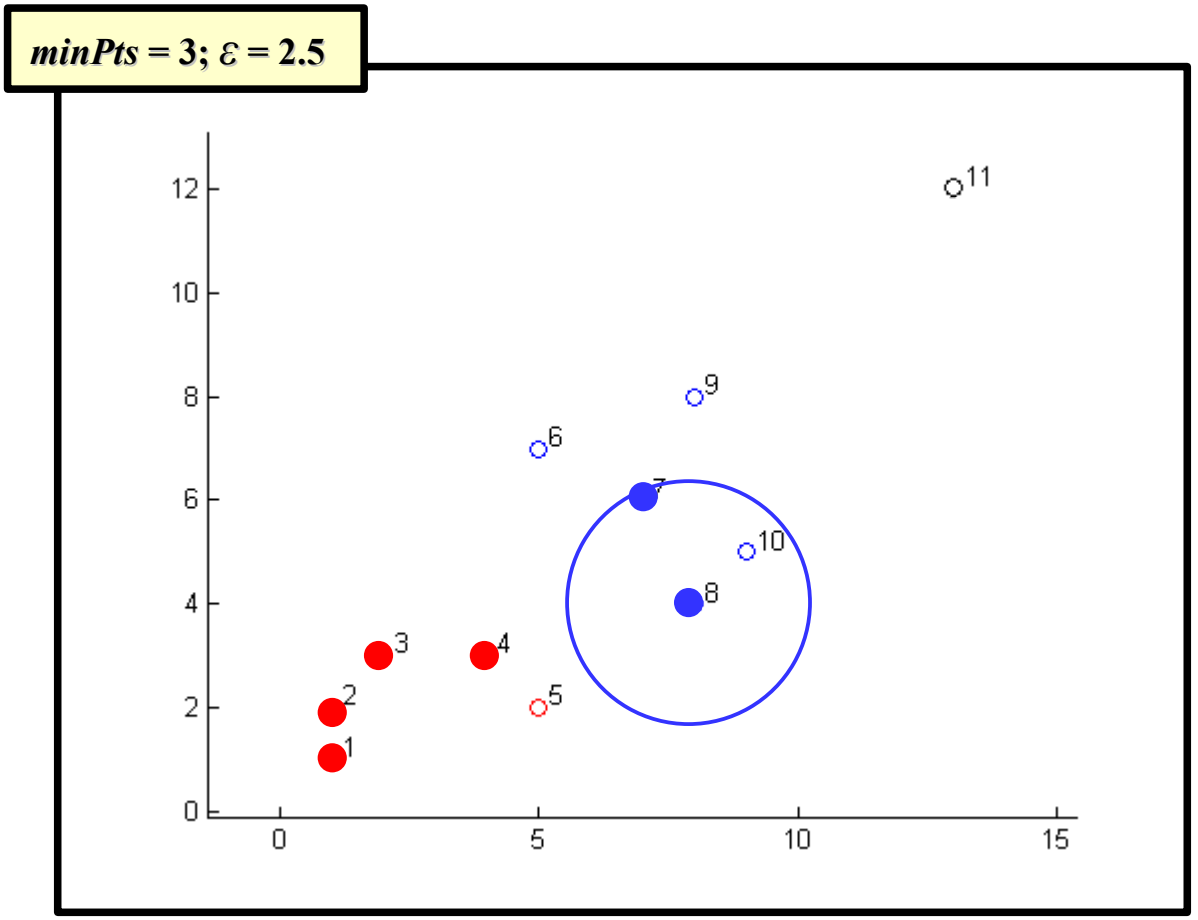
$$X = \begin{bmatrix} 1.0 & 1.0 & 2.0 & 4.0 & 5.0 & 5.0 & 7.0 & 8.0 & 8.0 & 9.0 & 13 \\ 1.0 & 2.0 & 3.0 & 3.0 & 2.0 & 7.0 & 6.0 & 4.0 & 8.0 & 5.0 & 12 \end{bmatrix}$$



Klasteryzacja: density-based clustering (algorytm DBSCAN)

Zbiór danych:

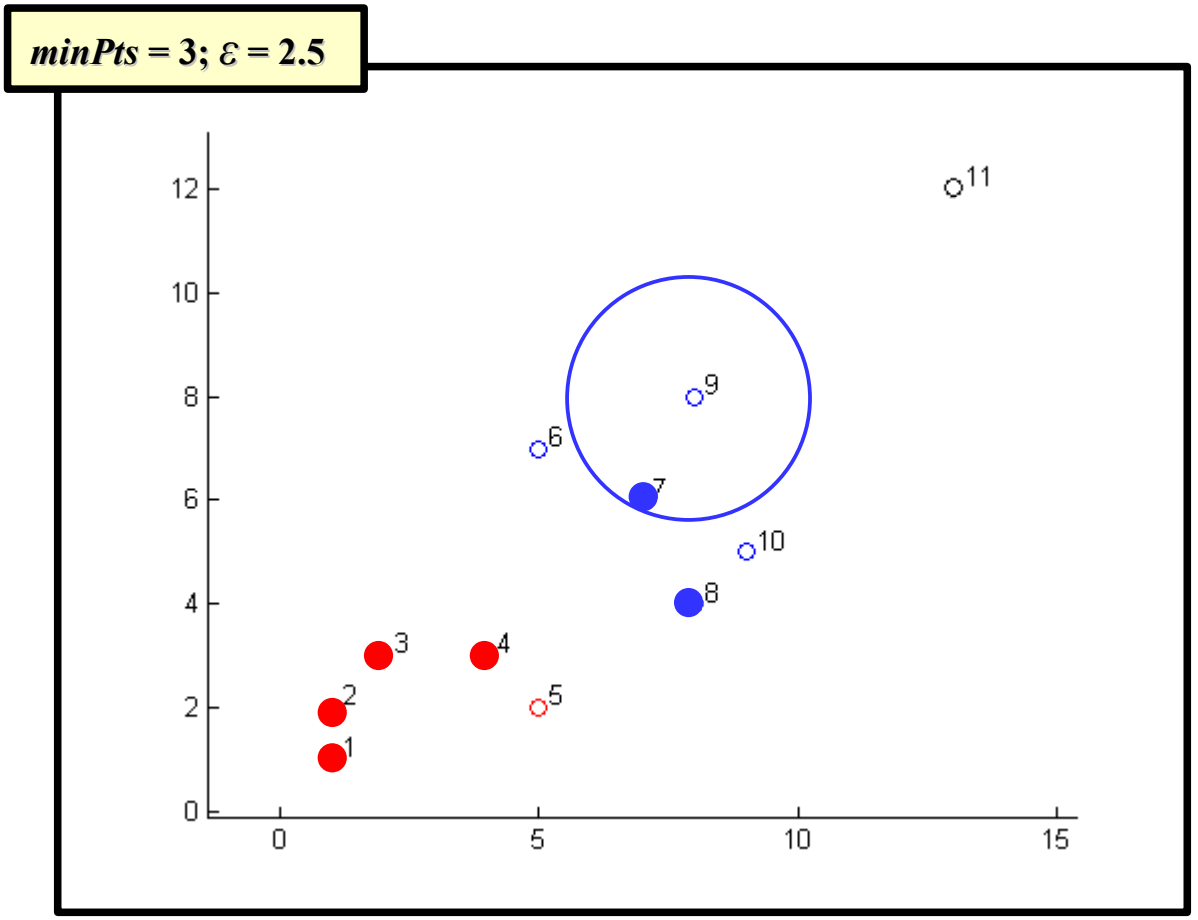
$$X = \left[\begin{array}{ccccc|ccccc|c} 1.0 & 1.0 & 2.0 & 4.0 & 5.0 & 5.0 & 7.0 & 8.0 & 8.0 & 9.0 & 13 \\ 1.0 & 2.0 & 3.0 & 3.0 & 2.0 & 7.0 & 6.0 & 4.0 & 8.0 & 5.0 & 12 \end{array}\right]$$



Klasteryzacja: density-based clustering (algorytm DBSCAN)

Zbiór danych:

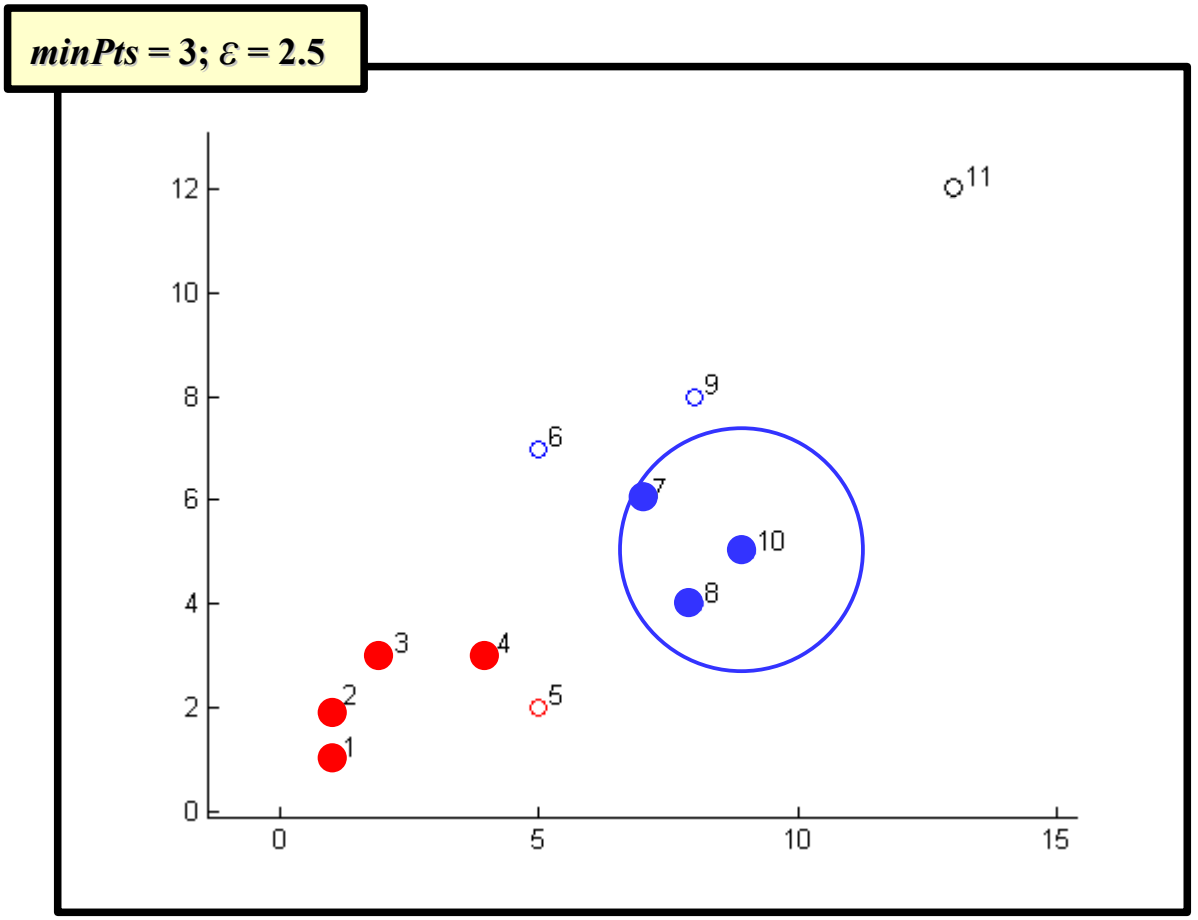
$$X = \left[\begin{array}{ccccc|ccccc|c} 1.0 & 1.0 & 2.0 & 4.0 & 5.0 & 5.0 & 7.0 & 8.0 & 8.0 & 9.0 & 13 \\ 1.0 & 2.0 & 3.0 & 3.0 & 2.0 & 7.0 & 6.0 & 4.0 & 8.0 & 5.0 & 12 \end{array}\right]$$



Klasteryzacja: density-based clustering (algorytm DBSCAN)

Zbiór danych:

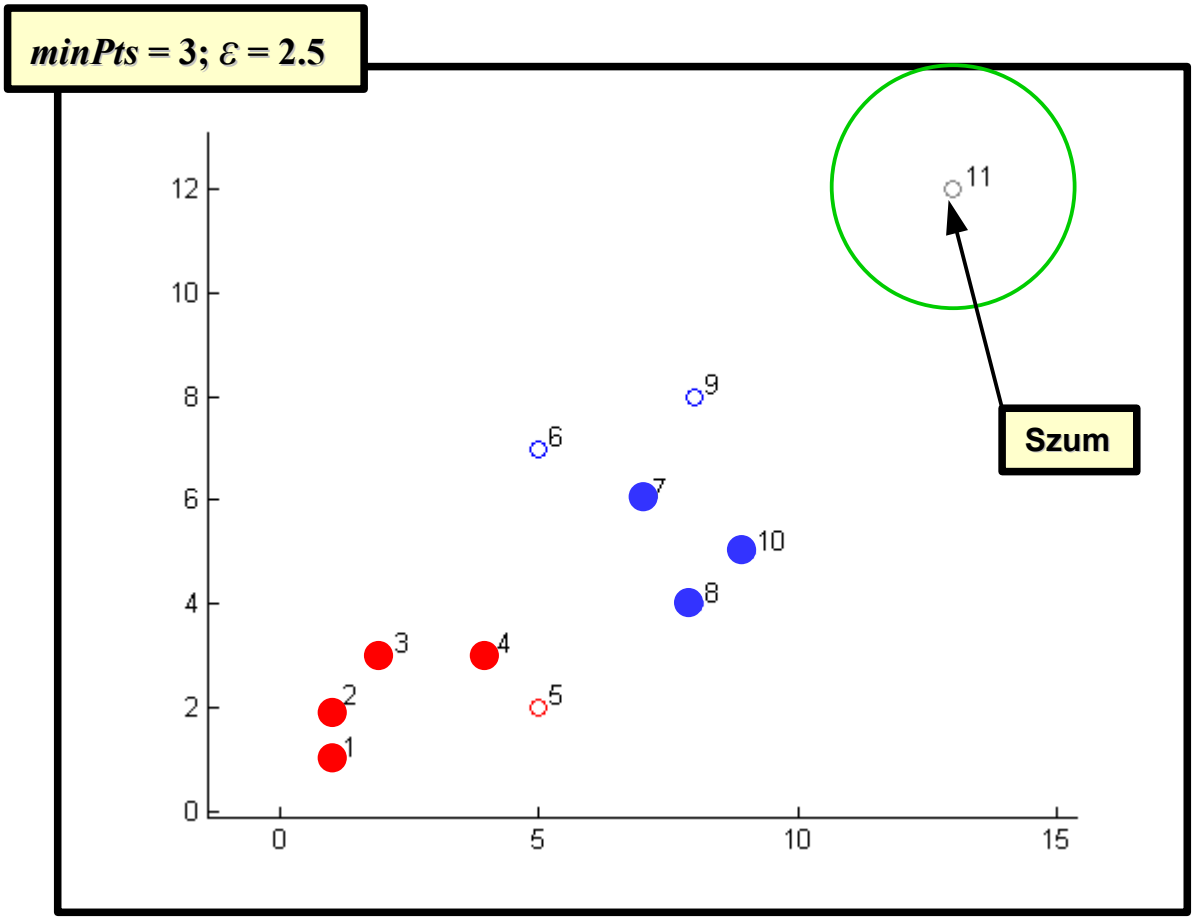
$$X = \begin{bmatrix} 1.0 & 1.0 & 2.0 & 4.0 & 5.0 & 5.0 & 7.0 & 8.0 & 8.0 & 9.0 & 13 \\ 1.0 & 2.0 & 3.0 & 3.0 & 2.0 & 7.0 & 6.0 & 4.0 & 8.0 & 5.0 & 12 \end{bmatrix}$$



Klasteryzacja: density-based clustering (algorytm DBSCAN)

Zbiór danych:

$$X = \begin{bmatrix} 1.0 & 1.0 & 2.0 & 4.0 & 5.0 & 5.0 & 7.0 & 8.0 & 8.0 & 9.0 & 13 \\ 1.0 & 2.0 & 3.0 & 3.0 & 2.0 & 7.0 & 6.0 & 4.0 & 8.0 & 5.0 & 12 \end{bmatrix}$$



Uczenie maszynowe w bioinformatyce

Wykład 6: klasteryzacja (algorytmy hierarchiczne)

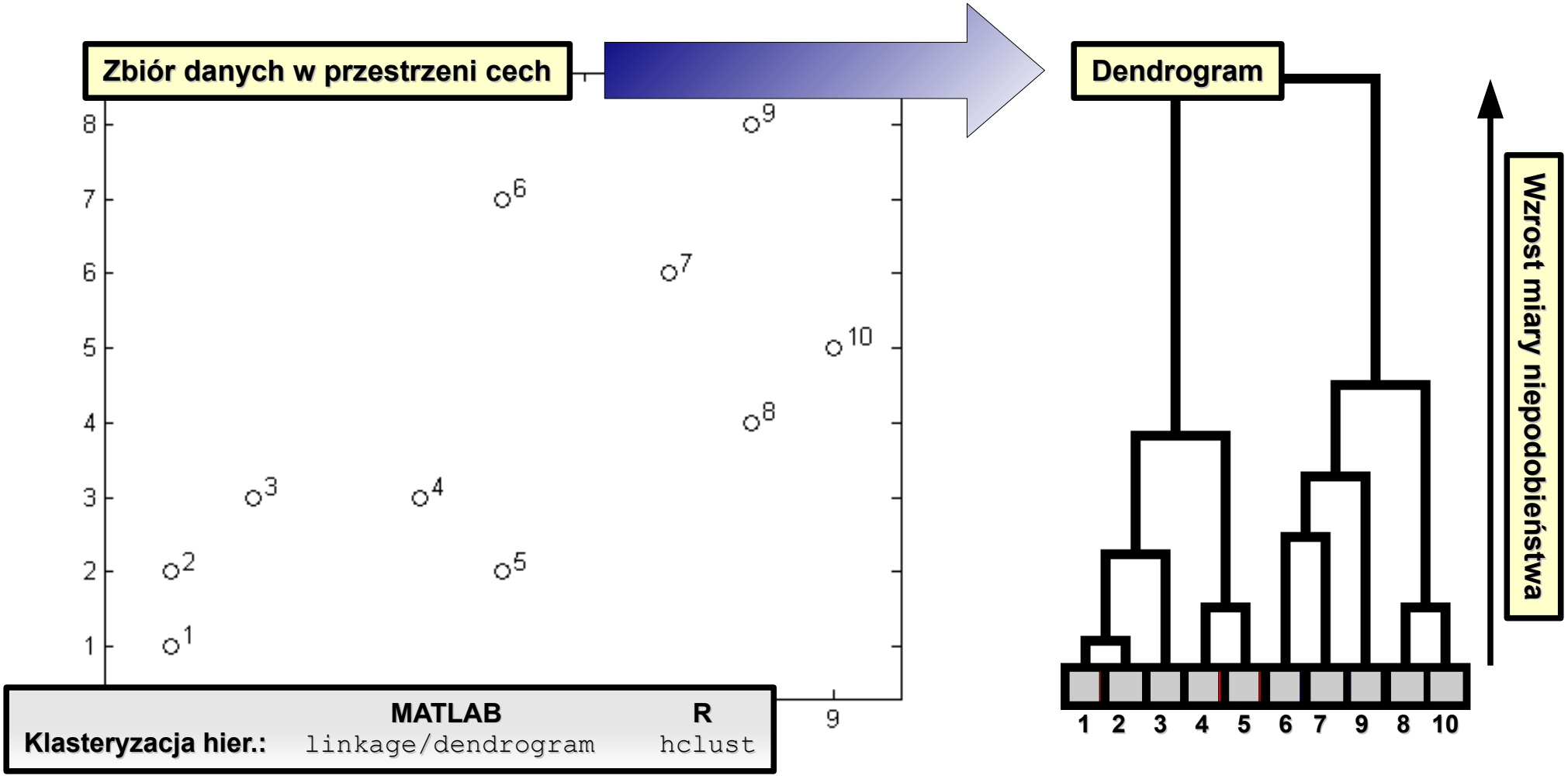
Tymon Rubel

Zakład Elektroniki Jądrowej i Medycznej
Instytut Radioelektroniki i Technik Multimedialnych PW

Klasteryzacja: algorytmy hierarchiczne

Podstawą działania algorytmów hierarchicznej klasteryzacji aglomeracyjnej jest iteracyjne łączenie grup powstałych w poprzednich krokach, przy czym początkowo każdy z obiektów stanowi osobną grupę.

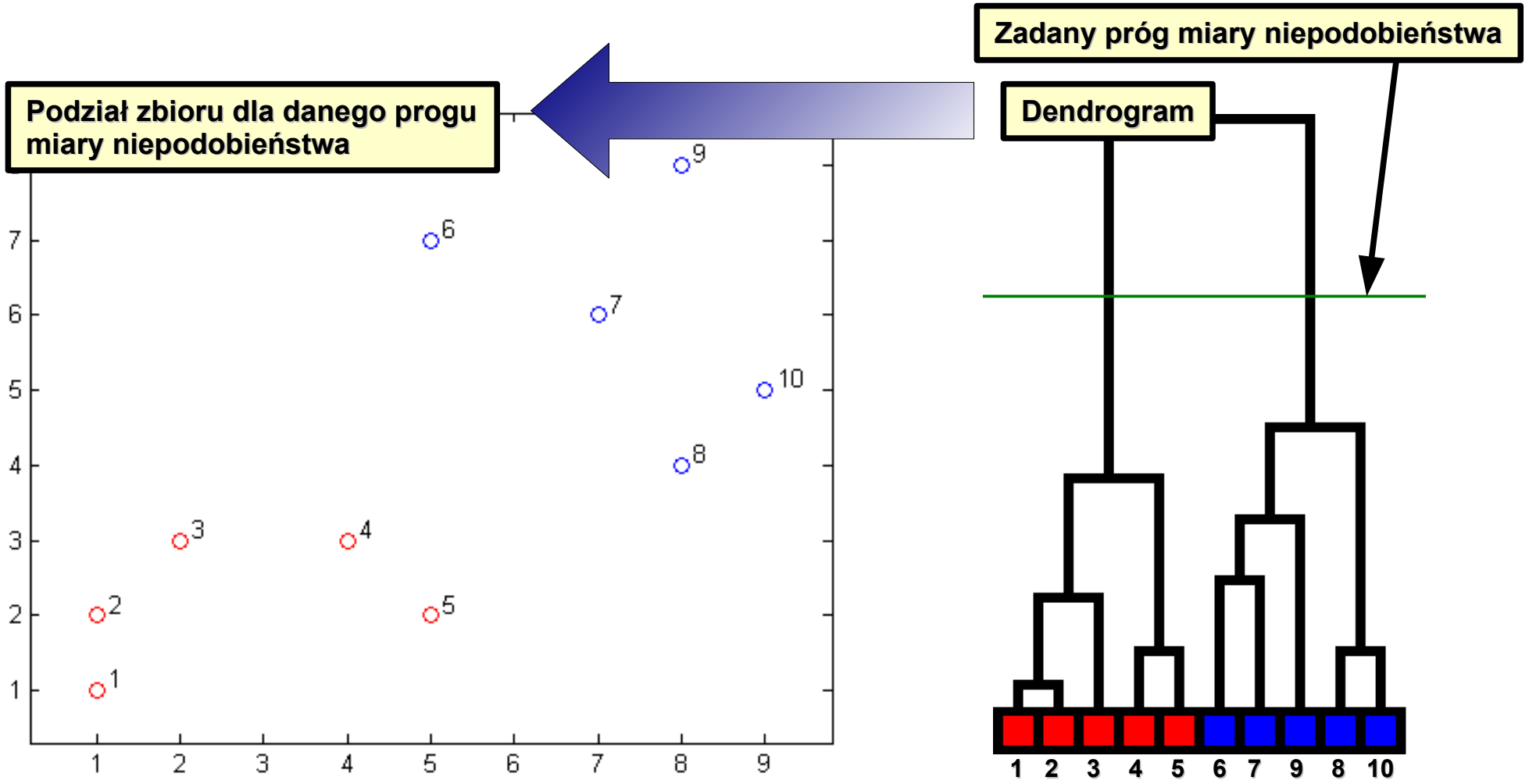
Graficzną reprezentacją efektów działania algorytmu jest dendrogram, czyli drzewo binarne, w którym długości gałęzi odwzorowują odległości pomiędzy obiektami.



	MATLAB	R
Klasteryzacja hier.:	`linkage/dendrogram`	`hclust`

Klasteryzacja: algorytmy hierarchiczne

W przypadku klasteryzacji hierarchicznej nie operujemy wprost definicją klastra, jednak na podstawie dendrogramu można podzielić zbiór danych na zadaną liczbę grup. Dokonujemy tego wybierając najwyżej położone węzły lub utworzyć klastry złożone z obiektów o niepodobieństwie mniejszym od pewnego progu.



Klasteryzacja: algorytmy hierarchiczne

Ogólny schemat działania algorytmów aglomeracyjnej klasteryzacji hierarchicznej:

- **utworzenie N klastrów odpowiadających N obiektom zbioru danych;**
- **wyznaczenie macierzy niepodobieństwa obiektów ze zbioru danych;**
- **iteracyjne powtarzanie dwóch kroków, aż do momentu gdy wszystkie obiekty połączone zostaną w pojedynczy klaster:**

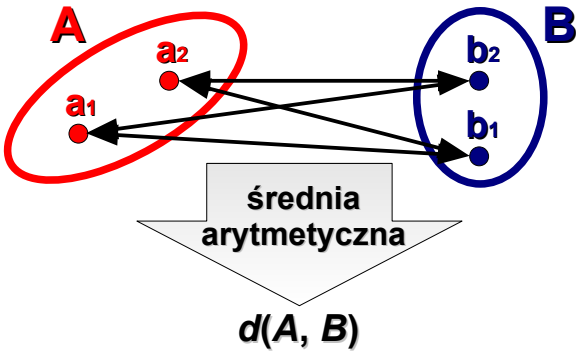
1. **połączenie klastrów A i B, pomiędzy którymi jest najmniejsza wartość miary niepodobieństwa w jeden klaster $A \cup B$ (zawierający elementy obu łączonych klastrów). Towarzyszy temu utworzenie nowego węzła w drzewie o długości gałęzi zależnej od wartości miary niepodobieństwa połączonych klastrów.**
2. **aktualizacja macierzy niepodobieństwa poprzez obliczenie wartości miary niepodobieństwa nowego klastra $A \cup B$ do wszystkich pozostałych (usunięcie wierszy i kolumn odpowiadających klastrom A i B oraz dodanie kolumny i wiersza odpowiadających połączonemu klastrowi $A \cup B$).**

Elementem różniącym poszczególne rodzaje algorytmów jest sposób wyznaczania odległości pomiędzy nowym klastrem $A \cup B$ a pozostałymi.

Klasteryzacja: algorytmy hierarchiczne (średniego wiązania)

W algorytmie średniego wiązania (*average linkage*, UPGMA – *Unweighted Pair-Group Method with Arithmatic Mean*) miara niepodobieństwa $d(A, B)$ pomiędzy klastrami A i B definiowana jest jako średnia arytmetyczna miar niepodobieństw $d(a,b)$ pomiędzy wszystkimi parami elementów należących do tych klastrów:

$$d(A, B) = \frac{1}{N_A N_B} \sum_{i=1}^{N_A} \sum_{j=1}^{N_B} d(a_i, b_j)$$



gdzie N_A i N_B oznaczają liczby elementów klastrów A i B.

Wartość miary niepodobieństwa $d(C, (A \cup B))$ pomiędzy połączonym klastrem $A \cup B$ a klastrem C może również zostać wyznaczona jako:

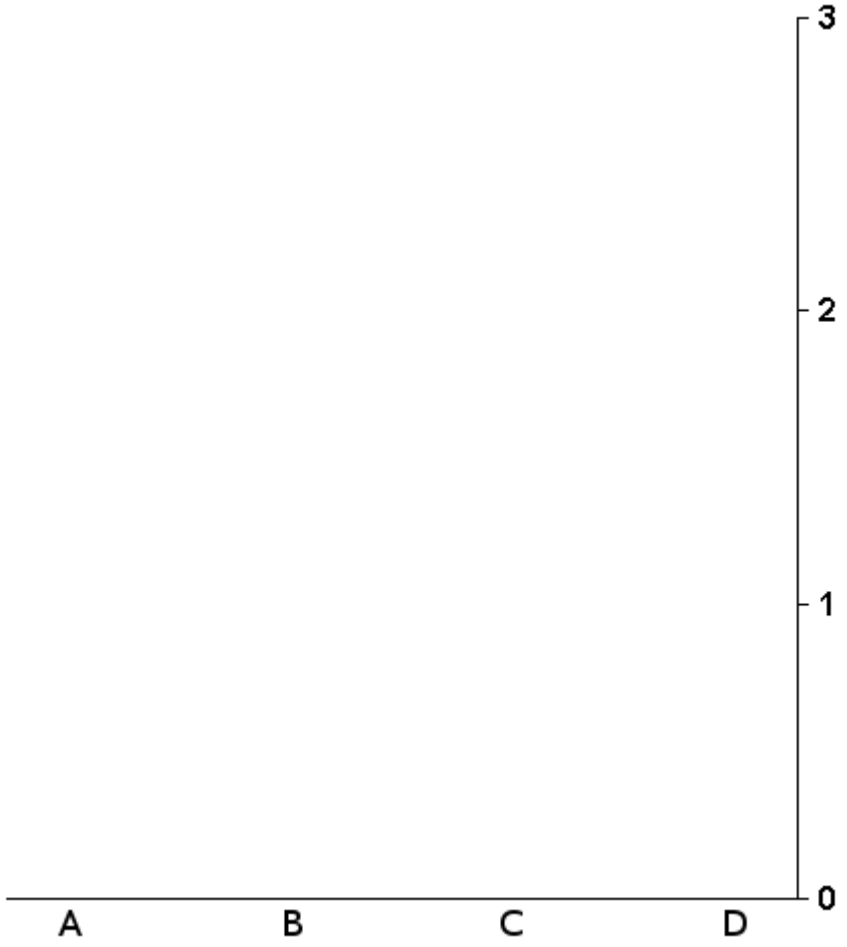
$$d(C, (A \cup B)) = \frac{N_A}{N_A + N_B} d(C, A) + \frac{N_B}{N_A + N_B} d(C, B)$$

Wzór ten jest równoważny temu podanemu na górze, ale w praktyce wygodniejszy, gdyż nie wymaga przechowywania pierwotnej macierzy niepodobieństwa (w każdej iteracji nową macierz wyznacza się w oparciu o wartości z poprzedniej iteracji).

Klasteryzacja: algorytmy hierarchiczne (średniego wiązania)

Inicjalizacja: utworzenie klastra dla każdego obiektu i wyznaczenie wartości miar niepodobieństwa pomiędzy każdą parą obiektów

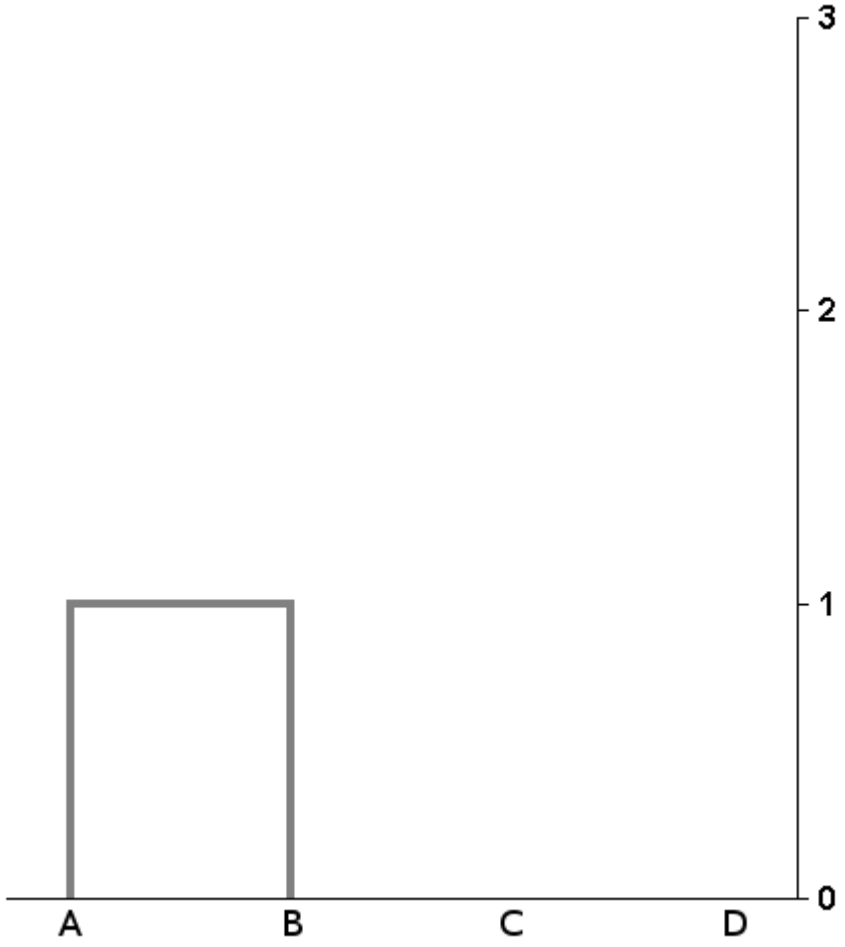
	A	B	C	D
A	0			
B	2	0		
C	6	6	0	
D	6	6	4	0



Klasteryzacja: algorytmy hierarchiczne (średniego wiązania)

Iteracja 1: wybór i połączenie pary klastrów o najmniejszym niepodobieństwie

	A	B	C	D
A	0			
B	2	0		
C	6	6	0	
D	6	6	4	0



Klasteryzacja: algorytmy hierarchiczne (średniego wiązania)

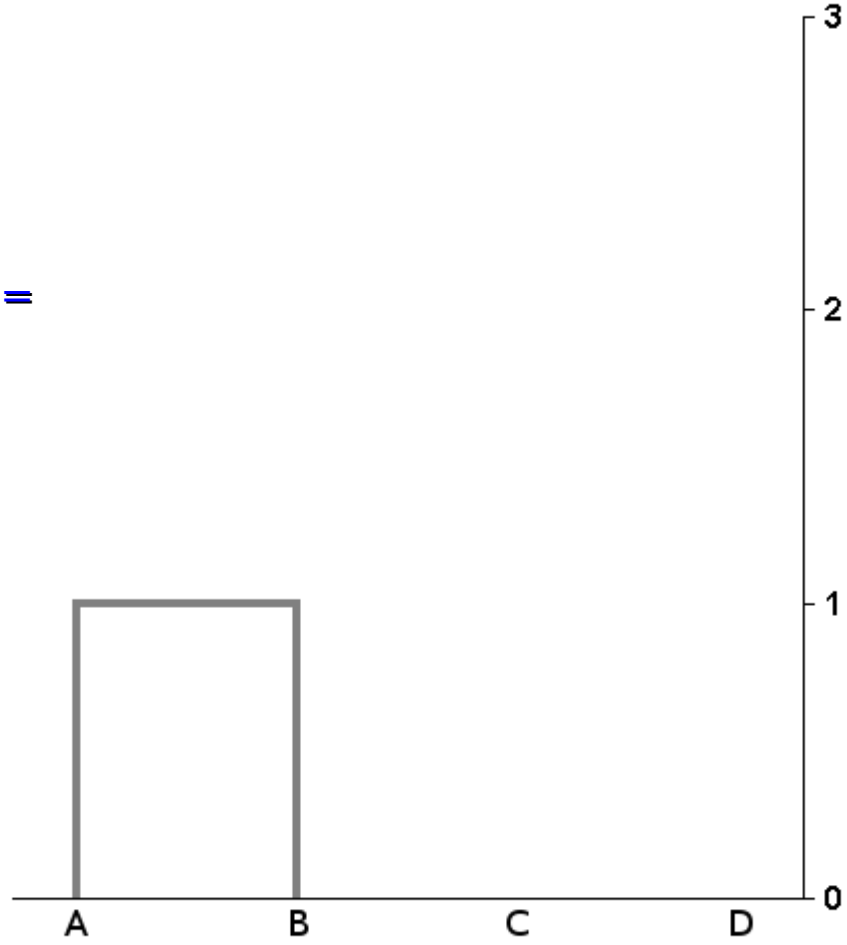
Iteracja 1: aktualizacja macierzy niepodobieństwa (zgodnie ze schematem UPGMA)

	A	B	C	D
A	0			
B	2	0		
C	6	6	0	
D	6	6	4	0

	AB	C	D
AB	0		
C	6	0	
D	6	4	0

$$d(AB,C) = N_A / (N_A+N_B) \cdot d(A,C) + N_B / (N_A+N_B) \cdot d(B,C) =$$

$$= 1 / (1+1) \cdot 6 + 1 / (1+1) \cdot 6 = 6$$



Klasteryzacja: algorytmy hierarchiczne (średniego wiązania)

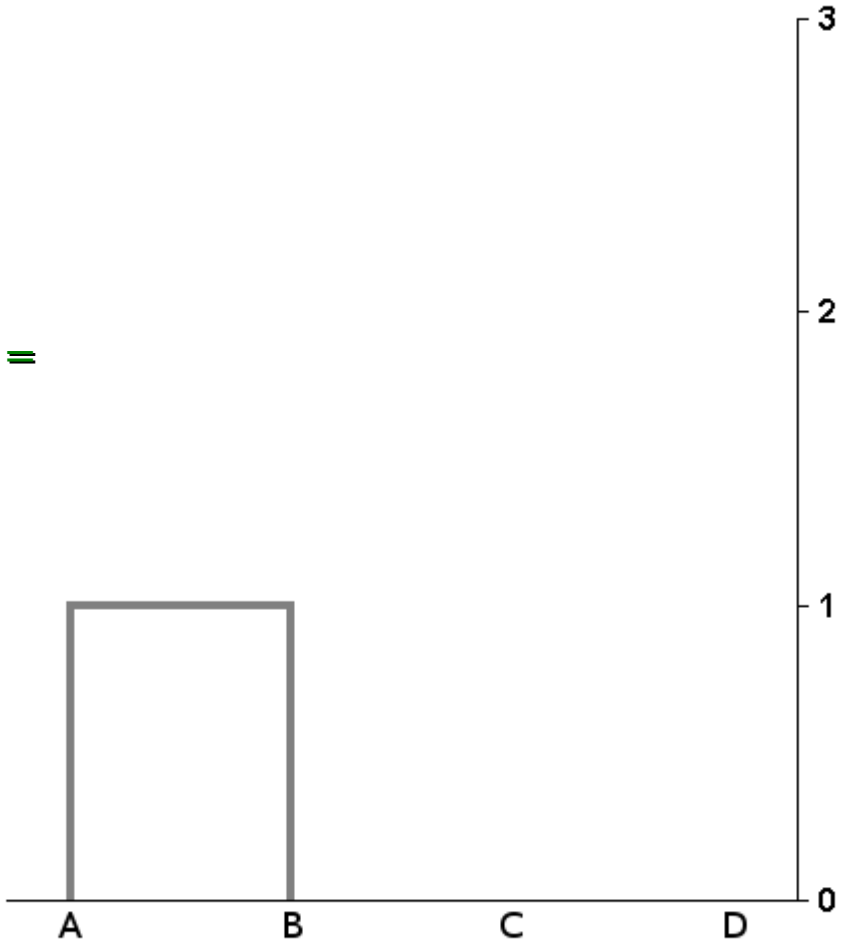
Iteracja 1: aktualizacja macierzy niepodobieństwa (zgodnie ze schematem UPGMA)

	A	B	C	D
A	0			
B	2	0		
C	6	6	0	
D	6	6	4	0

	AB	C	D
AB	0		
C	6	0	
D	6	4	0

$$d(AB,D) = N_A / (N_A+N_B) \cdot d(A,D) + N_B / (N_A+N_B) \cdot d(B,D) =$$

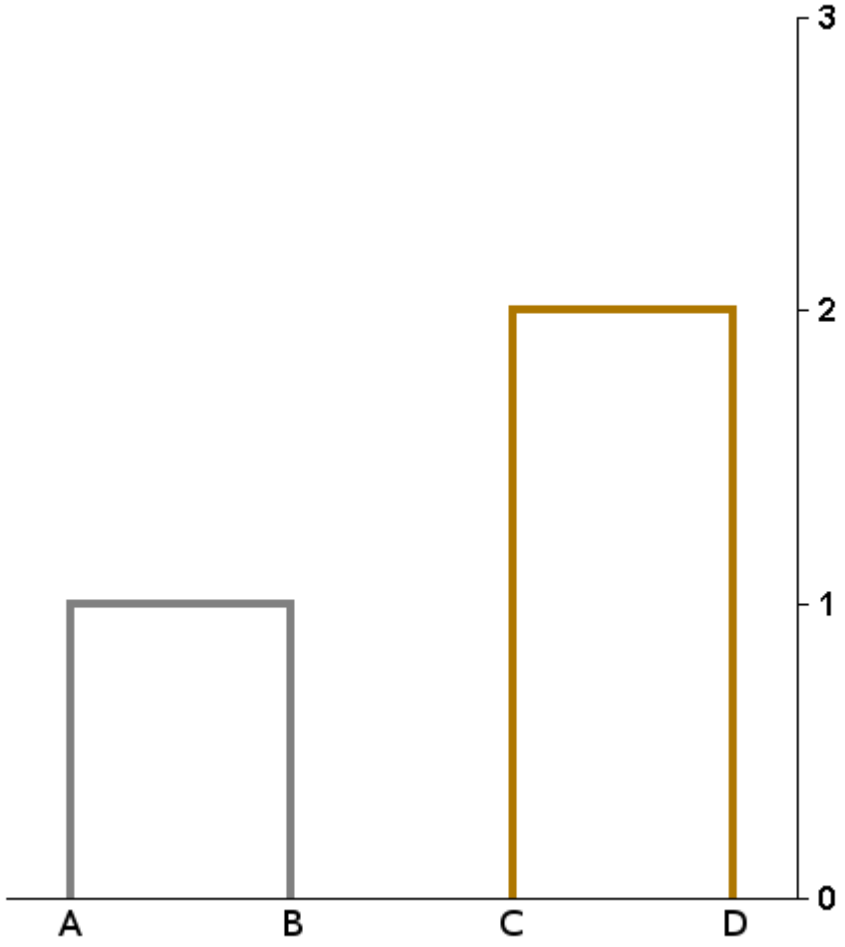
$$= 1 / (1+1) \cdot 6 + 1 / (1+1) \cdot 6 = 6$$



Klasteryzacja: algorytmy hierarchiczne (średniego wiązania)

Iteracja 2: wybór i połączenie pary klastrów o najmniejszym niepodobieństwie

	AB	C	D
AB	0		
C	6	0	
D	6	4	0



Klasteryzacja: algorytmy hierarchiczne (średniego wiązania)

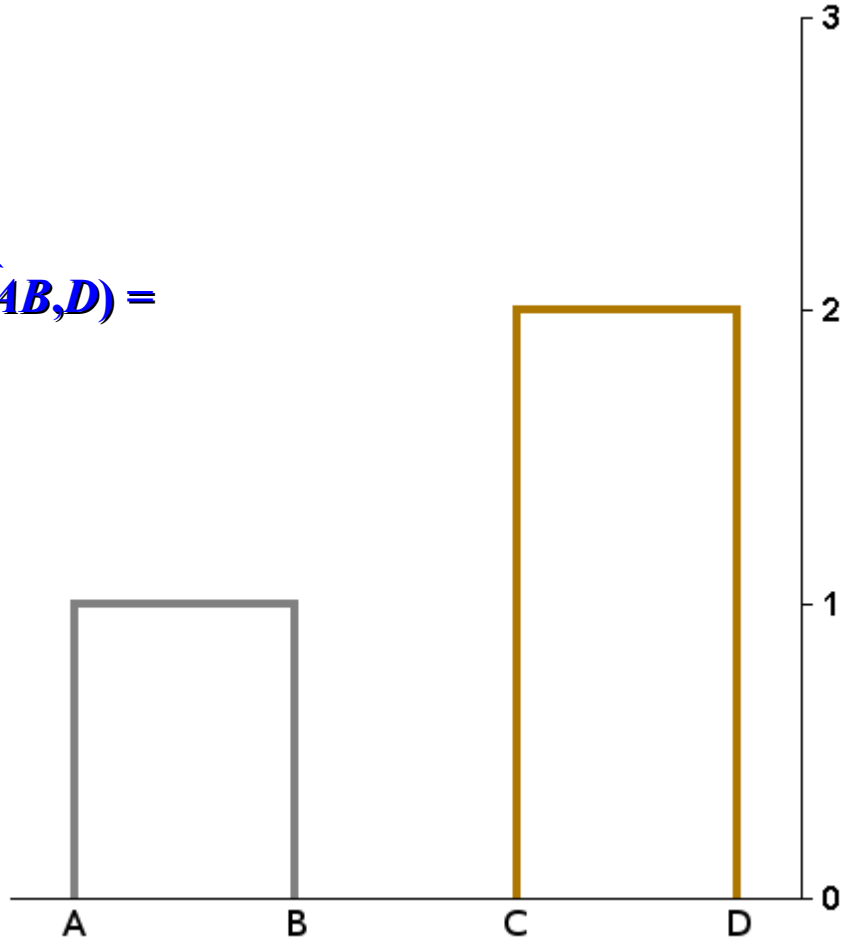
Iteracja 2: aktualizacja macierzy niepodobieństwa (zgodnie ze schematem UPGMA)

	AB	C	D
AB	0		
C	6	0	
D	6	4	0

	AB	CD
AB	0	
CD	6	0

$$d(CD, AB) = N_C / (N_C+N_D) \cdot d(AB,C) + N_D / (N_C+N_D) \cdot d(AB,D) =$$

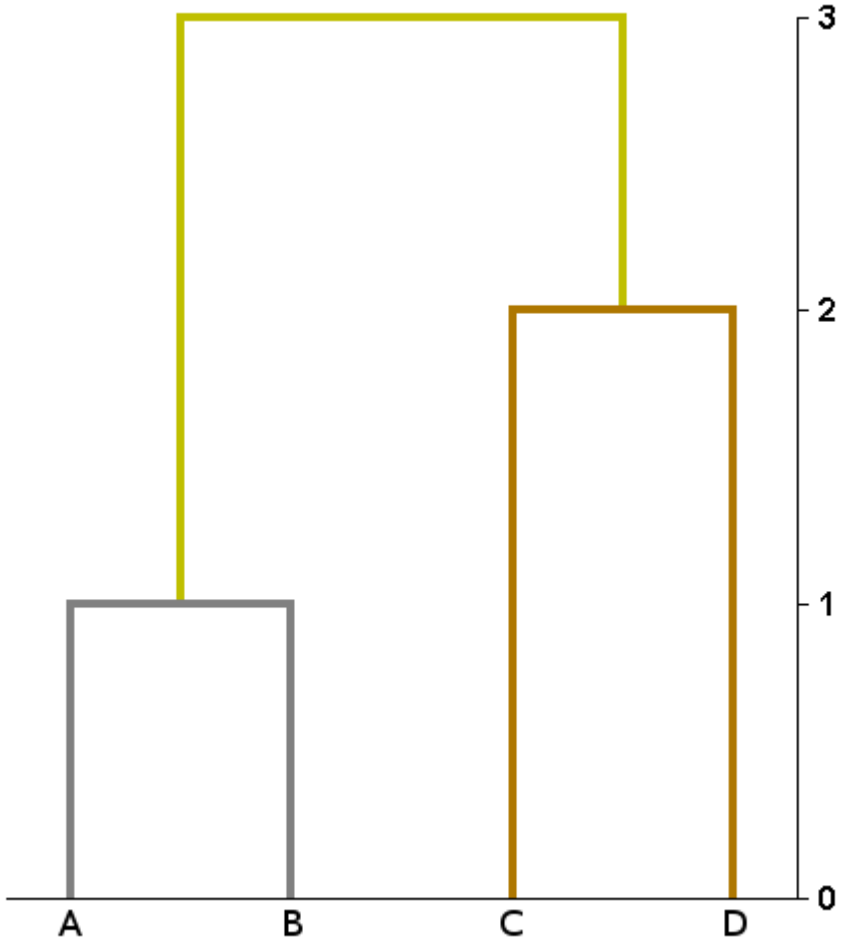
$$= 1/(1+1) \cdot 6 + 1/(1+1) \cdot 6 = 6$$



Klasteryzacja: algorytmy hierarchiczne (średniego wiązania)

Iteracja 3: wybór i połączenie pary klastrów o najmniejszym niepodobieństwie

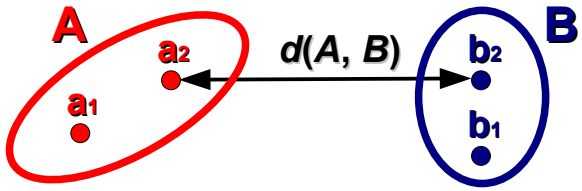
	AB	CD
AB	0	
CD	6	0



Klasteryzacja: algorytmy hierarchiczne (pojedynczego wiązania)

W algorytmie pojedynczego wiązania (najbliższego sąsiedztwa, *single linkage*) miara niepodobieństwa $d(A, B)$ pomiędzy klastrami A i B definiowana jest jako najmniejsza wartość niepodobieństwa pomiędzy elementami tych klastrów:

$$d(A, B) = \min_{a \in A, b \in B} d(a, b)$$



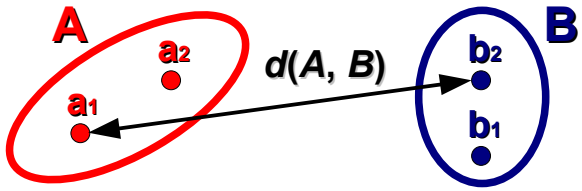
Aktualizacja niepodobieństwa $d(C, (A \cup B))$ pomiędzy nowo utworzonym klastrem $A \cup B$ a klastrem C następuje zgodnie z zależnością:

$$d(C, (A \cup B)) = \min\{d(C, A), d(C, B)\} = \frac{1}{2}(d(C, A) + d(C, B) - |d(C, A) - d(C, B)|)$$

Klasteryzacja: algorytmy hierarchiczne (pełnego wiązania)

W algorytmie pełnego wiązania (najdalszego sąsiedztwa, *complete linkage*) miara niepodobieństwa $d(A, B)$ pomiędzy klastrami A i B definiowana jest jako największa wartości niepodobieństwa pomiędzy elementami tych klastrów:

$$d(A,B) = \max_{a \in A, b \in B} d(a,b)$$



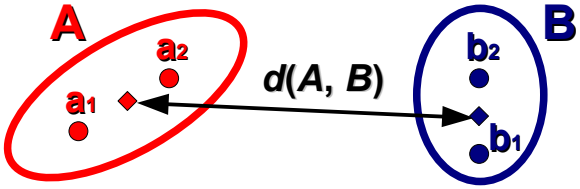
Aktualizacja niepodobieństwa $d(C, (A \cup B))$ pomiędzy nowo utworzonym klastrem $A \cup B$ a klastrem C następuje zgodnie z zależnością:

$$d(C,(A \cup B)) = \max\{d(C,A), d(C,B)\} = \frac{1}{2}\left(d(C,A) + d(C,B) + |d(C,A) - d(C,B)|\right)$$

Klasteryzacja: algorytmy hierarchiczne (środków ciężkości)

W algorytmie środków ciężkości (centroidów, UPGMC – *Unweighted Pair-Group Method with Centroid*) odległość $d(A, B)$ pomiędzy klastrami A i B definiowana jest jako odległość pomiędzy centroidami tych klastrów (będących wektorami wartości średnich elementów należących do klastrów):

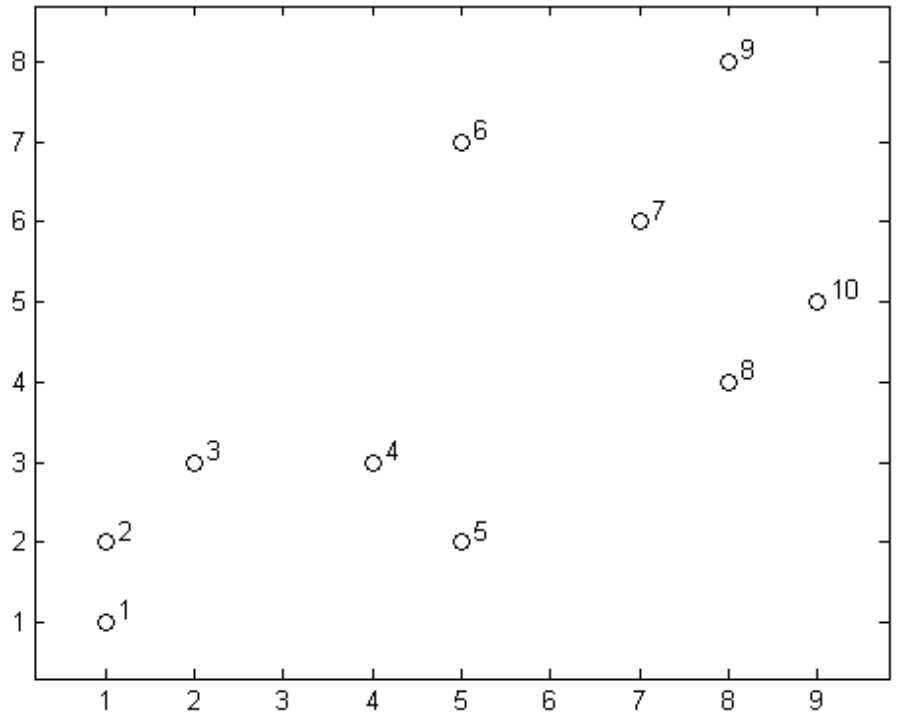
$$d(A, B) = d(\bar{a}, \bar{b})$$

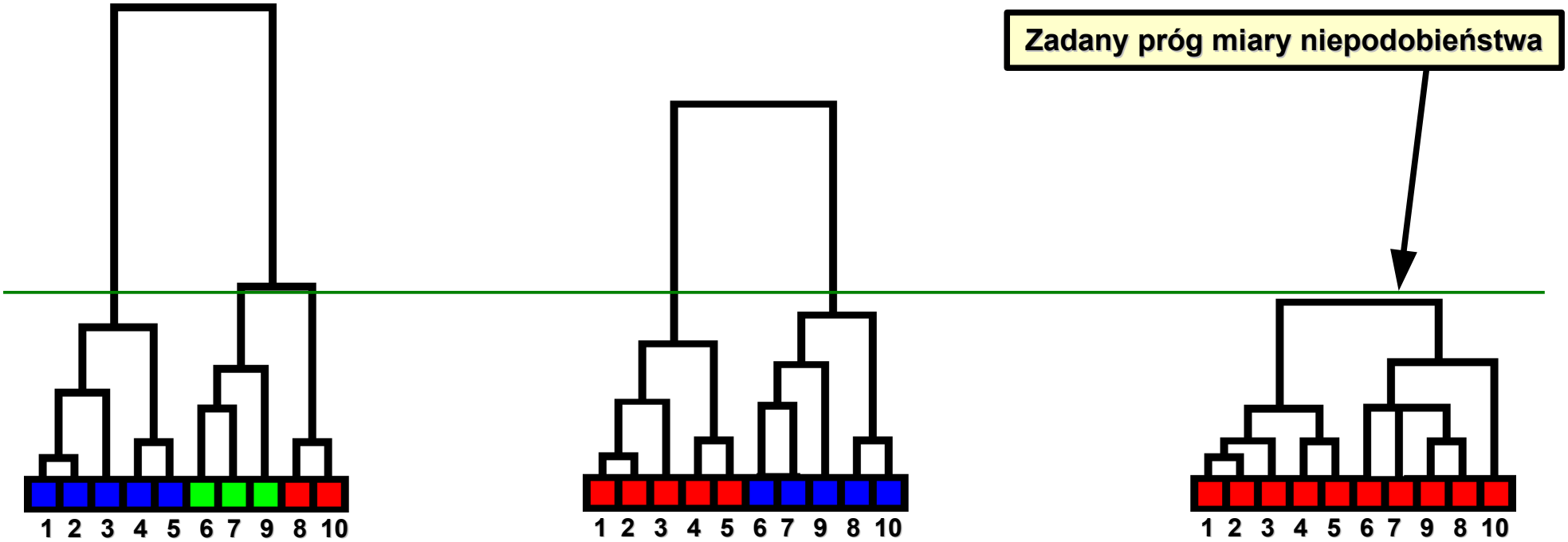


W przypadku stosowania odległości Euklidesa jako miary niepodobieństwa aktualizacja $d(C, (A \cup B))$ pomiędzy nowo utworzonym klastrem $A \cup B$ a klastrem C następuje zgodnie z zależnością:

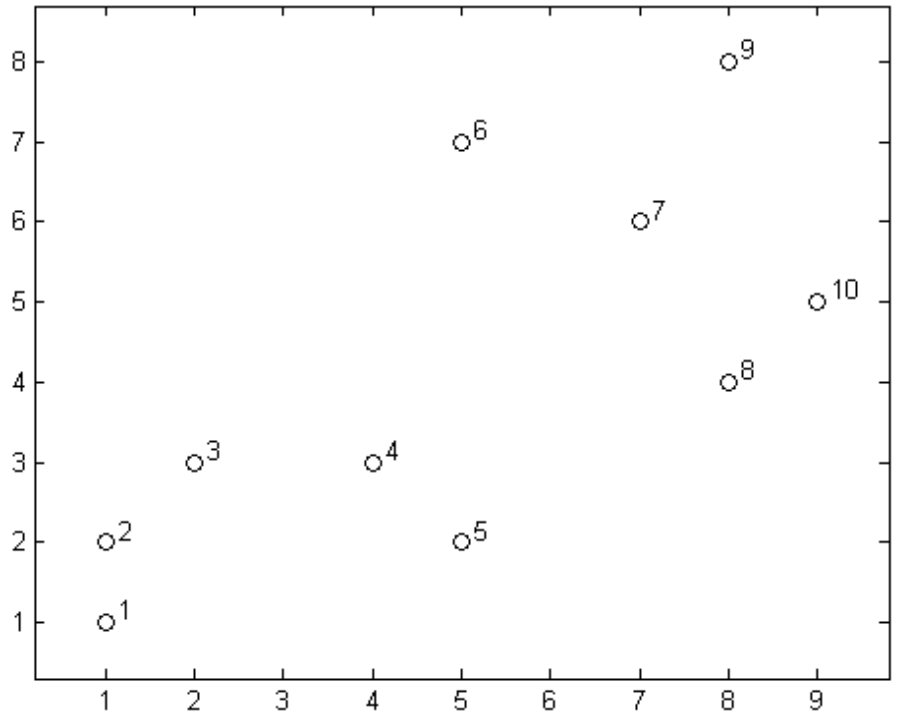
$$d(C, (A \cup B)) = \frac{N_A}{N_A + N_B} d(C, A) + \frac{N_B}{N_A + N_B} d(C, B) - \frac{N_A N_B}{(N_A + N_B)^2} d(A, B)$$

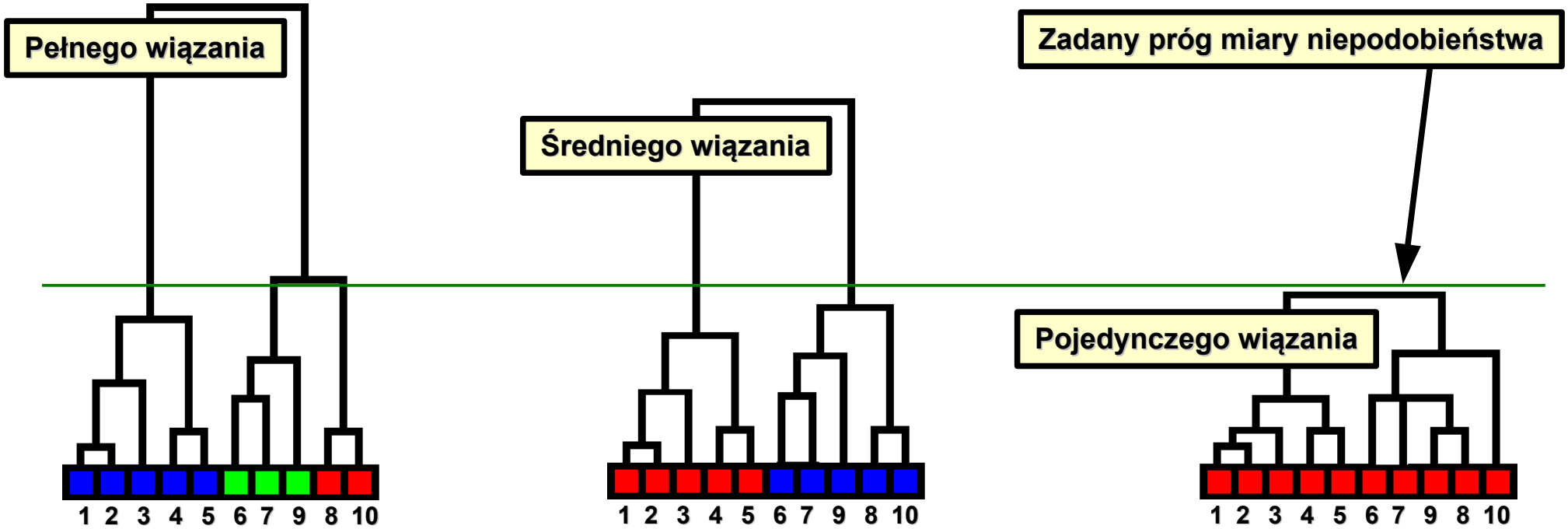
Klasteryzacja: porównanie algorytmów hierarchicznych





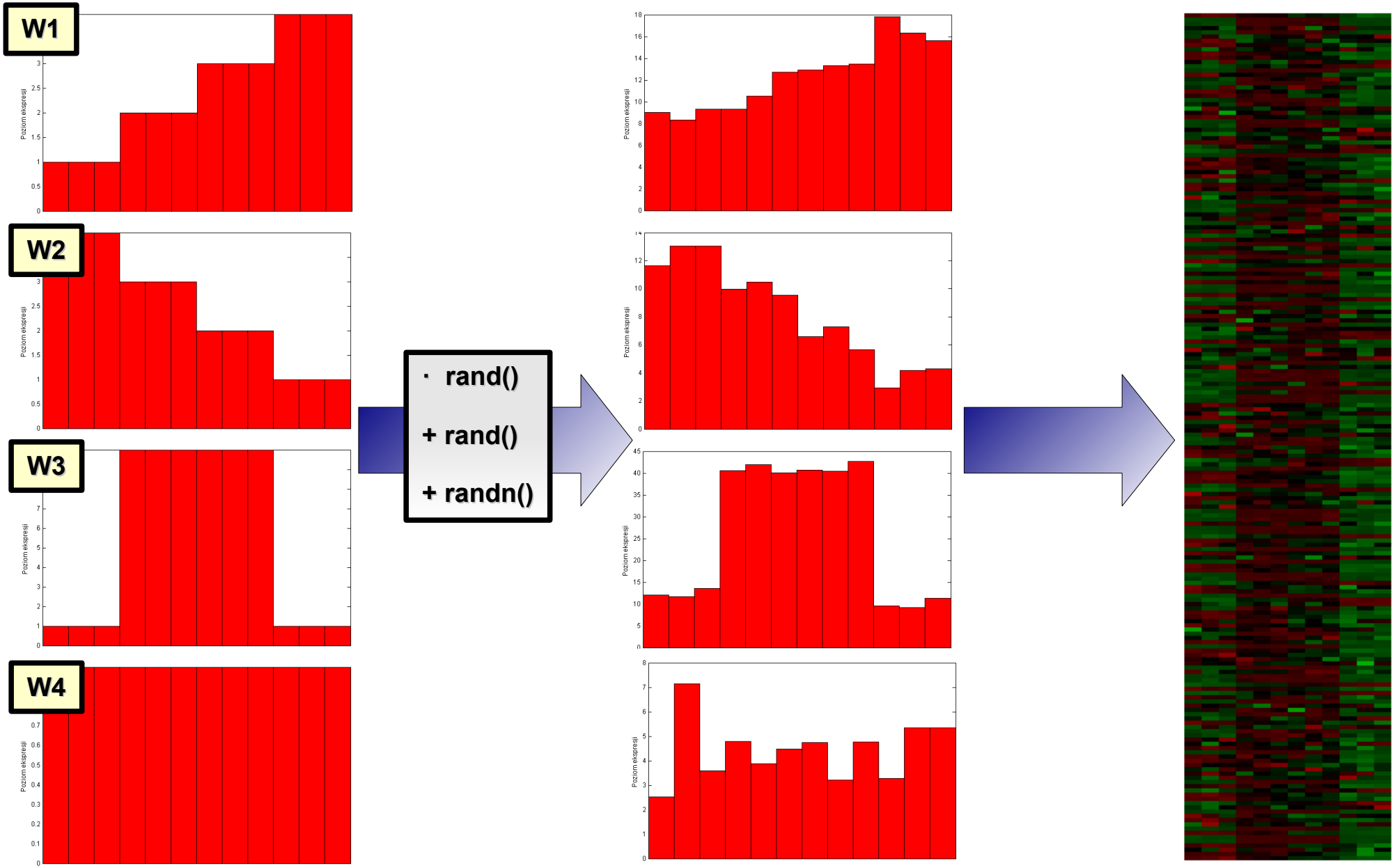
Klasteryzacja: porównanie algorytmów hierarchicznych





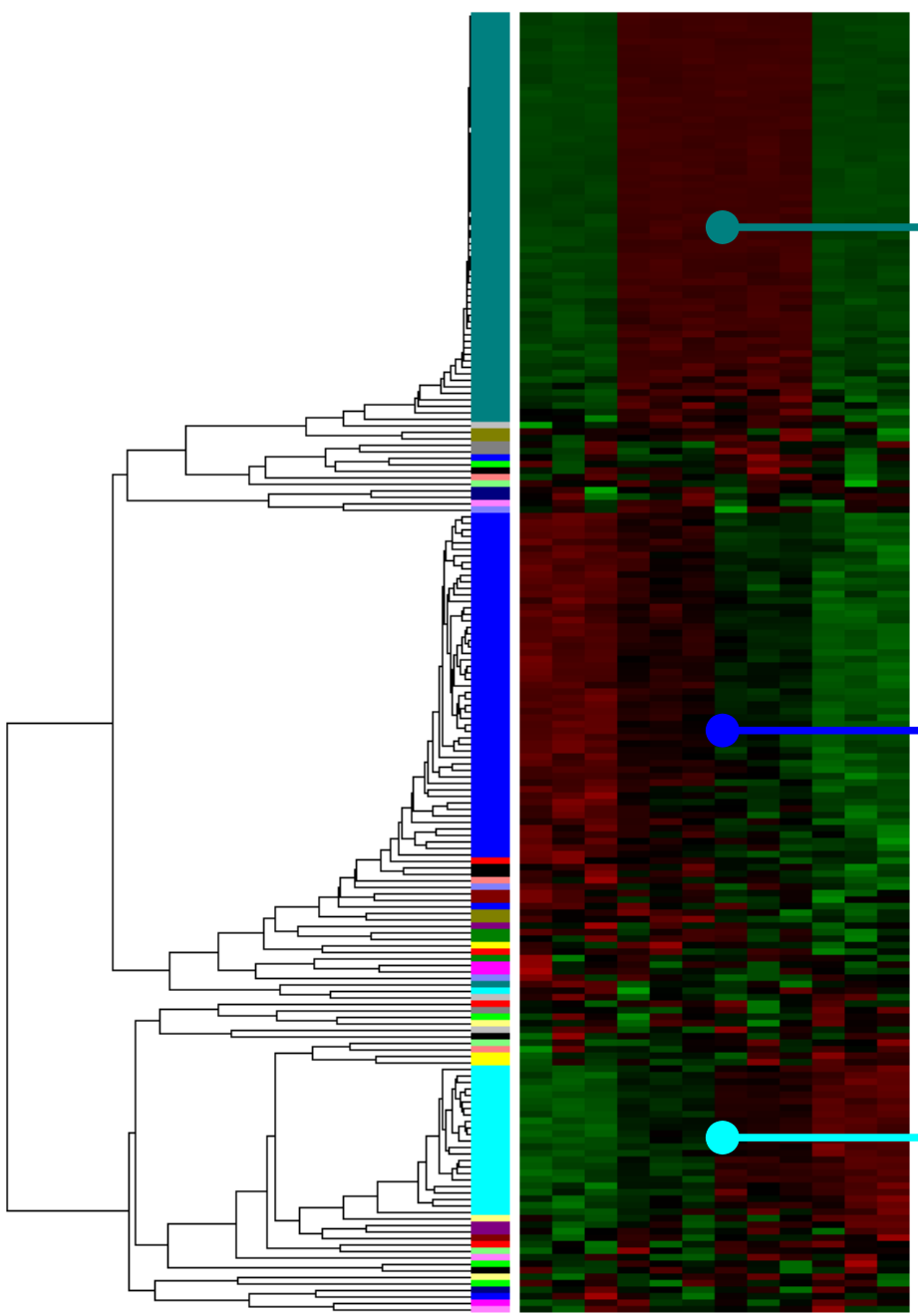
Klasteryzacja: algorytmy hierarchiczne (przykład działania)

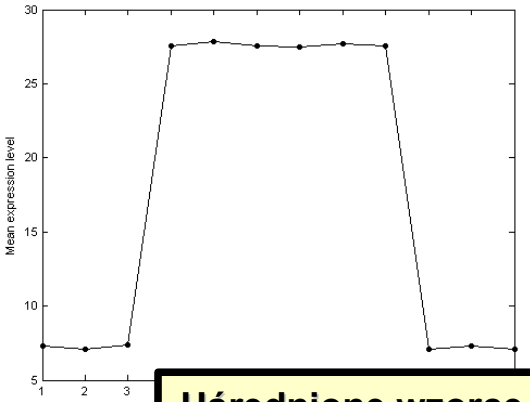
Przykład klasteryzacji UPGMA sztucznego zbioru danych złożonego z 200 „genów” o wartościach ekspresji wygenerowanych przez losowe przeskalowanie, dodanie składowej stałej oraz szumu Gaussowskiego do czterech wzorców:

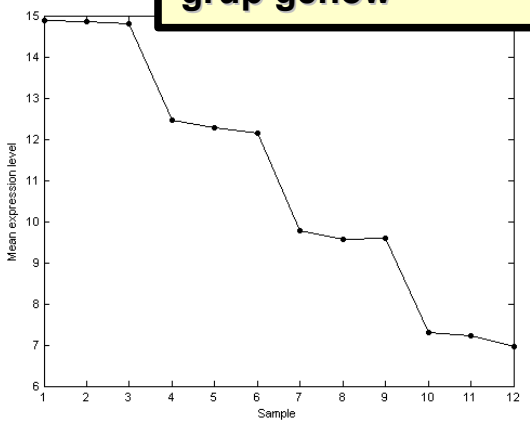


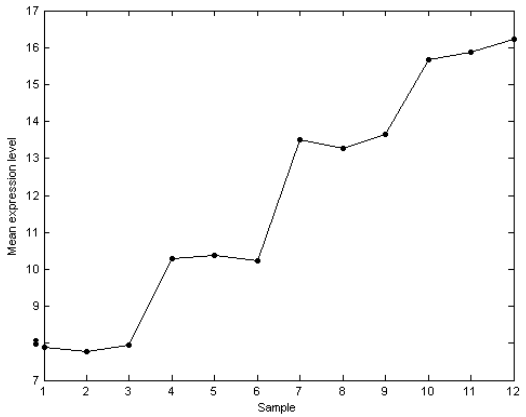
Klasteryzacja: algorytmy hierarchiczne (przykład działania)

Wynik klasteryzacji dla progu korelacyjnej miary niepodobieństwa równego 0.2**:**









Uśrednione wzorce ekspresji grup genów