Priporočanje z algoritmom Slope One

↓↓↓
Mladen Borovič

→ Priporočilni sistemi se v zadnjem desetletju vse bolj uporabljajo v različne namene. Najpogosteje jih najdemo v spletnih trgovinah, socialnih omrežjih, iskalnikih in tudi v storitvah za pretočnost video in avdio vsebin. V tem prispevku opisujemo pristop sodelovalnega priporočanja Slope One, ki deluje na podlagi povratnih informacij, pridobljenih s strani drugih uporabnikov.

Denimo, da se odpravljate na morje in želite na plaži prebrati dobro knjigo. Da bi izbrali knjigo, ki bi jo res z veseljem brali, omejite izbiro na žanre, avtorje ali pa tudi dolžino knjige. Velika možnost je, da bo po začetni omejitvi izbire na voljo še vedno preveč knjig in potrebujete še nekaj več informacij, da bi izbrali tisto pravo knjigo, ki vam bo ustrezala na plaži. Velikokrat se v takšnih primerih obrnemo na prijatelje oziroma pregledamo mnenja in recenzije knjig na spletu. Ker se mnenja velikokrat razlikujejo, saj smo ljudje različni, želimo ugotoviti, kateri ljudje imajo podoben okus kot mi.

V opisani situaciji hitro vidimo, da izvajamo neke vrste filtriranje med veliko množico knjig. V svetu priporočilnih sistemov (*recommender systems*) poznamo dva osnovna pristopa k pridobivanju ustreznega rezultata. Prvi pristop je vsebinsko filtriranje (*content-based filtering*). Pri tem pristopu nastopa računalniški sistem, ki mu podamo potrebne informacije (npr. tematiko knjige), vrne pa nam seznam priporočil, ki jih računalniški sistem izračuna s pomočjo metrike vsebinske podobnosti. Drugi pristop je sodelovalno filtriranje (*collaborative filtering*), kjer pa računalniškemu sistemu podamo le naše pretekle stike (npr. oceno knjige na lestvici med 1 in 5), dobimo pa seznam priporočil, ki jih računalniški sistem izračuna s pomočjo metrike podobnosti. Pri tem računalniški sistem upošteva tudi ostale bralce istih knjig. V nadaljevanju prispevka bomo predstavili algoritem sodelovalnega filtriranja Slope One, ki bo znal glede na pretekle ocene knjig priporočati najustreznejšo knjigo, ki je še nismo prebrali.

Algoritem Slope One

Področje sodelovalnega filtriranja pozna veliko algoritmov, ki se samostojno ali kot skupek večih algoritmov uporabljajo še danes v zelo razširjenih spletnih storitvah, kot sta YouTube in Amazon. Algoritem Slope One se prvič pojavi leta 2005 kot nov, preprost in hiter način za priporočanje s sodelujočim filtriranjem. Prednosti algoritma sta natančnost priporočil in možnost enostavne paralelizacije.

Algoritem Slope One deluje tako, da na podlagi kombinacije uporabnikov in njihovih že obstoječih ocen poskuša predvidevati, kako bi drugi uporabniki ocenili tiste elemente, ki si jih še niso ogledali. V našem primeru so elementi priporočanja knjige, ki so jih bralci že ocenili, z algoritmom Slope One pa želimo predvideti našo oceno za vse knjige, ki jih še nismo ocenili. Ideja algoritma Slope One je ustvarjanje linearne relacije med elementi priporočanja in uporabniki, ki je podobna linearni funkciji $f(x) = ax + b$. Gre za linearno funkcijo, kjer je smerni koeficient oz. naklon a enak 1; od tod izvira tudi ime algoritma Slope One. Prvi korak algoritma je torej ustrezna predstavitev uporabniških ocen in knjig. Le-to zelo elegantno predstavimo z matriko, kjer vrstice predstavljajo uporabnike, stolpci predstavljajo knjige, številčne vrednosti pa oceno med 1 in 5, s katero je uporabnik ocenil knjigo. Oznaka – pomeni, da oseba še ni ocenila knjige.

	K_A	K_B	K_C
Marko	5	2	3
Mateja	3	4	–
Tina	–	3	4

Iz takšne predstavitve lahko tvorimo vektorje, ki jih uporabimo v naslednjih korakih algoritma. Recimo, da želimo izračunati, kako bi Tina ocenila knjigo K_A. Najprej izračunamo povprečni razdalji med K_A in K_B ter med K_A in K_C. To storimo z operacijo odštevanja vektorjev, kjer v primeru, da za knjigo

nimamo podane ocene (oznaka –), razlike med ocenami ne bo mogoče izračunati. V tem primeru izločimo pripadajočo vrstico v vektorjih (enačba 1):

- $d_{i,j} = K_i - K_j.$ (1)

Rezultat odštevanja dveh vektorjev, kjer ni podane ocene (oznaka –), je nov vektor, kjer izločitev vrstice (in s tem tudi knjige) ponazorimo z oznako –, pri nadaljnjih izračunih pa uporabimo ustrezno preoblikovan vektor (v tem primeru je to vektor $[4 \quad 0]^T$):

- $\begin{bmatrix} 7 \\ 4 \\ - \end{bmatrix} - \begin{bmatrix} 3 \\ 4 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 0 \\ - \end{bmatrix}$

Hkrati si beležimo število ocenjenih skupnih knjig $N_{i,j}$, saj želimo izračunati povprečno razliko med ocenami za obravnavane knjige. $N_{i,j}$ ustreza številu nemanjkajočih komponent vektorja. Če delamo v prostoru z dimenzijo D, potem povprečno razliko med ocenami izračunamo po enačbi 2. Pri tem velja, da ocena obstaja ($d^k_{i,j} \neq -$):

- $\bar{d}_{i,j} = \frac{1}{N_{i,j}} \sum_{k=0}^{D} d^k_{i,j}$ (2)

V tem koraku izračunamo povprečne razlike v ocenah:

- $d_{A,B} = \begin{bmatrix} 5 \\ 3 \\ - \end{bmatrix} - \begin{bmatrix} 2 \\ 4 \\ 3 \end{bmatrix} = \begin{bmatrix} 3 \\ -1 \\ - \end{bmatrix}; N_{A,B} = 2$

  $d_{A,C} = \begin{bmatrix} 5 \\ 3 \\ - \end{bmatrix} - \begin{bmatrix} 3 \\ - \\ 4 \end{bmatrix} = \begin{bmatrix} 2 \\ - \\ - \end{bmatrix}; N_{A,C} = 1$

- $\bar{d}_{A,B} = \frac{(3 + (-1))}{2} = \frac{2}{2} = 1$

  $\bar{d}_{A,C} = \frac{2}{1} = 2$

Zadnji korak je izračun vrednosti sprememb ocen r, kjer upoštevamo povprečno razliko v ocenah $\bar{d}_{i,j}$ in oceno uporabnika y (enačba 3). Vrednosti sprememb ocen za uporabnika in knjige $r(y)_{i,j}$ uporabimo v izračunu predvidene ocene, ki pove, kako bi uporabnik y ocenil knjigo x. To izračunamo z enačbo 4, ki predpostavlja linearno relacijo med uporabnikom y in njegovo oceno knjige x na podlagi ocen preostalih uporabnikov, ki so preostale knjige ocenili podobno kot uporabnik x:

- $r(y)_{i,j} = \text{ocena}(y)_j + \bar{d}_{i,j}$ (3)

- $\text{SlopeOne}(x, y) = \frac{\sum_{i=1}^{n} N_{x,i} \cdot r(y)_{x,i}}{\sum_{i=1}^{n} N_{x,i}}$ (4)

Za Tino in knjigo A je izračun v zadnjem koraku sledeč:

- $r(\text{Tina})_{A,B} = \text{ocena}(\text{Tina})_B + \bar{d}_{A,B} = 3 + 1 = 4$

  $r(\text{Tina})_{A,C} = \text{ocena}(\text{Tina})_C + \bar{d}_{A,C} = 4 + 2 = 6$

- $\text{SlopeOne}(A, \text{Tina}) =$

  $= \frac{N_{A,B} \cdot r(\text{Tina})_{A,B} + N_{A,C} \cdot r(\text{Tina})_{A,C}}{N_{A,B} + N_{A,C}}$

  $= \frac{2 \cdot 4 + 1 \cdot 6}{2 + 1} = \frac{14}{3} = 4{,}667$

Z algoritmom Slope One predvidevamo, da bi Tina ocenila knjigo A z oceno 4,667, kar je zelo blizu Markovi oceni za knjigo A. Če primerjamo njune ocene ostalih knjig, opazimo, da so tudi te zelo podobne. Iz tega lahko sklepamo, da imata Tina in Marko zelo podoben okus. Vidimo tudi, kako je na Tinino predvideno oceno knjige A vplivala nižja ocena Mateje.

Prikazan postopek v praksi izvedemo za vsako neocenjeno knjigo. Tako dobimo predvidene ocene neocenjenih knjig, ki jih lahko uredimo po velikosti padajoče. S tem tvorimo seznam priporočil, ki ga lahko omejimo s številom zadetkov na izhodu θ. Seznam priporočil ponavadi omejimo na 3 do 5 zadetkov.

Zgled

Priporočanje z algoritmom Slope One prikažimo še na praktičnem zgledu, kjer imamo pet oseb (Luka, Jakob, Nika, Sara in Anja) in osem knjig (označimo jih s $K_1 - K_8$). Tvorimo tabelo ocen, kjer oznaka – pomeni, da oseba še ni ocenila knjige, številčna vrednost pa predstavlja oceno med 1 (zelo slabo) in 5 (odlično).

Ustvarili bomo priporočila za Jakoba, ki je relativno kritičen in nov bralec, kot je razvidno iz tabele ocen. Izračunali bomo predvidene ocene za K_1, K_3, K_5, K_6, K_7 in K_8. Oglejmo si postopek izračuna predvidenih ocen.

	K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8
Luka	3	–	–	2	–	4	5	5
Jakob	–	1	–	3	–	–	–	–
Nika	5	4	5	–	2	–	4	–
Sara	2	2	–	–	5	–	–	–
Anja	–	–	4	4	–	–	–	4

Najprej z enačbo 1 izračunajmo razliko med ocenami knjig, ki jih je Jakob že ocenil ($K_2 = 1$, $K_4 = 3$), in ocenami knjig, ki jih želimo priporočati. V primeru, da za knjigo nimamo podane ocene (oznaka –), razlike med ocenami ne bo možno izračunati:

- $$d_{1,2} = K_1 - K_2 = \begin{bmatrix} 3 \\ - \\ 5 \\ 2 \\ - \end{bmatrix} - \begin{bmatrix} - \\ 1 \\ 4 \\ 2 \\ - \end{bmatrix} = \begin{bmatrix} - \\ - \\ 1 \\ 0 \\ - \end{bmatrix}; N_{1,2} = 2$$

$$d_{1,4} = K_1 - K_4 = \begin{bmatrix} 3 \\ - \\ 5 \\ 2 \\ - \end{bmatrix} - \begin{bmatrix} 2 \\ 3 \\ - \\ - \\ 4 \end{bmatrix} = \begin{bmatrix} 1 \\ - \\ - \\ - \\ - \end{bmatrix}; N_{1,4} = 1$$

Hkrati si beležimo tudi število ocenjenih skupnih knjig $N_{i,j}$, ki ga bomo potrebovali za izračun povprečne razlike med ocenami za obravnavane knjige (enačba 2):

- $$\bar{d}_{1,2} = \frac{1}{N_{1,2}} \cdot \sum_{d_k \in d_{1,2}} d_k = 0.5$$

$$\bar{d}_{1,4} = \frac{1}{N_{1,4}} \cdot \sum_{d_k \in d_{1,4}} d_k = 1$$

Sledi izračun predvidene ocene za knjigo K_1 z enačbama 3 in 4:

- $$r(\text{Jakob})_{1,2} = \text{ocena(Jakob)}_2 + \bar{d}_{1,2} = 1{,}5$$
$$r(\text{Jakob})_{1,4} = \text{ocena(Jakob)}_4 + \bar{d}_{1,4} = 4$$

- $$\text{SlopeOne}(1, \text{Jakob}) = \frac{\sum_{i=1}^{n} N_{1,i} \cdot r(\text{Jakob})_{1,i}}{\sum_{i=1}^{n} N_{1,i}}$$
$$= \frac{N_{1,2} \cdot r(\text{Jakob})_{1,2} + N_{1,4} \cdot r(\text{Jakob})_{1,4}}{N_{1,2} + N_{1,4}}$$
$$= 2{,}333$$

Z izračunom vrednosti Slope One predvidevamo, da bi Jakob knjigo K_1 ocenil z oceno 2,333. Na enak način pridobimo še vrednosti za preostale knjige:

- $$R_{\text{Jakob}} = \left\langle \begin{matrix} \mathbf{K_1} & K_2 & \mathbf{K_3} & K_4 & \mathbf{K_5} & \mathbf{K_6} & \mathbf{K_7} & \mathbf{K_8} \\ \mathbf{2{,}333} & 1 & \mathbf{2} & 3 & \mathbf{1{,}5} & \mathbf{5} & \mathbf{4} & \mathbf{4{,}5} \end{matrix} \right\rangle.$$

Knjige s pripadajočimi predvidenimi ocenami zapišemo v seznam, ki ga uredimo po velikosti ocene padajoče. Seznam lahko omejimo na pet zadetkov (npr. $\theta = 5$), ki ga nato posredujemo Jakobu.

- $$R^{\theta}_{\text{Jakob}} = \left\langle \begin{matrix} K_6 & K_8 & K_7 & K_1 & K_3 \\ 5 & 4{,}5 & 4 & 2{,}333 & 2 \end{matrix} \right\rangle.$$

V tem prispevku opisan algoritem sodelovalnega priporočanja lahko uporabimo tudi za priporočanje filmov, pesmi, ljudi, izdelkov v trgovinah in drugih izdelkov, ki jih uporabniki lahko ocenijo. Iz tega sledi tudi glavna pomanjkljivost vsakega algoritma sodelovalnega priporočanja. To je t. i. problem hladnega začetka (*cold-start problem*), saj bomo na začetku vedno potrebovali množico ocen, da bomo sploh lahko izvedli priporočanje. Vsebinsko filtriranje rešuje ta problem, vendar gre za kompleksnejše metode računanja podobnosti, hkrati pa se lahko po določenem času priporočila začnejo ponavljati. Iz tega razloga se sodobni priporočilni sistemi poslužujejo tako vsebinskega kot sodelovalnega filtriranja. To so hibridni priporočilni sistemi (*hybrid recommender systems*), ki vedno bolj uporabljajo pristope globokega učenja in se tudi že uporabljajo v praksi.

Literatura

[1] D. Lemire in A. Maclachlan, *Slope One Predictors for Online Rating-Based Collaborative Filtering*, Proceedings of the 2005 SIAM International Conference on Data Mining, 471–475, 2005.

[2] P. Melville in V. Sindhwani, *Recommender Systems*, Encyclopedia of Machine Learning, Springer, 829–838, 2010.

[3] F. Ricci, L. Rokach in B. Shapira, *Introduction to Recommender Systems Handbook*, Recommender Systems Handbook, Springer, 1–35, 2011.

[4] R. Burke, *Hybrid Recommender Systems: Survey and Experiments*, User Modeling and User-Adapted Interaction, 12(4), 331–370, 2002.

×××