

Rekomendacje dla grup

Systemy Rekomendacyjne 2023/2024

Definicja problemu

- Wszystkie dotychczasowe algorytmy generowały rekomendacje dla pojedynczych użytkowników
 - Nawet jeśli uczyliśmy się na danych z populacji
- Jak tworzyć rekomendacje dla arbitralnych 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

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

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

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

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

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

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 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

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 = []

	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 = [C]

	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 = [C, E]

	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

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://dl.acm.org/doi/10.1145/3341105.3375766>

Miara zadowolenia

$$\textit{sat}(u_i, Gr_j) = \frac{\textit{GroupListSat}(u_i, Gr_j)}{\textit{UserListSat}(u_i, A_{u_i,j})}$$

$$\textit{GroupListSat}(u_i, Gr_j) = \sum_{d_z \in Gr_j} p_j(u_i, d_z)$$

$$\textit{UserListSat}(u_i, A_{u_i,j}) = \sum_{d_z \in A_{u_i,j}} p_j(u_i, d_z)$$

Miara zadowolenia

$$satO(u_i, \mathcal{GR}) = \frac{\sum_{j=1}^{\mu} sat(u_i, Gr_j)}{\mu}$$

$$groupSat(G, Gr_j) = \frac{\sum_{u_i \in G} sat(u_i, Gr_j)}{|G|}$$

$$groupSatO(G, \mathcal{GR}) = \frac{\sum_{u_i \in G} satO(u_i, \mathcal{GR})}{|G|}$$

Miara niespójności zadowolenia

$$\begin{aligned} groupDis(G, \mathcal{GR}) = \\ \max_{u_i \in G} satO(u_i, \mathcal{GR}) - \min_{u_i \in G} satO(u_i, \mathcal{GR}) \end{aligned}$$

Algorytm

Sequential Hybrid Aggregation

- Algorytm optymalizuje równocześnie średnie oraz minimalne zadowolenie z elementów rekomendacji

$$\begin{aligned} \text{score}(G, d_z, j) = \\ (1 - \alpha_j) * \text{avgScore}(G, d_z, j) + \alpha_j * \text{leastScore}(G, d_z, j) \end{aligned} \quad (8)$$

- Parametr α zmienia się w czasie

$$\alpha_j = \max_{u \in G} \text{sat}(u, Gr_{j-1}) - \min_{u \in G} \text{sat}(u, Gr_{j-1}) \quad (9)$$

Podsumowanie

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