

WYKŁAD 3

Inteligentne systemy internetowe

Klastrowanie treści internetowych

Prowadzący: Jarosław Szkoła

Plan wykładu

- 1 Wprowadzenie do klastrowania treści
- 2 Przygotowanie i reprezentacja danych tekstowych
- 3 Grupowanie hierarchiczne
- 4 Algorytm k-średnich (k-means)
- 5 Inne algorytmy: DBSCAN, mieszaniny Gaussa
- 6 Ocena jakości grupowania i dobór liczby grup
- 7 Zastosowania w inteligentnych systemach internetowych

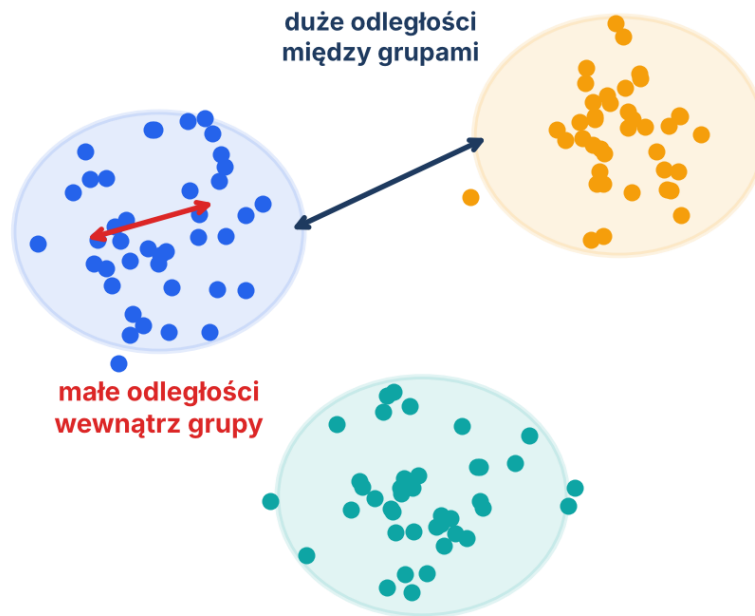
01

Wprowadzenie do klastrowania treści

Czym jest klastrowanie?

- Cel: podzielić zbiór obiektów na grupy (klastry) tak, by obiekty były podobne wewnątrz grupy i różne między grupami
- Grupowanie to uczenie nienadzorowane: nie ma etykiet ani „poprawnej” odpowiedzi, szukamy struktury ukrytej w danych
- Wynik zależy od reprezentacji obiektów, miary podobieństwa i algorytmu – inne wybory dają inne grupy
- Grupy mogą być rozłączne (twarde) lub z przynależnością częściową (miękkie)

Idea grupowania



Zastosowania klastrowania

- Segmentacja użytkowników i klientów: grupy o podobnych zachowaniach i potrzebach
- Grupowanie dokumentów, artykułów i newsów w tematy; tematyczne porządkowanie wyników wyszukiwania
- Wykrywanie anomalii: obiekty odległe od wszystkich grup lub oznaczone jako szum
- Redukcja wymiarowości i kompresja: centroid grupy zastępuje wiele podobnych obiektów

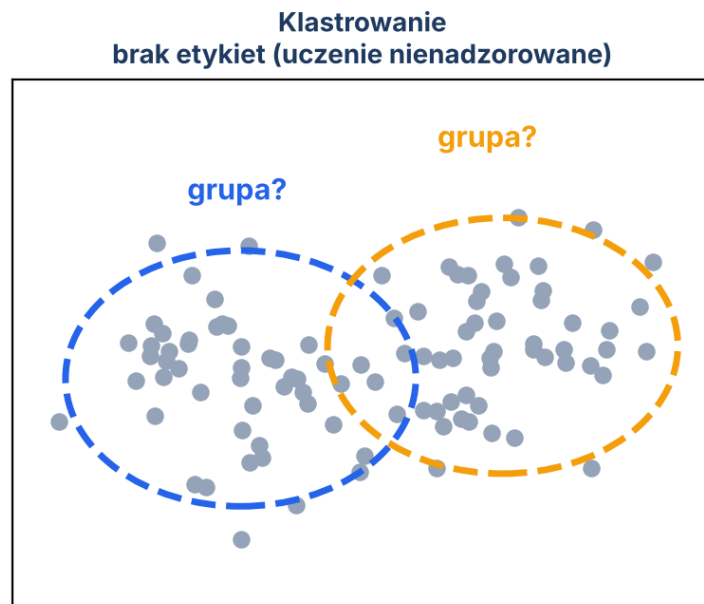
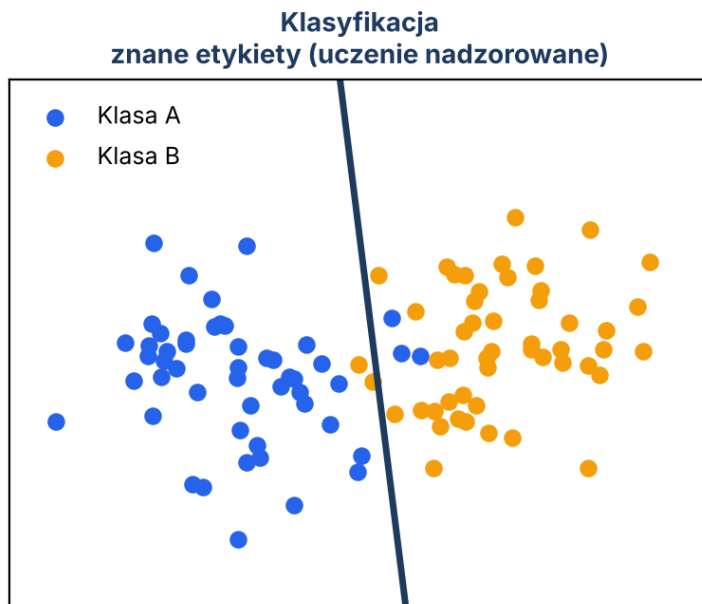
Obszary zastosowań klastrowania



Klastrowanie a klasyfikacja

- Klasyfikacja: dane z etykietami, model uczy się przewidywać znane klasy (uczenie nadzorowane)
- Klastrowanie: brak etykiet, liczba i znaczenie grup są nieznane i trzeba je odkryć
- Ocena klasyfikacji porównuje wynik z etykietami; ocena grupowania jest trudniejsza i częściowo subiektywna
- Typowy schemat: grupowanie odkrywa strukturę, a grupy opisujemy i używamy jako nowych cech

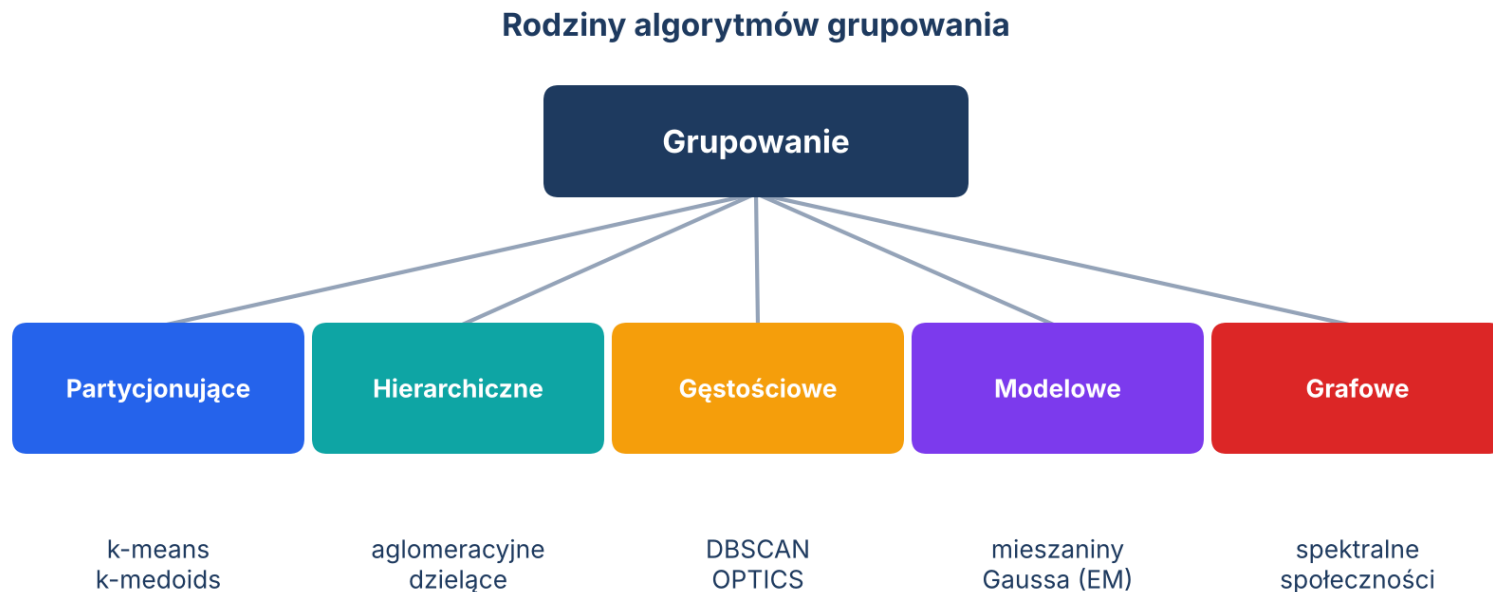
Klasyfikacja a klastrowanie



Rodziny algorytmów grupowania

- Partycjonujące (k-means, k-medoids): podział na k grup optymalizujący kryterium, np. sumę kwadratów odległości
- Hierarchiczne: drzewo zagnieżdżonych grup (dendrogram), od pojedynczych obiektów do jednej grupy
- Gęstościowe (DBSCAN, OPTICS): grupy to gęste obszary, rzadkie obszary traktujemy jak szum
- Modelowe (mieszanie Gaussa, EM) oraz grafowe i spektralne: grupy jako rozkłady lub społeczności w grafie

Rodziny algorytmów grupowania



Wyzwania klastrowania treści internetowych

- Skala: miliony dokumentów i użytkowników wymagają algorytmów o niskiej złożoności lub przybliżonych
- Wysoka wymiarowość i rzadkość reprezentacji tekstu utrudniają pomiar podobieństwa
- Brak jednej „właściwej” liczby grup oraz grupy o różnych rozmiarach, gęstościach i kształtach
- Dane zmieniają się w czasie (nowe artykuły, trendy), więc grupowanie trzeba aktualizować

02

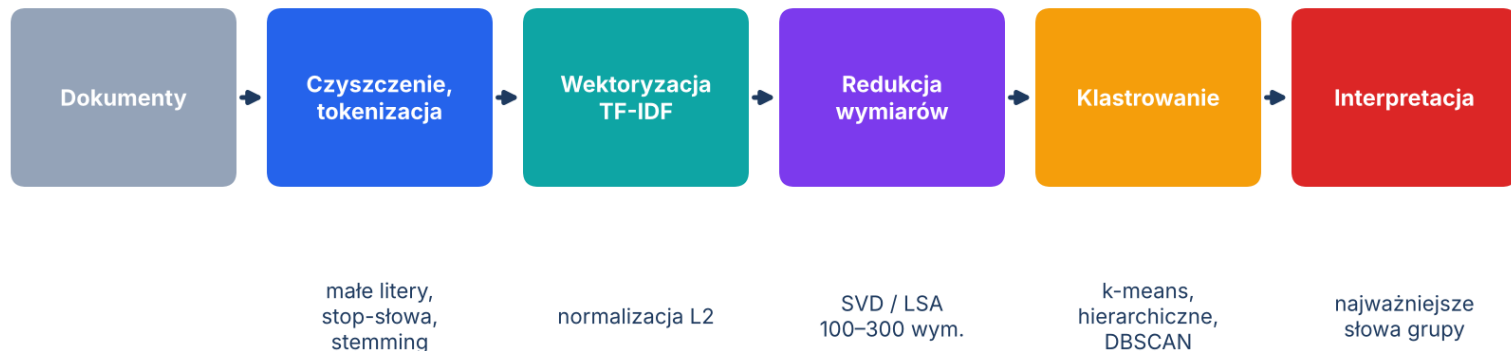
Przygotowanie i reprezentacja danych tekstowych

Potok przetwarzania tekstu

- Czyszczenie: usunięcie znaczników HTML, reklam i nawigacji, ujednolicenie wielkości liter
- Tokenizacja i normalizacja: podział na słowa, usunięcie stop-słów, stemming lub lematyzacja
- Wektoryzacja: zamiana dokumentu na wektor liczb (worek słów, TF-IDF, osadzenia)
- Opcjonalna redukcja wymiarów, a następnie grupowanie i interpretacja wyników

Potok przetwarzania treści

Potok przetwarzania treści tekstowych



każdy etap wpływa na jakość grup – błędy na początku potoku przenoszą się na wynik

Worek słów i TF-IDF

- Worek słów: dokument to wektor liczników słów, kolejność słów jest ignorowana
- $TF\text{-}IDF(t, d) = tf(t, d) \cdot \log(N / df(t))$: słowa częste w dokumencie, a rzadkie w kolekcji, są ważne (wykład 2)
- Przykład: $tf = 3$, $N = 1000$, $df = 10$, logarytm dziesiętny: $idf = 2$, więc $TF\text{-}IDF = 3 \cdot 2 = 6$
- Wektory normalizujemy do długości 1 (norma L2), by długość dokumentu nie wpływała na podobieństwo

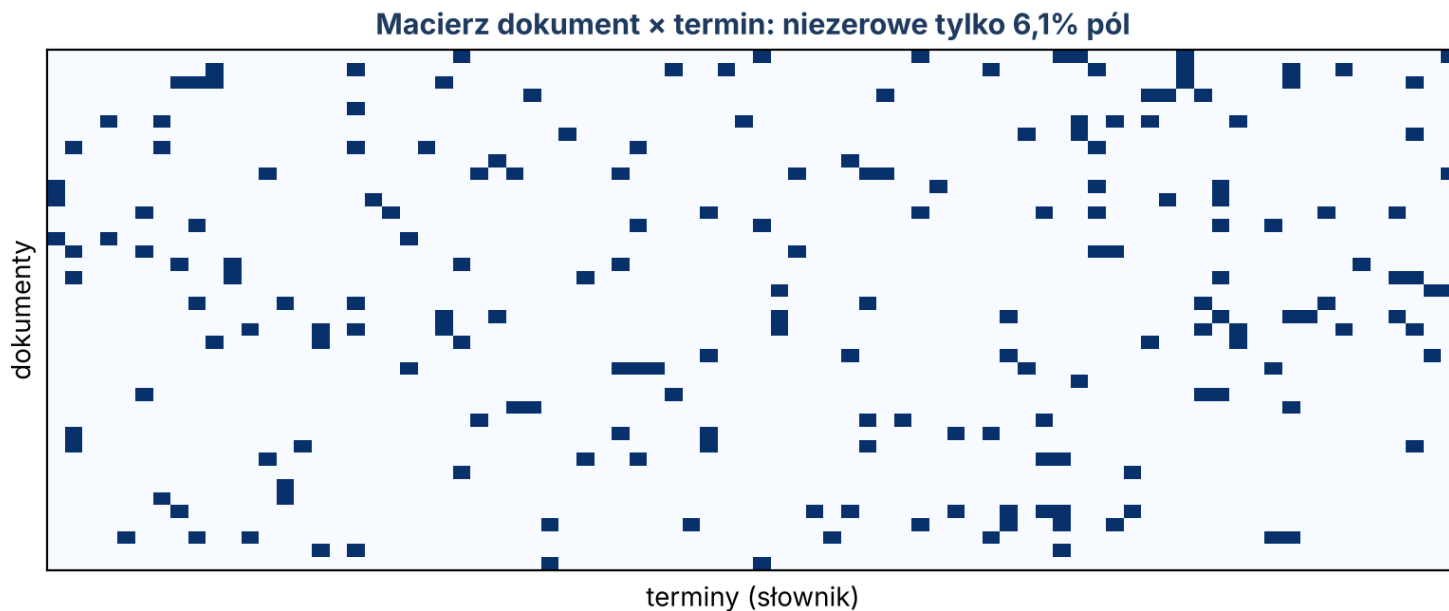
Cechy liczbowe i tekstowe, normalizacja

- Dane użytkowników łączą cechy liczbowe (liczba wizyt, wartość koszyka) i tekstowe lub kategorialne
- Standaryzacja: $z = (x - \text{średnia}) / \text{odchylenie standardowe}$; min-max: $(x - \text{min}) / (\text{max} - \text{min})$
- Bez skalowania cecha o dużych wartościach (np. kwota w zł) zdominuje odległość euklidesową
- Cechy kategorialne kodujemy one-hot, a tekst reprezentujemy osobno (TF-IDF) i łączymy z wagami

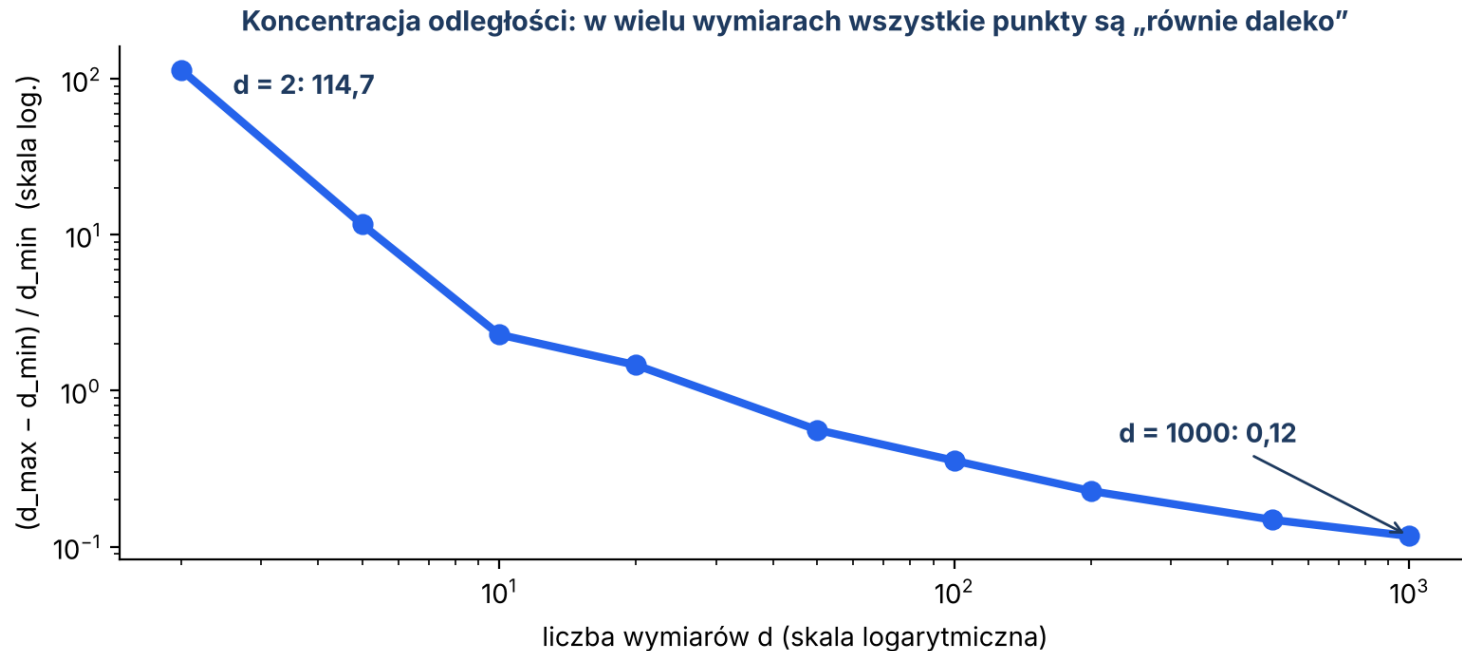
Wysoka wymiarowość i rzadkość

- Słownik liczy dziesiątki tysięcy terminów, a pojedynczy dokument zawiera tylko ich niewielki ułamek
- Macierz dokument $\times$ termin jest rzadka: niezerowe są zwykle ułamki procenta pól
- Klątwa wymiarowości: w wielu wymiarach odległości do najbliższego i najdalszego punktu stają się niemal równe
- Skutek: słaba rozróżnialność grup oraz duży koszt pamięci i obliczeń

Rzadka macierz dokument $\times$ termin



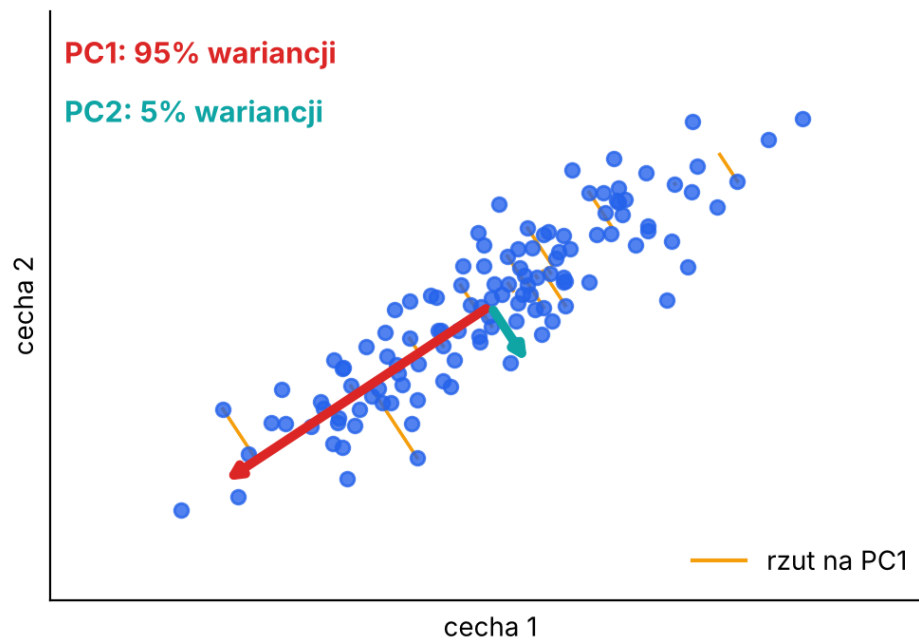
Koncentracja odległości w wysokich wymiarach



Redukcja wymiarów: PCA

- PCA znajduje nowe, wzajemnie prostopadłe osie (składowe główne) o największej wariancji
- Składowe to wektory własne macierzy kowariancji; wariancja wyjaśniona = wartość własna / suma wartości własnych
- Zachowujemy pierwsze k składowych wyjaśniających np. 90–95% wariancji
- PCA wymaga centrowania danych, co niszczy rzadkość – dla macierzy TF-IDF stosuje się raczej SVD

Składowe główne PCA

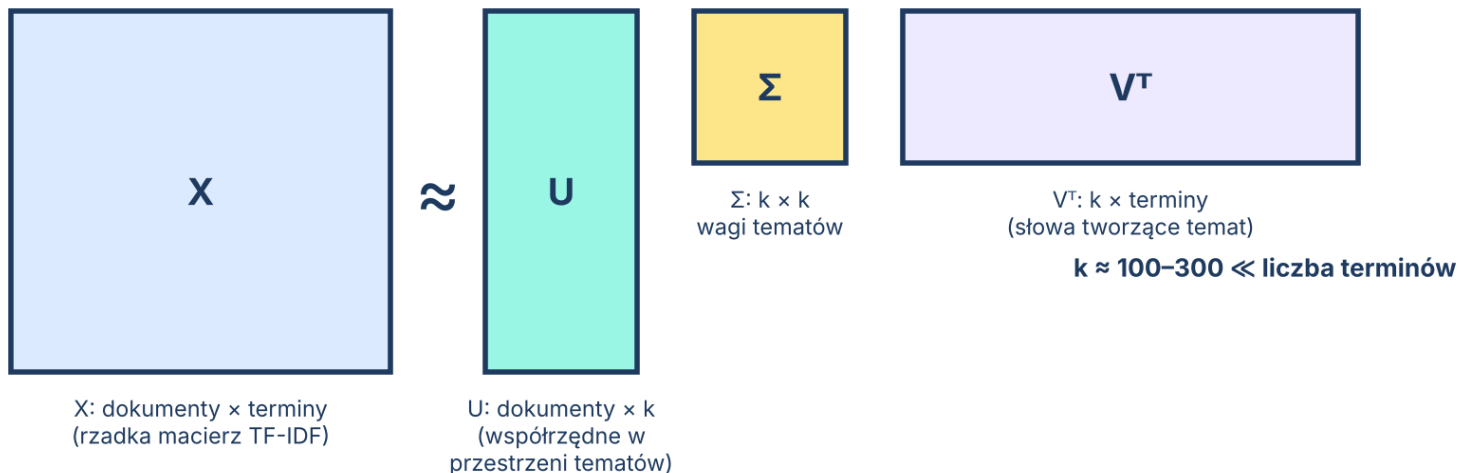


SVD i analiza semantyczna LSA

- SVD rozkłada macierz $X \approx U \cdot \Sigma \cdot V^T$; k największych wartości osobliwych daje najlepszą aproksymację rzędu k
- LSA: SVD na macierzy TF-IDF; wymiary odpowiadają ukrytym tematom (zwykle $k = 100\text{--}300$)
- Dokumenty o podobnym znaczeniu zbliżają się, nawet gdy nie mają wspólnych słów (synonimy)
- TruncatedSVD działa na macierzach rzadkich i jest standardem przed grupowaniem tekstu

Rozkład SVD i LSA

Rozkład SVD i analiza semantyczna LSA



Podobieństwo cosinusowe i k-means dla tekstu

- Dla wektorów o długości 1 zachodzi $\|x - y\|^2 = 2 - 2 \cdot \cos(x, y)$, więc odległość euklidesowa odpowiada cosinusowej
- Dlatego po normalizacji L2 można używać k-means do dokumentów (spherical k-means)
- Odległość cosinusowa = $1 - \cos(x, y)$ ignoruje długość dokumentu, liczy się rozkład tematów
- Ta sama miara jest podstawą rankingu opartego na treści w wyszukiwarkach

03

Grupowanie hierarchiczne

Idea grupowania hierarchicznego

- Wynikiem jest hierarchia zagnieżdżonych grup, a nie jeden podział; liczbę grup wybieramy po zbudowaniu drzewa
- Aglomeracyjne (od dołu): start od pojedynczych obiektów, łączenie najbliższych grup aż do jednej
- Dzielące (od góry): start od jednej grupy i rekurencyjny podział (np. bisecting k-means)
- W praktyce dominuje wariant aglomeracyjny

Algorytm aglomeracyjny

- Krok 1: każdy obiekt tworzy własną grupę; obliczamy macierz odległości
- Krok 2: znajdujemy dwie najbliższe grupy i łączymy je w jedną
- Krok 3: uaktualniamy odległości nowej grupy do pozostałych według wybranej metody łączenia
- Krok 4: powtarzamy kroki 2–3, aż zostanie jedna grupa; zapisujemy wysokości łączeń

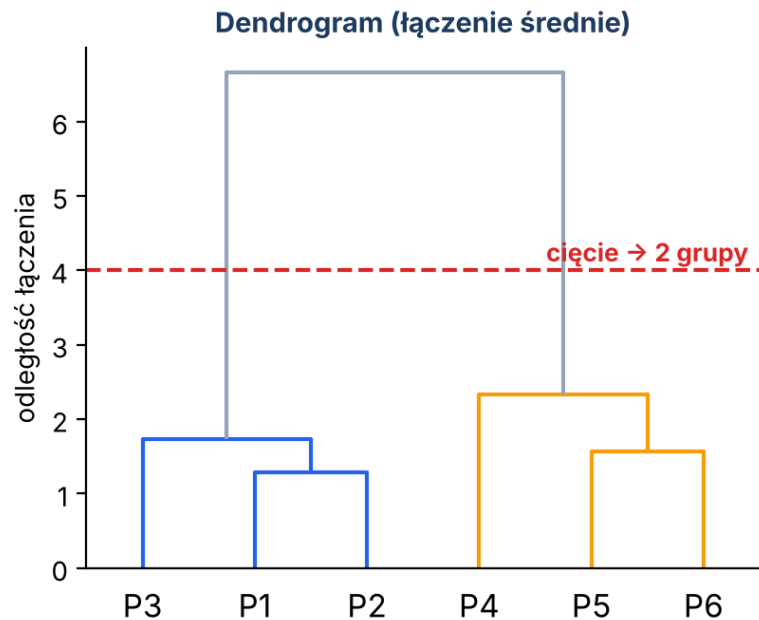
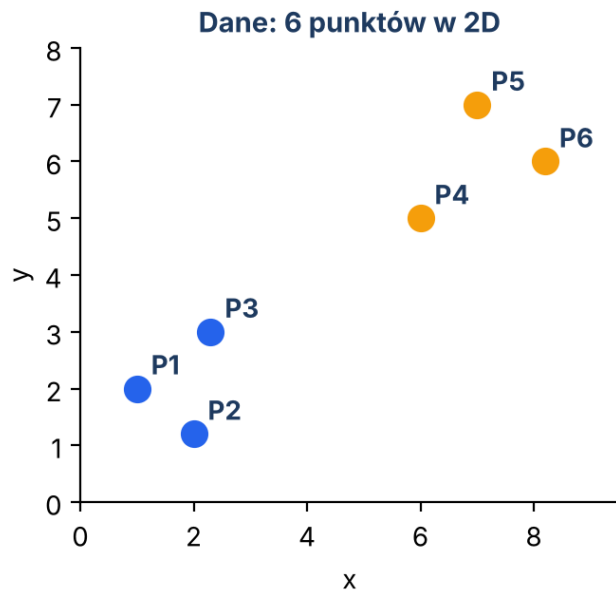
Dendrogram

- Dendrogram to drzewo łączy: wysokość złączenia oznacza odległość między łączonymi grupami
- Poziome „cięcie” dendrogramu daje podział na grupy: im wyżej tnjemy, tym mniej grup
- Duży skok wysokości łączy sugeruje naturalną liczbę grup
- Kolejność liści jest umowna – liczą się wysokości i struktura połączeń

Przykład liczbowy: sześć punktów

- Punkty: $P1(1; 2)$, $P2(2; 1,2)$, $P3(2,3; 3)$, $P4(6; 5)$, $P5(7; 7)$, $P6(8,2; 6)$; odległość euklidesowa
- Łączenie średnie: $P1 + P2$ przy 1,28; $P5 + P6$ przy 1,56; $P3$ dołącza do $\{P1, P2\}$ przy 1,73
- Następnie $P4$ dołącza do $\{P5, P6\}$ przy 2,33, a obie grupy łączą się dopiero przy 6,65
- Cięcie na wysokości 4 daje dwie grupy: $\{P1, P2, P3\}$ i $\{P4, P5, P6\}$

Dane i dendrogram

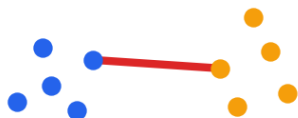


Metody łączenia (linkage)

- Pojedyncze (single): minimalna odległość między parą punktów z dwóch grup – wykrywa wydłużone grupy, ale „łańcuchuje”
- Pełne (complete): maksymalna odległość między parami – zwarte grupy podobnej wielkości
- Średnie (average): średnia odległość po wszystkich parach – kompromis, odporny na wartości odstające
- Warda: łączy grupy, które najmniej zwiększają sumę kwadratów odchyleń wewnątrz grup (podobne do k-means)

Metody łączenia grup

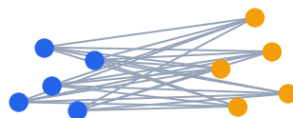
Pojedyncze (single)
min = 3,01



Pełne (complete)
max = 6,40



Średnie (average)
śr. = 4,78



Warda (Ward)
min. przyrostu
wariancji



czerwone: odległość (lub środek) uwzględniana przy łączeniu dwóch grup

Przykład: różne metody łączenia

- Dla dwóch grup z rysunku: single = 3,01; average = 4,78; complete = 6,40
- Ta sama para grup jest więc „blisko” lub „daleko” zależnie od metody, co zmienia kolejność łączeń
- Single często tworzy jedną wielką grupę (efekt łańcucha); Ward i average dają bardziej zrównoważone grupy
- Ward wymaga odległości euklidesowej; pozostałe metody działają z dowolną miarą odległości

Zalety i wady grupowania hierarchicznego

- Zalety: nie trzeba podawać k z góry, wynik jest deterministyczny, a dendrogram pokazuje strukturę na wielu poziomach
- Pozwala na dowolną miarę odległości, w tym cosinusową dla tekstu
- Wady: wymaga macierzy odległości – pamięć i czas co najmniej $O(n^2)$ (naiwnie $O(n^3)$)
- Nie skaluje się do milionów dokumentów; decyzji o połączeniu nie można cofnąć

04

Algorytm k-średnich (k-means)

Idea algorytmu k-means

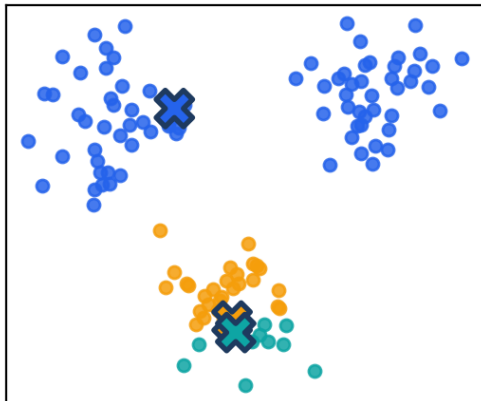
- Dzieli n obiektów na k grup; każda grupa jest reprezentowana przez centroid (średnią swoich punktów)
- Obiekt trafia do grupy o najbliższym centroidzie (zwykle odległość euklidesowa)
- Liczbę grup k podajemy z góry; algorytm jest prosty, szybki i bardzo popularny
- To uczenie nienadzorowane i grupowanie twarde: każdy punkt należy do dokładnie jednej grupy

Kroki algorytmu

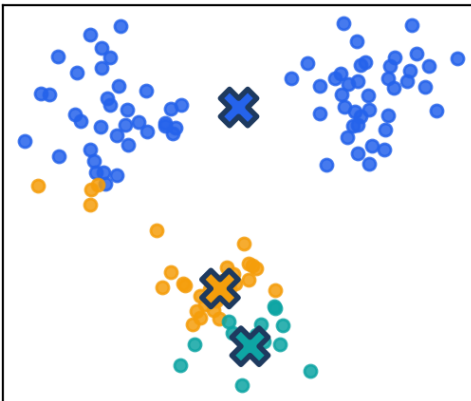
- Krok 1: inicjalizacja – wybór k początkowych centroidów (losowo lub metodą k-means++)
- Krok 2: przypisanie – każdy punkt do najbliższego centroidu
- Krok 3: aktualizacja – nowy centroid to średnia punktów przypisanych do grupy
- Krok 4: powtarzanie kroków 2–3 do zbieżności; koszt jednej iteracji to $O(n \cdot k \cdot d)$

Kolejne iteracje k-means

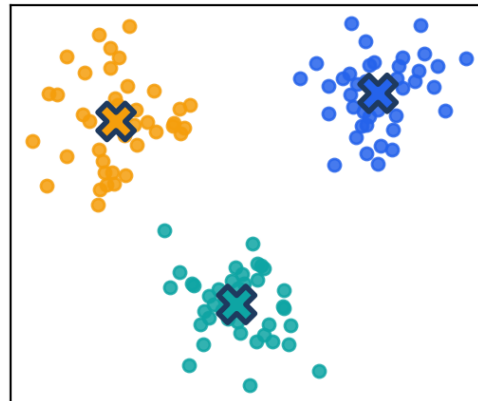
Inicjalizacja
(3 losowe centroidy)



Iteracja 1



Zbieżność (iteracja 6)



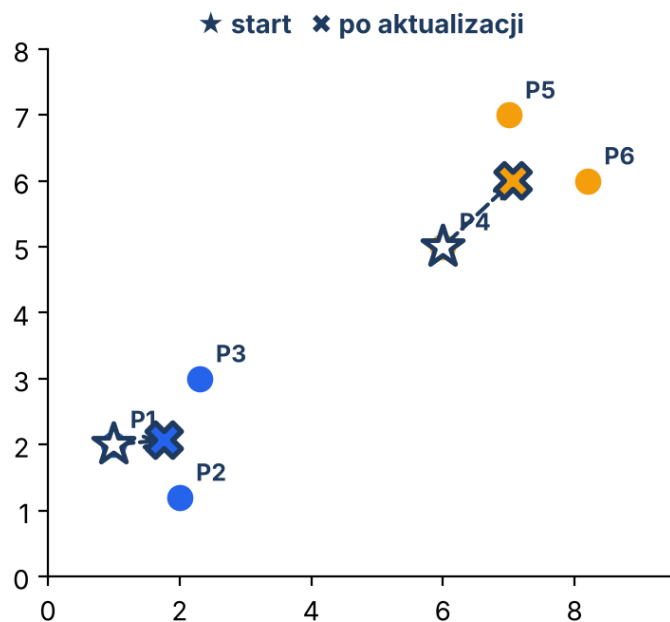
Przykład liczbowy: dane i inicjalizacja

- Te same punkty: $P1(1; 2)$, $P2(2; 1,2)$, $P3(2,3; 3)$, $P4(6; 5)$, $P5(7; 7)$, $P6(8,2; 6)$;
 $k = 2$
- Inicjalizacja: $c1 = P1 = (1; 2)$ oraz $c2 = P4 = (6; 5)$
- Przypisanie 1: $P2 (1,28 / 5,52)$ i $P3 (1,64 / 4,21)$ do grupy 1; $P5 (7,81 / 2,24)$ i $P6 (8,24 / 2,42)$ do grupy 2
- Grupa 1 = $\{P1, P2, P3\}$, grupa 2 = $\{P4, P5, P6\}$

Przykład liczbowy: aktualizacja i zbieżność

- Nowe centroidy: $c1 = (1,77; 2,07)$ oraz $c2 = (7,07; 6,00)$ – średnie współrzędnych punktów grupy
- Iteracja 2: P1–P3 najbliżej $c1$ (0,77; 0,90; 1,07), a P4–P6 najbliżej $c2$ (1,46; 1,00; 1,13)
- Przypisania się nie zmieniły, więc algorytm zakończył pracę po dwóch iteracjach
- Suma kwadratów odległości do centroidów (WCSS): 15,17 dla centroidów początkowych, 6,98 po aktualizacji

Przykład liczbowy k-means



Przypisanie w iteracji 1			
punkt	$d(\cdot, c_1)$	$d(\cdot, c_2)$	grupa
P1	0,00	5,83	1
P2	1,28	5,52	1
P3	1,64	4,21	1
P4	5,83	0,00	2
P5	7,81	2,24	2
P6	8,24	2,42	2

Kryterium: suma kwadratów wewnątrz grup

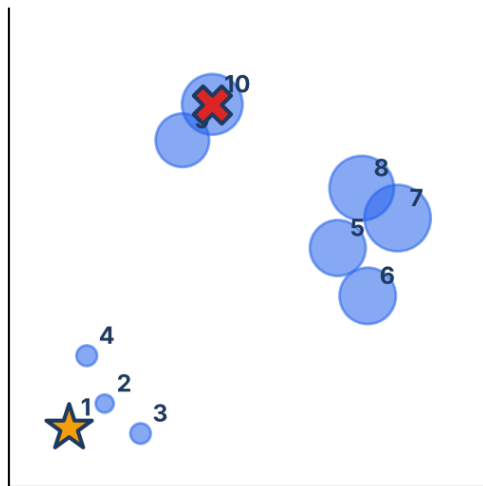
- k-means minimalizuje WCSS = suma po grupach i punktach grupy z $\|x - \text{centroid}\|^2$ (inercja)
- Każdy krok (przypisanie, aktualizacja) nie zwiększa WCSS, więc algorytm zawsze się zatrzymuje
- Znajduje jednak tylko minimum lokalne, zależne od inicjalizacji
- Dlatego uruchamiamy algorytm wielokrotnie (n_{init}) i wybieramy wynik o najmniejszej inercji

Inicjalizacja k-means++

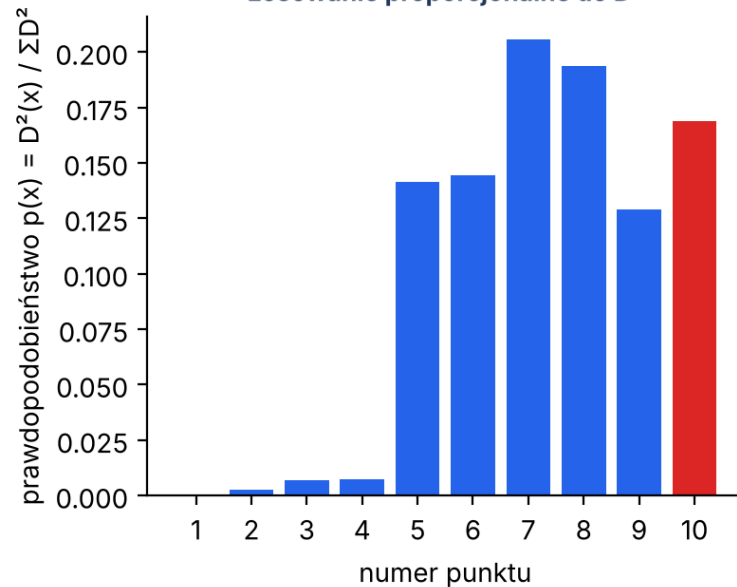
- Pierwszy centroid: losowy punkt; kolejne losujemy z prawdopodobieństwem proporcjonalnym do $D^2(x)$
- $D(x)$ to odległość punktu od najbliższego już wybranego centroidu, więc centroidy leżą daleko od siebie
- Przykład: po wyborze punktu 1 punkty 1–4 (tuż obok) mają łącznie ok. 1,7% szans, a punkty 5–10 ok. 98%
- Gwarancja: oczekiwana inercja jest $O(\log k)$ razy większa od optimum; to domyślna inicjalizacja w scikit-learn

Losowanie centroidów w k-means++

★ 1. centroid; ✕ wylosowany 2. centroid
rozmiar punktu $\propto D^2$



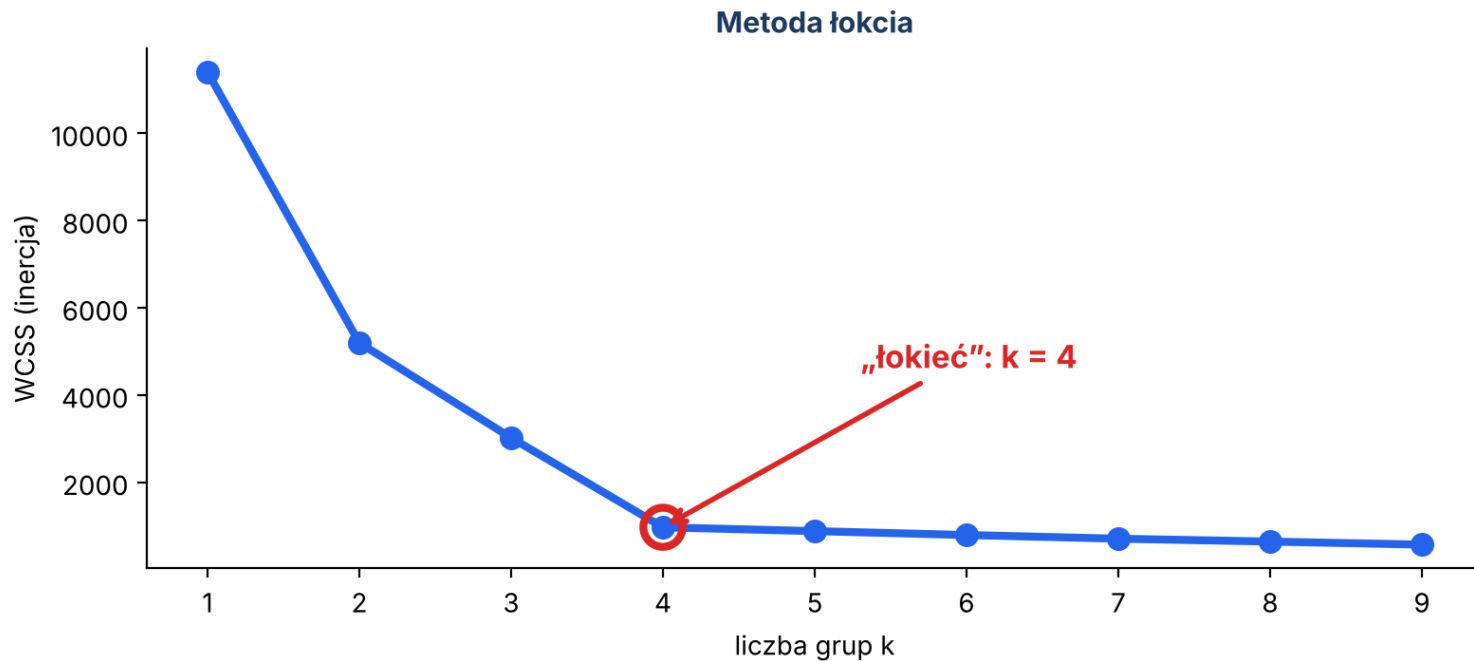
Losowanie proporcjonalne do D^2



Metoda łokcia

- Obliczamy WCSS dla $k = 1, 2, 3, \dots$ i rysujemy wykres $WCSS(k)$
- WCSS maleje zawsze wraz z k (dla $k = n$ wynosi 0), więc szukamy „łokcia”, gdzie spadek wyraźnie słabnie
- Na rysunku łokieć wypada przy $k = 4$: dalsze grupy poprawiają WCSS już tylko nieznacznie
- Wada: łokieć bywa niejednoznaczny, dlatego łączymy go z innymi miarami, np. sylwetką (sekcja 6)

Metoda łokcia



Zalety, wady i warianty k-means

- Zalety: prostota, skalowalność (Mini-Batch k-means) i łatwa interpretacja centroidów
- Wady: trzeba znać k ; wrażliwość na inicjalizację, wartości odstające i skalę cech
- Zakłada grupy kuliste i podobnej wielkości; nie radzi sobie z grupami o dowolnych kształtach
- Warianty: k-medoids (reprezentant to rzeczywisty obiekt), spherical k-means (cosinus dla tekstu), bisecting k-means

05

**Inne algorytmy: DBSCAN, mieszaniny
Gausa**

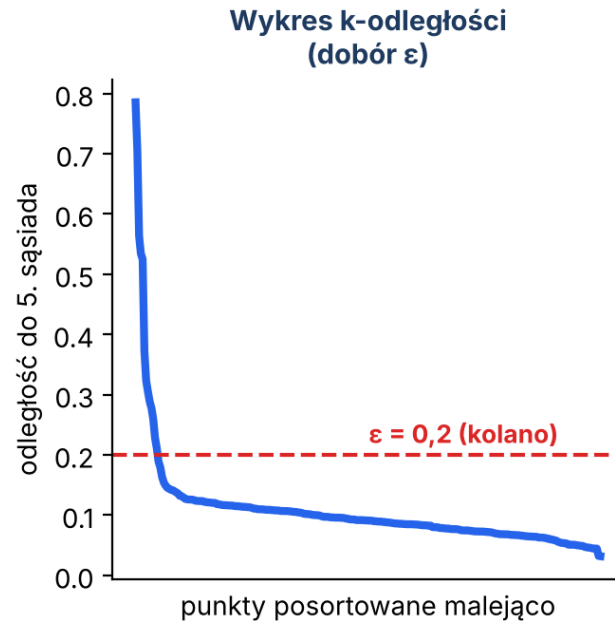
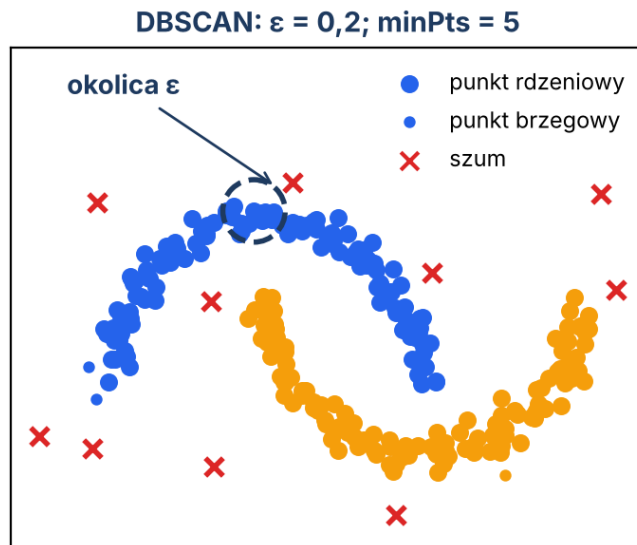
DBSCAN – idea

- Grupy to obszary o dużej gęstości punktów rozdzielone obszarami rzadkimi; punkty z rzadkich obszarów to szum
- Parametry: ϵ (promień sąsiedztwa) i minPts (minimalna liczba punktów w okolicy ϵ)
- Punkt rdzeniowy: w okolicy ϵ ma co najmniej minPts punktów (wraz z samym sobą)
- Punkt brzegowy leży w okolicy punktu rdzeniowego, ale sam nie jest rdzeniowy; pozostałe punkty to szum

DBSCAN – algorytm i dobór parametrów

- Dla nieodwiedzonego punktu rdzeniowego zakładamy nową grupę i dołączamy punkty osiągalne gęstościowo
- Grupa rozrasta się przez kolejne punkty rdzeniowe aż do brzegów; liczbę grup wyznacza sam algorytm
- minPts: zwykle co najmniej $d + 1$, np. 4–10 (więcej dla danych zaszumionych)
- ϵ : z wykresu k-odległości (odległość do minPts-tego sąsiada, sortowana malejąco) wybieramy „kolano”

DBSCAN i wykres k-odległości



DBSCAN – zalety i wady

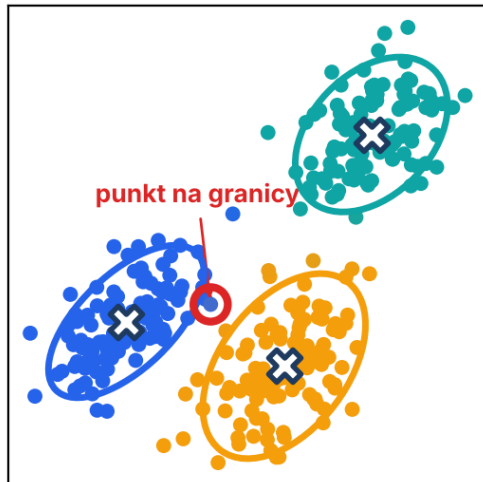
- Zalety: nie wymaga podania k , wykrywa grupy o dowolnym kształcie i oznacza szum (anomalie)
- Złożoność $O(n \log n)$ z indeksem przestrzennym (np. drzewo k -d), w przeciwnym razie $O(n^2)$
- Wady: trudny dobór ϵ przy grupach o różnych gęstościach (rozwiązania: OPTICS, HDBSCAN)
- W wysokich wymiarach (TF-IDF) pojęcie gęstości traci sens – warto najpierw zredukować wymiary (SVD)

Mieszaniny Gaussa (GMM)

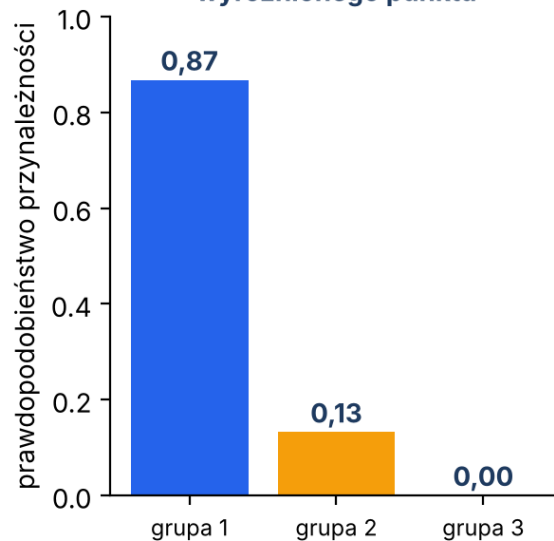
- Zakładamy, że dane pochodzą z mieszaniny k rozkładów normalnych: $p(x) = \sum_k \pi_k \cdot N(x \mid \mu_k, \Sigma_k)$
- Każda grupa ma własną średnią, macierz kowariancji (kształt elipsy) i wagę π_k
- Miękkie przypisanie: każdy punkt ma prawdopodobieństwo przynależności do każdej grupy
- Liczbę składowych wybieramy kryteriami informacyjnymi BIC lub AIC

Mieszanina Gaussa i miękkie przypisanie

Mieszanina 3 rozkładów Gaussa (elipsy 2σ)



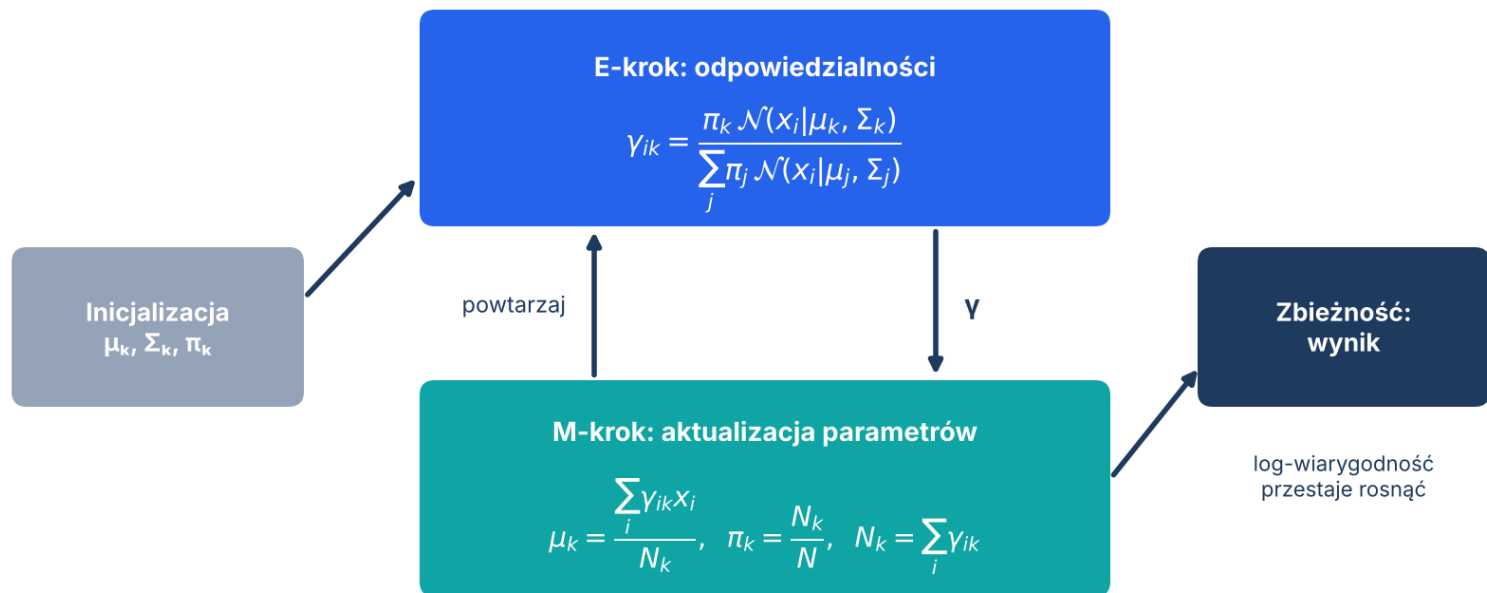
Miękkie przypisanie
wyróżnionego punktu



Algorytm EM

- Parametry (π , μ , Σ) estymujemy metodą największej wiarygodności, ale przynależność do grup jest zmienną ukrytą
- E-krok: obliczamy odpowiedzialności, czyli prawdopodobieństwa przynależności punktów do składowych
- M-krok: aktualizujemy średnie, kowariancje i wagi jako średnie ważone odpowiedzialnościami
- Powtarzamy do zbieżności log-wiarygodności; wynik zależy od inicjalizacji (często z k-means), możliwe minima lokalne

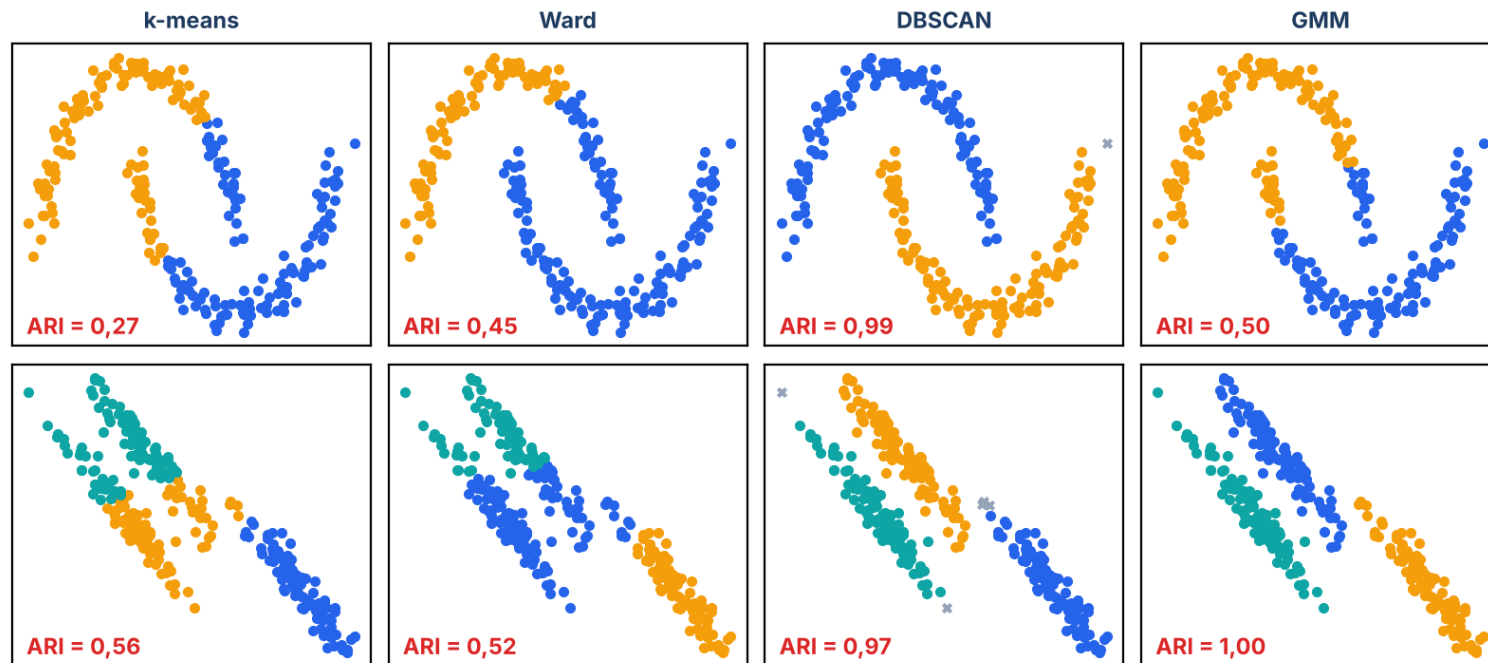
Schemat algorytmu EM



Porównanie algorytmów

- k-means: szybki i prosty, grupy kuliste, wymaga k; GMM: grupy eliptyczne, miękkie przypisanie, wymaga k
- Hierarchiczne: dendrogram i dowolna miara odległości, ale kosztowne dla dużych zbiorów
- DBSCAN: dowolne kształty, szum, bez k; wrażliwy na dobór ϵ i różne gęstości
- Dla tekstu: najpierw TF-IDF i SVD, potem spherical k-means lub hierarchiczne; DBSCAN po redukcji wymiarów

Porównanie algorytmów na dwóch zbiorach



Wnioski z porównania

- Zbiór „księżyców”: DBSCAN odtwarza grupy niemal idealnie ($ARI = 0,99$), k-means tylko na poziomie 0,27
- Wydłużone elipsy: GMM wypada najlepiej ($ARI = 1,00$), a k-means i Ward gorzej (0,56 i 0,52)
- ARI porównuje wynik z prawdziwymi grupami: 1 oznacza zgodność, 0 wynik losowy
- Parametry DBSCAN dobrano ręcznie, aby pokazać możliwości metody; nie ma algorytmu uniwersalnie najlepszego

06

Ocena jakości grupowania i dobór liczby grup

Jak oceniać grupowanie?

- Miary wewnętrzne: korzystają tylko z danych (spójność w grupach, rozdzielenie grup) i nie wymagają etykiet
- Miary zewnętrzne: porównują grupy ze znanymi etykietami (ARI, NMI, czystość) – do walidacji metod
- Ocena użytkowa: czy grupy są interpretowalne i przydatne w danym zastosowaniu (np. spójne tematy)
- Stabilność: powtarzamy grupowanie na podpróbkach i porównujemy wyniki

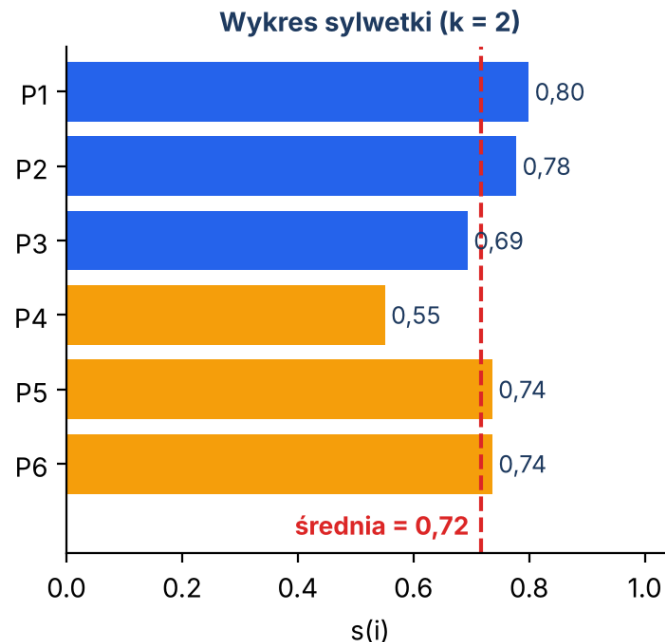
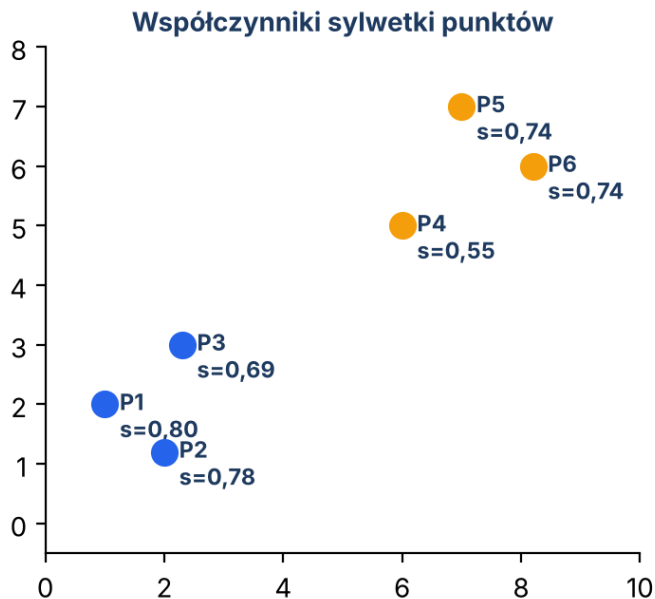
Współczynnik sylwetki (silhouette)

- Dla punktu i : $a(i)$ – średnia odległość do reszty własnej grupy, $b(i)$ – najmniejsza średnia odległość do innej grupy
- $s(i) = (b(i) - a(i)) / \max(a(i), b(i))$; wartości od -1 do 1
- s bliskie 1 : punkt dobrze dopasowany; bliskie 0 : na granicy grup; ujemne: prawdopodobnie w złej grupie
- Średnia sylwetka (reguła orientacyjna): $> 0,7$ silna struktura, $0,5-0,7$ rozsądna, $0,25-0,5$ słaba, $< 0,25$ brak

Przykład liczbowy sylwetki

- Grupy {P1, P2, P3} i {P4, P5, P6} z poprzednich przykładów; liczymy punkt P1
- $a(P1) = (1,28 + 1,64) / 2 = 1,46$; $b(P1) = (5,83 + 7,81 + 8,24) / 3 = 7,29$
- $s(P1) = (7,29 - 1,46) / 7,29 \approx 0,80$
- Pozostałe punkty: 0,78; 0,69; 0,55; 0,74; 0,74; średnia sylwetka $\approx 0,72$, czyli silna struktura

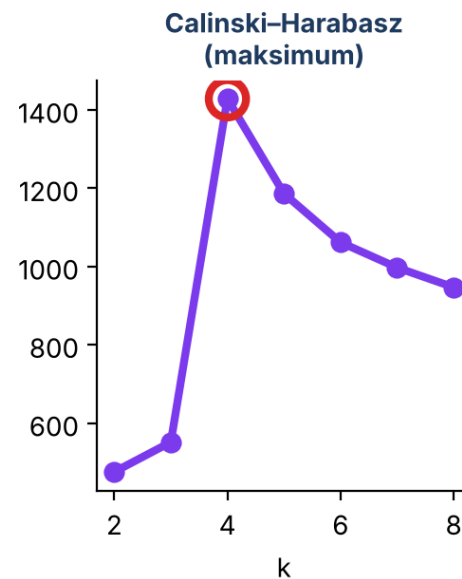
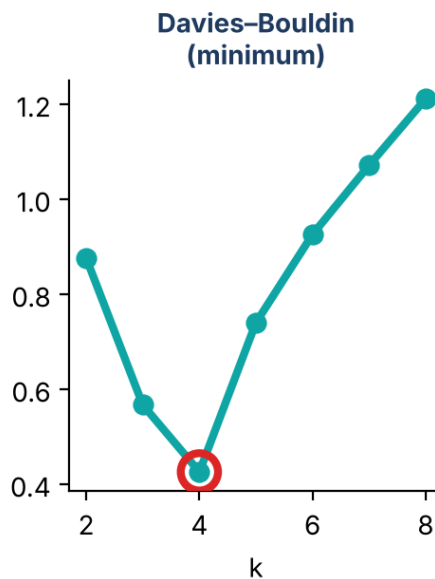
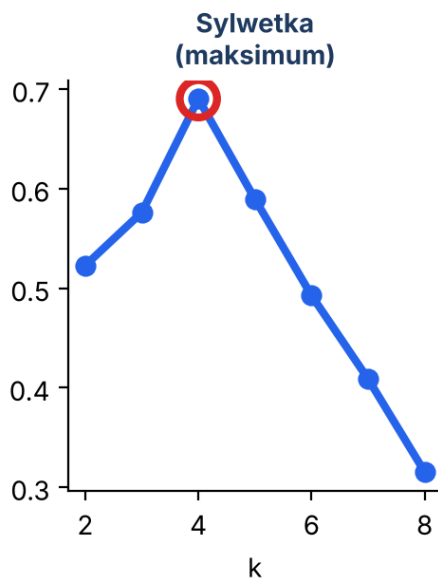
Współczynniki sylwetki



Indeksy Daviesa–Bouldina i Calinskiego–Harabasza

- Davies–Bouldin: średnia po grupach największego stosunku (rozrzut i + rozrzut j) / odległość centroidów; im mniej, tym lepiej
- Calinski–Harabasz: $CH = (B / (k - 1)) / (W / (n - k))$, gdzie B – rozrzut między grupami, W – wewnątrz; im więcej, tym lepiej
- Obie miary są szybkie (bez macierzy odległości), ale faworyzują grupy kuliste i wypukłe
- Najlepsze k wskazujemy, porównując wynik dla różnych liczb grup na tych samych danych

Wskaźniki jakości w funkcji k

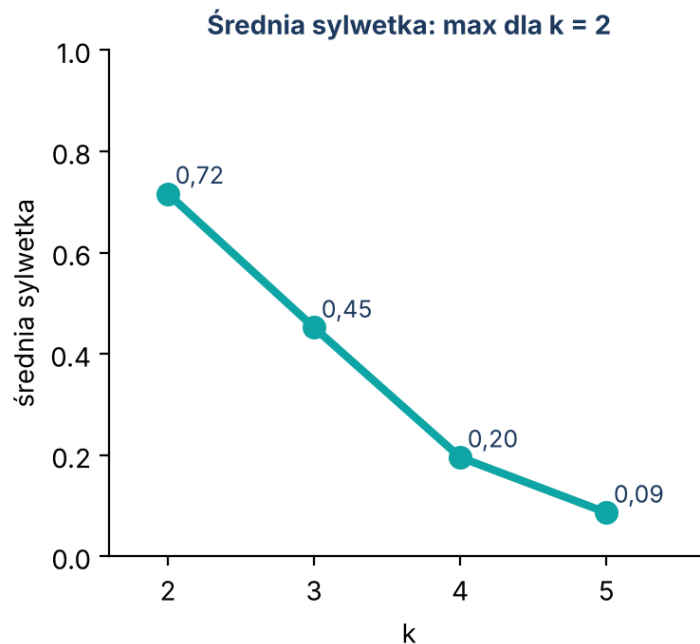
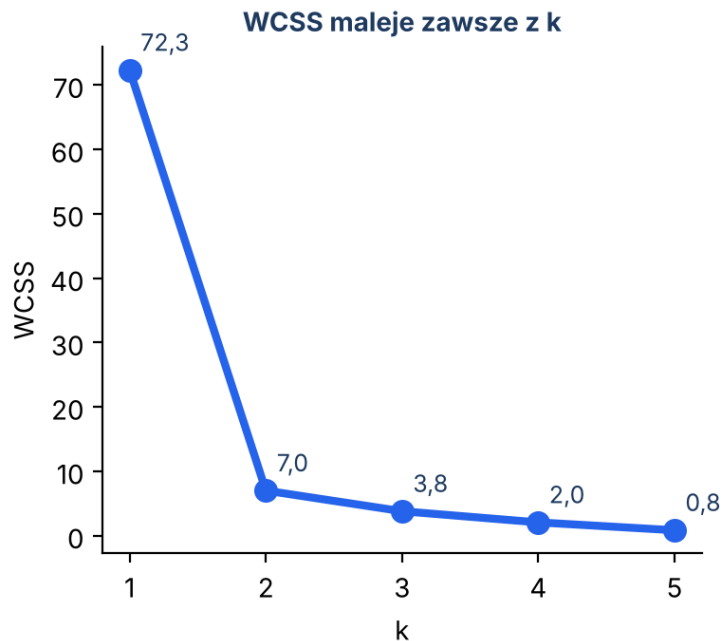


Dane z 4 grupami: wszystkie trzy wskaźniki wskazują $k = 4$

Dobór k – przykład liczbowy

- Dla sześciu punktów WCSS dla $k = 1...5$ wynosi 72,32; 6,98; 3,77; 2,04; 0,82 – maleje zawsze, więc sama nie rozstrzyga
- Średnia sylwetka dla $k = 2...5$: 0,72; 0,45; 0,20; 0,09 – wyraźne maksimum przy $k = 2$
- Łokieć WCSS także wypada przy $k = 2$ (spadek z 72 do 7), co zgadza się z sylwetką
- Wniosek: dwie grupy; dla $n = 6$ większe k oznacza dzielenie naturalnych grup na kawałki

WCSS i sylwetka dla $k = 1...5$



Miary zewnętrzne: czystość, ARI, NMI

- Czystość: każdej grupie przypisujemy dominującą klasę; czystość = suma największych licznosci / n
- ARI: indeks Randa skorygowany o przypadek – 1 zgodność, około 0 wynik losowy, może być ujemny
- NMI: znormalizowana informacja wzajemna, od 0 do 1, niezależna od numeracji grup
- Czystość rośnie z liczbą grup (dla $k = n$ wynosi 1), więc do porównań lepsze są ARI i NMI

Tablica kontyngencji i miary zewnętrzne

Tablica kontyngencji (12 dokumentów)

Grupa 1	4	0	1
Grupa 2	0	3	1
Grupa 3	0	1	2
	Sport	Polityka	Technologia
	rzeczywista klasa		

Czystość

0,75

ARI

0,36

NMI

0,50

Przykład liczbowy: czystość, ARI, NMI

- 12 dokumentów, 3 klasy (sport, polityka, technologia) i 3 grupy z tablicy kontyngencji
- Czystość = $(4 + 3 + 2) / 12 = 0,75$
- ARI $\approx 0,36$ – zgodność wyraźnie lepsza od losowej, ale daleka od doskonałej; NMI $\approx 0,50$
- Błędne przypisania: po jednym dokumencie technologicznym w grupach 1 i 2 oraz jeden polityczny w grupie 3

07

Zastosowania w inteligentnych systemach internetowych

Grupowanie artykułów i newsów

- Cel: automatyczne wykrywanie tematów w strumieniu artykułów, bez ręcznego tagowania
- Potok: TF-IDF $\rightarrow$ SVD $\rightarrow$ k-means (lub hierarchiczne); temat opisują słowa o najwyższych wagach w centroidzie
- Agregatory newsów grupują doniesienia o tym samym wydarzeniu (klastry zdarzeń, wykrywanie duplikatów)
- Wyzwanie: strumień danych – stosuje się grupowanie przyrostowe i okna czasowe

Tematy z grupowania artykułów

Strumień artykułów



TF-IDF
+ k-means



Automatyczne tematy (przykład ilustracyjny)

Grupa 1: Sport

najważniejsze słowa: mecz, liga, trener, gol, sezon

Grupa 2: Gospodarka

najważniejsze słowa: inflacja, stopy, bank, kurs, budżet

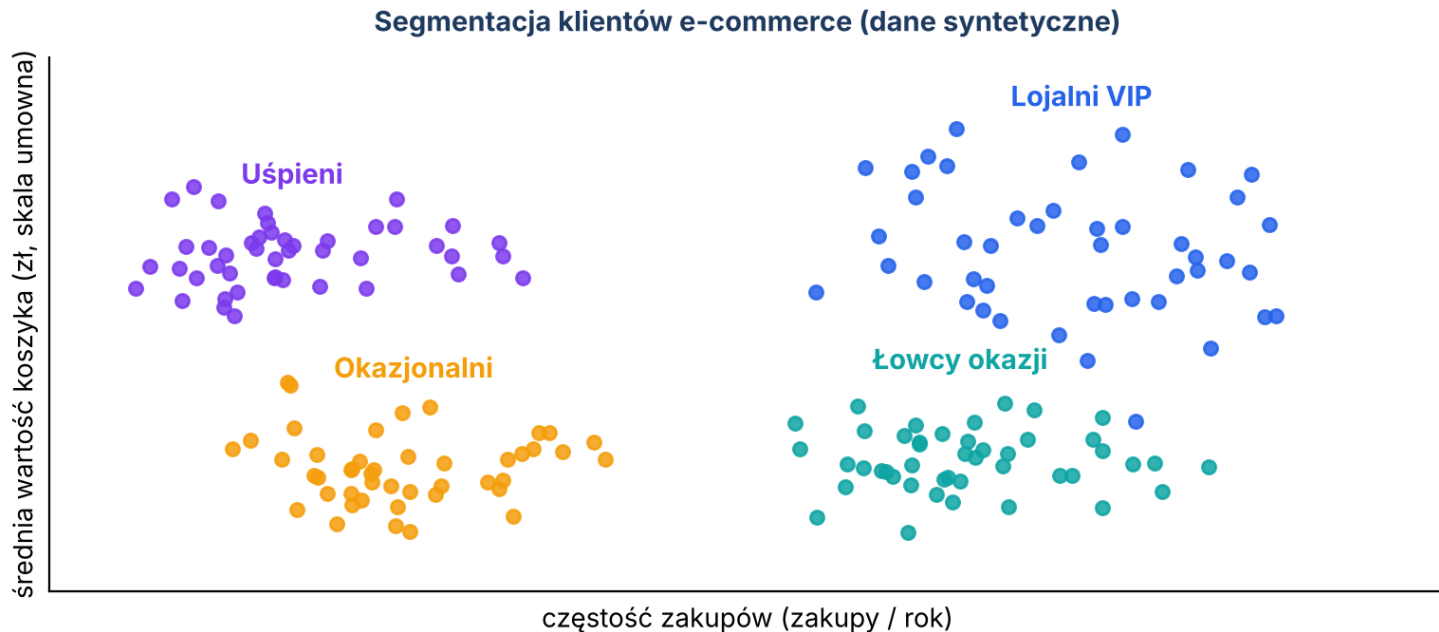
Grupa 3: Technologia

najważniejsze słowa: sieci, model, procesor, aplikacja, dane

Segmentacja klientów w e-commerce

- Cechy: częstość i świeżość zakupów, wartość koszyka (analiza RFM), kategorie produktów, źródło ruchu
- Cechy skalujemy, a grupy interpretujemy na podstawie centroidów (np. lojalni, okazjonalni, uśpieni)
- Segmenty wspierają personalizację: oferty, newslettery, ceny i kampanie retencyjne
- Liczbę segmentów dobieramy miarami jakości i potrzebami biznesu (zwykle kilka do kilkunastu)

Segmentacja klientów

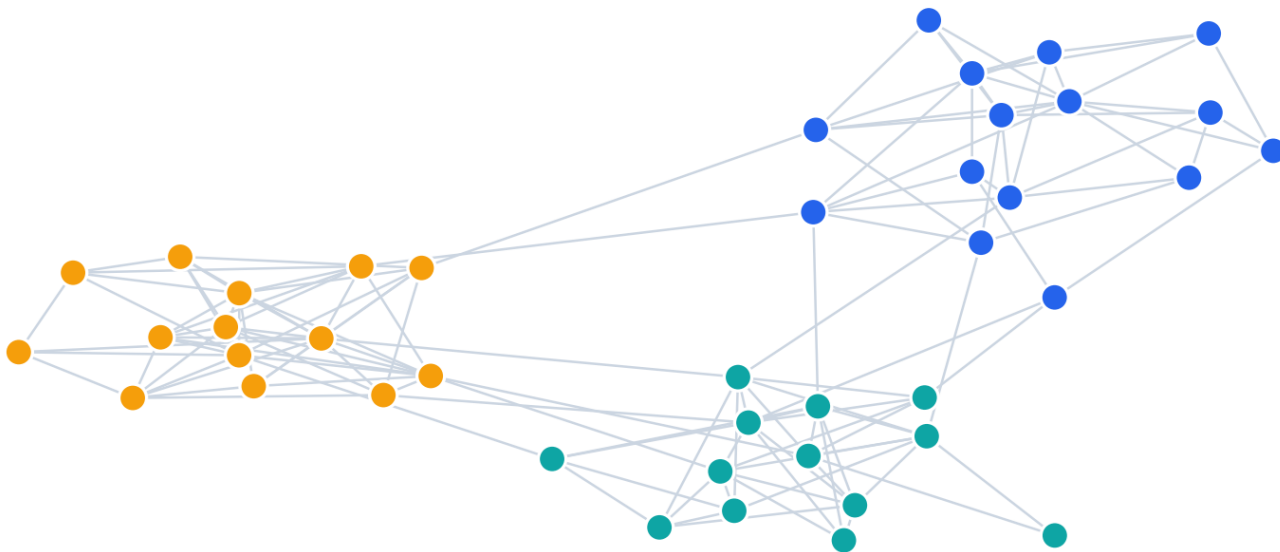


Odkrywanie społeczności w mediach

- Sieć społecznościowa to graf: użytkownicy to wierzchołki, powiązania to krawędzie
- Społeczność to zbiór wierzchołków gęsto połączonych wewnątrznie i rzadko z resztą grafu
- Algorytmy: Louvain, maksymalizacja modularności, grupowanie spektralne, propagacja etykiet
- Zastosowania: grupy zainteresowań, analiza rozprzestrzeniania informacji, wykrywanie skoordynowanych kont

Społeczności w grafie

Wykryte społeczności: 3 (modularność $Q = 0,57$)



Klastrowanie wyników wyszukiwania

- Wyszukiwarka grupuje listę wyników zapytania w tematy, np. „jaguar”: zwierzę, samochód, system operacyjny
- Grupowane są tytuły i fragmenty (snippets) wyników, więc algorytm musi działać w ułamku sekundy
- Etykiety grup tworzy się z najważniejszych fraz (np. algorytmy Suffix Tree Clustering i Lingo)
- Pomaga przy zapytaniach niejednoznacznych: użytkownik zawęży wyniki, wybierając grupę

Makieta klastrowania wyników

Klastrowanie wyników wyszukiwania (makieta)

Grupy wyników

- **Zwierzęta (14)**
- Samochody (11)
- System operacyjny (6)
- Sport i drużyny (4)
- Inne (5)

Wyniki w grupie: Zwierzęta

Jaguar – największy kot Ameryki
tryb życia, występowanie, ochrona gatunku

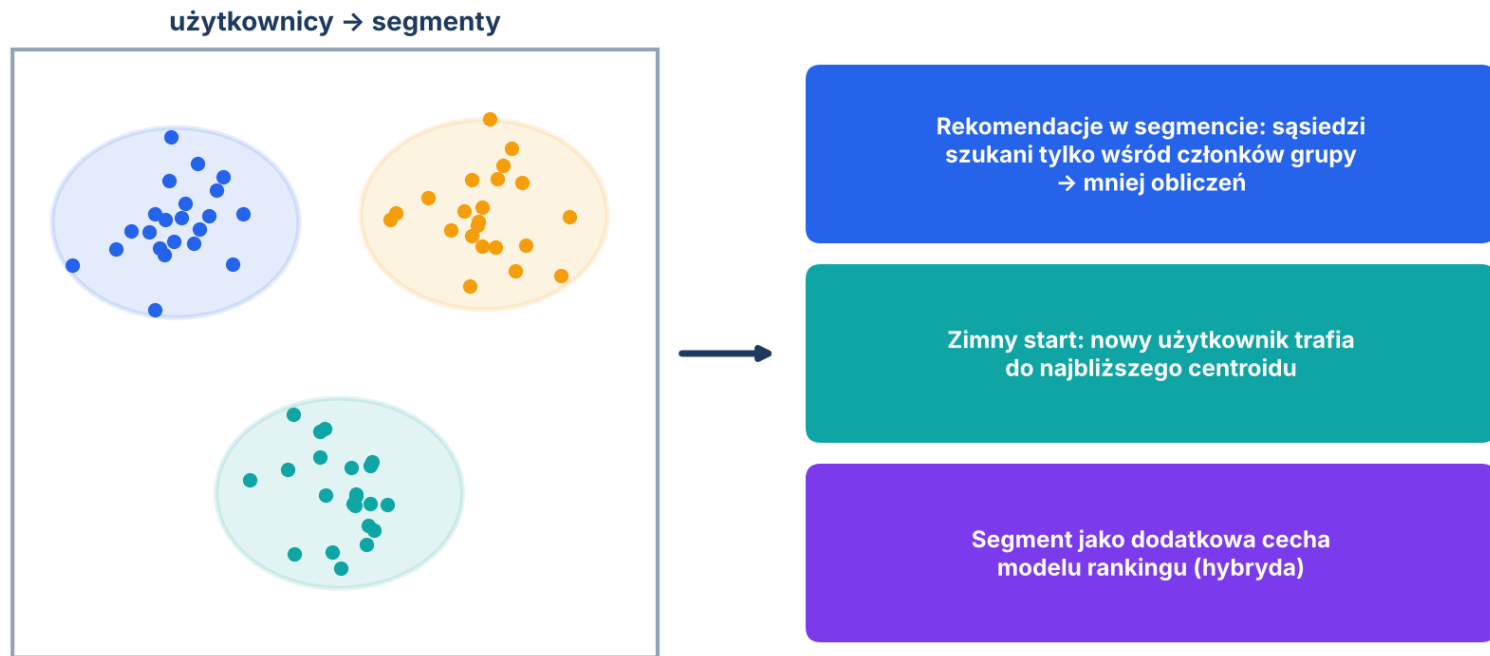
Gatunki wielkich kotów
porównanie jaguara, pantery i lamparta

Jaguar w lasach Amazonii
siedlisko i ochrona drapieżnika

Integracja z systemami rekomendacyjnymi

- Segmentacja użytkowników zmniejsza koszt filtrowania kolaboratywnego: sąsiadów szukamy tylko w obrębie segmentu
- Grupowanie obiektów (produktów, artykułów) pozwala rekomendować pozycje z tej samej grupy i zwiększa różnorodność
- Zimny start: nowego użytkownika lub obiekt przypisujemy do najbliższego centroidu i korzystamy z preferencji grupy
- Hybrydy: etykieta grupy staje się cechą modelu rankingowego lub składową wyniku rekomendacji

Grupowanie w systemie rekomendacyjnym



Dobre praktyki grupowania

- Zacznij od celu: od tego, do czego potrzebne są grupy, zależy wybór reprezentacji, algorytmu i oceny
- Przygotuj dane: czyszczenie, skalowanie, TF-IDF + SVD; porównaj kilka miar podobieństwa
- Uruchom kilka algorytmów i wartości k , oceń miarami wewnętrznymi i ręcznie przejrzyj próbki grup
- Monitoruj stabilność i dryf: grupy trzeba okresowo przeliczać, gdy dane się zmieniają

Podsumowanie i literatura

- Grupowanie odkrywa strukturę w danych bez etykiet; wynik zależy od reprezentacji, miary i algorytmu
- Dla tekstu: TF-IDF, normalizacja i redukcja wymiarów (SVD/LSA) przed grupowaniem
- k-means, hierarchiczne, DBSCAN i GMM dobieramy do kształtu, rozmiaru grup i szumu
- Ocena: sylwetka, Davies–Bouldin, Calinski–Harabasz oraz miary zewnętrzne; liczbę k trzeba uzasadnić
- Manning, Raghavan, Schütze – Introduction to Information Retrieval (rozd. 16–17); Aggarwal, Reddy – Data Clustering
- Narzędzia: scikit-learn, SciPy, HDBSCAN, gensim, NetworkX, Carrot2