

Rekomendacje dla grup

Systemy Rekomendacyjne 2024/2025

Definicja problemu

- Wszystkie dotychczasowe algorytmy generowały rekomendacje dla pojedynczych użytkowników
 - Nawet jeśli uczyliśmy się na danych z populacji
- Jak tworzyć rekomendacje dla dowolnych grup użytkowników?
- Jakie powinny być funkcje celu?
- Jak godzić preferencje, zwłaszcza te rozbieżne?

Definicja problemu

- Dane są przewidywane oceny każdego elementu przez każdego z użytkowników
- Chcemy dobrać rekomendację tak, by zadowolenie w całej grupie było jak największe
- Źródło: <https://towardsdatascience.com/an-introduction-to-group-recommender-systems-8f942a06db56>

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10

Algorytmy proste

Dyktatura

- Decyduje jedna osoba

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10

Dyktatura

- Decyduje jedna osoba

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10

Dyktatura

- Decyduje jedna osoba

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10

Metoda sprawiedliwa (a.k.a. dyktatura przechodnia)

- Osoby z grupy ustalają między sobą kolejność
- W każdej iteracji o wyborze elementu decyduje kolejna osoba

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10

Metoda sprawiedliwa (a.k.a. dyktatura przechodnia)

- Osoby z grupy ustalają między sobą kolejność
- W każdej iteracji o wyborze elementu decyduje kolejna osoba

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10

Metoda sprawiedliwa (a.k.a. dyktatura przechodnia)

- Osoby z grupy ustalają między sobą kolejność
- W każdej iteracji o wyborze elementu decyduje kolejna osoba

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10

Metoda sprawiedliwa (a.k.a. dyktatura przechodnia)

- Osoby z grupy ustalają między sobą kolejność
- W każdej iteracji o wyborze elementu decyduje kolejna osoba

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10

Maksymalizacja zadowolenia

- Wybieramy filmy o najwyższej maksymalnej ocenie

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10

Maksymalizacja zadowolenia

- Wybieramy filmy o najwyższej maksymalnej ocenie

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10

Minimalizacja niezadowolenia

- Wybieramy filmy o najwyższej minimalnej ocenie

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10

Minimalizacja niezadowolenia

- Wybieramy filmy o najwyższej minimalnej ocenie

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10

Algorytmy konsensusu

Średnia

- Waga każdego filmu to średnia ocen wystawionych przez użytkowników
- Wybieramy filmy z najwyższą wagą

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10
Score	$4\frac{1}{3}$	$5\frac{2}{3}$	$8\frac{2}{3}$	$7\frac{1}{3}$	$7\frac{1}{3}$	7

Średnia

- Waga każdego filmu to średnia ocen wystawionych przez użytkowników
- Wybieramy filmy z najwyższą wagą

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10
Score	$4\frac{1}{3}$	$5\frac{2}{3}$	$8\frac{2}{3}$	$7\frac{1}{3}$	$7\frac{1}{3}$	7

Średnia bez niezadowolenia (*without misery*)

- Najpierw eliminujemy wszystkie filmy, które uzyskały co najmniej jedną ocenę poniżej wartości granicznej
- Waga każdego filmu to średnia ocen wystawionych przez użytkowników
- Wybieramy filmy z najwyższą wagą

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10
Score	$4\frac{1}{3}$	$5\frac{2}{3}$	$8\frac{2}{3}$	$7\frac{1}{3}$	$7\frac{1}{3}$	7

Średnia bez niezadowolenia (*without misery*)

- Najpierw eliminujemy wszystkie filmy, które uzyskały co najmniej jedną ocenę poniżej wartości granicznej
- Waga każdego filmu to średnia ocen wystawionych przez użytkowników
- Wybieramy filmy z najwyższą wagą

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10
Score	$4\frac{1}{3}$	$5\frac{2}{3}$	$8\frac{2}{3}$	$7\frac{1}{3}$	$7\frac{1}{3}$	7

Średnia bez niezadowolenia (*without misery*)

- Najpierw eliminujemy wszystkie filmy, które uzyskały co najmniej jedną ocenę poniżej wartości granicznej
- Waga każdego filmu to średnia ocen wystawionych przez użytkowników
- Wybieramy filmy z najwyższą wagą

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10
Score	$4\frac{1}{3}$	$5\frac{2}{3}$	$8\frac{2}{3}$	$7\frac{1}{3}$	$7\frac{1}{3}$	7

Metoda multiplikatywna

- Waga każdego filmu to iloczyn ocen wystawionych przez użytkowników
- Wybieramy filmy z najwyższą wagą
- Zamiast iloczynu możemy także użyć np. średniej geometrycznej lub harmonicznej

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10
Score	48	180	630	384	378	100

Metoda multiplikatywna

- Waga każdego filmu to iloczyn ocen wystawionych przez użytkowników
- Wybieramy filmy z najwyższą wagą
- Zamiast iloczynu możemy także użyć np. średniej geometrycznej lub harmonicznej

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10
Score	48	180	630	384	378	100

Algorytmy wyborcze

Głosowanie proste

- Każdy film dostaje tyle punktów, ile ma ocen powyżej wartości granicznej
- Wybieramy filmy o największej sumie punktów

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10
Scores ≥ 5	1	3	3	3	3	2

Głosowanie proste

- Każdy film dostaje tyle punktów, ile ma ocen powyżej wartości granicznej
- Wybieramy filmy o największej sumie punktów

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10
Scores ≥ 5	1	3	3	3	3	2

Algorytm Bordy

- Każdy użytkownik przyznaje elementom punkty – od 0 punktów dla elementu najmniej lubianego do N punktów dla elementu najbardziej lubianego
- Sumujemy liczby punktów dla każdego elementu
- Wybieramy elementy z największą sumą punktów

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10

Algorytm Bordy

- Każdy użytkownik przyznaje elementom punkty – od 0 punktów dla elementu najmniej lubianego do N punktów dla elementu najbardziej lubianego
- Sumujemy liczby punktów dla każdego elementu
- Wybieramy elementy z największą sumą punktów

	A	B	C	D	E	F
Alice	3 → 0	6 → 2	10 → 5	8 → 3	6 → 2	10 → 5
Bob	8 → 4	6 → 1	7 → 2	8 → 4	9 → 5	1 → 0
Carol	2 → 0	5 → 1	9 → 4	6 → 2	7 → 3	10 → 5

Algorytm Bordy

- Każdy użytkownik przyznaje elementom punkty – od 0 punktów dla elementu najmniej lubianego do N punktów dla elementu najbardziej lubianego
- Sumujemy liczby punktów dla każdego elementu
- Wybieramy elementy z największą sumą punktów

	A	B	C	D	E	F
Alice	0	2	5	3	2	5
Bob	4	1	2	4	5	0
Carol	0	1	4	2	3	5

Algorytm Bordy

- Każdy użytkownik przyznaje elementom punkty – od 0 punktów dla elementu najmniej lubianego do N punktów dla elementu najbardziej lubianego
- Sumujemy liczby punktów dla każdego elementu
- Wybieramy elementy z największą sumą punktów

	A	B	C	D	E	F
Alice	0	2	5	3	2	5
Bob	4	1	2	4	5	0
Carol	0	1	4	2	3	5
Score	4	4	11	9	10	10

Algorytm Bordy

- Każdy użytkownik przyznaje elementom punkty – od 0 punktów dla elementu najmniej lubianego do N punktów dla elementu najbardziej lubianego
- Sumujemy liczby punktów dla każdego elementu
- Wybieramy elementy z największą sumą punktów

	A	B	C	D	E	F
Alice	0	2	5	3	2	5
Bob	4	1	2	4	5	0
Carol	0	1	4	2	3	5
Score	4	4	11	9	10	10

Algorytm Copelanda

- Dla każdej pary elementów i, j obliczamy różnicę między liczbą zwolenników elementu i oraz elementu j
 - Punkt dostaje ten element, który ma więcej zwolenników
 - W przypadku remisu elementy otrzymują po pół punktu
- Wybieramy elementy z największą sumą punktów

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10

Algorytm Copelanda

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10

	A	B	C	D	E	F
A	-					
B		-				
C			-			
D				-		
E					-	
F						-

Algorytm Copelanda

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10

	A	B	C	D	E	F
A	-	<	<	<	<	<
B		-	<	<	<	<
C			-	>	>	=
D				-	<	<
E					-	<
F						-

	Wygrane
A	0
B	1
C	4½
D	2
E	3
F	4½

Algorytm Copelanda

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10

	A	B	C	D	E	F
A	-	<	<	<	<	<
B		-	<	<	<	<
C			-	>	>	=
D				-	<	<
E					-	<
F						-

Wygrane	
A	0
B	1
C	4½
D	2
E	3
F	4½

Proportional Approval Voting (PAV)

- Każdy użytkownik decyduje, które filmy może zaakceptować, a które nie
- Zakładamy, że każdy kolejny element akceptowany przez użytkownika daje mu mniejszą satysfakcję
 - Kolejno 1 punkt, $1/2$, $1/3$, $1/4$, ...
- Dobieramy elementy tak, by zmaksymalizować zadowolenie
- Algorytm NP-trudny, ale istnieje zachłanna aproksymacja
 - W każdej iteracji dobieramy ten element, który najbardziej zwiększa sumę zadowolenia

Proportional Approval Voting (PAV)

- ocena ≥ 7
- Rekomendacja = []

Zadowolenie	
Alice	0
Bob	0
Carol	0

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10

Proportional Approval Voting (PAV)

- ocena ≥ 7
- Rekomendacja = []

Zadowolenie	
Alice	0
Bob	0
Carol	0

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10
Score	0+1+0 = 1	0+0+0 = 0	1+1+1 = 3	1+1+0 = 2	0+1+1 = 2	1+0+1 = 2

Proportional Approval Voting (PAV)

- ocena ≥ 7
- Rekomendacja = [C]

Zadowolenie	
Alice	1
Bob	1
Carol	1

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10
Score	0+1+0 = 1	0+0+0 = 0	3	1+1+0 = 2	0+1+1 = 2	1+0+1 = 2

Proportional Approval Voting (PAV)

- ocena ≥ 7
- Rekomendacja = [C]

Zadowolenie	
Alice	1
Bob	1
Carol	1

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10
Score	$0 + \frac{1}{2} + 0 = \frac{1}{2}$	$0 + 0 + 0 = 0$	3	$\frac{1}{2} + \frac{1}{2} + 0 = 1$	$0 + \frac{1}{2} + \frac{1}{2} = 1$	$\frac{1}{2} + 0 + \frac{1}{2} = 1$

Proportional Approval Voting (PAV)

- ocena ≥ 7
- Rekomendacja = [C, D]

Zadowolenie	
Alice	2
Bob	2
Carol	1

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10
Score	$0 + \frac{1}{2} + 0 = \frac{1}{2}$	$0 + 0 + 0 = 0$	3	1	$0 + \frac{1}{2} + \frac{1}{2} = 1$	$\frac{1}{2} + 0 + \frac{1}{2} = 1$

Proportional Approval Voting (PAV)

- ocena ≥ 7
- Rekomendacja = [C, D]

Zadowolenie	
Alice	2
Bob	2
Carol	1

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10
Score	$0 + \frac{1}{3} + 0 = \frac{1}{3}$	$0 + 0 + 0 = 0$	3	1	$0 + \frac{1}{3} + \frac{1}{2} = \frac{5}{6}$	$\frac{1}{3} + 0 + \frac{1}{2} = \frac{5}{6}$

Proportional Approval Voting (PAV)

- ocena ≥ 7
- Rekomendacja = [C, D, E]

Zadowolenie	
Alice	2
Bob	3
Carol	2

	A	B	C	D	E	F
Alice	3	6	10	8	6	10
Bob	8	6	7	8	9	1
Carol	2	5	9	6	7	10
Score	$0 + \frac{1}{3} + 0 = \frac{1}{3}$	$0 + 0 + 0 = 0$	3	1	$\frac{5}{6}$	$\frac{1}{3} + 0 + \frac{1}{2} = \frac{5}{6}$

Funkcje celu

Dlaczego potrzebujemy funkcji celu?

- Optymalizowanie zadowolenia lub sumy zadowolenia jest niewystarczające
 - Nie uwzględnia tego, że część grupy może być wciąż niezadowolona
- Nie ma jednej, uniwersalnej i uznanej funkcji celu
- Najlepiej zdefiniować kilka różnych funkcji celu obrazujących różne aspekty
- Źródło:
<https://homepages.tuni.fi/konstantinos.stefanidis/docs/sac20.pdf>

Miara zadowolenia

$$GroupListSat(u_i, R) = \sum_{d \in R} p(u_i, d)$$

$$UserListSat(u_i) = \sum_{d \in TopN(u_i)} p(u_i, d)$$

$$sat(u_i, R) = \frac{GroupListSat(u_i, R)}{UserListSat(u_i)}$$

$$GroupSat(G, R) = \sum_{u_i \in G} sat(u_i, R)$$

Miara niespójności zadowolenia

$$\begin{aligned} GroupDis(G, R) = \\ \max_{u_i \in G} (sat(u_i, R)) - \min_{u_i \in G} (sat(u_i, R)) \end{aligned}$$

Algorytm

Sequential Hybrid Aggregation

- Algorytm optymalizuje równocześnie średnie oraz minimalne zadowolenie z elementów rekomendacji

$$\begin{aligned} score_j(G, d) = \\ (1 - \alpha_j) * avgScore(G, d) + \alpha_j * leastScore(G, d) \end{aligned}$$

- Parametr α zmienia się w czasie

$$\alpha_j = \max_{u_i \in G} sat(u_i, R) - \min_{u_i \in G} sat(u_i, R)$$

Podsumowanie

- Definicja problemu rekomendacji dla grup
- Algorytmy proste
- Algorytmy konsensusu
- Algorytmy wyborcze
- Metryki oceny jakości rekomendacji
- Dynamiczny algorytm optymalizujący kilka metryk jednocześnie