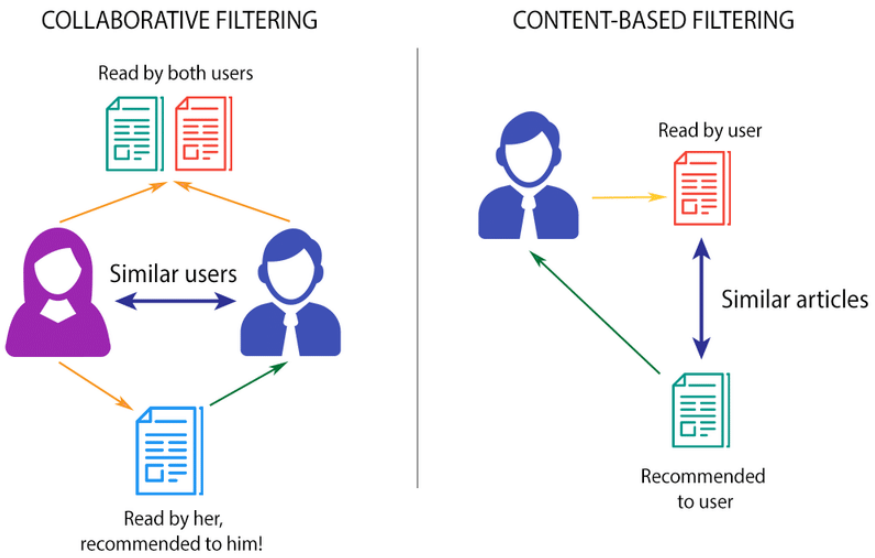
Системи засновани на сарадњи

Систем заснован на сарадњи кориснику препоручује оне производе које су њему слични корисници оценили као веома добре. Заправо, овај приступ претпоставља да ако корисник *А* има слично мишљење као корисник *Б* о неком скупу производа, већа је вероватноћа да ће *А* имати исто мишљење као *Б* о одређеном производу са којим *А* није упознат.



Дакле, за разлику од система заснованим на садржају, овде је фокус на сличностима оцена два корисника о истим производима. Вектор корисника се представља одговарајућим редом из матрице корисности. Са друге стране, вектор производа се конструише помоћу одговарајуће колоне из поменуте матрице. Празна поља у матрици корисности замењујемо нулама. Степен сличности два корисника или два производа се и овде такође мери помоћу *Jaccard similarity, Cosine similarity* или *Пирсоновог коефицијента корелације.* Суштина овог приступа јесте у претрази сличних корисника кориснику *А* и препоруци оних производа који су се допали овим корисницима. Управо тај процес идентификовања сличних корисника и препоручивања оних производа који су се њима највише допали се назива **сарадничко филтрирање** (*collaborative filtering*).

У зависности од ентитета између којих се испитује сличност разликују се две варијанте овог приступа:

- *User Based Collaborative Filtering* (Сарадничко филтрирање засновано на кориснику) и
- *Item Based Collaborative Filtering* (Сарадничко филтрирање засновано на производу).

Недостаци *User Based Collaborative Filtering* су:

- Рачунање сличности између свих корисника са одређеним корисником је веома рачунски захтевно и изискује доста времена.
- Навике и укуси људи се мењају,

Код *Item Based Collaborative Filtering* израчунава се сличност између производа који су се кориснику највише допали и оних производа које још увек није искусио. Након тога се кориснику, у виду *Top-N* листе, представи одређени број најбоље рангираних производа.

Због ових недостатака, сложени системи (нпр. Амазон) се окрећу *Item Based Collaborative Filtering* приступу. Наиме, фокусирање на сличности између непроменљивих објеката може довести до бољих резултата, него ослањање на сличности укуса различитих људи, што је променљива категорија. Може се десити да се једној особи свидео производ са одређеним својствима, а након кратког периода нешто сасвим супротно. Будући да је код сложенијих система укупан број производа доста мањи од броја корисника, рачунање матрице сличности између производа захтева значајно мање рачунарских ресурса. Такође, ефикасније генерисање матрице сличности значи и брже прилагођавање система када се у каталог додају нови производи.

Ако је матрица корисности испуњена имплицитним подацима, онда је корисно користити *Jaccard similarity.* Ова мера сличности се представља односом броја производа који су на оба корисника оставила исти утисак (рецимо, 1 ако је купљен) према укупном броју производа са којима су оба корисника интераговала[1]. Са друге стране, ако је реч о експлицитним подацима, најбоље је користити *Cosine Similarity*. Као у случају система базираним на садржају (*Content-Based)*, потребно је прво нормализовати матрицу корисности, третирати празна поља као да је уписана нула, па тек тада рачунати *Cosine Similarity.*

Основни недостатак кориснички орјентисаних система за препоруку се односи на тзв. **Cold Start** проблем, када се нови корисник прикључи систему, а још увек није дао оцене садржајима на сајту. Тада се не може одредити сличност са другим корисницима.

Највећи проблем сарадничког филтрирања јесте празна матрица (*sparse utility matrix*), тј. када нема довољно података за утврђивање тачног степена сличности између корисника или производа. Због математичке природе формуле, највећи проблем приликом рачунања *Cosine Similarity* се јавља ако постоји само један производ са којим су оба корисника интераговала. У том случају, сличност је једнака јединици без обзира на исказани степен допадљивости једног и другог корисника. Како би се избегао овај сценарио, рачуна се сличност само са корисницима са којима постоје два или више заједничких артикала. Један од начина за превазилажење *sparsity* проблема јесте кластеровање корисника или производа.

Изазови (challanges): https://en.wikipedia.org/wiki/Collaborative_filtering

[1] Под интеракцијом подразумевамо експлицитно остављање рејтинга од стране корисника или неког имплицитног податка, који указује на заинтересованост корисника за тај објекат.

Одређивање сличности између корисника

Jaccard vs Cosinus similarity

Ако матрица корисности осликава куповину производа, Jaccard сличност је добар избор. Ипак, ако матрица садржи детаљне оцене, Jaccard сличност губи одређене информације.

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

A и B имају пресек величине 1, а унија је 5. Тада, Jaccard сличност је 1/5, а њихова Jaccard удаљеност је 4/5; тј. веома су удаљени. У супротном, A и C имају Jaccard сличност од 2/4, тако да је њихова Jaccard удаљеност иста, 1/2. То значи да је A ближе C него B. Шта каже интуиција? A и C имају различито мишљење о два филма која су гледали, док се A и B слажу у вези једног филма који су обоје гледали.

$$cosSim(A, B) = \frac{4 \times 5}{\sqrt{4^2 + 5^2 + 1^2}\sqrt{5^2 + 5^2 + 4^2}} = 0.380$$

$$cosSim(A, C) = 0.322$$

Заокруживање података

Доста се често у пракси дешава да корисници дају или врло високе или врло ниске оцене производима (items). Како би се у потпуности елеминисале овакве појаве, на пример, можемо оцене 3, 4, 5 да сматрамо да имају вредност „1“, а оне са оценама 1, 2 да сматрамо неоцењеним производима. Тада, *JaccDist(A, B) = ¾*, а *JaccDist(A, C) = 1*. Применом косинусне сличности над оваквим подацима долази се до истог закључка.

Нормализација оцена

Нормализација оцена корисника се спроводи одизимањем просечног рејтинга тог корисника од свих датих оцена. Ниски рејтинг постаје негативан, а високи рејтинг остаје позитиван број.

cosSim(A, B) = 0.092
cosSim(A, C) = -0.559

Дуалност сличности (The Duality of Similarity)

Матрица сличности даје информације о корисницима, производима, или обе ствари у исто време. Свака од техника за тражење сличности између корисника, може да се примени и за тражење сличности између производа. Постоје два разлога због којих је ова симетрија у пракси нарушена.

1. Сличност између корисника се може користити за препоруку производа. За датог корисника може се наћи одређени број сличних корисника. Кориснику се могу понудити производи које је већина сличних корисника високо оценила. Ипак, ту не постоји симетрија. Ако пронађемо парове сличних производа, потребан је додатни корак да би се производ препоручио корисницима.
2. Постоји разлика између сличности између корисника и између производа. Производи се могу класификовати веома једноставно на различите начине. Једна музичка нумера не може да припада року 60-их и 1700 бароку. На другој страни постоје корисници који воле обе поменуте категорије. Последица је да је лакше одредити сличност између производа, него између корисника јер они могу волети један исти жанр, док у исто време могу волети тотално различите жанрове.

(*) Као што је раније речено, један од начина да се предвиди вредност којој би корисник U оценио производ I је да се пронађе n корисника (за неко унапред задато n) који су најсличнији кориснику U и одреди њихов просек оцена за производ I. Свакако на почетку матрицу треба нормализовати. То заправо значи да се најпре одређује просек оцена (тј. просек добијене разлике за сваку оцену) за оне кориснике који су оценили производ I, а затим се овај просек додаје просеку свих оцена које је корисник U дао свим његовим производима.

На овај начин се пролагођава процена у случају да корисник U има тенденцију давања веома високих или веома ниских оцена или за случај да велики део сличних корисника има ту исту тененцију.

Кластеризација корисника и производа

Некада је тешко одредити сличности било између корисника, било између производа, јер имам недовољно имформација у ретко попуњеној матрици корисности. Ако имамо два слична производа коа припадају истој категорији, мали су изгледи да ће постојати пуно корисника који су купили или оценили те производе.

Један од приступа да се реши овај проблем је кластеризација производа и/или корисника. На пример, може се урадити кластеризација производа из палете доступних производа. Филмови се могу кластеризовати у три категорије HP, TW, SW.

Када су производи подељени по кластерима, нова матрица корисности сада уместо колона за производе има колоне кластера, при чему је оцена за корисника U и кластер C просечна оцена коју је корисник U дао само оцењеним производима из C. У случају да корисник није оценио ниједан производ из C, тада оцена остаје празна.

Сада се ова нова матрица може даље мењати, при чему се сада могу кластеризовати корисници. Као за случај производа, вредност за U-кластер i I-кластер је просек оцена корисника у посматраном кластеру производа. Процес се може понављати више пута ако има потребе, и за кориснике и за производе.

Уколико желимо да предвидимо оцену за корисника U и производ I:

a) Пронаћи кластере којима корисник U и производ I припадају, нпр, C и D.

b) Ако вредност за C и D није празно поље, тада узети ту вредност као процењену оцену за U-I поље.

c) Ако је вредност за C и D празно поље, тада да би се проценила оцена за U-I поље треба искористити приступ (*) где се ова оцена одређује на основу сличних кластера за C или D. За више информација на који начин се све може дефинисати сличност између скупова (кластера), погледати поглавље 3 књиге Mining of Massive Datasets, Jure Leskovec.

Feature weighting in content-based recommendation systems

Шта ово значи? Да ли смо већ ово видели негде раније?

Одговор: TFIDF

Сада се поставља питање да ли је доменско знање довољно да ми сами можемо да дефинишемо тежине и тако осликамо значај одређених особина (нпр. очекивано је да главни глумац има већу тежину од глумца са споредном улогом или камермана).

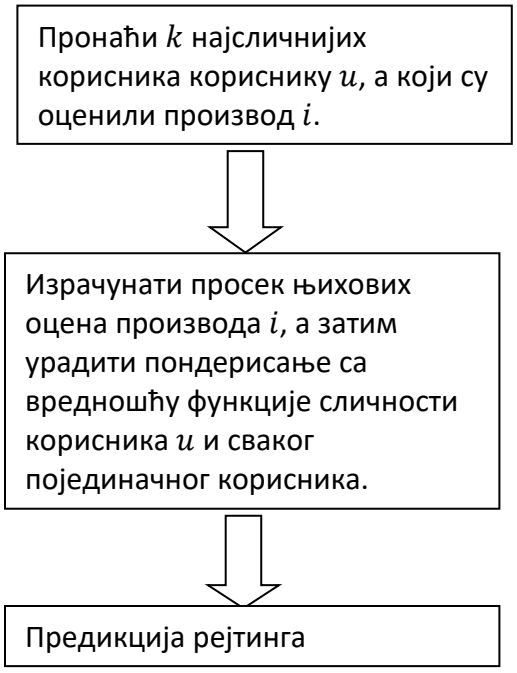
1.1.1. KNN системи за препоруку

Системи засновани на сарадњи генеришу листу препоручених производа на основу сличности између објеката (корисника или производа). Међутим, овакви системи немају могућност да предвиде рејтинг неког производа, који може послужити као критеријум на основу кога се формира *Top-N* листа.

KNN (***K** **N**earest **N**eighbours* – K најближих суседа) системи управо отклањају наведени недостатак. Као и у случају сарадничког филтрирања, постоје две врсте KNN система:

- KNN системи засновани на кориснику (*user-based KNN*)
- KNN системи засновани на производу (*item-based KNN*)

У наставку је дата шема KNN система заснованих на кориснику, која приказује предвиђање оцене којом би корисник u оценио производ i.



$$\hat{r}_{ui} = \frac{\sum_{v \in N_i^k(u)} (sim(u,v) \cdot r_{vi})}{\sum_{v \in N_i^k(u)} sim(u,v)}$$

$\hat{r}_{ui}$ - предикција рејтинга корисника u за производ i

$N_i^k(u)$ - скуп k најближих корисника кориснику u који су оценили производ i

$sim(u,v)$ - функција сличности која рачуна степен сличности између корисника u и v (нпр. *Cosine Similarity*).

r_{vi}- рејтинг којим је корисник v оценио производ i

Идентичан поступак се примењује и код *item-based KNN* система.

Литература: Jure Leskovec, Anand Rajaraman, Jeff Ullman, *Mining of Massive Datasets*, Second edition. Cambridge Universiy Press, 2014.