Miary odległości i klasteryzacja

Paweł Gliwny

Miary odległości

Wiele metod eksploracji danych opiera się na miarach **podobieństwa** (lub **odległości**) między obiektami.

Sposoby uzyskania miar podobieństwa:

Bezpośrednio – ocena ekspercka lub badanie (np. ankieta marketingowa o podobieństwie produktów).

Pośrednio – obliczenie z wektorów cech opisujących obiekty (wymaga zdefiniowania metryki).

Typowe metryki

Typowe metryki wykorzystywane w grupowaniu to m.in.

- **Euklidesowa** – odległość w linii prostej, $d = \sqrt{\sum (x_i - y_i)^2}$
- **Manhattan** – suma różnic wzdłuż osi, $d = \sum |x_i - y_i|$
- **Minkowskiego** – uogólnienie powyższych (parametr p)
- **Cosinusowa** – kąt między wektorami (dla danych tekstowych)

Odległość euklidesowa

Koncepcja wywodząca się ze starożytnej greckiej matematyki, dziś narzędzie w nauce o danych i uczeniu maszynowym. Mierzy odległość w linii prostej między punktami – najkrótszą ścieżkę w przestrzeni euklidesowej.

Odległość w 2D:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Odległość w n wymiarach:

$$d = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

gdzie p i q są wektorami punktów w przestrzeni n-wymiarowej.

Zastosowania:

KNN – znajdowanie najbliższych sąsiadów (np. rozpoznawanie spamu, rekomendacje)

K-means – przypisywanie punktów do centrów klastrów (np. segmentacja klientów)

Ograniczenia odległości euklidesowej

Wrażliwość na skalę cech:

- Cechy o dużych wartościach dominują w obliczeniach odległości.
- Przykład: w zbiorze danych z dochodem (tysiące PLN) i wiekiem (0-100), dochód będzie miał nieproporcjonalnie duży wpływ.

Rozwiązanie: Normalizacja lub standaryzacja danych przed obliczeniem odległości.

Przykład w Pythonie:

```
1 from sklearn.preprocessing import StandardScaler
2 scaler = StandardScaler()
3 X_scaled = scaler.fit_transform(X)
4
```

Odległość Manhattan

- Suma bezwzględnych różnic wzdłuż każdej osi – jak poruszanie się po ulicach w układzie siatki miasta (stąd nazwa ódległość miejskich bloków").
- Przydatna w danych wielowymiarowych, gdzie odległość euklidesowa bywa zawodna.

Odległość Manhattan (metryka l1)

Definicja: Odległość Manhattan, znana także jako *odległość taksówkowa* lub *metryka l1*, mierzy odległość między dwoma punktami, poruszając się jedynie wzdłuż osi współrzędnych.

Wzór w przestrzeni dwuwymiarowej:

$$d_{\text{Manhattan}}(A, B) = |x_2 - x_1| + |y_2 - y_1|$$

gdzie $A = (x_1, y_1)$ oraz $B = (x_2, y_2)$.

Wzór w przestrzeni n-wymiarowej:

$$d_{\text{Manhattan}}(A, B) = \sum_{i=1}^{n} |x_i - y_i|$$

gdzie $A = (x_1, x_2, \ldots, x_n)$ oraz $B = (y_1, y_2, \ldots, y_n)$.

Zastosowania odległości Manhattan

Klasteryzacja i detekcja anomalii

- Klasyfikacja tekstów, klasteryzacja dokumentów (dane rzadkie)
- K-Means – lepsza odporność na outliers niż metryka euklidesowa
- Wykrywanie oszustw, bezpieczeństwo sieciowe – identyfikacja anomalii bez nadmiernego wpływu wartości ekstremalnych

Systemy GIS i logistyka

- Modelowanie ruchu w sieci ulic – planowanie miejskie
- Optymalizacja lokalizacji obiektów (minimalizacja odległości podróży)

Wybór metryki: Manhattan vs Euklidesowa

Manhattan:

- Ruch w siatce (bloki miejskie, PCB)
- Dane wysokowymiarowe i rzadkie
- Dane dyskretne/porządkowe
- Mniejsza wrażliwość na wartości odstające

Euklidesowa:

- Fizyczne odległości w otwartej przestrzeni
- Dane ciągłe
- Ruchy po przekątnej równie ważne jak wzdłuż osi

Odległość Minkowskiego

Odległość Minkowskiego to miara odległości między dwoma punktami w przestrzeni, będąca uogólnieniem odległości euklidesowej i manhattańskiej. Parametr p pozwala na dostosowanie miary do różnych przypadków.

- Gdy $p = 1$, odległość Minkowskiego staje się **odległością manhattańską**.
- Gdy $p = 2$, odległość Minkowskiego to **odległość euklidesowa**.
- Ogólnie, wartość p określa wpływ różnic między współrzędnymi punktów na ich odległość.

$$d(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}$$

gdzie:

- x i y to dwa punkty w przestrzeni n-wymiarowej,
- x_i i y_i to współrzędne punktów x i y,
- p to parametr określający rodzaj odległości.

Obliczanie metryk odległości w Pythonie

```python
import numpy as np
from scipy.spatial.distance import euclidean, cityblock
from scipy.spatial.distance import minkowski, cosine

# Dwa punkty w przestrzeni 3D
p1 = np.array([1, 2, 3])
p2 = np.array([4, 5, 6])

# Odleglosc euklidesowa
d_euc = euclidean(p1, p2)  # 5.196
# Odleglosc Manhattan
d_man = cityblock(p1, p2)  # 9
# Odleglosc Minkowskiego (p=3)
d_mink = minkowski(p1, p2, p=3)  # 4.327

```

Odległość cosinusowa

- Mierzy kąt między dwoma punktami lub wektorami, skupiając się na ich orientacji względem siebie, a nie na długości linii między nimi.
- Przydatna w analizie tekstu lub systemach rekomendacji, gdzie kierunek danych (np. częstotliwości słów w artykułach lub preferencje użytkowników) jest ważniejszy niż ich wielkość.
- im bliżej kąt jest do 0°, tym większe podobieństwo między stronami,
- im kąt jest bliższy 90°, tym mniejsze podobieństwo.

Odległości cosinusowa

Przykład 1: Podobieństwo stron WWW

Rozważmy zbiór stron internetowych, które możemy reprezentować jako punkty w **przestrzeni wielowymiarowej**. W tej przestrzeni:

- każdy wymiar odpowiada jednemu słowu z ustalonego słownika,
- każda strona jest reprezentowana jako wektor, gdzie wartość w danym wymiarze odzwierciedla występowanie danego słowa.

Definicja podobieństwa: Odległość $d(x, y)$ między stronami x i y definiujemy jako **znormalizowany iloczyn skalarny** ich wektorów:

$$d(x, y) = \frac{x \cdot y}{\|x\| \cdot \|y\|}$$

gdzie:

- $x \cdot y$ oznacza iloczyn skalarny wektorów x i y,
- $\|x\|$ oraz $\|y\|$ to długości (normy) wektorów x i y.

Tworzenie słownika słów

Aby porównać dokumenty tekstowe, tworzymy najpierw **słownik**, czyli zbiór unikalnych słów występujących w analizowanych dokumentach.

Przykład: Rozważmy dwa dokumenty:

- Dokument 1: "kosmos galaktyka planeta gwiazda"
- Dokument 2: "galaktyka planeta kometa"

Proces tworzenia słownika:

1. Zapisujemy wszystkie unikalne słowa ze wszystkich dokumentów.
2. Każde słowo z dokumentów dodajemy do słownika, jeśli jeszcze w nim nie występuje.

Słownik dla powyższych dokumentów:

$$\text{Słownik} = \{\text{kosmos, galaktyka, planeta, gwiazda, kometa}\}$$

Każde słowo w słowniku reprezentuje **jeden wymiar** w przestrzeni wektorowej.

Reprezentacja wektorowa dokumentów

Po stworzeniu słownika możemy przedstawić każdy dokument jako **wektor liczbowy** w przestrzeni wymiarowej, gdzie każda wartość oznacza liczbę wystąpień danego słowa w dokumencie.

Przykład:

- Słownik: { "kosmos", "galaktyka", "planeta", "gwiazda", "kometa"}
- Dokument 1: "kosmos galaktyka planeta gwiazda"
  Reprezentacja wektorowa: $x = [1, 1, 1, 1, 0]$
- Dokument 2: "galaktyka planeta kometa"
  Reprezentacja wektorowa: $y = [0, 1, 1, 0, 1]$

Interpretacja:

- Każdy element wektora odpowiada liczbie wystąpień konkretnego słowa z naszego słownika w danym dokumencie.
- Tak utworzone wektory możemy porównać, aby obliczyć **podobieństwo** między dokumentami.

Obliczanie podobieństwa między dokumentami

Dla wektorów z poprzedniego slajdu:

- $x = [1, 1, 1, 1, 0]$ (Dokument 1)
- $y = [0, 1, 1, 0, 1]$ (Dokument 2)

Podobieństwo cosinusowe:

$$\cos(\theta) = \frac{x \cdot y}{\|x\| \cdot \|y\|}$$

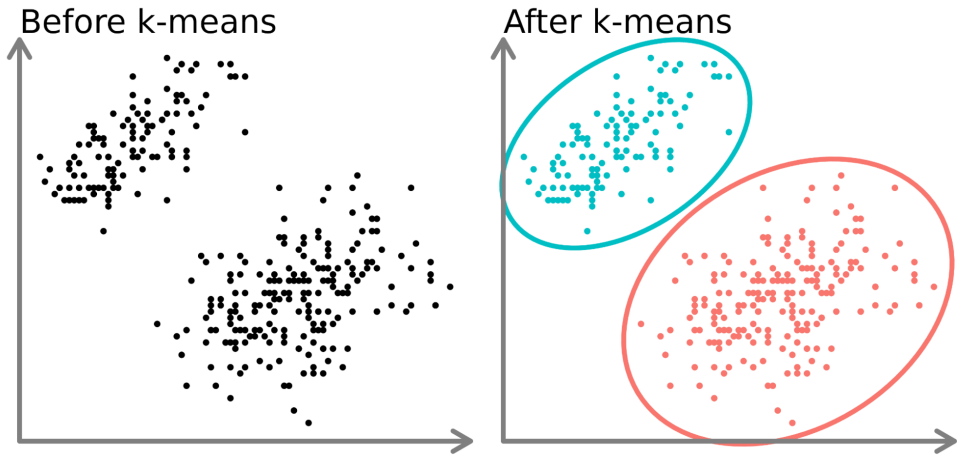
Obliczenia:

- Iloczyn skalarny: $x \cdot y = 1 \cdot 0 + 1 \cdot 1 + 1 \cdot 1 + 1 \cdot 0 + 0 \cdot 1 = 2$
- Norma x: $\|x\| = \sqrt{1^2 + 1^2 + 1^2 + 1^2 + 0^2} = 2$
- Norma y: $\|y\| = \sqrt{0^2 + 1^2 + 1^2 + 0^2 + 1^2} = \sqrt{3}$

$$\cos(\theta) = \frac{2}{2 \cdot \sqrt{3}} = \frac{1}{\sqrt{3}} \approx 0{,}577$$

Interpretacja: Dokumenty mają umiarkowane podobieństwo (wspólne słowa: "galaktyka", "planeta").

Klasteryzacja: idea



Rysunek: Źródło: DataCamp

Uczenie Nienadzorowane

Uczenie nienadzorowane odnosi się do metod statystycznych, które wydobywają sens z danych bez użycia modelu trenowanego na danych z etykietami (tj. danych, gdzie wynik jest znany).

- W uczeniu nadzorowanym (ang. *supervised learning*) model jest trenowany do przewidywania zmiennej odpowiedzi na podstawie zbioru zmiennych predykcyjnych.
- W uczeniu nienadzorowanym nie ma rozróżnienia na zmienną odpowiedzi i zmienne predykcyjne.

Uczenie nienadzorowane w EDA

Cel: Wgląd w dane i odkrywanie relacji między zmiennymi – bez etykietowanych danych.

Zastosowania:

Grupowanie – segmentacja użytkowników na podstawie aktywności i danych demograficznych, co pomaga w personalizacji treści i ofert.

Redukcja wymiarowości – uproszczenie danych przez sprowadzenie wielu zmiennych (np. z czujników) do kilku kluczowych cech, ułatwiając wizualizację i budowę modeli.

Techniki te pozwalają na analizę dużych zbiorów danych i odkrywanie ukrytych struktur.

Czym jest klasteryzacja?

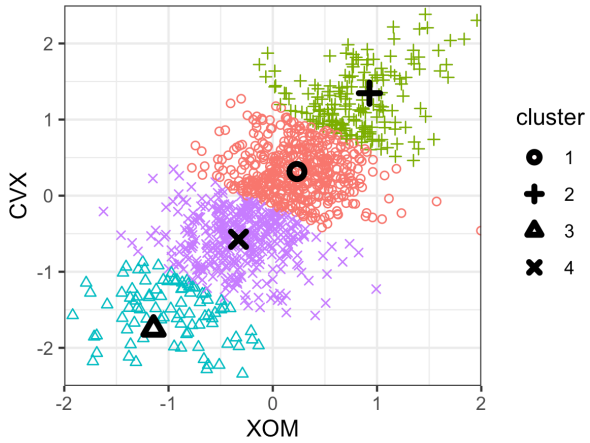
Klasteryzacja to proces grupowania obiektów w taki sposób, aby obiekty w tej samej grupie (nazywanej klastrem) były bardziej podobne do siebie niż do obiektów z innych grup.

Zastosowanie: Klasteryzacja jest często wykorzystywana w fazie EDA do odkrywania nowych informacji i wzorców w danych.

Zastosowaniami w różnych dziedzinach, takich jak:

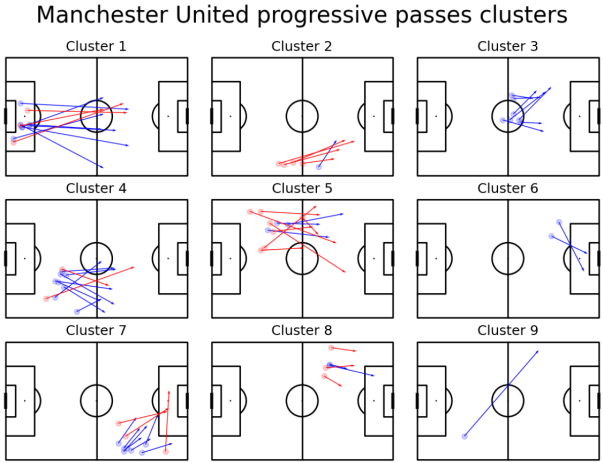
- analiza obrazu,
- analiza zachowań klientów,
- segmentacja rynku,
- analiza sieci społecznościowych.

Przykład K-means



Rysunek: Klastry K-means zastosowane do dziennych zwrotów akcji ExxonMobil i Chevron (centra klastrów są wyróżnione czarnymi symbolami)

Przykład K-means, piłka nożna



Rysunek: Manchester United progressive passes clusters z soccermatics

Kluczowe wyzwania w klasteryzacji

W odróżnieniu od klasyfikacji czy regresji, klasteryzacja nie może być w pełni zautomatyzowana – wymaga wiedzy dziedzinowej i ludzkiego osądu.

Problem oceny jakości: Brak etykiet uniemożliwia stosowanie typowych miar (dokładność, AUC, RMSE) – ocena jest subiektywna.

Kryteria sukcesu:

- Czy model jest interpretowalny?
- Czy wyniki są użyteczne biznesowo?
- Czy odkryto nowe wzorce w danych?

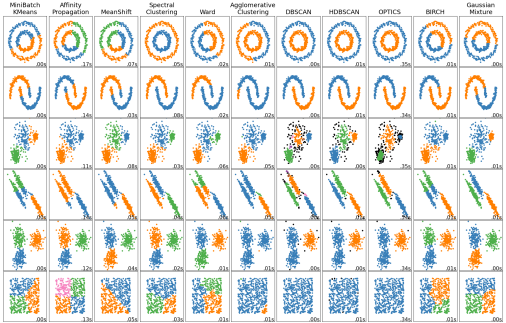
Porównanie różnych algorytmów klasteryzacji

Różnice między algorytmami: algorytmy klasteryzacji można porównać pod kątem czterech kluczowych aspektów:

- **Parametry wymagane przez model** - ile i jakie parametry należy ustawić,
- **Skalowalność** - jak algorytm radzi sobie z dużymi zbiorami danych,
- **Przypadki użycia** - zastosowania i obszary, w których algorytm sprawdza się najlepiej,
- **Geometria** - metryka używana do obliczania odległości między punktami (np. metryka euklidesowa, manhattan).

Różne algorytmów klasteryzacji

W bibliotece `scikit-learn` - popularnej bibliotece uczenia maszynowego w Pythonie - zaimplementowano 10 algorytmów klasteryzacji nienadzorowanej. Każdy z tych algorytmów w odmienny sposób identyfikuje i przypisuje klastry w zbiorze danych.



Rysunek: scikit-learn.org

Algorytm K-Means – wprowadzenie

Najpopularniejszy algorytm klasteryzacji – dzieli dane na K rozłącznych klastrów poprzez minimalizację wewnątrzklastrowej sumy kwadratów.

Funkcja celu:

$$\mathrm{SS}_{\mathrm{total}} = \sum_{k=1}^{K} \sum_{i \in \mathrm{Cluster}\ k} \left((x_i - \bar{x}_k)^2 + (y_i - \bar{y}_k)^2 \right)$$

gdzie środek klastra k to średnia jego punktów:

$$\bar{x}_k = \frac{1}{n_k} \sum_{i \in \mathrm{Cluster}\ k} x_i, \quad \bar{y}_k = \frac{1}{n_k} \sum_{i \in \mathrm{Cluster}\ k} y_i$$

Minimalizacja $\mathrm{SS}_{\mathrm{total}}$ prowadzi do klastrów zwartych i dobrze odseparowanych.

Algorytm K-Means – kroki

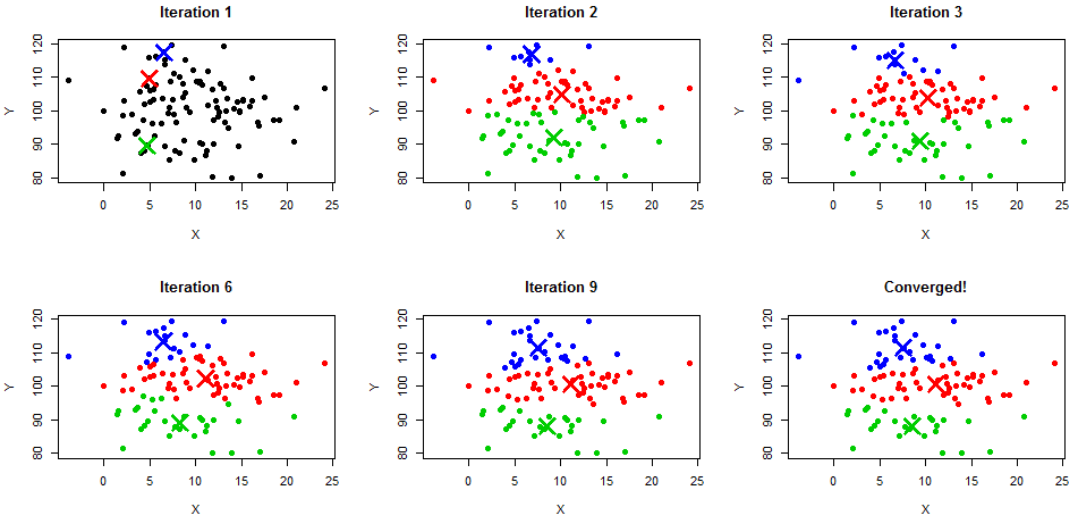
Przebieg algorytmu:

1. Wybierz liczbę klastrów K i zainicjuj środki klastrów (losowo).
2. Przypisz każdy punkt do najbliższego środka (odległość euklidesowa).
3. Przelicz środki klastrów jako średnie przypisanych punktów.
4. Powtarzaj kroki 2–3, aż przypisania przestaną się zmieniać.

Uwagi praktyczne:

- Liczbę klastrów K określa użytkownik (wiedza dziedzinowa lub testowanie).
- Wynik zależy od inicjalizacji – warto uruchomić algorytm kilkukrotnie i wybrać rozwiązanie o najmniejszym SS_{total}.

K-Means: wizualizacja



Wybór liczby klastrów K

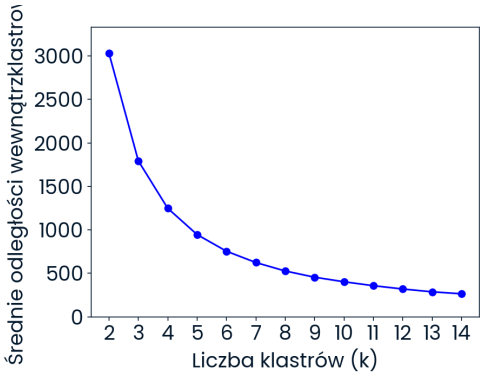
Algorytm K-means wymaga określenia liczby klastrów K:

- W niektórych zastosowaniach liczba klastrów może wynikać z praktycznych potrzeb.
- Przykład: firma zarządzająca siłami sprzedaży może podzielić klientów na segmenty, aby lepiej dopasować działania sprzedażowe.
  - 2 segmenty mogą nie być wystarczająco zróżnicowane,
  - 8 segmentów może być zbyt trudnych do zarządzania.
- W przypadku braku liczby klastrów wynikającej z potrzeb menedżerskich, można zastosować podejście statystyczne.

Metoda łokcia - wybór liczby klastrów K

Metoda łokcia to popularne podejście do wyboru liczby klastrów:

- Polega na znalezieniu takiej liczby klastrów, dla której dodanie kolejnego klastra przestaje znacząco zmniejszać średnie odległości wewnątrzklastrowe.
- Punkt załamania na wykresie oznacza „łokieć", gdzie średnie odległości wewnątrzklastrowe stabilizują się po początkowym szybkim spadku.



Ograniczenia algorytmu K-means

Wrażliwość na inicjalizację:

- Losowy wybór początkowych centroidów może prowadzić do różnych wyników.
- Rozwiązanie: użycie `n_init` (wielokrotne uruchomienie) lub metody `k-means++`.

Założenie o kształcie klastrów:

- K-means zakłada, że klastry są sferyczne (kuliste) i mają podobną wielkość.
- Nie radzi sobie dobrze z klastrami o nieregularnych kształtach.

Wrażliwość na wartości odstające:

- Outliery mogą znacząco przesunąć położenie centroidów.
- Rozwiązanie: usunięcie outlierów przed klasteryzacją.

Konieczność określenia K z góry:

- Wymaga wiedzy dziedzinowej lub metod takich jak metoda łokcia.

Klasteryzacja hierarchiczna – wprowadzenie

Alternatywa dla K-means – tworzy hierarchię klastrów w formie drzewa (dendrogramu), bez konieczności określania liczby klastrów z góry.

Kluczowe pojęcia:

- **Dendrogram**: drzewo pokazujące hierarchię klastrów; liście to rekordy, długość gałęzi to stopień niepodobieństwa.
- **Metryka odległości** $d_{i,j}$: odległość między rekordami i i j.
- **Metryka niepodobieństwa** $D_{A,B}$: różnica między klastrami A i B (na podstawie $d_{i,j}$).

Zalety: analiza struktury na różnych poziomach, intuicyjna wizualizacja.
Wady: wysoki koszt obliczeniowy, słaba skalowalność.

Algorytm aglomeracyjny

Podejście "bottom-up-- każdy rekord zaczyna jako osobny klaster, następnie klastry są sukcesywnie łączone.

Kroki algorytmu:

1. Utwórz początkowy zbiór klastrów – każdy rekord to osobny klaster.
2. Oblicz niepodobieństwo D_{A_i,A_j} między wszystkimi parami klastrów.
3. Połącz dwa najmniej niepodobne klastry.
4. Powtarzaj kroki 2–3, aż pozostanie jeden klaster.

Odczyt dendrogramu: Pozioma linia przecinająca drzewo określa liczbę klastrów – każde przecięcie z gałęzią pionową to osobny klaster.

Zastosowania klasteryzacji hierarchicznej

Biologia – analiza relacji genetycznych, grupowanie organizmów, taksonomia.

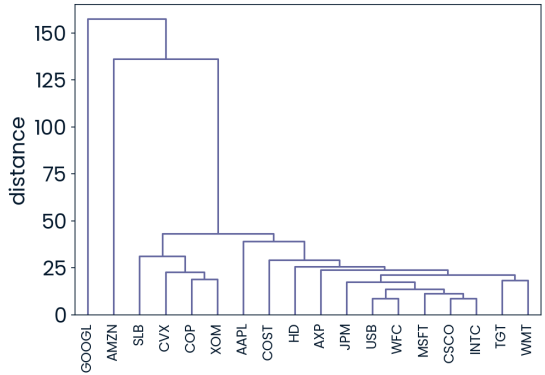
Przetwarzanie obrazów – segmentacja przez grupowanie pikseli wg koloru/intensywności.

Marketing – hierarchie klientów wg nawyków zakupowych, personalizacja strategii.

Sieci społeczne – identyfikacja społeczności i analiza struktury relacji.

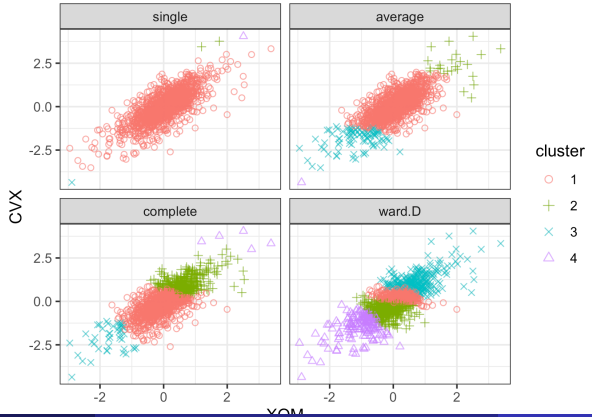
Interpretacja Dendrogramu

- Akcje Google i Amazon są bardziej różne od pozostałych.
- Akcje ropy (SLB, CVX, XOM, COP) tworzą osobny klaster.
- Apple (AAPL) stanowi odrębny klaster.



Miary Niepodobieństwa *Dissimilarity*

Istnieją cztery popularne miary niepodobieństwa: **kompletne połączenie**, **pojedyncze połączenie**, **średnie połączenie** oraz **minimalna wariancja**. Te miary (oraz inne) są obsługiwane przez większość oprogramowania do grupowania hierarchicznego, w tym `hclust` i `linkage`.



Miary niepodobieństwa: porównanie

- **Kompletne połączenie (ang. complete linkage)** – oblicza odległość między dwoma klastrami C_1 i C_2 jako maksymalną odległość między parami elementów w tych klastrach.
- **Pojedyncze połączenie (ang. single linkage)** – wykorzystuje minimalną odległość między elementami dwóch klastrów C_1 i C_2 jako miarę odległości między klastrami.
- **Średnie połączenie (ang. average linkage)** – odległość między dwoma klastrami C_1 i C_2 to średnia odległość między wszystkimi parami elementów w tych klastrach.
- **Minimalna wariancja (ang. minimum variance)** (metoda Warda) – podobna do K-means, minimalizuje sumę kwadratów odchyleń wewnątrz klastrów.

Porównanie: K-means vs Klasteryzacja hierarchiczna

Kryterium	**K-means**	**Hierarchiczna**
Liczba klastrów	Wymagana z góry	Można określić po analizie
Złożoność	$O(n \cdot K \cdot i)$	$O(n^2 \log n)$
Skalowalność	Dobra (duże zbiory)	Słaba (małe/średnie zbiory)
Kształt klastrów	Sferyczne	Dowolne
Wynik	Zależny od inicjalizacji	Deterministyczny
Wizualizacja	Brak hierarchii	Dendrogram

Kiedy używać K-means: Duże zbiory danych, znana liczba klastrów.
Kiedy używać hierarchicznej: Eksploracja danych, potrzeba wizualizacji struktury, małe/średnie zbiory.

Podsumowanie: K-means vs klasteryzacja hierarchiczna

K-means:

- Użytkownik wybiera liczbę klastrów K
- Iteracyjne przypisywanie do najbliższego centroidu
- Wybór K zależy od wymagań praktycznych

Hierarchiczna:

- Nie wymaga określenia K z góry
- Aglomeracja: od pojedynczych rekordów do jednego klastra
- Dendrogram pokazuje strukturę na różnych poziomach

Źródła

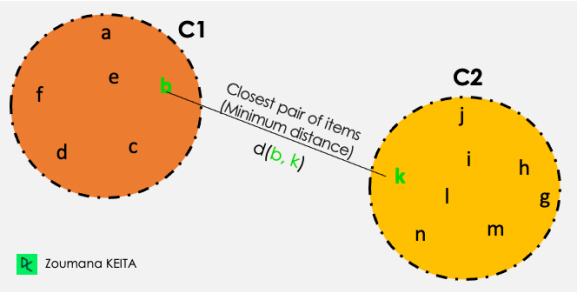
- Książka Practical Statistics for Data Scientists. 50+ Essential Concepts Using R and Python. 2nd Edition, by Peter Bruce, Andrew Bruce, Peter Gedeck
- Materiały z Data Camp tutorials

Dodatek: Single Linkage

Single Linkage wykorzystuje minimalną odległość między elementami dwóch klastrów C_1 i C_2 jako miarę odległości między klastrami.

$$\text{Odległość}(C_1, C_2) = \min\{d(i,j) \mid i \in C_1, j \in C_2\}$$

Spośród wszystkich par elementów w klastrach C_1 i C_2, odległość między tymi, które mają najmniejszą odległość, jest wykorzystywana jako odległość między klastrami.

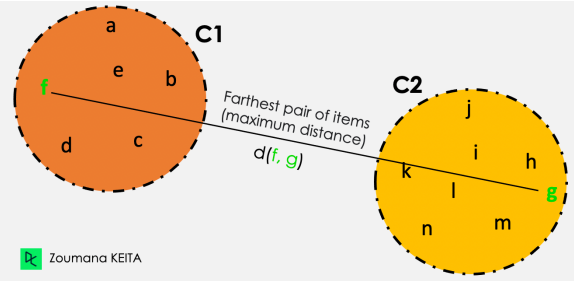


Dodatek: Complete Linkage

Complete Linkage oblicza odległość między dwoma klastrami C_1 i C_2 jako maksymalną odległość między parami elementów w tych klastrach.

$$\text{Odległość}(C_1, C_2) = \max\{d(i,j) \mid i \in C_1, j \in C_2\}$$

Spośród wszystkich par elementów w klastrach C_1 i C_2, odległość między tymi, które mają największą odległość, jest traktowana jako odległość między klastrami.



Dodatek: Average Linkage

W **Average Linkage** odległość między dwoma klastrami C_1 i C_2 to średnia odległość między wszystkimi parami elementów w tych klastrach.

$$\text{Odległość}(C_1, C_2) = \frac{\sum d(i,j)}{\text{Liczba par}}$$

Średnia odległość reprezentuje dystans między klastrami.