

Metody probabilistyczne w uczeniu maszynowym

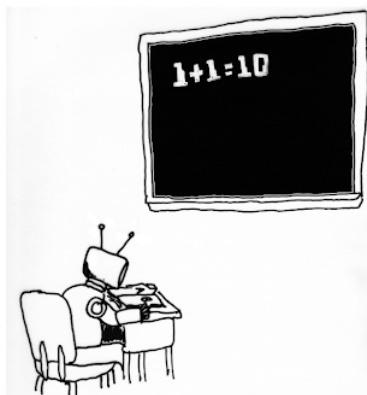
Wykład 11: uczenie nienadzorowane – grupowanie hierarchiczne,
algorytm k -średnich, klasteryzacja spektralna

Katarzyna Grygiel

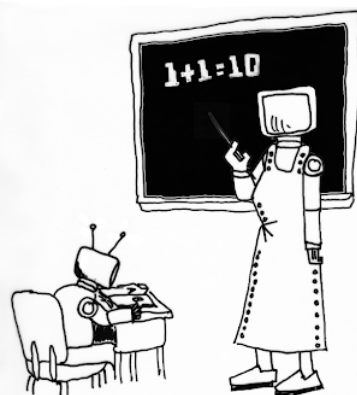
Semestr zimowy 2023/2024

Uczenie bez nadzoru

UNSUPERVISED MACHINE LEARNING



SUPERVISED MACHINE LEARNING



PROFFGADSWHIMMY.BLOGSPOT.CA

Uczenie bez nadzoru

W uczeniu nienadzorowanym opieramy się na zbiorze danych $\{x^{(i)} : i = 1, \dots, m\}$, które nie mają przypisanych etykiet.

Przykładowe zastosowania:

- grupowanie genów, rozpoznawanie różnych rodzajów tkanek, badania rynku, systemy rekomendacji, wykrywanie anomalii
→ **klasteryzacja**
(hierarchiczna, analiza skupień, probabilistyczna...)
- kompresja/wizualizacja danych, segmentacja obrazu
→ **redukcja wymiarów**
(analiza głównych składowych PCA, rozkład SVD...)
- tworzenie realistycznych obrazów, przekształcanie tekstu na obraz, usuwanie szumu ze zdjęć
→ **modele generatywne**
(autoenkodery wariacyjne, generatywne sieci przeciwstawne...)

Grupowanie hierarchiczne

Grupowanie hierarchiczne: algorytm

Dla danych $\{x^{(i)} : i = 1, \dots, m\}$ tworzymy macierz odległości D : $D_{ij} = d(x^{(i)}, x^{(j)})$.

Grupowanie hierarchiczne: algorytm

Dla danych $\{x^{(i)} : i = 1, \dots, m\}$ tworzymy macierz odległości D : $D_{ij} = d(x^{(i)}, x^{(j)})$.

Na początku każdy element tworzy osobny klaster.

W każdym kolejnym kroku sprawdzamy, które dwa klastry są sobie najbliższe według ustalonego kryterium i łączymy je ze sobą w jeden większy klaster.

Algorytm kończymy, gdy otrzymamy jeden klaster zawierający wszystkie dane.

Grupowanie hierarchiczne: algorytm

Dla danych $\{x^{(i)} : i = 1, \dots, m\}$ tworzymy macierz odległości D : $D_{ij} = d(x^{(i)}, x^{(j)})$.

Na początku każdy element tworzy osobny klaster.

W każdym kolejnym kroku sprawdzamy, które dwa klastry są sobie najbliższe według ustalonego kryterium i łączymy je ze sobą w jeden większy klaster.

Algorytm kończymy, gdy otrzymamy jeden klaster zawierający wszystkie dane.

Zauważmy, że nie musimy wstępnie definiować liczby klastrów.

Algorytm opiera się na podobieństwach między danymi. Ale jak mierzyć „podobieństwo”?

Musimy zdefiniować odległość oraz kryterium łączenia.

Grupowanie hierarchiczne: metryka i łączenie

Wybór metryki d :

- dla danych numerycznych: najczęściej euklidesowa (ale może być dowolna),
- dla danych nienumerycznych: na przykład odległość Hamminga lub edycyjna.

Grupowanie hierarchiczne: metryka i łączenie

Wybór metryki d :

- dla danych numerycznych: najczęściej euklidesowa (ale może być dowolna),
- dla danych nienumerycznych: na przykład odległość Hamminga lub edycyjna.

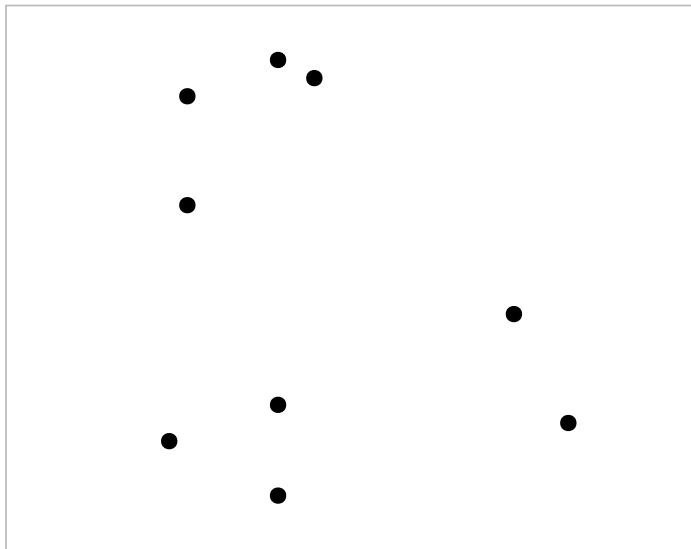
Możliwe kryteria wyboru klastrów do łączenia:

- pojedyncze: $\min\{d(a, b) : a \in A, b \in B\}$,
- pełne: $\max\{d(a, b) : a \in A, b \in B\}$,
- średnie: $\frac{1}{|A||B|} \sum_{a \in A} \sum_{b \in B} d(a, b)$,
- centroidalne: $d(c_A, c_B)$,
- Warda: $\sum_{a \in A \cup B} d(a, c_{A \cup B})^2$,

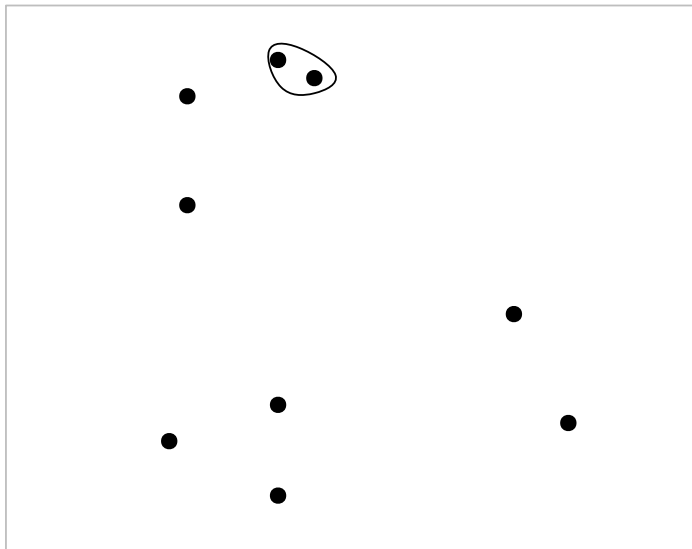
gdzie $c_{A \cup B}$ jest *centroidem* klastra $A \cup B$:

$$c_{A \cup B} = \min_{x \in \mathcal{X}} \sum_{a \in A \cup B} d(a, x)^2.$$

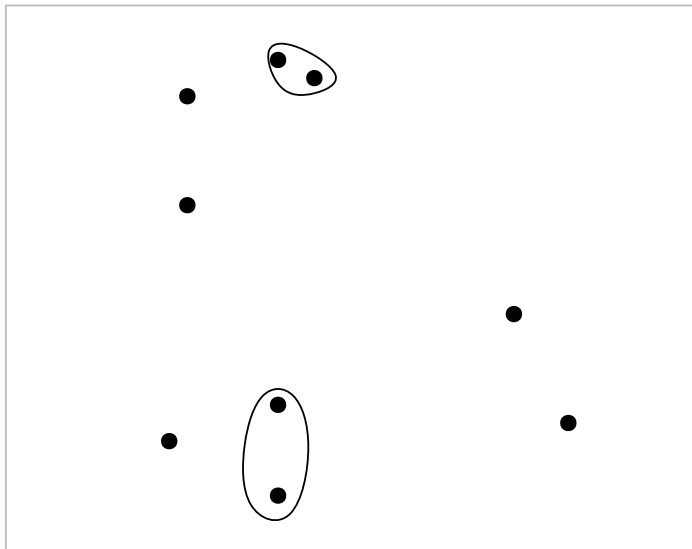
Grupowanie hierarchiczne: przykład



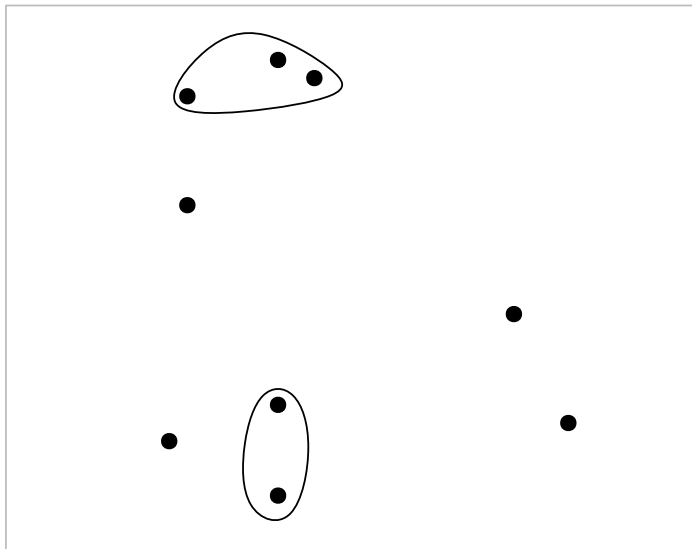
Grupowanie hierarchiczne: przykład



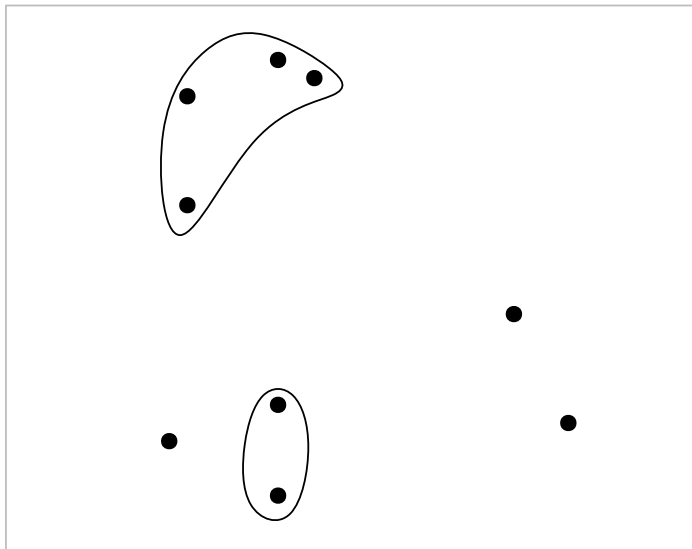
Grupowanie hierarchiczne: przykład



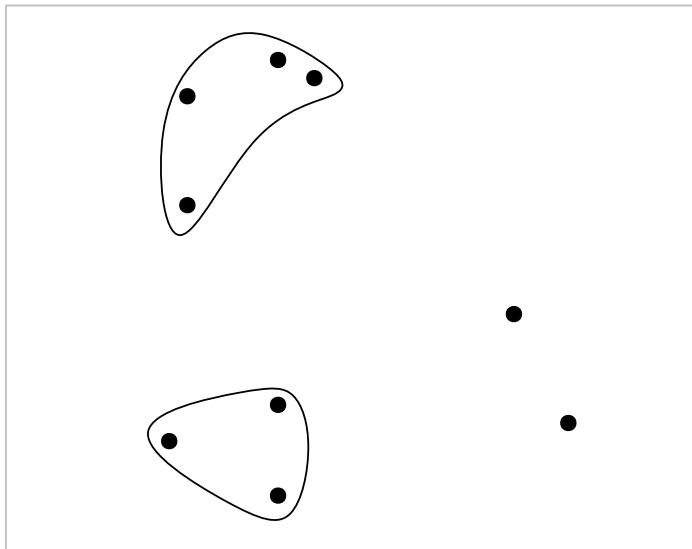
Grupowanie hierarchiczne: przykład



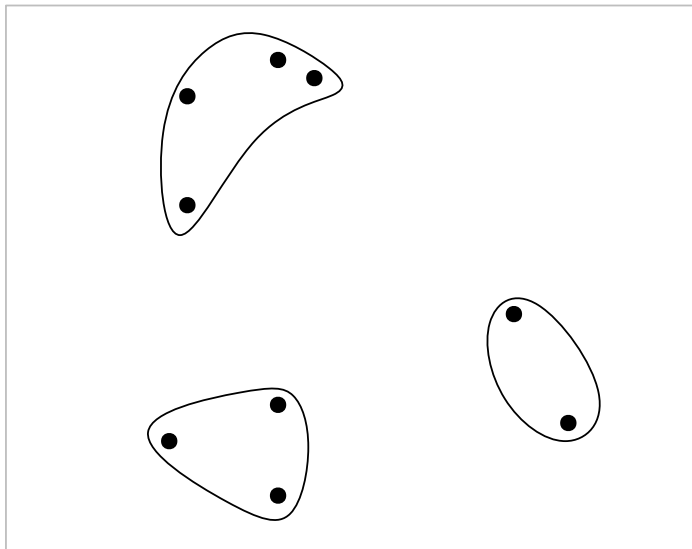
Grupowanie hierarchiczne: przykład



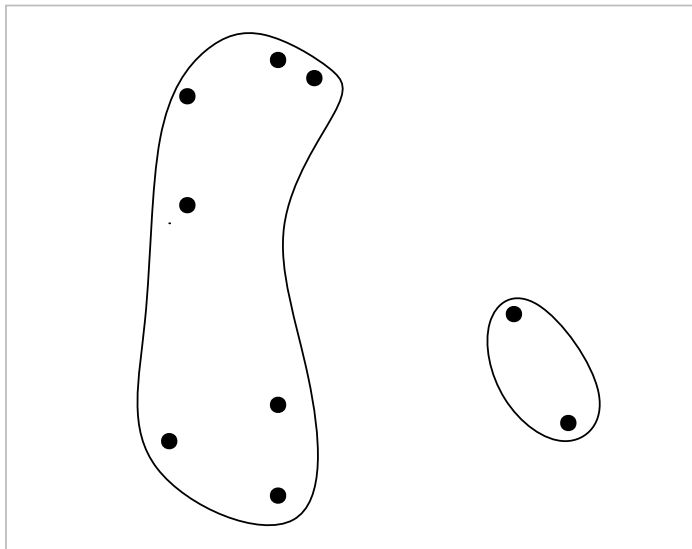
Grupowanie hierarchiczne: przykład



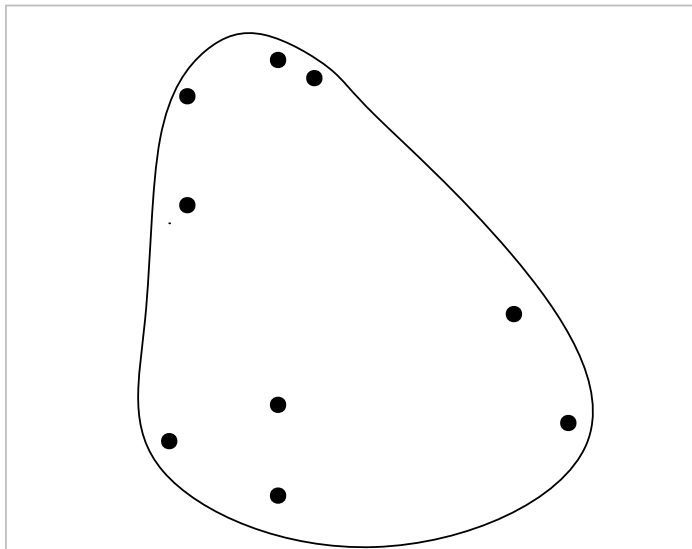
Grupowanie hierarchiczne: przykład



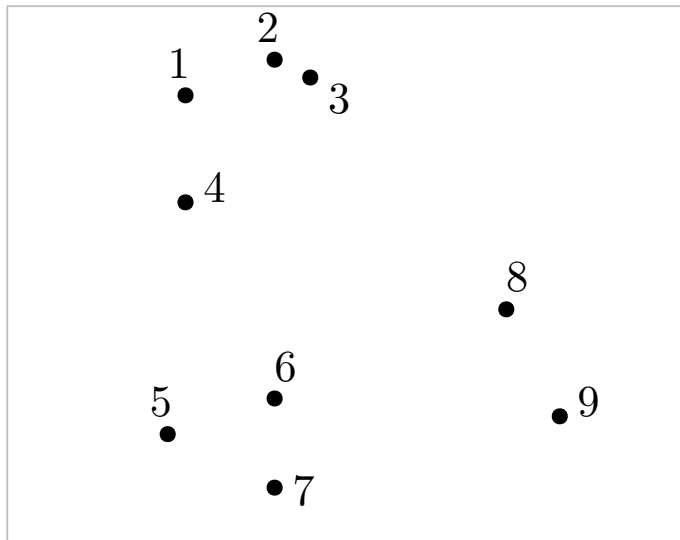
Grupowanie hierarchiczne: przykład



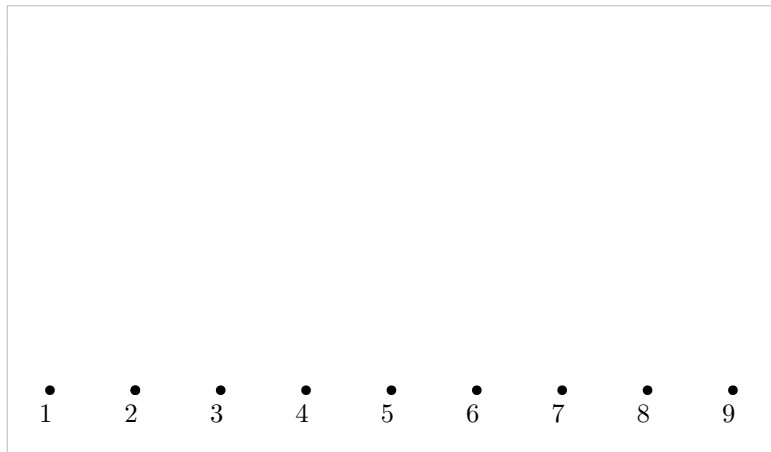
Grupowanie hierarchiczne: przykład



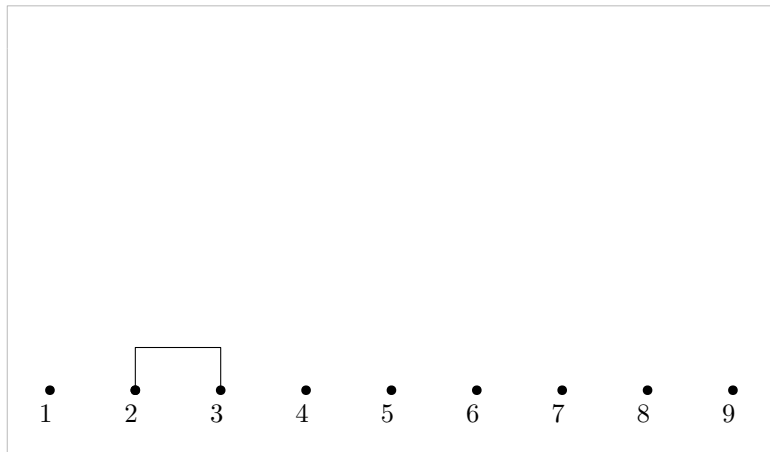
Grupowanie hierarchiczne: dendrogram



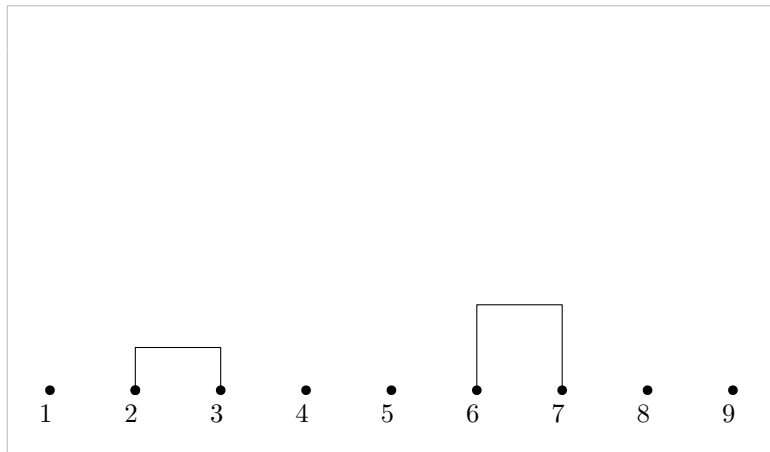
Grupowanie hierarchiczne: dendrogram



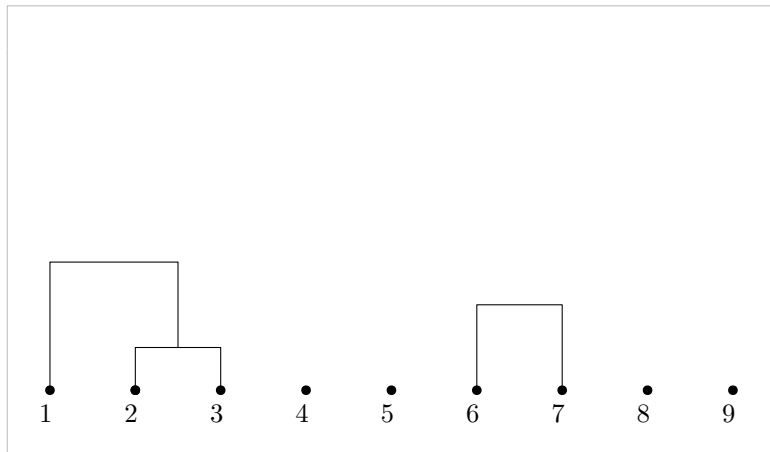
Grupowanie hierarchiczne: dendrogram



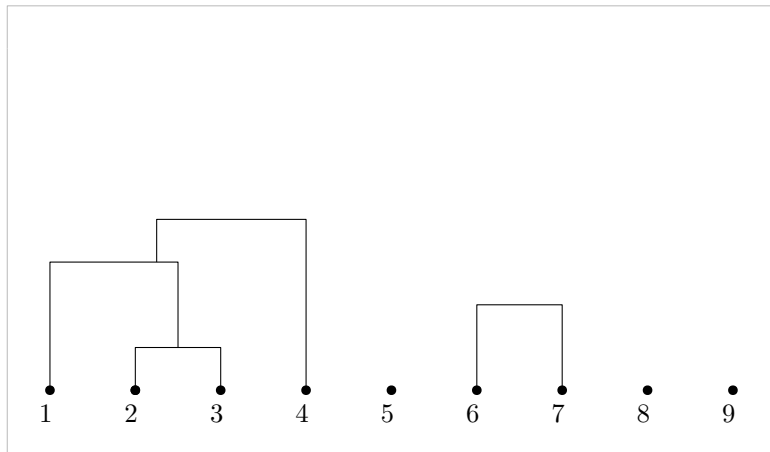
Grupowanie hierarchiczne: dendrogram



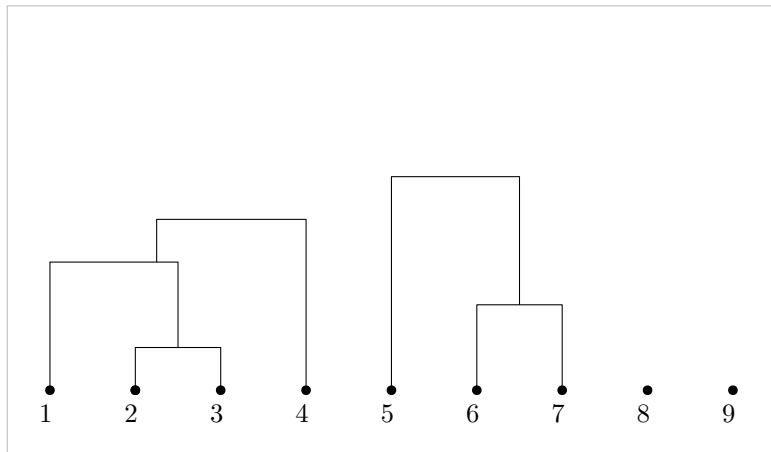
Grupowanie hierarchiczne: dendrogram



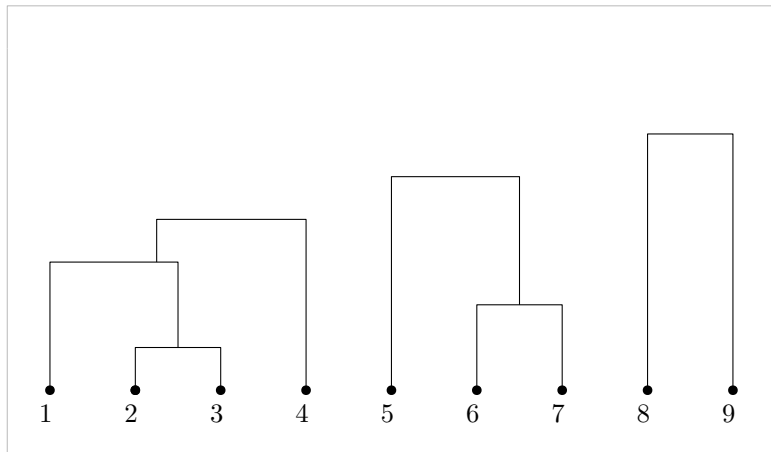
Grupowanie hierarchiczne: dendrogram



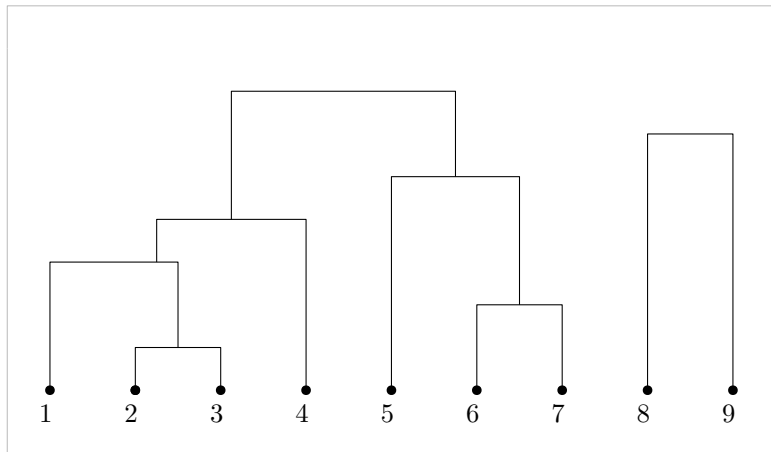
Grupowanie hierarchiczne: dendrogram



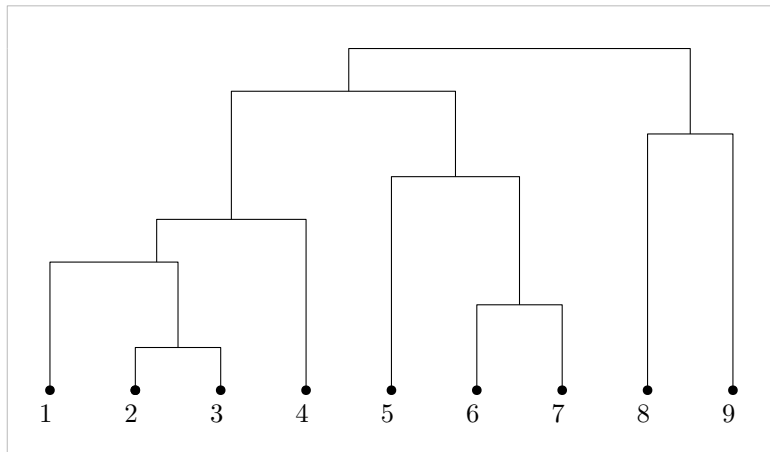
Grupowanie hierarchiczne: dendrogram



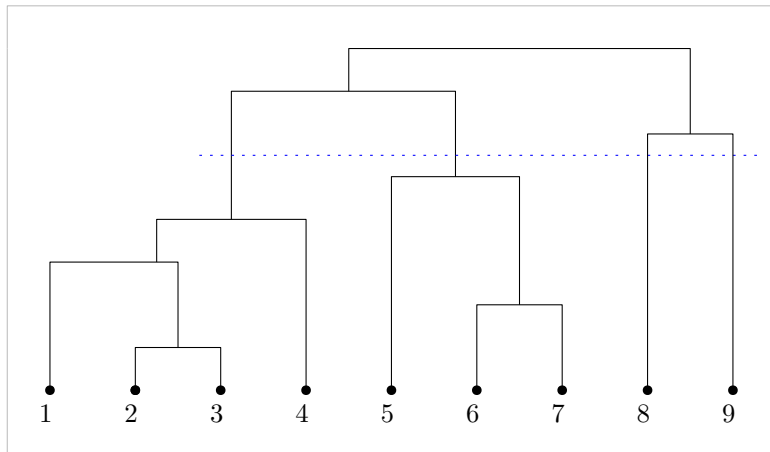
Grupowanie hierarchiczne: dendrogram



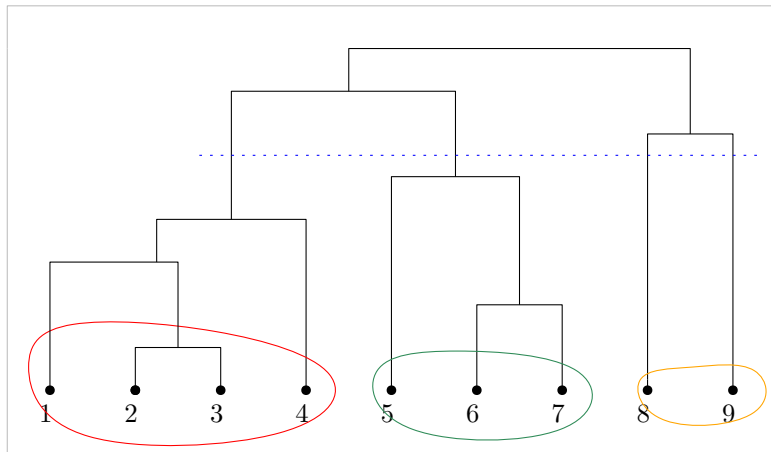
Grupowanie hierarchiczne: dendrogram



Grupowanie hierarchiczne: dendrogram



Grupowanie hierarchiczne: dendrogram



Którą metodę łączenia wybrać?

- Łączenie pojedyncze:
 - + radzi sobie z niekulistymi kształtami,
 - jest czułe na obserwacje odstające.
- Łączenie pełne:
 - + jest mniej czułe na obserwacje odstające niż pojedyncze,
 - ma tendencję do rozbijania dużych klas.
- Łączenie średnie:
 - + jest mniej podatne na obserwacje odstające niż pojedyncze,
 - ma tendencje kuliste.
- Łączenie centroidalne:
 - + sprawdza się dobrze w praktyce,
 - znacznie wolniejsze niż pojedyncze,
- Łączenie Warda:
 - + jest mniej czułe na obserwacje odstające niż pojedyncze,
 - ma tendencje kuliste.

Algorytm Lance'a-Williamsa

Na początku macierz D jest macierzą odległości pomiędzy klastrami jednoelementowymi, czyli $D_{ij} = d(x^{(i)}, x^{(j)})$.

Po każdym połączeniu dwóch klastrów A i B w jeden klaster C dodajemy wiersz i kolumnę odpowiadającą klastrowi C w ten sposób, że dla każdego klastra K zachodzi

$$D_{KC} = \alpha_A D_{KA} + \alpha_B D_{KB} + \beta D_{AB} + \gamma |D_{KA} - D_{KB}|.$$

Na końcu usuwamy wiersze i kolumny dla klastrów A i B .

Algorytm Lance'a-Williamsa

Na początku macierz D jest macierzą odległości pomiędzy klastrami jednoelementowymi, czyli $D_{ij} = d(x^{(i)}, x^{(j)})$.

Po każdym połączeniu dwóch klastrów A i B w jeden klaster C dodajemy wiersz i kolumnę odpowiadającą klastrowi C w ten sposób, że dla każdego klastra K zachodzi

$$D_{KC} = \alpha_A D_{KA} + \alpha_B D_{KB} + \beta D_{AB} + \gamma |D_{KA} - D_{KB}|.$$

Na końcu usuwamy wiersze i kolumny dla klastrów A i B .

metoda łączenia	α_A	α_B	β	γ
pojedyncze	$1/2$	$1/2$	0	$-1/2$
pełne	$1/2$	$1/2$	0	$1/2$
średnie	$\frac{ A }{ A + B }$	$\frac{ B }{ A + B }$	0	0
centroidalne	$\frac{ A }{ A + B }$	$\frac{ B }{ A + B }$	$-\frac{ A B }{(A + B)^2}$	0
Warda	$\frac{ A + C }{ A + B + C }$	$\frac{ B + C }{ A + B + C }$	$-\frac{ C }{ A + B + C }$	0

Wady i zalety grupowania hierarchicznego

Zalety:

- nie wymaga uprzedniego określenia liczby klastrów,
- algorytm jest stabilny: nie zależy od warunków początkowych,
- intuicyjny i prosty w implementacji.

Wady:

- ze względu na złożoność nie sprawdza się dla dużych zbiorów danych,
- nie działa dla danych z brakującymi wartościami,
- nie sprawdza się dobrze w przypadku danych z cechami o różnych typach.

Metoda k -średnich

Algorytm k -średnich (k -means)

W przypadku metody k -średnich dane $D = \{x^{(i)} : i = 1, \dots, m\}$ muszą być numeryczne. Najczęściej przestrzeń wejść to $\mathcal{X} = \mathbb{R}^n$.

Określamy metrykę d na \mathcal{X} (najczęściej euklidesową).

Ustalamy liczbę klastrów k (stąd k w nazwie algorytmu).

Naszym celem jest podział danych D na klastry C_1, \dots, C_k w ten sposób, aby zminimalizować funkcję błędu zdefiniowaną następująco:

$$\begin{aligned} \mathcal{J}_k(C_1, \dots, C_k) &= \sum_{j=1}^k \sum_{x \in C_j} d(x, \mu(C_j))^2 \\ &= \min_{\mu_1, \dots, \mu_k \in \mathcal{X}} \sum_{j=1}^k \sum_{x \in C_j} d(x, \mu_j)^2, \end{aligned}$$

gdzie $\mu_j = \mu(C_j)$ jest centroidem klastra C_j .

Algorytm k -średnich (k -means)

Wybieramy losowo k różnych punktów μ_1, \dots, μ_k ze zbioru D – są to początkowe centroidy.

Do momentu stabilizacji centroidów powtarzamy następującą operację: dla każdego $j = 1, \dots, k$ definiujemy

$$C_j := \{x \in D: j = \arg \min_{\ell} d(x, \mu_{\ell})\},$$

$$\mu_j := \arg \min_{\mu \in \mathcal{X}} \sum_{x \in C_j} d(x, \mu)^2.$$

Algorytm k -średnich (k -means)

Wybieramy losowo k różnych punktów μ_1, \dots, μ_k ze zbioru D – są to początkowe centroidy.

Do momentu stabilizacji centroidów powtarzamy następującą operację: dla każdego $j = 1, \dots, k$ definiujemy

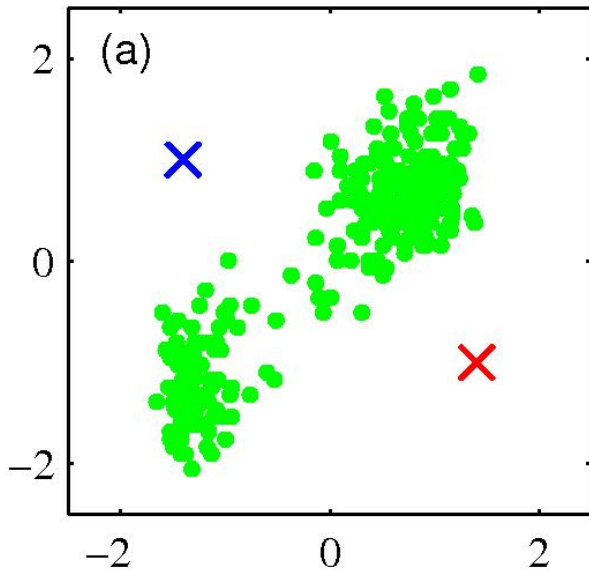
$$C_j := \{x \in D: j = \arg \min_{\ell} d(x, \mu_{\ell})\},$$

$$\mu_j := \arg \min_{\mu \in \mathcal{X}} \sum_{x \in C_j} d(x, \mu)^2.$$

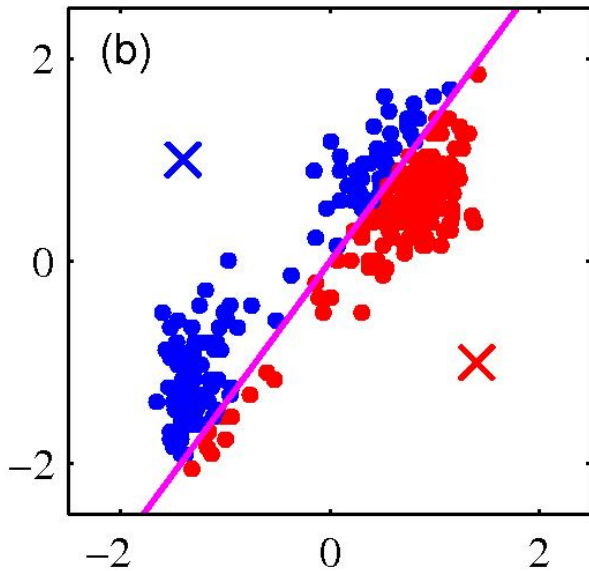
W przypadku remisów ustalamy jakąś strategię rozstrzygania (na przykład losowo).

W przypadku metryki euklidesowej nowy centroid jest wyznaczany jako średnia arytmetyczna ze wszystkich danych należących do danego klastra.

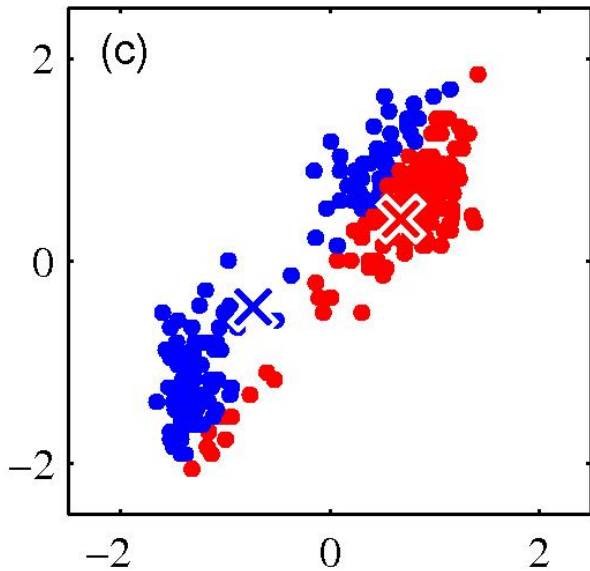
Przykład



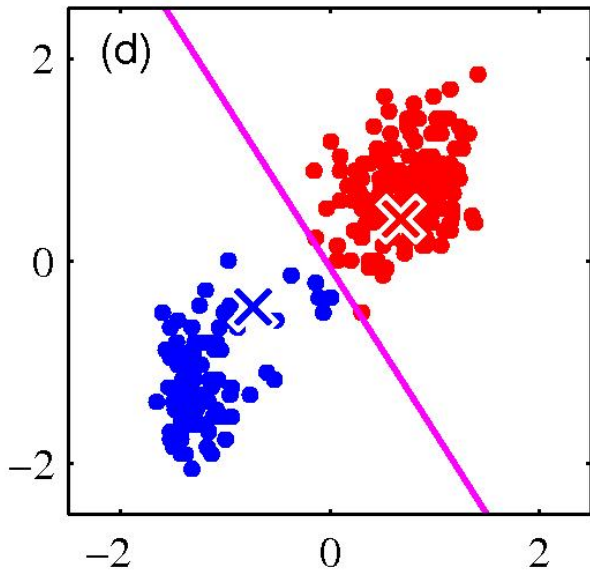
Przykład



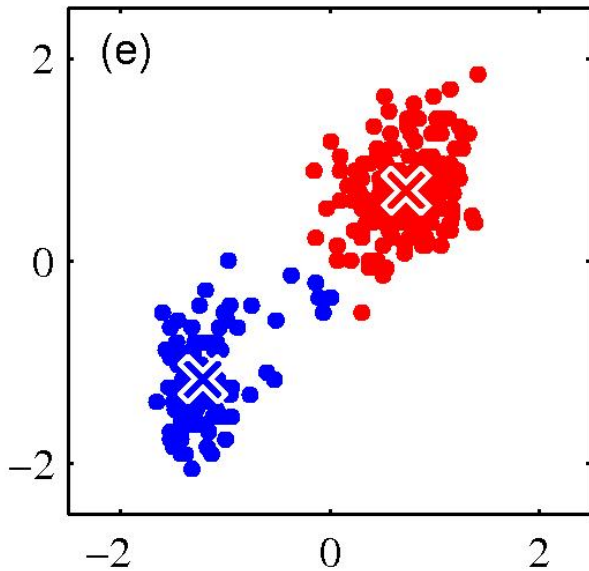
Przykład



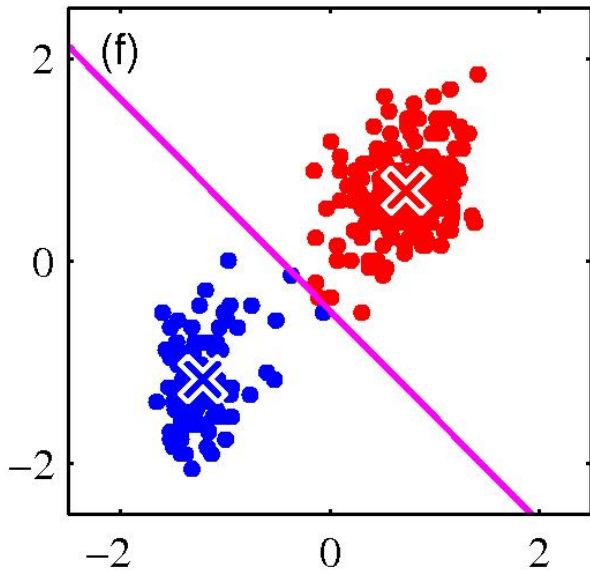
Przykład



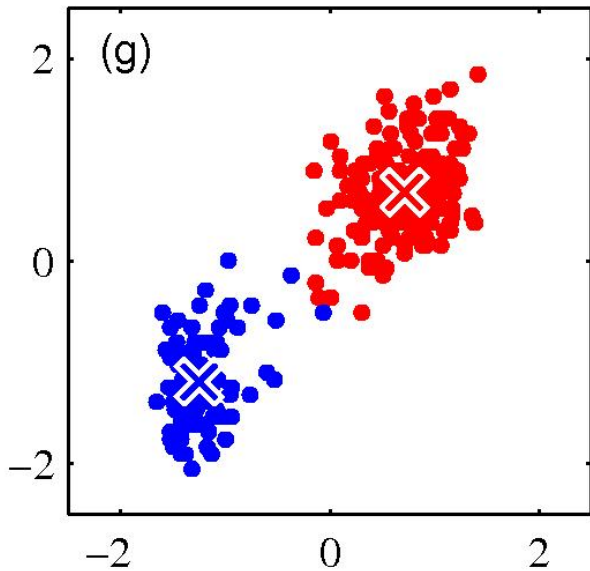
Przykład



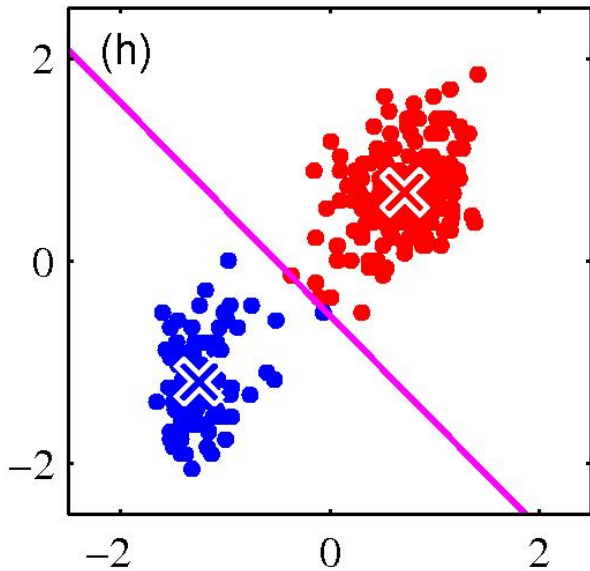
Przykład



Przykład



Przykład



Problemy z metodą k -średnich

Zależność od wartości początkowych



Problemy z metodą k -średnich

Zależność od wartości początkowych



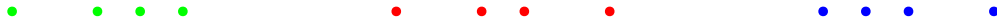
Problemy z metodą k -średnich

Zależność od wartości początkowych



Problemy z metodą k -średnich

Zależność od wartości początkowych



Przy odpowiednim wyborze początkowych centroidów otrzymaliśmy poprawny podział.

Problemy z metodą k -średnich

Zależność od wartości początkowych



Przy odpowiednim wyborze początkowych centroidów otrzymaliśmy poprawny podział.



Problemy z metodą k -średnich

Zależność od wartości początkowych



Przy odpowiednim wyborze początkowych centroidów otrzymaliśmy poprawny podział.



Problemy z metodą k -średnich

Zależność od wartości początkowych



Przy odpowiednim wyborze początkowych centroidów otrzymaliśmy poprawny podział.



Problemy z metodą k -średnich

Zależność od wartości początkowych



Przy odpowiednim wyborze początkowych centroidów otrzymaliśmy poprawny podział.



W powyższym przypadku wybór centroidów doprowadził do złego podziału.

Problemy z metodą k -średnich

Zależność od wartości początkowych



Przy odpowiednim wyborze początkowych centroidów otrzymaliśmy poprawny podział.

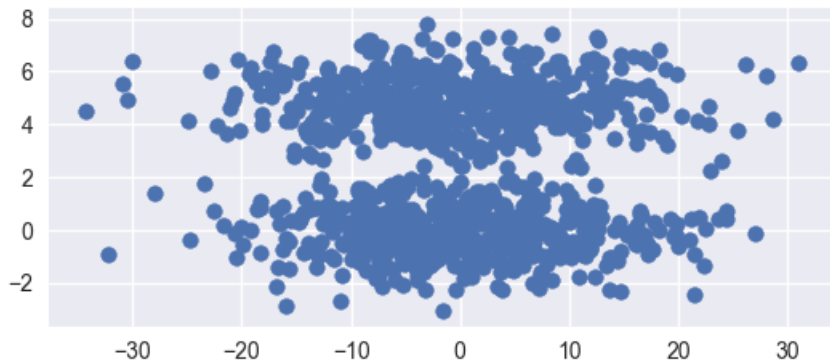


W powyższym przypadku wybór centroidów doprowadził do złego podziału.

Algorytm k -średnich należy **uruchomić wielokrotnie** dla losowo wybieranych początkowych centroidów i wybrać najlepszą klasteryzację.

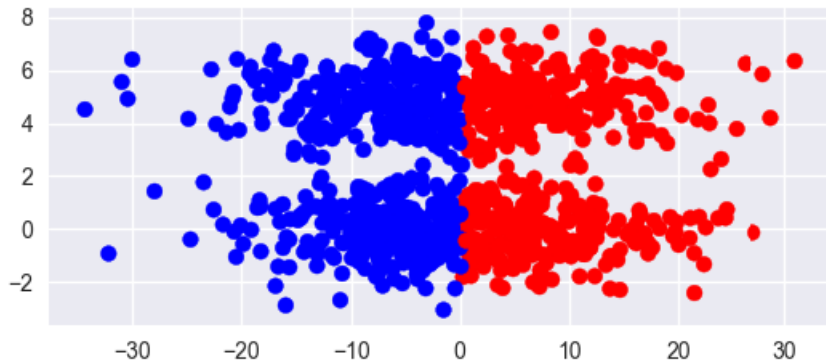
Problemy z metodą k -średnich

Dane w różnej skali



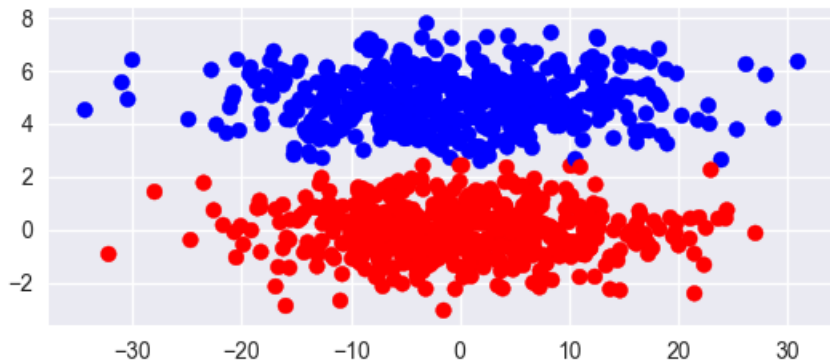
Problemy z metodą k -średnich

Dane w różnej skali



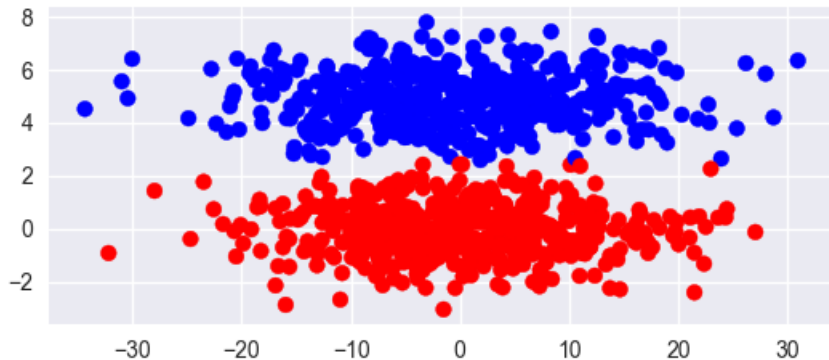
Problemy z metodą k -średnich

Dane w różnej skali



Problemy z metodą k -średnich

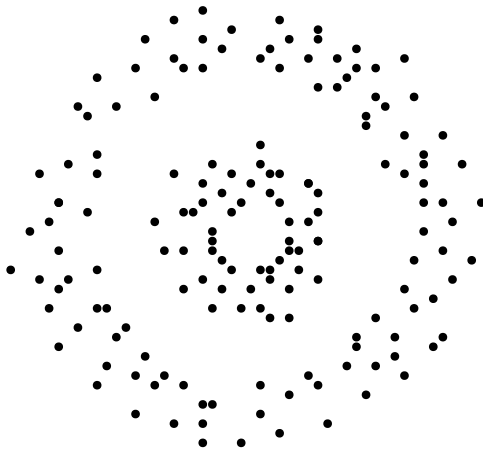
Dane w różnej skali



Należy zatem **skalować dane**.

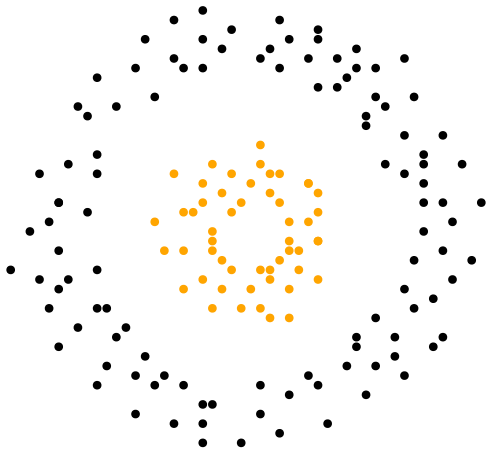
Problemy z metodą k -średnich

Klastry niekuliste



Problemy z metodą k -średnich

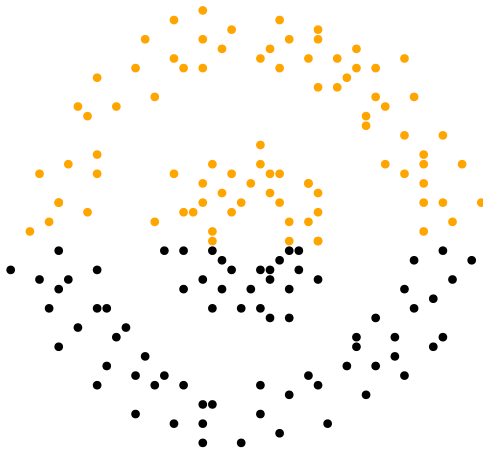
Klastry niekuliste



Chcielibyśmy uzyskać taki podział na klastry.

Problemy z metodą k -średnich

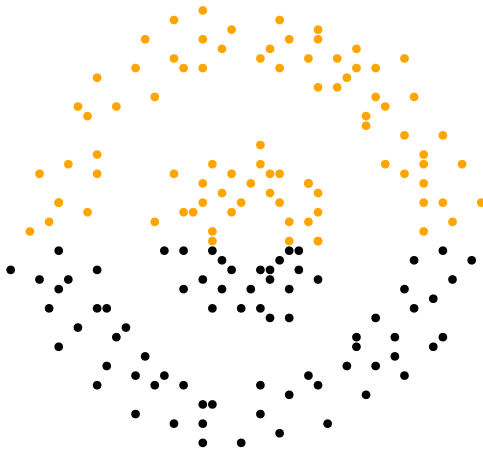
Klasy niekuliste



Algorytm k -średnich zwraca jednak taki podział.

Problemy z metodą k -średnich

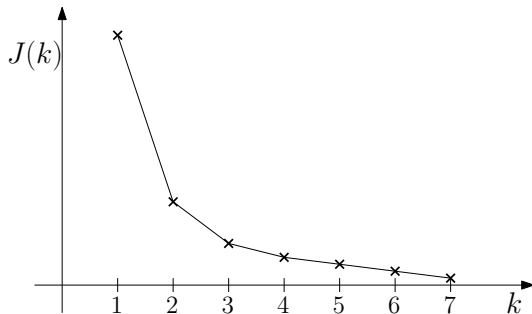
Klasy niekuliste



Algorytm k -średnich zwraca jednak taki podział. Możemy zastosować **funkcje jądrowe**.

Ile klastrow? Metoda łokcia

Dla różnej liczby klastrow k wyznaczamy błąd klasteryzacji $J(k)$. Na podstawie wykresu wybieramy taką wartość k , dla której krzywa ma najbardziej wyraźne „przełamanie”.



W tym przypadku wybór dwóch lub trzech klastrow wydaje się rozsądny. Z wykresu nie zawsze wynika, która wartość k jest najlepsza. Metodą łokcia niekoniecznie otrzymamy dobry wynik, dlatego należy podchodzić do niej z dystansem.

Inicjalizacja centroidów algorytmem *k-means++*

W żadnej iteracji algorytmu *k*-średnich funkcja błędu $J_k(C_1, \dots, C_k)$ nie rośnie. Nie mamy jednak żadnej pewności, że osiągniemy minimum globalne – wszystko zależy od wyboru początkowych centroidów.

Inicjalizacja centroidów algorytmem *k-means++*

W żadnej iteracji algorytmu *k*-średnich funkcja błędu $J_k(C_1, \dots, C_k)$ nie rośnie. Nie mamy jednak żadnej pewności, że osiągniemy minimum globalne – wszystko zależy od wyboru początkowych centroidów.

Algorytm *k-means++* zmienia sposób wyboru centroidów:

- Pierwszy centroid wybieramy losowo.
- Po wybraniu $n < k$ centroidów $C = \{c_1, \dots, c_n\}$ obliczamy odległość każdego z pozostałych punktów $x \in D \setminus C$ od zbioru C . Kolejny centroid wybieramy ze zbioru $D \setminus C$ losowo, przy czym prawdopodobieństwo wyboru punktu x_0 wynosi

$$\frac{d(x_0, C)^2}{\sum_{x \in D \setminus C} d(x, C)^2}.$$

Inicjalizacja centroidów algorytmem *k-means++*

W żadnej iteracji algorytmu *k*-średnich funkcja błędu $J_k(C_1, \dots, C_k)$ nie rośnie. Nie mamy jednak żadnej pewności, że osiągniemy minimum globalne – wszystko zależy od wyboru początkowych centroidów.

Algorytm *k-means++* zmienia sposób wyboru centroidów:

- Pierwszy centroid wybieramy losowo.
- Po wybraniu $n < k$ centroidów $C = \{c_1, \dots, c_n\}$ obliczamy odległość każdego z pozostałych punktów $x \in D \setminus C$ od zbioru C . Kolejny centroid wybieramy ze zbioru $D \setminus C$ losowo, przy czym prawdopodobieństwo wyboru punktu x_0 wynosi

$$\frac{d(x_0, C)^2}{\sum_{x \in D \setminus C} d(x, C)^2}.$$

Niech J_k^* oznacza minimalny błąd dla klasteryzacji otrzymanej metodą *k*-średnich. Przy powyższej inicjalizacji zachodzi

$$\mathbb{E}[J_k(C_1, \dots, C_k)] \leq 8(\log_2 k + 2)J_k^*.$$

Wady i zalety metody k -średnich

Zalety:

- prosty intuicyjnie i w implementacji,
- dobrze się sprawdza dla dużych zbiorów danych, szczególnie w przypadku klastrów kulistych,
- jest liniowy ze względu na liczbę obserwacji,
- zwraca zwarte klastry.

Wady i zalety metody k -średnich

Zalety:

- prosty intuicyjnie i w implementacji,
- dobrze się sprawdza dla dużych zbiorów danych, szczególnie w przypadku klastrów kulistych,
- jest liniowy ze względu na liczbę obserwacji,
- zwraca zwarte klastry.

Wady:

- jest bardzo zależny od początkowego wyboru centroidów, w związku z czym wymaga wielu przebiegów,
- jest czuły na zakres zmiennych, wymaga skalowania,
- działa tylko dla danych numerycznych,
- nie radzi sobie z klastrami o mało kulistych kształtach.

Klasteryzacja spektralna

Idea klasteryzacji spektralnej

Algorytm klasteryzacji spektralnej opiera się na grafie podobieństwa utworzonym dla danych wejściowych.

Dane te zostają zanurzone w przestrzeni o mniejszym wymiarze w taki sposób, że klastry są łatwiej wykrywalne. W tym celu wykorzystuje się wektory własne laplasjanu grafu podobieństwa.

Dla nowej reprezentacji danych stosuje się np. algorytm k -średnich.

Idea klasteryzacji spektralnej

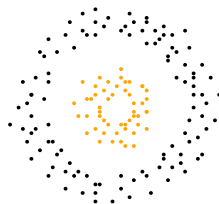
Algorytm klasteryzacji spektralnej opiera się na grafie podobieństwa utworzonym dla danych wejściowych.

Dane te zostają zanurzone w przestrzeni o mniejszym wymiarze w taki sposób, że klastry są łatwiej wykrywalne. W tym celu wykorzystuje się wektory własne laplasjanu grafu podobieństwa.

Dla nowej reprezentacji danych stosuje się np. algorytm k -średnich.



metoda k -średnich



klasteryzacja spektralna

Graf podobieństwa

Dla zaobserwowanych danych $\{x^{(i)} : i = 1, \dots, m\}$ tworzymy graf i odpowiadającą mu macierz sąsiedztwa.

Jak zbudować graf? Jak rozumieć sąsiedztwo?

Graf podobieństwa

Dla zaobserwowanych danych $\{x^{(i)} : i = 1, \dots, m\}$ tworzymy graf i odpowiadającą mu macierz sąsiedztwa.

Jak zbudować graf? Jak rozumieć sąsiedztwo?

Przykładowe możliwe podejścia:

- otoczka epsilonowa: dla ustalonej metryki d oraz $\varepsilon > 0$ łączymy te punkty, które są oddalone od siebie o mniej niż ε (graf nieskierowany, najczęściej bez wag);

Graf podobieństwa

Dla zaobserwowanych danych $\{x^{(i)} : i = 1, \dots, m\}$ tworzymy graf i odpowiadającą mu macierz sąsiedztwa.

Jak zbudować graf? Jak rozumieć sąsiedztwo?

Przykładowe możliwe podejścia:

- otoczka epsilonowa: dla ustalonej metryki d oraz $\varepsilon > 0$ łączymy te punkty, które są oddalone od siebie o mniej niż ε (graf nieskierowany, najczęściej bez wag);
- k -najbliższych sąsiadów: dla danego punktu łączymy go z k najbliższymi sąsiadami (najczęściej graf nieskierowany po usunięciu kierunków, z wagami);

Graf podobieństwa

Dla zaobserwowanych danych $\{x^{(i)} : i = 1, \dots, m\}$ tworzymy graf i odpowiadającą mu macierz sąsiedztwa.

Jak zbudować graf? Jak rozumieć sąsiedztwo?

Przykładowe możliwe podejścia:

- otoczka epsilonowa: dla ustalonej metryki d oraz $\varepsilon > 0$ łączymy te punkty, które są oddalone od siebie o mniej niż ε (graf nieskierowany, najczęściej bez wag);
- k -najbliższych sąsiadów: dla danego punktu łączymy go z k najbliższymi sąsiadami (najczęściej graf nieskierowany po usunięciu kierunków, z wagami);
- graf pełny z wagami.

Wagi są proporcjonalnie do podobieństwa między punktami. Często jako funkcję podobieństwa wybieramy jądro gaussowskie lub odwrotność odległości.

Laplasjan grafu

Dla macierzy sąsiedztwa grafu $G = (V, E)$:

$$A_{ij} = \begin{cases} w_{ij}, & \text{gdy } \{i, j\} \in E, \\ 0, & \text{gdy } \{i, j\} \notin E \end{cases}$$

oraz diagonalnej macierzy D :

$$D_{ij} = \begin{cases} \sum_{j: \{i, j\} \in E} w_{ij}, & \text{gdy } i = j, \\ 0, & \text{gdy } i \neq j \end{cases}$$

tworzymy *laplasjan* grafu G , czyli macierz $L = D - A$.

Laplasjan grafu

Dla macierzy sąsiedztwa grafu $G = (V, E)$:

$$A_{ij} = \begin{cases} w_{ij}, & \text{gdy } \{i, j\} \in E, \\ 0, & \text{gdy } \{i, j\} \notin E \end{cases}$$

oraz diagonalnej macierzy D :

$$D_{ij} = \begin{cases} \sum_{j: \{i,j\} \in E} w_{ij}, & \text{gdy } i = j, \\ 0, & \text{gdy } i \neq j \end{cases}$$

tworzymy *laplasjan* grafu G , czyli macierz $L = D - A$.

Macierz L jest symetryczna i dodatnio półokreślona. Zatem wszystkie wartości własne L są nieujemne.

Wektor Fiedlera i dwa klastry

Wyznaczamy wartości i wektory własne macierzy L . Porządkujemy wartości własne rosnąco:

$$\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_m.$$

Odpowiadające im wektory własne oznaczamy przez v_1, \dots, v_m .

Łatwo można pokazać, że $\lambda_1 = 0$. Ponadto jeśli dokładnie n wartości $\lambda_1, \dots, \lambda_n$ jest zerowych, to graf ma n składowych spójnych.

Wektor Fiedlera i dwa klastry

Wyznaczamy wartości i wektory własne macierzy L . Porządkujemy wartości własne rosnąco:

$$\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_m.$$

Odpowiadające im wektory własne oznaczamy przez v_1, \dots, v_m .

Łatwo można pokazać, że $\lambda_1 = 0$. Ponadto jeśli dokładnie n wartości $\lambda_1, \dots, \lambda_n$ jest zerowych, to graf ma n składowych spójnych.

Wartość λ_2 opisuje *algebraiczną spójność* grafu. Im jest większa, tym bardziej spójny jest graf.

Wektor Fiedlera i dwa klastry

Wyznaczamy wartości i wektory własne macierzy L . Porządkujemy wartości własne rosnąco:

$$\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_m.$$

Odpowiadające im wektory własne oznaczamy przez v_1, \dots, v_m .

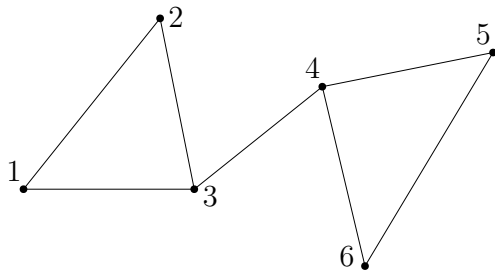
Łatwo można pokazać, że $\lambda_1 = 0$. Ponadto jeśli dokładnie n wartości $\lambda_1, \dots, \lambda_n$ jest zerowych, to graf ma n składowych spójnych.

Wartość λ_2 opisuje *algebraiczną spójność* grafu. Im jest większa, tym bardziej spójny jest graf.

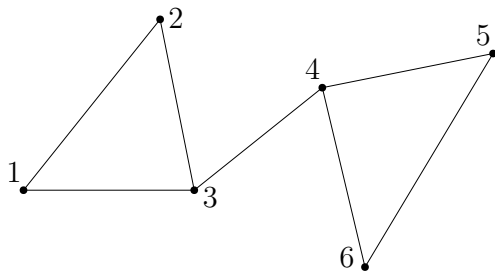
Odpowiadający jej wektor własny v_2 to tzw. *wektor Fiedlera*. Gdy $\lambda_2 \neq 0$, to na jego podstawie dzielimy graf na dwa klastry C_1 i C_2 . Dla każdego $i = 1, \dots, m$ rozpatrujemy i -tą wartość v_2 :

- jeżeli $(v_2)_i > 0$, to $x^{(i)} \in C_1$,
- jeżeli $(v_2)_i < 0$, to $x^{(i)} \in C_2$.

Podział grafu na dwa klastry: przykład

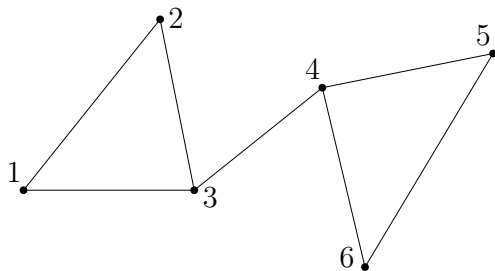


Podział grafu na dwa klastry: przykład



$$L = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix}$$

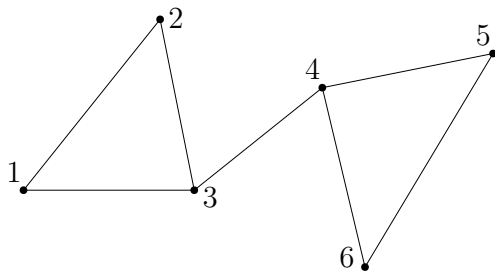
Podział grafu na dwa klastry: przykład



$$L = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix}$$

$$\lambda_2 = \frac{5 - \sqrt{17}}{2}$$

Podział grafu na dwa klastry: przykład

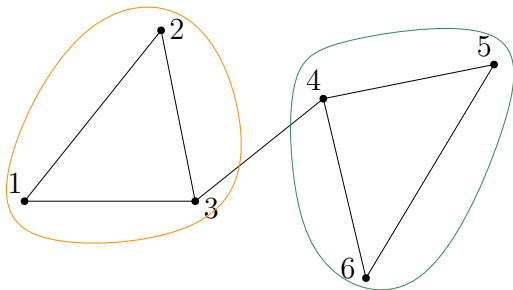


$$L = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix}$$

$$\lambda_2 = \frac{5 - \sqrt{17}}{2}$$

$$v_2 = \begin{bmatrix} -1 \\ -1 \\ -0.56 \\ 0.56 \\ 1 \\ 1 \end{bmatrix}$$

Podział grafu na dwa klastry: przykład



$$L = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix}$$

$$\lambda_2 = \frac{5 - \sqrt{17}}{2}$$

$$v_2 = \begin{bmatrix} -1 \\ -1 \\ -0.56 \\ 0.56 \\ 1 \\ 1 \end{bmatrix}$$

Więcej klastrów

Jak rozszerzyć powyższy pomysł na k -klasteryzację?

Normalizujemy laplasjan, czyli wyznaczamy macierz

$$L_{\text{norm}} = D^{-1/2} L D^{-1/2} \quad \text{lub} \quad L_{\text{norm}} = D^{-1/2} L.$$

Dla macierzy L_{norm} wyznaczamy pierwsze (według porządku na wartościach własnych) k wektorów własnych v_1, \dots, v_k .

Tworzymy macierz $U \in \mathbb{R}^{m \times k}$ zawierającą jako kolumny wektory v_1, \dots, v_k .

Danej $x^{(i)}$ odpowiada teraz i -ty wiersz macierzy U . Dla takiej nowej reprezentacji stosujemy dowolny algorytm klasteryzacji (na przykład metodę k -średnich).

Wady i zalety klasteryzacji spektralnej

Zalety:

- klastry mogą mieć różnorodne kształty,
- nie trzeba przechowywać wszystkich danych, wystarczy macierz sąsiedztwa lub sam laplasjan,
- ma solidne podstawy teoretyczne.

Wady i zalety klasteryzacji spektralnej

Zalety:

- klastry mogą mieć różnorodne kształty,
- nie trzeba przechowywać wszystkich danych, wystarczy macierz sąsiedztwa lub sam laplasjan,
- ma solidne podstawy teoretyczne.

Wady:

- wymaga wyboru liczby klastrów na początku,
- może być kosztowna obliczeniowo dla dużych zbiorów danych.