SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1136

Preporučiteljski sustavi u sveprisutnom računarstvu

Branimir Pervan

Zagreb, svibanj 2015.

Umjesto ove stranice umetnite izvornik Vašeg rada.
Da bi ste uklonili ovu stranicu obrišite naredbu `\izvornik`.

Ovdje dolazi zahvala

SADRŽAJ

1. Uvod

U posljednjih dvadesetak godina razvoj Interneta stvari (*eng. Internet of Things*) uhvatio je gotovo eksponencijalni zamah, a pojedini izvori navode da će broj uređaja priključenih na ovu sveprisutnu mrežu do 2020. g. doseći 26 milijardi [1] odnosno 30 milijardi [2]. Tomu značajno doprinosi i konstantno opadanje cijene proizvodnog procesa tehnologije koja naizgled obične stvari na neki način čini inteligentnima i sposobnima za komunikaciju. Sveprisutno računarstvo, kao koncept u računarskoj znanosti gdje je računarstvo prisutno svugdje [3], opisuje upravo takve vrste stvari i uređaja, ali i takve principe gdje računalo može biti ugrađeno u bilo kojem uređaju, na bilo kojoj lokaciji i u bilo kojem obliku.

Tu još negjde mora doći dosta toga o preporučiteljima i njihovoj mega korisnosti

Potreba i smisao izučavanja ovog područja dolazi iz očitog primjera za ulaganjem u bolje i efikasnije algoritme jer je u nepreglednoj masi informacija, kakav je Internet stvari idealan izvor, procesna moć današnjih računala davno izgubila bitku.

Motiv ovog diplomskog rada jest manjak dostupnih algoritama za ovakvu specifičnu vrstu preporučitelja. U ovom radu dat će se teorijska podlogua bazičnih algoritama za filtriranje sadržaja te analizirati prednosti i nedostatke takvih pristupa. Prikazat će se principi primjene preporučiteljskih sustava u sveprisutnim aplikacijama te će se prikazati i analizirati posebni zahtjevi na preporučiteljske sustave od strane takvih aplikacija na konkretnom scenariju. Na kraju će biti dan prikaz razvijenog preporučiteljskog sustava za takvu primjenu.

2. Preporučiteljski sustavi

2.1. Uvod u preporučiteljske sustave

Preporučiteljski sustavi su, ukratko rečeno, skup programskih alata i tehnika koji iz relativno velikog i nepreglednog skupa karakteriziranih informacija krajnjem korisniku, koji je opet na neki način karakteriziran, filtriraju informacije o mogućim preferencijama tog korisnika. Općenito, možemo reći da je jednostavan model preporučiteljskog sustava dan formulom:

$$R \leftarrow U \times I \tag{2.1}$$

gdje je R rezultat, tj. predikcija (*eng. prediction, recommendation*), preferencije korisnika U (*eng. user*) koji je zatražio preporuku, tj. filtriranje sadržaja , a I predmet nad kojim se vrši predikcija preferencije (*eng. item*). Traženje potencijalnih preporuka za korisnika U tada se u najjednostavnijem slučaju svodi na kombiniranje profila njegovih preferencija s profilima predmeta u skupu svih predmeta dostupnih algoritmu za filtriranje. Krajnji rezultat na kraju jest najčešće lista od n najboljih preporuka, tj. pretpostavki da bi korisnik te predmete ocjenio najbolje (*eng. top – N list*) Gornji model ima dva osnovna i lako uočljiva ograničenja:

1. Traženje preporuka za korisnika svodi se na iscrpno pretraživanje prostora predmeta dostupnih algoritmu za filtriranje

2. Rezultat je preporuka kojoj fali bilo kakav kontekst.

Da bismo doskočili ovim problemima, razmotrit ćemo razne modele preporuke od kojih su neke već dobro poznati i korišteni algoritmi. Za početak, uvedimo u formulu 2.1 općeniti kontekst preporuke:

$$R \leftarrow U \times I \times C \tag{2.2}$$

gjde je C kontekst u kojem korisnik U traži preporuku.

Predmet se shvaća generički i on može varirati ovisno o kontekstu primjene, primjerice, artikli u internet trgovini, knjige u digitalnim knjižnicama, pjesme i filmovi namultimedijalnim servisima, rezultati pretraživanja na tražilicama, osobe na društvenim mrežama, smjerovi kretanja u prostoru i u ovisnosti s vremenom itd. Svaki predmet u korišten od strane algoritma za filtriranje obično je opisan nekim karakteristikama koji variraju u ovisnosti o kontekstu predmeta. Tako primjerice neka pjesma može biti opisana žanrom, trajanjem i izvođačem, a knjiga isto tako žanrom, autorom i brojem stranica. Unositi težine za pojedine ocjene karakteristika predmeta nije uobičajeno jer na taj način dolazi do subjektiviziranja rezultata filtriranja na manji skup osoba, ali s druge strane gledano, nije ni nemoguće.

S druge strane, korisnici sustava imaju različite scenarije korištenja preporučitelja od kojih su osnovni filtriranje neželjenog sadržaja iz velikih baza podatak i savjetovanje pri nedostatku vlastite kompetencije za izbor sadržaja (referenca: ono čudo koje te pita kaj te interesira). Korisnici imaju svoje preferencije koje su u ovom slučaju uglavnom opisane težinama jer prema različitim potrebama određene karakteristike predmeta nad kojima se vrši filtriranje mogu biti zanimljivije, odnosno manje zanimljive.

Interakcijom korisnika sa sustavom omogućuje se praćenje njegovih odabira, treniranje preporučitelja te kroz analizu profila korisnika i njegovih osobnih preferencija stvaranje modela za preporuku predmeta na nekoliko načina. Podaci koje korisnik ostavlja u sustavu u osnovi se mogu podijeliti u dva skupa:

1. Implicitni
2. Eksplicitni

Implicitni podaci su oni podaci koje je sustav prikupio od korisnika bez da ga je to eksplicitno zatražio. Takvi podaci mogu biti primjerice, demogravski podaci, točnije, šire područje iz kojeg korisnik koristi sustav a jednostavno se doznaje iz baze podataka dodijeljenih područja (*eng. scope*) IP adresa. Također, pod implicitne podatke spadaju i akcije korisnika u sustavu koje se mogu doznati iz sjedničkih zapisa, kao i tzv. klikovi na određene poveznice unutar sustava.

S druge strane, eksplicitni podaci su oni koje korisnik ostavlja s namjerom, primjerice koristeći ankete o svojim preferencijama, ostavljajući povratnu informaciju na ponuđene predmete (*eng. feedback*) ili odgovarajući na bilo koji način na upite o pojedinim predmetima.

U općem slučaju, preporučiteljske sustave razlikujemo prema načinu filtriranja i analiziranja informacija, a razlikujemo četiri osnovna načina (tj. jedan način pseudo-

filtriranja i tri načina filtriranja):

1. Filtriranje neovisno o korisniku (*eng. Non – personalized filtering*)
2. Filtriranje zasnovano na sadržaju (*eng. Content – based filtering*)
3. Filtriranje zasnovano na suradnji (*eng. Collaborative filtering*)
4. Hibridne tehnike filtriranja odnosno preporučivanja

Povijesno gledano, razvoj preporučiteljskih sustava započeo je devedesetih godina prošlog stoljeća, a nemalo je popularizirana $2006.$ g. svojevrsnim natjecanjem „The Netflix Prize "kada je poznati pružatelj multimedije na zahtjev ponudio nagradu od $\$1,000,000$ američkih dolara za tim koji razvije preporučitelj bolji od taga postojećeg sustava „Cinematch "za određeni postotak. Ovo je ostavilo velik utjecaj na razvoj preporučitelja prvenstveno zbog činjenice da je u uvjetima natjecanja navedeno da rezultati i principi rada razvijenih preporučitelja moraju biti javno objavljeni i dostupni

2.2. Filtriranje neovisno o korisniku

Osnovni model filtriranja jest filtriranje neovisno o korisniku. Model preporuke koji proizlazi iz ovakvog načina filtriranja, strogo gledano, ne može biti preporučitelj jer preporuka ne ovisi strogo o korisniku. Drugim riječima, svaki korisnik koji zatraži preporuku od ove vrste filtriranja dobit će istu preporuku. Ovu tvrdnju mogli bi jednostavnije prikazati relacijom:

$$R \leftarrow I \tag{2.3}$$

gdje je R predikcija, tj. preporuka, a I predmet. Iz relacije 2.3 očigledno je da je predikcija funkcija isključivo predmeta, pa kao takva ne može biti smatrana punokrvnim preporučiteljem.

Razmjerno jednostavna predožba ovog modela jest rejting (*eng. Rating*). Uzmimo za primjer neki servis za ocjenjivanje i korisničke recenzije ugostiteljskih objekata. Neka svaki korisnik koji je koristnio uslugu nekog od objekata ima pristup sustavu u kojem može u više kategorija ostaviti ocjenu iz nekog intervala s prisanom recenzijom. Također, neka svaki korisnik ima mogućnost ocijeniti uslugu brojčanom ocjenom iz intervala od jedan do pet. Model preporučitelja u tom slučaju bio bi opisan s:

$$S = \{1, 2, 3, 4, 5\} \tag{2.4}$$

$$R = \lfloor \frac{\sum_{i=1}^{N} s_i}{N} * 10 \rfloor \tag{2.5}$$

gdje je S skup mogućih ocjena, R konačan rejting predmeta, N ukupan broj korisnika koji su ocjenili taj predmet, a s_i ocjena i-tog korisnika. Iako, izgrađeni model preporuke strogo gledano nije preporučitelj, on to ipak čini posredno nudeći korisniku ono što su drugi korisnici obilježili kao poželjnije. Ovakav model obično koriste usluge s povratnom informacijom korisnika (*eng. Feedback*), npr. *eBay*, *Tripadvisor* i *Zagat*. Elemente ovog preporučitelja prikazane relacijama 2.4 i 2.5 možemo varirati kako bi prilagodili izgrađeni model drugim sustavima, primjerice:

- Skup ocjena S. Ovisno o potrebi, moguće je skup proširiti do potrebnog broja ocjena, imajući na umu da veća granulacija nije nužno bolja, kao i da može biti beskorisna u vidu onemogućenja korisnika da predmet ocjeni spontano, a da neće biti vidljiva u krajnjem rezultatu. Također, granulaciju je moguće povećati dozvoljavanjem ocjena van skupa cijelih brojeva.
- Prikaz rezultata R. U formuli 2.5 prije zaokruživanja prosjek je pomnožen faktorom 10 radi eliminacije decimala. Moguće je odabrati neki drugi prikaz rezultata, primjerice u postotcima.

Sam način ocjenjivanja ne mora nužno biti eksplicitna dodjela ocjene. Moguće je primjerice koristiti sustav glasovanja (*eng. Vote up/down*). Najpoznatiji primjeri koji koriste takve ocjene su *Reddit* i *StackOverflow*.

U općem slučaju modeli preporučitelja zasnovani na ovakvoj vrsti filtriranja imaju dvije mane:

- Zavaravanje korisnika od strane rejtinga koji je, neovisno o načinu prikaza, i dalje samo prosjek pojedinačnih ocjena.
- Nedostatak konteksta za preporuke.

Potonji problem posebno se manifestira prilikom asocijativnog preporučivanja. Uzmimo za primjer da tražimo preporuku za neki predmet koji je dodatak na neki već postojeći predmet, primjerice, tražimo preporuku za preljev za sladoled. Izgradimo sada jednostavan model prema kojem bi mogli dobiti nepersonaliziranu preporuku za predmet koji ovisi o drugom predmetu. Neka su X i Y skupovi svih korisnika koji su kupili proizvode X i Y respektivno.

Intuitivno se nameće da ukoliko je više korisnika kupilo proizvod X i uz njega proizvod Y da će i ostalim korisnicima (koji traže preporuku) proizvod Y odgovarati uz proizvod X. Elementarnom algebrom skupova možemo dakle zaključiti da se predikcija za nekog korisnika može izraziti relacijom:

$$R = \frac{\mid X \cap Y \mid}{\mid X \mid} \tag{2.6}$$

Osnovni problem proizlazi iz činjenice da su određeni predmeti neovisno popularni. Primjerice, u nekom dućanu, moguće je da većina korisnika kupuje neki predmet pa može doći do lažne korelacije popularnog predmeta s nekim drugim predmetom. Drugi problem je nepostojanost veze između predmeta iz skupa X i Y u smislu asocijativnosti. Ukoliko je X skup sladoleda, a Y skup preljeva, korisniku bi kao preporuku trebalo izdvojiti preljeve za sladoled iz skupa Y.

$$R = \frac{\frac{|X \cap Y|}{|X|}}{\frac{|\overline{X} \cap Y|}{|\overline{X}|}} \tag{2.7}$$

Preporuka korisniku na kraju se jednostavno svodi na prikaz prvih N najboljih prosječnih ocjena.

Naivni preporučitelj preporuku može dati korištenjem intuitivno izvedive formule:

$$R = \frac{X \cap Y}{X} \tag{2.8}$$

gdje su X i Y skupovi svih korisnika koji su kupili proizvode

2.3. Filtriranje ovisno o korisniku

Za razliku od filtriranja neovisnog o korisniku, filtriranje ovisno o korisniku uzima u obzir korisnika i njegove preferencije.

2.3.1. Filtriranje zasnovano na sadržaju

Razmotrimo situaciju kada ne tretiramo korisnike ili predmete kao osnovne (atomarne) jedinice, nego ih možemo opisati nekim kategorijama, primjerice, demografskim podacima za korisnika, autorom i izdavačem ako je predmet neka knjiga i sl. Preporučivanje zasnovano na sadržaju u osnovi dovoi u vezu prikupljene preferencije korisnika, bilo eksplicitno, bilo implicitno. Neka je u sustavu koji koristi preporučitelj zasnovan na sadržaju svaki predmet opisan tekstualnim medapodacima i vektorom:

$$\boldsymbol{X_i} = \left[\boldsymbol{w_{1,i}}, \boldsymbol{w_{2,i}}, \boldsymbol{w_{3,i}}, \ldots, \boldsymbol{w_{N,i}}\right]^T \tag{2.9}$$

gdje je $\boldsymbol{w_{j,i}}$ kvantitativni, tj. brojčani opis neke j-te karakteristike za i-ti predmet. Težina je neka proizvoljno odabrana metrika koja može varirati od jednostavnog broja pojavljivanja, uključujući $0/1$ pristup (karakteristika je primjenjiva, odnosno karakteristika nije primjenjiva) do precijznijih metrika. Sličnost između dva predmeta moguće

je tada izraziti kosinusom kuta između vektora njihovih karakteristika.

$$\boldsymbol{cos}(\boldsymbol{V}_1, \boldsymbol{V}_2) = \frac{\boldsymbol{V}_1 * \boldsymbol{V}_2}{\|\boldsymbol{V}_1\| \times \|\boldsymbol{V}_2\|} \tag{2.10}$$

Tu još negdje ubaci da svaka karakteristika može imati svoju težinu, u smislu da neka karakteristika može biti važnija. Također, neka za svakog korisnika postoji korisnički profil sa dostupnim preferencijama korisnika dostupnim u vektorskom zapisu gdje i-ta komponenta vektora predstavlja težinu te karakteristike za korisnika. Tada je na sličan način moguće izraziti kompatibilnost promatranog korisnika i predmeta. U ovisnosti o kontekstu primjene, vektore je poželjno i normalizirati. Jedna od većih prednosti ovog načina filtriranja je što može stvarati preporuke neovisno o tome je li za predmet davana povratna informacija ili ne. Drugim riječima, ovaj način filtriranja iznimno je prikladan na početku rada sustava jer nema problema s takozvanim hladnim početkom (*eng. Cold start*). Isto tako, prikladan je za primjene gdje je moguće relativno dobro strukturiranim karakteristikama opisati predmete. S druge strane, nepogodan je ukoliko ga se implementira u sustave gdje korisnici dolaze rijetko ili relativno često mijenjaju preferencije. Zbog svega navedenog, ova vrsta filtriranja uglavnom se upotrebljava u sustavima za pregledavanje vijesti, personaliziranim servisima za multimediju, video na zahtjev i sl.

Nešto tu ne štima vezano uz težine. Komponente vektora zapravo su kvantitativni opis, brojka, to nisu težine. Težine, koje predstavljaju koja je karakteristika korisniku koliko važna su u drugom vektoru?

2.3.2. Filtriranje zasnovano na suradnji

Suradnički pristup dijametralno je suprotan sadržajnom pristupu. Princip rada ove vrste filtriranja suradnja je između pojedinih korisnika odnosno predmeta. Definicija ove vrste suradnje zapravo leži u određivanju sličnosti između dvaju korisnika ili predmeta, a glavna premisa jest da preferencije predmeta uglavnom važe za sve korisnike koji imaju iste interese ili su slično ocijenili slične predmete. Primjerice, razmatranjem slučaja gdje dva različita korisnika dodijele dvije relativno slične ocjene nekom predmetu, zaključak jest da je vjerojatnost da su ta dva korisnika slično ocjenila i neke druge predmete razmjerno velika. S druge strane, veća je vjerojatnost da će neki korisnik ocjeniti slično neka dva predmeta ako su ih i ostali korisnici slično ocjenili.

Zbog usporedbi i rada na dvije različite razine, korisničkoj i predmetnoj, ovaj preporučitelj se u osnovi dijeli na dvije moguće tehnike:

– Korisnik-korisnik (*eng. User-user, Neighbourhood-based, Memory-based*)

– Predmet-predmet (*eng. Item-item, Item-based, Model-based*)

Korisnik – Korisnik

Suradničko filtriranje na relaciji korisnik - korisnik

$$R = \frac{\sum_{i \epsilon I} \left[(r_{k,i} - \overline{r_k}) * (r_{u,i} - \overline{r_u}) \right]}{\sqrt{\sum_{i \epsilon I} (r_{k,i} - \overline{r_k})^2} * \sqrt{\sum_{i \epsilon I} (r_{u,i} - \overline{r_u})^2}} \tag{2.11}$$

Neka je K skup svih korisnika nekog sustava. Također, neka je k korisnik iz skupa K koji traži preporuku za neki predmet. Tada je susjedstvo N korisnika k definirano kao

$$N = K \backslash \{k\} \tag{2.12}$$

$$w_{k,u} = \cos(\vec{r_k}, \vec{r_u}) = \frac{\vec{r_k} * \vec{r_u}}{\|\vec{r_k}\| \times \|\vec{r_u}\|} = \frac{\sum_{i=1}^{m} r_{k,i} * r_{u,i}}{\sqrt{\sum_{i=1}^{m} r_{k,i}^2} * \sqrt{\sum_{i=1}^{m} r_{u,i}^2}} \tag{2.13}$$

Algoritam:

Algorithm 1 Korisnik-korisnik filtriranje

Ulaz: k - korisnik za kojeg se traži predikcija. N - susjedstvo korisnika

Izlaz: Predikcija p ocjene korisnika K za predmet P.

topN := 20

n := length(N)

w := initVector()

for ($i := 0; i < n; inc(i)$) **do**

  $w_i := calculatePearson(k, N_i)$

end for

sort(w)

$suma := 0; tezine := 0$

for ($i := 0; i < topN; inc(i)$) **do**

  $suma := suma + w_i * N_i$

  $tezine := tezine + w_i$

end for

$p := suma/tezine$

return p

2.3.3. Hibridni tehnike

2.3.4. Moguća područja primjene

3. Modeliranje podataka

3.1. Hijerarhijsko modeliranje

3.2. Modeliranje korisnika

3.3. Modelrianje predmeta

4. Problem vremena i prostora

Osnovni problem s kojim se konvencionalni preporučiteljski sustavi susreću (a time i popularniji radni okviri i biblioteke koji ih implementiraju) jest izostanak bilo kakve potpore za prostorne i vremenske komponente koje su se pokazale neophodne za rad sa sveprisutnim sustavima.

4.1. Vremenska komponenta

4.2. Prostorna komponenta

5. Razvoj algoritma i radnog okvira

6. Testiranje i evaluacija

6.1. Metodologija

6.2. Testiranje

6.3. Evaluacija preporučitelja

7. Zaključak

Zaključak.

LITERATURA

[1] Gartner. *Forecast: The Internet of Things, Worldwide, 2013.*, December 2013. URL `http://www.gartner.com/newsroom/id/2636073`.

[2] ABI Research. *More Than 30 Billion Devices Will Wirelessly Connect to the Internet of Everything in 2020*, May 2013. URL `https://www.abiresearch.com/press/more-than-30-billion-devices-will-wirelessly-conne/`.

[3] Mark Weiser. The computer for the 21st century. *ACM SIGMOBILE Mobile Computing and Communications Review*, 1999.

Preporučiteljski sustavi u sveprisutnom računarstvu

Sažetak

Sažetak na hrvatskom jeziku.

Ključne riječi: Ključne riječi, odvojene zarezima.

Recommender systems in ubiquituous computing

Abstract

Abstract.

Keywords: Keywords.