

WYKŁAD 2

Systemy rekomendacyjne

Miary podobieństwa i dopasowanie obiektów

Prowadzący: Jarosław Szkoła

Plan wykładu

- 1 Wprowadzenie do systemów rekomendacyjnych
- 2 Miary podobieństwa
- 3 Filtrowanie kolaboratywne
- 4 Metody oparte na treści i dopasowanie obiektów
- 5 Faktoryzacja macierzy i modele latentne
- 6 Systemy hybrydowe i wyzwania praktyczne
- 7 Ewaluacja systemów rekomendacyjnych

01

Wprowadzenie do systemów rekomendacyjnych

Po co systemy rekomendacyjne?

- Nadmiar wyboru: katalogi liczą miliony filmów, produktów, utworów i artykułów
- Cel: wskazać użytkownikowi obiekty, które z dużym prawdopodobieństwem go zainteresują
- Zastosowania: sklepy internetowe, serwisy wideo i muzyczne, media społecznościowe, newsy, reklama
- Korzyści: większe zaangażowanie, sprzedaż i satysfakcja oraz lepsze odkrywanie oferty

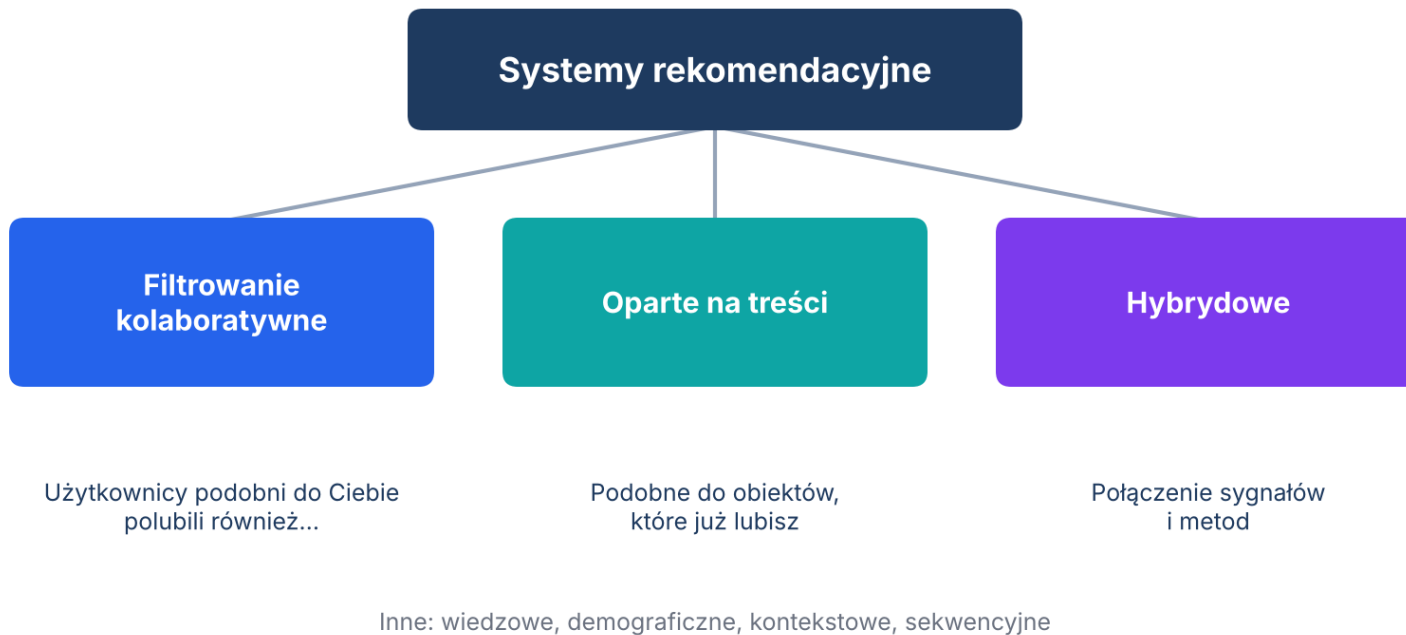
Historia i kamienie milowe

- Lata 90.: pierwsze systemy filtrowania kolaboratywnego (Tapestry, GroupLens)
- 2003: item-to-item collaborative filtering w Amazonie
- 2006–2009: Netflix Prize – milion dolarów za poprawę RMSE o 10%; triumf faktoryzacji macierzy
- Po 2015: głębokie sieci, osadzenia (embeddings), rekomendacje sekwencyjne i kontekstowe

Dane wejściowe: oceny jawne i niejawne

- Feedback jawny: oceny gwiazdkowe, polubienia – rzadki, ale jednoznaczny
- Feedback niejawny: kliknięcia, czas oglądania, zakupy – obfity, ale zaszumiony
- Macierz użytkownik $\times$ obiekt jest bardzo rzadka: zwykle wypełnione jest mniej niż 1% pól
- Dodatkowe dane: atrybuty obiektów i użytkowników oraz kontekst (czas, urządzenie, lokalizacja)

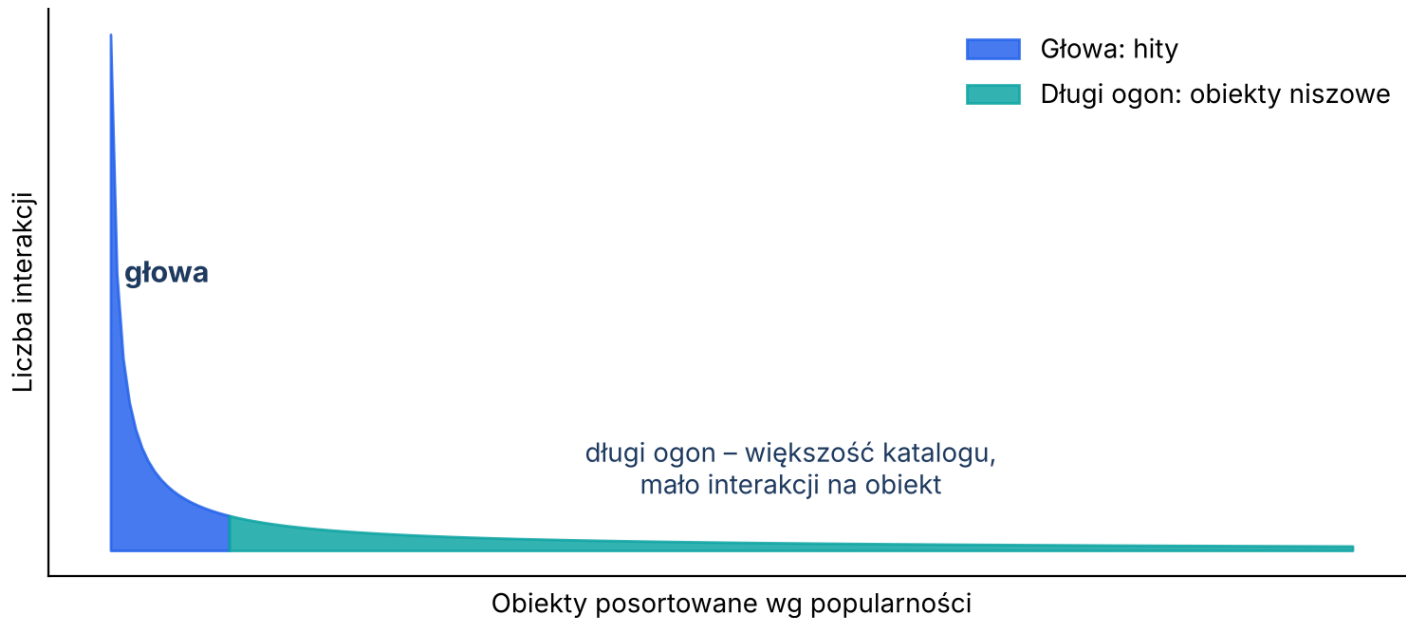
Rodzaje systemów rekomendacyjnych



Zadanie rekomendacji – ujęcie formalne

- Dane: zbiory użytkowników U i obiektów I oraz znane oceny $r(u,i)$ dla niewielkiej części par
- Predykcja oceny: oszacuj $\hat{r}(u,i)$ dla pary, której użytkownik jeszcze nie ocenił
- Rekomendacja top-N: uporządkuj nieocenione obiekty i zwróć N najlepszych
- Oba warianty oceniamy innymi miarami: RMSE i MAE vs. precision@k i NDCG

Efekt długiego ogona



Popularność i długi ogon

- Niewielka liczba hitów zbiera większość interakcji, reszta katalogu to „długi ogon”
- Ranking popularności to dobry punkt odniesienia (baseline), ale nie personalizuje
- Wartość biznesowa: dobre rekomendacje odsłaniają użytkownikom obiekty niszowe
- Problem: dla obiektów z ogona jest mało danych, więc predykcje są niepewne

02

Miary podobieństwa

Po co miary podobieństwa?

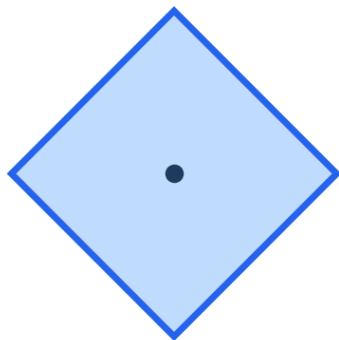
- Podobieństwo (sim) i odległość (d) opisują, jak bliskie są dwa obiekty lub użytkownicy
- Zależność: im większa odległość, tym mniejsze podobieństwo, np. $\text{sim} = 1 / (1 + d)$
- Obiekty reprezentujemy jako wektory: oceny, cechy, wagi TF-IDF lub osadzenia
- Wybór miary zależy od danych: liczbowe, binarne, tekstowe, rzadkie

Odległości Minkowskiego

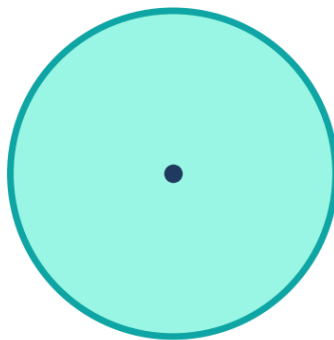
- Wzór: $d(x, y) = (\sum |x_i - y_i|^p)^{1/p}$
- $p = 1$: odległość Manhattan (suma różnic bezwzględnych); $p = 2$: odległość euklidesowa
- $p \rightarrow \infty$: odległość Czebyszewa, czyli największa różnica na jednej współrzędnej
- Wrażliwe na skalę cech (potrzebna standaryzacja) i tracą sens w bardzo wielu wymiarach

Jednostkowe „kule” odległości

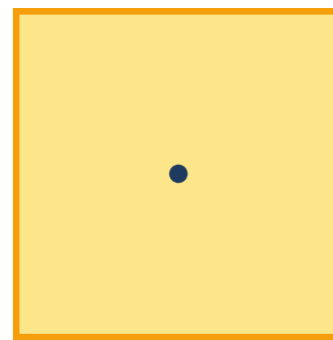
Manhattan (L1)
 $|\Delta x| + |\Delta y|$



Euklidesowa (L2)
 $\sqrt{(\Delta x)^2 + (\Delta y)^2}$



Czebyszewa (L ∞)
 $\max(|\Delta x|, |\Delta y|)$

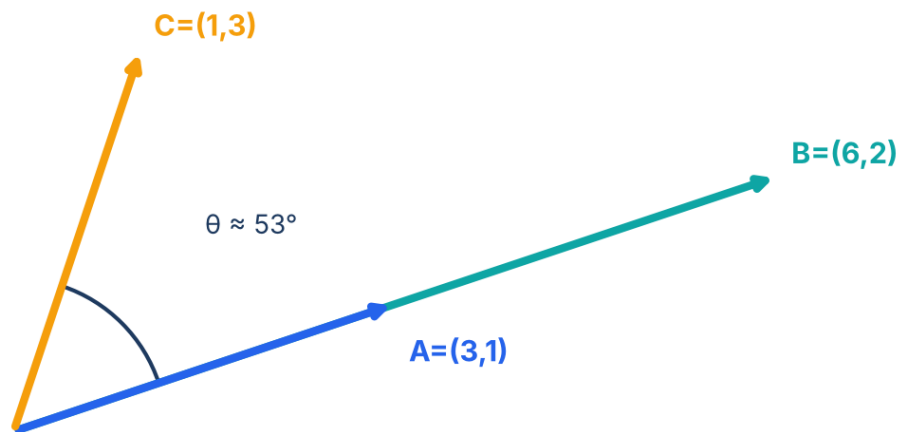


Zbiory punktów w tej samej odległości (=1) od środka

Podobieństwo cosinusowe

- Wzór: $\cos(x, y) = (x \cdot y) / (\|x\| \cdot \|y\|)$ – cosinus kąta między wektorami
- Wartości od -1 do 1; dla nieujemnych ocen i wag – od 0 do 1
- Mierzy kierunek, a nie długość wektora, więc nie zależy od liczby ocen ani skali
- Standard dla danych rzadkich: dokumenty (TF-IDF), oceny, osadzenia

Kąt i podobieństwo cosinusowe



$\cos(A, B) = 1,00$
ten sam kierunek, $d(A,B) \approx 3,2$

$\cos(A, C) = 0,60$
różne kierunki, ale $d(A,C) \approx 2,8$

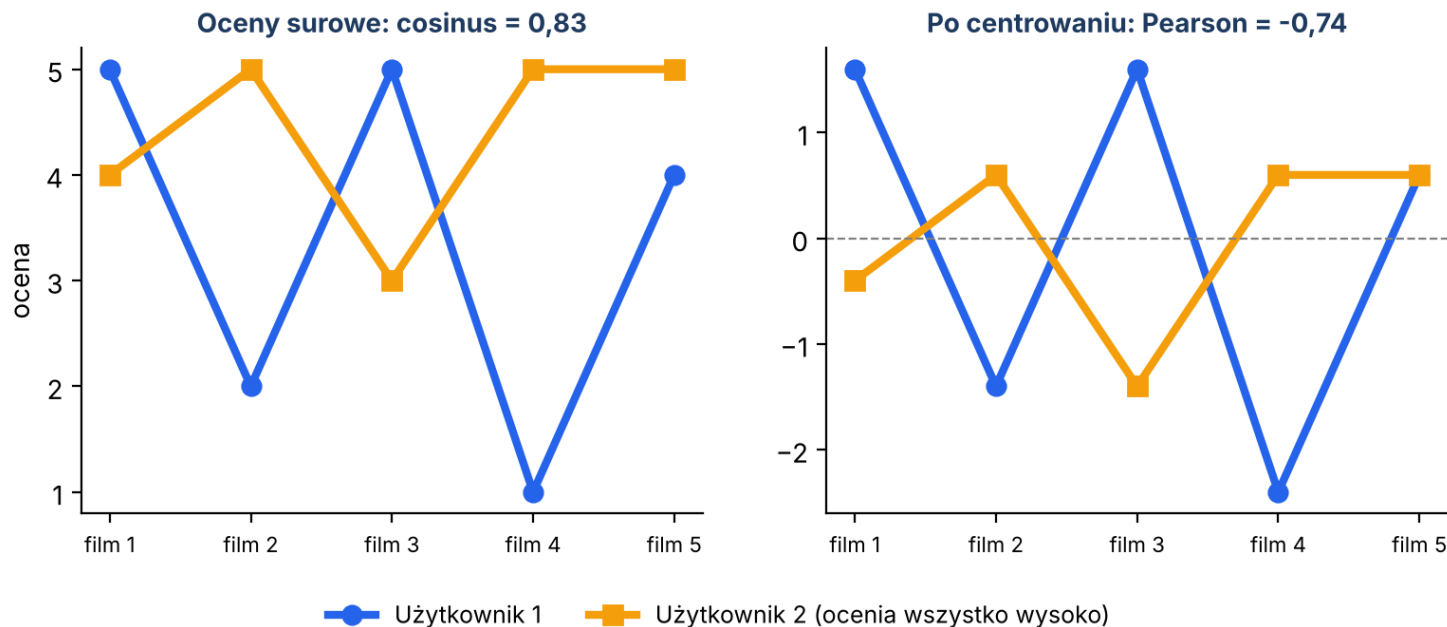
Korelacja Pearsona i skorygowany cosinus

- Pearson: cosinus wektorów po odjęciu średniej ocen, liczony po obiektach ocenionych przez obu
- Usuwa tendencję użytkowników do ocen wysokich lub niskich
- Zakres od -1 (przeciwnie gusta) do 1 (zgodne gusta)
- Skorygowany cosinus (adjusted cosine) – centrowanie względem średniej użytkownika, dla podobieństwa obiektów

Przykład liczbowy: cosinus a Pearson

- Użytkownik 1 ocenił 5 filmów na (5, 2, 5, 1, 4), użytkownik 2 na (4, 5, 3, 5, 5)
- Cosinus $\approx 0,83$ – sugeruje duże podobieństwo, bo oba zestawy mają wysokie oceny
- Pearson $\approx -0,74$ – po usunięciu średnich widać, że gusta są przeciwne
- Wniosek: bez centrowania miara myli się, gdy użytkownicy używają różnych skal ocen

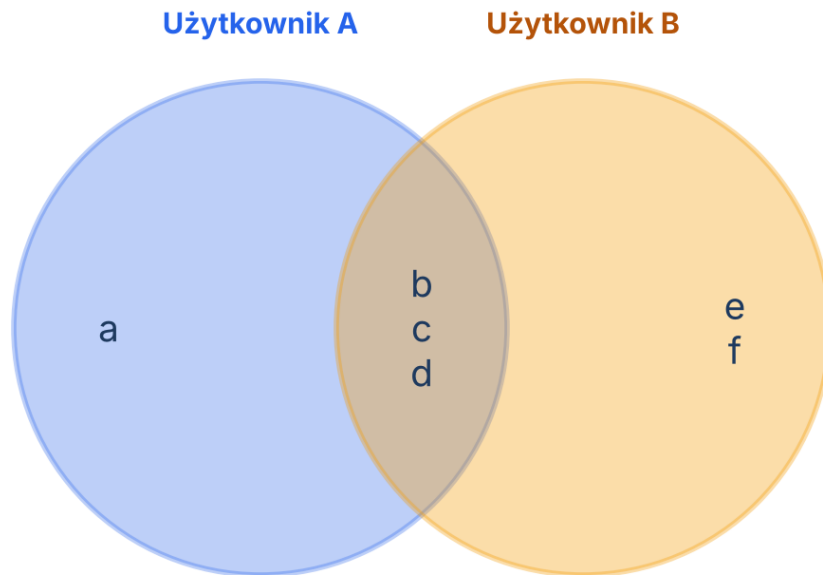
Surowy cosinus a korelacja Pearsona



Indeks Jaccarda i miary dla danych binarnych

- Jaccard: $J(A, B) = |A \cap B| / |A \cup B|$ – dla zbiorów, np. kupionych produktów
- Odległość Jaccarda = $1 - J$; wspólne braki (nieobecne obiekty) nie zwiększają podobieństwa
- Współczynnik Dice'a: $2|A \cap B| / (|A| + |B|)$; odległość Hamminga dla wektorów binarnych
- Dla dużych zbiorów: MinHash i LSH szybko szacują podobieństwo Jaccarda

Indeks Jaccarda



$$J(A, B) = |A \cap B| / |A \cup B|$$

$$= 3 / 6 = 0,5$$

Odległość Jaccarda
 $1 - J = 0,5$

Dane binarne: kupić / nie kupić,
obejrzeć / nie obejrzeć

Inne miary i dobór miary

- Tekst: cosinus wektorów TF-IDF; odległość edycyjna Levenshteina dla krótkich ciągów
- Mahalanobis uwzględnia korelacje cech; miary Kullbacka–Leiblera i Jensena–Shannona porównują rozkłady
- Oceny: Pearson lub cosinus; zakupy i kliknięcia: Jaccard lub cosinus; cechy liczbowe: euklidesowa
- Najlepszą miarę wybieramy empirycznie, porównując kilka na danych walidacyjnych

03

Filtrowanie kolaboratywne

Idea filtrowania kolaboratywnego

- Założenie: użytkownicy zgodni w przeszłości będą zgodni także w przyszłości
- Wykorzystuje wyłącznie macierz interakcji – nie potrzebuje opisu obiektów
- Dwa warianty: metody pamięciowe (sąsiedztwo) i modelowe (faktoryzacja, sieci)
- Zalety: odkrywa nieoczywiste powiązania; wady: zimny start i rzadkość danych

User-based CF – algorytm

- Krok 1: oblicz podobieństwo $\text{sim}(u, v)$ użytkownika u do pozostałych użytkowników
- Krok 2: wybierz k najbliższych sąsiadów, którzy ocenili obiekt i
- Krok 3: $\hat{r}(u,i) = \text{średnia}(u) + \frac{\sum \text{sim}(u,v) \cdot (r(v,i) - \text{średnia}(v))}{\sum |\text{sim}(u,v)|}$
- Krok 4: uporządkuj nieocenione obiekty według $\hat{r}(u,i)$ i zwróć N najlepszych

Przykład liczbowy: user-based

- Ty oceniłeś filmy A–D na (5, 4, 2, 1); film E nie został jeszcze oceniony
- Najbliżsi sąsiedzi ($k = 2$): użytkownik 3 (cosinus $\approx 0,99$) i użytkownik 2 (cosinus $\approx 0,96$)
- Ich oceny filmu E to 4 i 5; średnia ważona podobieństwem: $(0,96 \cdot 5 + 0,99 \cdot 4) / 1,95 \approx 4,5$
- Film E trafia wysoko na listę rekomendacji dla Ciebie

User-based CF – predykcja oceny

k = 2 najbliższych sąsiadów (obramowani), średnia ważona podobieństwem

Ty	5	4	2	1	?	Predykcja filmu E: ≈ 4,5	
Użytkownik 2	4	5	3	2	5		sim = 0,96
Użytkownik 3	5	3	2	1	4		sim = 0,99
Użytkownik 4	1	2	5	4	2		sim = 0,59
Użytkownik 5	2	1	4	5	1		sim = 0,59
	Film A	Film B	Film C	Film D	Film E		

Item-based CF

- Podobieństwo liczymy między obiektami (kolumnami macierzy ocen), a nie między użytkownikami
- Predykcja: średnia ważona ocen użytkownika dla obiektów podobnych do i
- Przykład: obiekty najbliższe E to B i A (cosinus 0,99 i 0,96), oceny T_y : 4 i 5, wynik $\approx 4,5$
- Podobieństwa obiektów są stabilniejsze i można je liczyć offline (Amazon, 2003)

Macierz podobieństwa obiektów

Podobieństwo obiektów: cosinus kolumn macierzy ocen (użytkownicy 2-5)

Film A	1,00	0,92	0,70	0,59	0,96
Film B	0,92	1,00	0,76	0,61	0,99
Film C	0,70	0,76	1,00	0,96	0,74
Film D	0,59	0,61	0,96	1,00	0,59
Film E	0,96	0,99	0,74	0,59	1,00
	Film A	Film B	Film C	Film D	Film E

Dobór sąsiadów, normalizacja i skalowanie

- Liczbę sąsiadów k (zwykle 20–100) i próg podobieństwa dobieramy walidacją
- Podobieństwa z kilku wspólnych ocen są niewiarygodne – stosuje się tłumienie (shrinkage)
- Normalizacja ocen: centrowanie względem średniej użytkownika lub z-score
- Skalowanie: przybliżone wyszukiwanie sąsiadów (LSH, HNSW) zamiast pełnego przeglądu

Zalety i wady metod sąsiedztwa

- Zalety: prostota, łatwa interpretacja („kupili też...”), brak etapu uczenia
- Nowe oceny można uwzględnić od razu, bez ponownego trenowania modelu
- Wady: wolne działanie online (user-based) i wrażliwość na rzadkość danych
- Słabo wykorzystują globalną strukturę danych – tę lukę wypełniają modele latentne

04

Metody oparte na treści i dopasowanie obiektów

Idea rekomendacji opartej na treści

- Rekomendujemy obiekty podobne do tych, które użytkownik wcześniej polubił
- Potrzebny jest opis obiektów: gatunek, cena, kategoria, tekst opisu, obraz
- Nie wymaga danych o innych użytkownikach i działa dla nowych obiektów
- Ryzyko nadmiernej specjalizacji: system poleca ciągle to samo

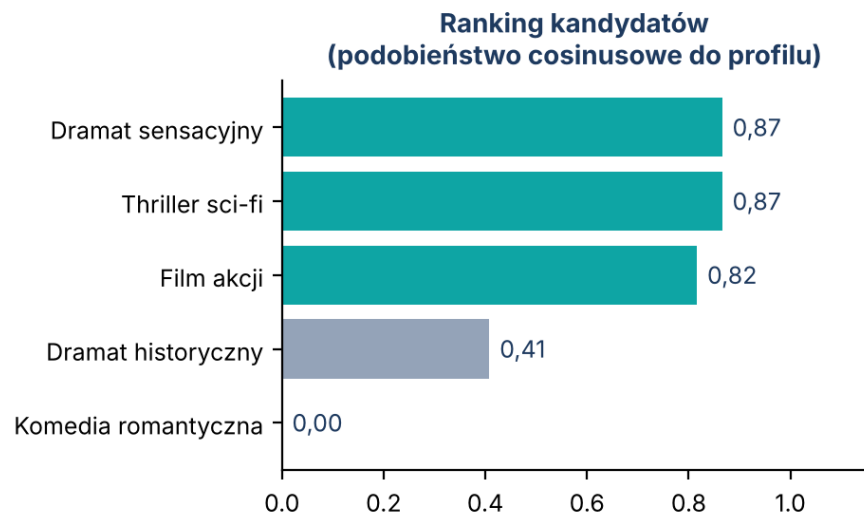
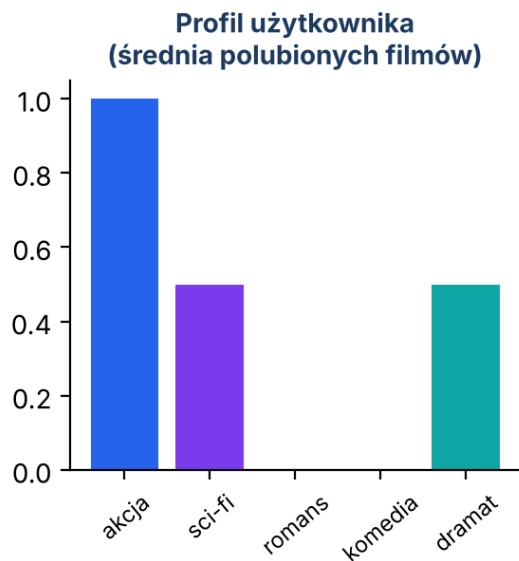
Reprezentacja obiektów

- Cechy kategoryjne: kodowanie one-hot lub multi-hot (np. gatunki filmu)
- Tekst: worek słów i TF-IDF, gdzie $TF\text{-}IDF(t, d) = tf(t, d) \cdot \log(N / df(t))$
- Obrazy i dźwięk: wektory cech z sieci neuronowych (np. CNN)
- Cechy liczbowe skalujemy, aby żadna nie dominowała w odległości

Profil użytkownika i ranking kandydatów

- Profil = średnia (ważona ocenami) wektorów obiektów, które użytkownik polubił
- Alternatywa: osobny klasyfikator lub regresor uczony dla każdego użytkownika
- Ranking: sortowanie kandydatów według podobieństwa do profilu
- Zanikanie wag sprawia, że nowsze preferencje liczą się bardziej

Profil użytkownika i ranking



Dopasowanie obiektów (item matching)

- Zadanie: znaleźć obiekty najbardziej podobne do danego, np. sekcja „Podobne produkty”
- Wyszukiwanie k najbliższych sąsiadów (k-NN) w przestrzeni cech lub osadzeń
- Pełne porównanie kosztuje $O(n)$ na zapytanie – dla milionów obiektów to za dużo
- Dopasowanie rekordów (entity resolution): rozpoznanie, że dwa opisy dotyczą tego samego obiektu

Osadzenia i przybliżone wyszukiwanie

- Embedding: gęsty wektor (np. 64–512 wymiarów) uczony tak, by podobne obiekty leżały blisko
- Źródła: faktoryzacja macierzy, item2vec na sekwencjach, sieci neuronowe, modele językowe
- ANN (approximate nearest neighbors): LSH, HNSW, kwantyzacja produktowa – szybko, lecz przybliżenie
- Biblioteki: FAISS, ScaNN, Annoy; kompromis między dokładnością a szybkością

Zalety i wady podejścia opartego na treści

- Zalety: brak zimnego startu obiektu, wyjaśnialność, niezależność od innych użytkowników
- Wady: potrzeba dobrych cech, mała różnorodność, słabe odkrywanie nowych zainteresowań
- Nowy użytkownik nadal potrzebuje kilku polubień, aby powstał profil
- W praktyce łączy się ją z filtrowaniem kolaboratywnym w systemach hybrydowych

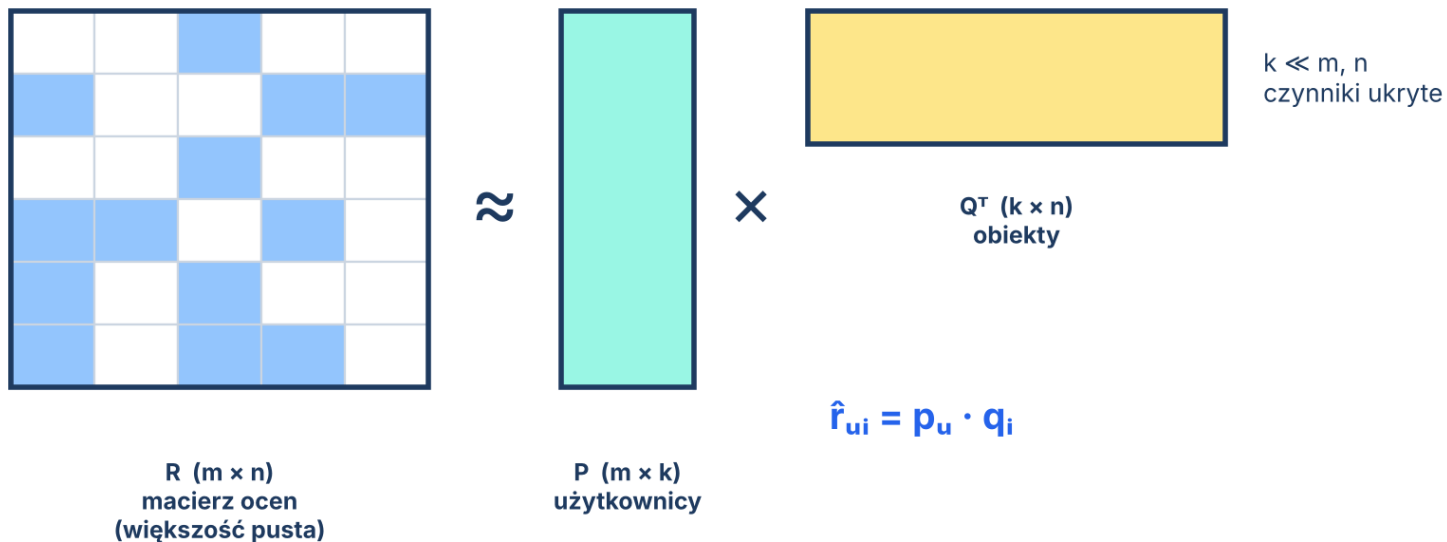
05

Faktoryzacja macierzy i modele latentne

Idea faktoryzacji macierzy

- Rozkład $R \approx P \cdot Q^T$: każdy użytkownik i obiekt dostaje wektor k czynników ukrytych
- Czynniki odpowiadają ukrytym cechom, np. „akcja vs. romans”, „poważny vs. lekki”
- Predykcja oceny: $\hat{r}(u,i) = p(u) \cdot q(i)$, czyli iloczyn skalarny wektorów
- Mała liczba czynników ($k \ll m, n$) kompresuje dane i uogólnia na brakujące oceny

Faktoryzacja macierzy ocen



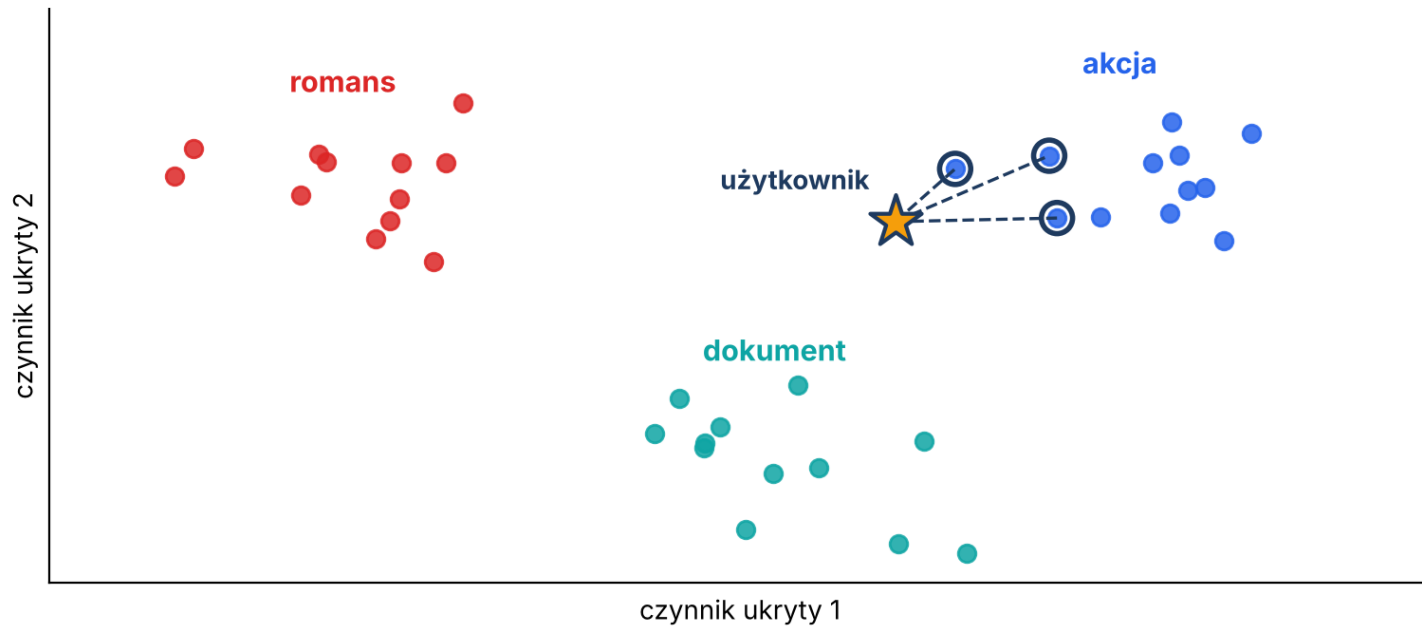
Funkcja celu i uczenie

- Minimalizujemy $\sum (r(u,i) - p(u) \cdot q(i))^2 + \lambda(\|p(u)\|^2 + \|q(i)\|^2)$ po znanych ocenach
- SGD: dla każdej oceny $e = r - p \cdot q$, potem $p \leftarrow p + \gamma(e \cdot q - \lambda p)$ i $q \leftarrow q + \gamma(e \cdot p - \lambda q)$
- ALS: naprzemienne rozwiązywanie układów równań dla P i Q – łatwe do zrównoleglenia
- Hiperparametry k , λ , γ i liczbę epok dobieramy walidacją krzyżową

Rozszerzenia modelu

- Obciążenia: $\hat{r} = \mu + b(u) + b(i) + p(u) \cdot q(i)$ – średnia, skłonność użytkownika, popularność obiektu
- SVD++ i modele zależne od czasu: niejawnny feedback i zmienność gustów
- Dane niejawne: ważony ALS (confidence) oraz BPR, czyli uczenie rankingu zamiast ocen
- Neural CF i modele dwuwieżowe (two-tower): sieci uczą osadzenia użytkowników i obiektów

Przestrzeń czynników ukrytych



06

Systemy hybrydowe i wyzwania praktyczne

Problem zimnego startu

- Nowy użytkownik: brak historii – pytanie o preferencje, popularność, kontekst
- Nowy obiekt: brak ocen – korzystamy z cech treści i metadanych
- Nowy system: brak danych – reguły, listy popularnych i eksperymenty
- Algorytmy typu multi-armed bandit równoważą eksplorację i eksploatację

Systemy hybrydowe

- Ważona: łączymy wyniki kilku metod z wagami uczonymi na danych walidacyjnych
- Przełączająca: wybór metody zależnie od sytuacji (np. treść dla nowych obiektów)
- Kaskadowa: wynik jednej metody jest wejściem drugiej
- Modele łączące cechy i interakcje: factorization machines, LightFM, sieci głębokie

Architektura wieloetapowa

Architektura wieloetapowa systemu rekomendacji



szybkie, przybliżone → dokładne, kosztowne → różnorodność i reguły biznesowe

Architektura produkcyjna

- Etap 1, generowanie kandydatów: szybkie ANN lub kilka źródeł, tysiące obiektów
- Etap 2, ranking: dokładniejszy model (gradient boosting, sieć) ocenia kandydatów
- Etap 3, re-ranking: różnorodność, świeżość, reguły biznesowe i filtry
- Offline: trenowanie i budowa indeksów; online: opóźnienia rzędu dziesiątek milisekund

Różnorodność, bańka filtrująca i etyka

- Bańka filtrująca: personalizacja zawęży zakres treści widzianych przez użytkownika
- Różnorodność, nowość i serendipity uzupełniają miary trafności
- Uprzedzenie popularności i sprzężenie zwrotne wzmacniają hity
- Prywatność i przejrzystość: zgoda na dane, wyjaśnialność, RODO i DSA

07

Ewaluacja systemów rekomendacyjnych

Ocena offline: błędy predykcji

- Podział danych: losowy, czasowy (przyszłość jako test) lub leave-one-out
- $MAE = \text{średnia z } |r - \hat{r}|$; $RMSE = \text{pierwiastek ze średniej z } (r - \hat{r})^2$
- RMSE silniej karze duże błędy; Netflix Prize oceniał właśnie RMSE
- Niski RMSE nie gwarantuje dobrej listy top-N, więc liczy się też jakość rankingu

Metryki rankingowe

- $\text{Precision@k} = \text{trafne w top-k} / k$; $\text{Recall@k} = \text{trafne w top-k} / \text{wszystkie trafne}$
- AP i MAP: średnia precision w pozycjach trafień, uśredniana po użytkownikach
- $\text{DCG@k} = \sum \text{rel}(i) / \log_2(i + 1)$; $\text{NDCG} = \text{DCG} / \text{IDCG}$ uwzględnia pozycję trafień
- MRR: odwrotność pozycji pierwszego trafnego obiektu

Metryki jakości list rekomendacji

Lista rekomendacji (✓ = obiekt trafny; w całym zbiorze są 4 trafne)

top-5



P@5	3/5 = 0,60	R@5	3/4 = 0,75	NDCG@5	0,74	AP@10	0,69
-----	------------	-----	------------	--------	------	-------	------

$$\text{NDCG@5} = \text{DCG} / \text{IDCG} = 1,89 / 2,56$$

$$\text{AP@10} = \text{średnia z precision w pozycjach trafień (1/1, 2/3, 3/5, 4/8)}$$

Przykład liczbowy metryk rankingowych

- Lista 10 obiektów, trafne na pozycjach 1, 3, 5 i 8; w zbiorze są 4 trafne obiekty
- $\text{Precision@5} = 3/5 = 0,60$; $\text{Recall@5} = 3/4 = 0,75$
- $\text{DCG@5} = 1 + 0,5 + 0,39 \approx 1,89$; $\text{IDCG@5} \approx 2,56$; $\text{NDCG@5} \approx 0,74$
- $\text{AP@10} = (1/1 + 2/3 + 3/5 + 4/8) / 4 \approx 0,69$

Metryki poza trafnością i testy online

- Pokrycie katalogu (coverage), różnorodność, nowość i serendipity
- Testy A/B: porównanie wariantów na prawdziwych użytkownikach
- Metryki biznesowe: CTR, konwersja, czas oglądania, retencja
- Ocena offline bywa obciążona, bo użytkownik widział tylko to, co pokazał system

Podsumowanie i literatura

- Systemy rekomendacyjne łączą filtrowanie kolaboratywne, treść, modele latentne i hybrydy
- Miara podobieństwa i reprezentacja obiektów decydują o jakości dopasowania
- Ewaluacja obejmuje trafność, różnorodność i testy online
- Ricci, Rokach, Shapira – Recommender Systems Handbook
- Aggarwal – Recommender Systems: The Textbook; Koren, Bell, Volinsky (2009) – Matrix Factorization Techniques
- Narzędzia: Surprise, LightFM, implicit, Spark MLlib, FAISS