

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1136

# **Preporučiteljski sustavi u sveprisutnom računarstvu**

Branimir Pervan

Zagreb, lipanj 2015.

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

*Ovdje dolazi zahvala*

# SADRŽAJ

<b>1. Uvod</b>	<b>1</b>
<b>2. Sveprisutno računarstvo</b>	<b>3</b>
2.1. Uvod u sveprisutno računarstvo . . . . .	3
2.2. Razvoj . . . . .	3
2.3. Zahtjevi . . . . .	4
2.4. Primjena . . . . .	5
<b>3. Preporučiteljski sustavi</b>	<b>6</b>
3.1. Uvod u preporučiteljske sustave . . . . .	6
3.2. Razvoj preporučiteljskih sustava . . . . .	8
3.3. Filtriranje neovisno o korisniku . . . . .	9
3.4. Filtriranje ovisno o korisniku . . . . .	12
3.4.1. Filtriranje zasnovano na sadržaju . . . . .	12
3.4.2. Filtriranje zasnovano na suradnji . . . . .	16
3.5. Redukcija dimenzija . . . . .	20
3.6. Hibridne tehnike . . . . .	21
3.7. Moguća područja primjene . . . . .	23
3.8. Preporučiteljski sustavi u sveprisutnom računarstvu . . . . .	24
<b>4. Izgrađeni model preporučiteljskog sustava</b>	<b>25</b>
4.1. Preporučiteljski sustavi s izraženom prostornom i vremenskom komponentom . . . . .	25
4.2. Prostorna komponenta . . . . .	25
4.3. Vremenska komponenta . . . . .	27
4.4. Modeliranje korisnika . . . . .	30
4.5. Modeliranje predmeta . . . . .	31
4.6. Izgrađeni model . . . . .	31

<b>5. Implementacija i ispitivanje modela</b>	<b>36</b>
5.1. Arhitektura implementacije . . . . .	36
5.2. Ispitni scenarij . . . . .	39
5.3. Simulacija primjene . . . . .	41
5.3.1. Ulazi . . . . .	42
5.3.2. Izlazi . . . . .	45
5.4. Rezultati ispitivanja . . . . .	46
5.4.1. Ispitivanje prostornog filtera . . . . .	46
5.4.2. Ispitivanje vremenskog filtera . . . . .	47
5.4.3. Ispitivanje sadržajnog filtera . . . . .	47
5.4.4. Ispitivanje suradničkog filtera . . . . .	48
5.4.5. Ispitivanje sustava za pojačavanje obilježja . . . . .	49
<b>6. Zaključak</b>	<b>51</b>
<b>Literatura</b>	<b>52</b>
<b>A. Primjer dnevničke datoteke</b>	<b>55</b>
<b>B. Datoteka s podacima o predmetima</b>	<b>56</b>
<b>C. Datoteka s matricom ocjena korisnika</b>	<b>57</b>

# 1. Uvod

U posljednjih dvadesetak godina razvoj Interneta stvari (engl. *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 [4] odnosno 30 milijardi [9]. Tomu značajno doprinosi i konstantno opadanje cijene proizvodnog procesa tehnologije koja naizgled obične stvari na neki način čini inteligentnima i sposobnima za komunikaciju. Internet stvari samo je jedna od mogućih manifestacija sveprisutnog računarstva. Sveprisutno računarstvo, kao koncept u računarskoj znanosti gdje je računarstvo prisutno svugdje [15], 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. Čovječanstvo je danas nepobitno zagazilo u treće računarsko doba koje nakon doba osobnih računala predviđa prožimanje računala u svakodnevnicu. To nedvojbeno znači da će se broj računala povećati, a time će se povećati i količina podataka koja će biti prikupljena ili generirana od strane takvih računala. Također, zbog činjenice sveprisutnosti, kontekst, a pogotovo prostor i vrijeme, u kojem se računalo nalazi postaje važna karakteristika za opisivanje i korištenje sustava.

S druge strane, preporučiteljski sustavi su se kroz posljednjih tridesetak godina nametnuli kao standard u filtriranju sadržaja u onim sustavima i uslugama koje imaju značajan udio dodira s krajnjim korisnicima. Moderne usluge i sustavi sa sadržajem gotovo su nezamislivi bez preporučitelja zbog količine informacija, u smislu predmeta preporuke, koja postoji u takvim sustavima. Primjer se može naći u modernijim internetskim dućanima u kojima korisnik obično ne može znati koji su sve predmeti dostupni i koji su sve predmeti povezani s onim predmetima koji su njemu od interesa. S druge strane, sustavi multimedije na zahtjev (engl. *multimedia-on-demand*) općenito sadrže velike količine multimedije za koju korisnik ne zna da postoji. Nerijetko se zbog toga zna dogoditi da korisnik eksplicitno traži od sustava preporuku za glazbu ili film koji bi mu se mogao svidjeti.

S obzirom na činjenicu da će sveprisutno računarstvo nastaviti razvoj u započetom smjeru, prirodno se nameće potreba za preporučiteljskim sustavima u takvoj vrsti

računarstva. Usluge koje sveprisutno računarstvo već sada nudi ili će tek nuditi neophodno će zahtijevati filtriranje informacija. Preporuke koje će takvi sustavi davati morat će biti svjesne konteksta, a pogotovo prostora i vremena u kojem se nalaze. Potreba i smisao izučavanja sinteze ova dva područja dolazi iz očitog primjera za ulaganjem u bolje i efikasnije algoritme jer je u nepreglednoj masi informacija, kakav je Internet stvari i sveprisutno računarstvo idealan izvor, procesna moć današnjih računala davno izgubila bitku.

Motiv ovog diplomskog rada jest manjak dostupnih algoritama i radnih okvira specifičnih za preporučiteljske sustave u sveprisutnom računarstvu. U ovom radu dat će se kratak uvod u sveprisutno računarstvo, teorijska podloga bazičnih algoritama za filtriranje sadržaja te analizirati prednosti i nedostatke takvih pristupa. Analizirat će se posebni zahtjevi na preporučiteljske sustave od strane sveprisutnih aplikacija. Prikazat će se principi rada osmišljenog preporučiteljskog sustava, te će se analizirati komponente istog sustava. Na kraju prikazat će se jednostavno testiranje izgrađenog sustava na konkretnom scenariju.

## 2. Sveprisutno računarstvo

### 2.1. Uvod u sveprisutno računarstvo

A šta koji drek ovdje da pričam. Što je sveprisutno računarstvo, tko ga je definirao, što u njemu dolazi do izražaja, koje su koristi?

Termin „Sveprisutno računarstvo“ prvi je upotrijebio Mark Weiser u svom vizionarskom članku u kojem je rekao kako su najkorisnije one one tehnologije koje nestaju, u smislu da korisnici izgube pojam o korištenju te tehnologije [14].

Cilj filozofije sveprisutnog računarstva nije staviti čovjeka u svijet računala nego integracija računala u svijet čovjeka. Praktična posljedica toga jest pojava velike količine mahom nestrukturiranih podataka koje mogu predstavljati izvrstan materijal za daljnju obradu i izučavanje algoritama kojima se ti podaci mogu upotrijebiti za analizu ponašanja čovjeka ili za pomoć u svakodnevnim poslovima [13].

### 2.2. Razvoj

Razvoj sveprisutnog računarstva započeo je devedesetih godina prošlog stoljeća a prethodile su mu dva računalna razdoblja. U općem slučaju mogu se razlikovati tri velika vala, a to su:

1. Radne stanice (engl. *Mainframe*) - jedno računalo, više osoba
2. Osobna računala (engl. *PC, Personal Computer*) - jedno računalo, jedna osoba
3. Sveprisutno računarstvo (engl. *Ubiquitous Computing*) - više računala, jedna osoba

Slika 2.1 prikazuje odnose računalnih razdoblja.





Slika 2.1: Tri vala u računarstvu

## 2.3. Zahtjevi

Analiziranjem zahtjeva na sveprisutno i prožimajuće računarstvo, može se zaključiti nešto:

- Mali uređaji
- Umrežavanje
- Sakupljanje podataka i razmjena u okruženju bez prisustva korisnika
- Otvoreno računarstvo radi podložnosti promjene softwarea i uređaja (Java ili CLR ili nekaj takvo -> bytecode)

Još jedna lista zahtjeva na sveprisutno

- Uređaj za prikaz, terminal, korisnički interface
- Ekonomična cijena
- Širok mrežni opseg
- Sistem nevidljivih datoteka (bez znanja o direktorijima, nazivu datoteka, lokaciji td.)
- Automatska instalacija - migracija s jednog računala na drugi
- Personaliziranost informacija
- Privatnost (ima puno personaliziranih informacija, što ako se netko dokopa?)

koje su glavne potrebe za rad sa sveprisutnim sustavima (mala energija, procesna moć, interakcija s okolinom)

## 2.4. Primjena

Sveprisutno računarstvo već sada nalazi svoju primjenu u raznim segmentima Udri brigu na veselje u ovom poglavlju natuće za sva vremena xD

Pametni gradovi, pametni prostori, internet stvari.

Aspekt integracije: postojeće stvari -> nova tehnologija (orijentacija u muzeju s mobitelom)

Primjene u urbanom računarstvu:

- Javne i komercijalne infrastrukture
- Mobilne i druge bežične mreže
- Kolektivni transport
- Plaćanje
- Sigurnost i nadgledanje
- Rasprostranjeno reklamiranje
- Pametna arhitektura
- Spektakularni urbani događaji
- Mobilni tehnološki uređaji
- Osobna komunikacija
- Servisi socijalnih mreža
- Urbana umjetnost?

Urbani krajolik -> skriveni slojevi mogu postati vidljivi, svojevrsni izum mikroskopa.

Urbano računarstvo -> ljudi koji žive u urbanoj sredini mogu imati mnogo različitih interesa ali imaju jednu stvar koja im je zajednička: mjesto gdje žive (ovo bi se moglo primijeniti na činjenicu da svi imaju isti cilj u šopingu)

## 3. Preporučiteljski sustavi

### 3.1. Uvod u preporučiteljske sustave

Preporučiteljski sustavi su, ukratko rečeno, podrazred sustava za filtriranje podataka kojima je svrha predviđanje ocjene (ili preferencije) kojom bi neki korisnik ocijenio neki predmet u sustavu [11]. Predmeti nad kojima takvi sustavi rade, kao i korisnici tog sustava opisani su određenim karakteristikama, odnosno važnostima tih karakteristika za korisnike. Općenito, može se reći da je jednostavan model preporučiteljskog sustava dan formulom:

$$R \leftarrow U \times I \quad (3.1)$$

gdje je  $R$  rezultat, tj. predikcija (engl. *prediction, recommendation*), ocjene korisnika  $U$  (engl. *user*) koji je zatražio preporuku, tj. filtriranje sadržaja, a  $I$  predmet nad kojim se vrši predikcija ocjene (engl. *item*). Drugim riječima, predikcija je posljedica karakteristike, tj. preferencije korisnika, te karakteristike predmeta. 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. Dakle, preporuka  $R$  zapravo je predviđanje ocjene korisnika  $U$  za traženi predmet  $I$ . Krajnji rezultat na kraju jest najčešće lista od  $n$  najboljih preporuka, tj. pretpostavki da bi korisnik te predmete ocjenio najbolje (engl. *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.

U svrhu rješavanja gore navedenih problema, razmotrit će se razni modeli preporuke od kojih su neki već dobro poznati i korišteni algoritmi. Neka je  $C$  kontekst u

kojem korisnik  $U$  traži preporuku. Formula 3.1 tada prelazi u:

$$R \leftarrow U \times I \times C \quad (3.2)$$

dok predikcija  $R$  postaje posljedica karakteristike, tj. preferencije korisnika, karakteristike predmeta i konteksta u kojem se vrši predikcija. Kontekst  $C$  se ne shvaća atomarno i u sebi može sadržavati više različitih komponenata koje mogu utjecati na preporuku, npr:

- Vrijeme predikcije
- Mjesto na kojem se traži predikcija
- Prisutnost drugih korisnika u trenutku predikcije
- Skupine predmeta, odnosno resursa, koje se nalaze u blizini

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 [1]. 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, demografski podaci, točnije, šire područje iz kojeg korisnik koristi sustav a jednostavno se doznaje iz baze podataka dodijeljenih područja (engl. *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 (engl. *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:

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

Iako se tehnički iz filtriranja neovisnog o korisniku može izgraditi preporučiteljski sustav, on to nije u punom smislu riječi jer praktički izostavlja ulogu korisnika u sustavu. Drugim riječima, svaki korisnik će dobiti istu preporuku. Zbog toga ga se može nazvati svojevrsnim *pseudopreporučiteljem*.

## **3.2. Razvoj preporučiteljskih sustava**

Povijesno gledano, razvoj preporučiteljskih sustava započeo je devedesetih godina prošlog stoljeća, a nemalo je populariziran 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 [2]. 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

### 3.3. 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. Ova tvrdnja može se jednostavnije prikazati relacijom:

$$R \leftarrow I \quad (3.3)$$

gdje je  $R$  predikcija, tj. preporuka, a  $I$  predmet. Iz relacije 3.3 očigledno je da je predikcija funkcija isključivo predmeta, pa kao takva ne može biti smatrana punokrvnim preporučiteljem. S druge strane, isplati se proučiti ovu vrstu filtriranja jer neki njezini koncepti vrijede u općem slučaju, npr. sustav ocjenjivanja i sustav prikazivanja rezultata.

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

$$S = \{1, 2, 3, 4, 5\} \quad (3.4)$$

$$R = \lfloor \frac{\sum_{i=1}^N s_i}{N} * 10 \rfloor \quad (3.5)$$

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 (engl. *Feedback*), npr. *eBay*, *Tripadvisor* i *Zagat*.

Recent Feedback ratings (last 12 months) ?				Detailed seller ratings (last 12 months) ?		
	1 month	6 months	12 months	Criteria	Average rating	Number of ratings
Positive	281	1550	2346	Item as described	★★★★★	2012
Neutral	0	6	9	Communication	★★★★★	1982
Negative	1	2	3	Shipping time	★★★★★	2000
				Shipping and handling charges	★★★★★	2122

Slika 3.1: Primjer rejtinga na internetskoj aukcijskoj kući *eBay*

Elementi ovog preporučitelja prikazani relacijama 3.4 i 3.5 mogu se varirati kako bi se prilagodio 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 3.5 prije zaokruživanja prosjek je pomnožen faktorom 10 radi eliminacije decimala. Moguće je odabrati neki drugi prikaz rezultata, primjerice u postotcima (slika 3.2).

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



**Slika 3.2:** Implicitna preporuka trgovca izražena postotcima

Sam način ocjenjivanja ne mora nužno biti eksplicitna dodjela ocjene. Moguće je primjerice koristiti sustav glasovanja (engl. *Vote up/down*) (engl. *Vote up/down*). Pomoću takvog sustava, korisnici masovno određuju koji je sadržaj kvalitetniji, odnosno, koji je manje kvalitetan te na taj način vrše implicitnu preporuku. Primjer je internet-ska stranica za programerska pitanja i odgovore *StackOverflow* (slika 3.3). Na ovoj stranici, nakon postavljenog pitanja, korisnici daju odgovore, a sustav na više pozicije postavlja one odgovore kojima su korisnici dali više glasova, a na niže pozicije lošije i negativno rangirane odgovore. Također, korisnici mogu glasovati na kvalitetu postavljenog pitanja pa time nekvalitetna i loše postavljena pitanja eliminirati (tj. ne preporučiti) za odgovaranje.

Drugi najpoznatiji primjer koji koristi takav sustav dodjele ocjena jest internetski portal *Reddit* (slika 3.4).

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.

1525 The following is an attempt to describe the Ukkonen algorithm by first showing what it does when the string is simple (i.e. does not contain any repeated characters), and then extending it to the full algorithm.



#### First, a few preliminary statements.



+100

1. What we are building, is *basically* like a search trie. So there is a root node, edges going out of it leading to new nodes, and further edges going out of those, and so forth
2. **But:** Unlike in a search trie, the edge labels are not single characters. Instead, each edge is labeled using a pair of integers `[from,to]`. These are pointers into the text. In this sense, each edge carries a string label of arbitrary length, but takes only  $O(1)$  space (two pointers).

Slika 3.3: Primjer rejtinga na internetskom portalu *StackOverflow*

The screenshot shows a list of posts on Stack Overflow. Each post has an upvote/downvote arrow on the left, a score, a title, a snippet of the body text, and a 'share' link. The posts are as follows:

- Score: 2110. Title: "Please copy/paste the relevant text from your source in the comments (+ a note on verifiable titles)". Submitted 7 months ago by TILMods. 127 comments.
- Score: 4808. Title: "TIL Sweden has instituted a country-wide program where citizens can enroll to receive an SMS when there is a heart attack victim nearby, allowing them to reach them faster than an ambulance and provide CPR. In 40% of the cases, SMSlifesavers arrived before ambulances and started providing CPR." Submitted 10 hours ago by TheMagnificentCazMo. 1031 comments.
- Score: 4415. Title: "TIL that every winter at a research station in the South Pole, 50 people are left totally isolated for 8 months. An annual tradition for the personnel is the back to back watching of the 3 'The Thing' films after the last plane has left for the winter". Submitted 13 hours ago by CRVaz. 837 comments.
- Score: 3415. Title: "TIL that guitarist Les Paul's right elbow was shattered in a car accident. Doctors were unable to repair it so he had them fuse his arm bones at a 90 degree angle so he could continue to play guitar." Submitted 14 hours ago by kirk13s. 137 comments.
- Score: 4120. Title: "TIL that every Dutch police car is 'equipped' with a teddy bear, in case something troubling happens to a child." Submitted 18 hours ago by Noticemenot. 1736 comments.

Slika 3.4: Primjer rejtinga na internetskom portalu *Reddit*

– Nedostatak konteksta za preporuke.

Nepersonalizirana preporuka može se izvesti i asocijativno, traženjem implicitne korelacije između dvaju ili više predmeta. Naivno gledano, može se zaključiti ako je relativna većina korisnika uz kupnju nekog proizvoda  $i_1$  kupila i proizvod  $i_2$ , da bi i budućim korisnicima trebalo uz proizvod  $i_1$  preporučiti proizvod  $i_2$ , i obrnuto. Ova tvrdnja može se opisati formulom 3.6:

$$R = \frac{|X \cap Y|}{|X|} \quad (3.6)$$

gdje je  $X$  skup svih korisnika koji su kupili predmet  $i_1$ ,  $Y$  skup svih korisnika koji su kupili predmet  $i_2$  a  $R$  ocjena ovisnosti između predmeta  $i_1$  i  $i_2$ . Zbog činjenice da je kardinalni broj presjeka dvaju skupova uvijek manji ili jednak kardinalnom broju bilo kojeg od ta dva skupa, ocjena  $R$  nalazi se u intervalu  $0 \leq R \leq 1$ .

Problem ove metode preporučivanja jest navedeni nedostatak konteksta za prepo-



ruke. Taj nedostatak može se manifestirati u činjenici da su neki predmeti neovisno popularni. Primjerice, ako je neki predmet popularan, naivni preporučitelj može koristeći 3.6 (opravdano) stvoriti vezu između tog predmeta i bilo kojeg drugog predmeta. Primjer dan u [5] pokazuje kako su u nekom sustavu posebno popularan predmet banane i kako naivni preporučitelj stvara veze između banana i nekih drugih generičkih predmeta, pa onda pri kupnji banana preporučuje te generičke predmete i obratno. Ovaj efekt može se eliminirati normalizacijom relacije 3.6:

$$R = \frac{\frac{|X \cap Y|}{|X|}}{\frac{|\bar{X} \cap Y|}{|\bar{X}|}} \quad (3.7)$$

gdje su  $X$  i  $Y$  skupovi svih korisnika koji su kupili predmete  $i_1$  i  $i_2$  respektivno.

### 3.4. Filtriranje ovisno o korisniku

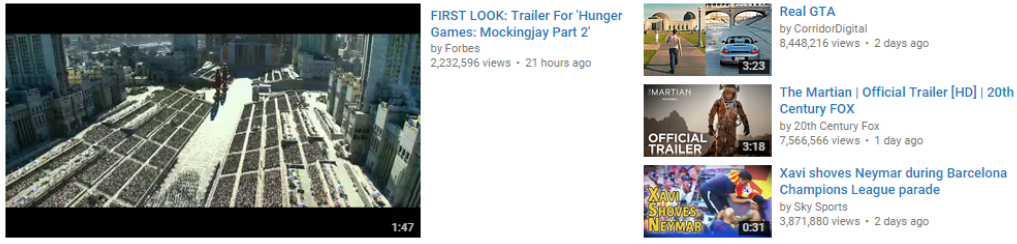
Grubi model filtriranja ovisnog o korisniku, prema relaciji 3.1 računa predikciju kao produkt karakteristike predmeta i preferencije korisnika, no zanemaruje kontekst, osim ako on nije implicitno uključen u komponentu korisnika ili predmeta. U općem slučaju razlikuju se dvije vrste filtriranja ovisnog o korisniku: filtriranje zasnovano na sadržaju te suradničko filtriranje.

#### 3.4.1. Filtriranje zasnovano na sadržaju

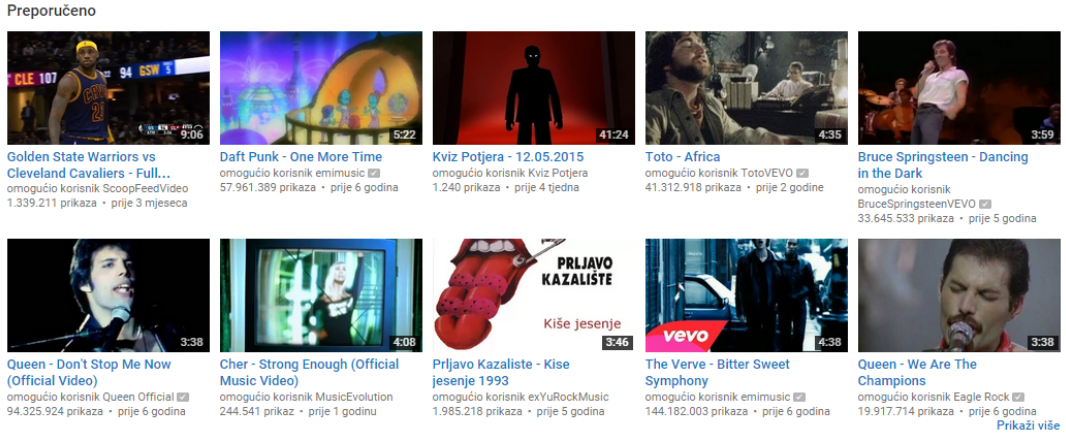
U gotovo svim primjenama preporučitelja, tj. u sustavima u kojima se oni koriste, predmeti ili korisnici nisu osnovne (atomarne) jedinice, nego ih se može opisati nekim kategorijama, primjerice, demografskim podacima za korisnika, autorom i izdavačem ako je predmet neka knjiga i sl. Preporučivanje zasnovano na sadržaju u osnovi dovodi u vezu prikupljene preferencije korisnika, bilo eksplicitno, bilo promatrajući povijest ponašanja, i karakteristike kojima je opisan neki predmet [11]. Na slici 3.5 vidljiv je primjer preporučiteljskog rezultata bez ikakvih informacija o korisniku.

Podaci na temelju kojih je generirana preporuka prikupljeni su implicitno, u vidu grube lokacije korisnika (na slici je lokacija Ujedinjeno Kraljevstvo). S druge strane, slika 3.6 prikazuje uključene podatke o korisniku, pa je lako vidljiva značajna razlika u generiranoj preporuci između dvije verzije preporučitelja u kojoj isti algoritam ima pristup različitim širinama skupova informacija.

Neka je u sustavu koji koristi preporučitelj zasnovan na sadržaju svaki predmet



Slika 3.5: Preporuka zasnovana na sadržaju i implicitno prikupljenim podacima



Slika 3.6: Preporuka zasnovana na sadržaju i eksplicitno personalizirana

opisan tekstualnim medapodacima i vektorom:

$$\mathbf{X}_i = [w_{1,i}, w_{2,i}, w_{3,i}, \dots, w_{N,i}]^T \quad (3.8)$$

gdje je  $w_{j,i}$  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 preciznijih metrika. Sličnost između dvaju predmeta,  $i_1$  i  $i_2$  moguće je tada izraziti kosinusom kuta između vektora njihovih karakteristika:

$$\cos(\mathbf{I}_1, \mathbf{I}_2) = \frac{\mathbf{I}_1 * \mathbf{I}_2}{\|\mathbf{I}_1\| \times \|\mathbf{I}_2\|} \quad (3.9)$$

Teoretski, svaka karakteristika može imati svoju težinu, u smislu da može biti važnija, odnosno manje važna za opis nekog predmeta. Taj pristup svodi se na algoritam *Redukcije dimenzija* i dekompoziciju matrice na singularne faktore, a bit će opisan naknadno.

Neka također 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:

$$\cos(\mathbf{U}, \mathbf{I}) = \frac{\mathbf{U} * \mathbf{I}}{\|\mathbf{U}\| \times \|\mathbf{I}\|} \quad (3.10)$$

Vektore je poželjno normalizirati, TFIDF

S obzirom na relacije 3.9 i 3.10 slijedi algoritam za preporučivanje zasnovano na sadržaju.

---

**Algorithm 1** Filtriranje zasnovano na sadržaju

---

```
1: Ulaz:  $k$  - vektor ocjena korisnika.  $I$  - matrica u kojoj su retci predmeti, a stupci
   karakteristike
2: Izlaz: Lista  $L$  najboljih  $N$  predmeta za korisnika  $k$ 
3:  $L := \text{initList}(N)$ ;
4:  $\text{idfPolje} := \text{initPolje}(\text{width}(I)); \text{userPolje} := \text{initPolje}(\text{width}(I))$ 
5: for ( $i := 0; i < \text{height}(I); \text{inc}(i)$ ) do
6:    $\text{suma} := 0$ 
7:   for ( $j := 0; j < \text{width}(I); \text{inc}(j)$ ) do
8:     if ( $i_j \neq 0$ ) then
9:        $\text{inc}(\text{suma})$ ;
10:    end if
11:  end for
12:  for ( $j := 0; j < \text{width}(I); \text{inc}(j)$ ) do
13:     $i_j := 1/\sqrt{\text{suma}}$ ;
14:  end for
15: end for
16:  $\text{count} := 0$ ;
17: for ( $i := 0; i < \text{width}(I); \text{inc}(i)$ ) do
18:   for ( $j := 0; j < \text{height}(I); \text{inc}(j)$ ) do
19:      $\text{userPolje}_i := \text{userPolje}_i + I_{i,j} * k_j$ ;
20:     if ( $I_{i,j} \neq 0$ ) then
21:        $\text{inc}(\text{count})$ ;
22:     end if
23:   end for
24:    $\text{idfPolje}_i := 1/\text{count}$ ;
25:    $\text{count} := 0$ ;
26: end for
27:  $L_i := L_i + i_j * \text{idf}_j * \text{userProfile}_j$ 
28: for ( $i := 0; i < \text{height}(I); \text{inc}(i)$ ) do
29:   for ( $j := 0; j < \text{width}(I); \text{inc}(j)$ ) do
30:      $L_i := L_i + I_j * \text{userProfile}_j * \text{idfPolje}_j$ ;
31:   end for
32: end for
33: return  $L$ 
```

---

Jedna od većih prednosti ovog načina filtriranja jest 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 (engl. *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.

### 3.4.2. 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 ocijenila i neke druge predmete razmjerno velika. S druge strane, veća je vjerojatnost da će neki korisnik ocijeniti slično neka dva predmeta ako su ih i ostali korisnici slično ocijenili.

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

- Korisnik-korisnik (engl. *User-user, Neighbourhood-based, Memory-based*)
- Predmet-predmet (engl. *Item-item, Item-based, Model-based*)

#### Korisnik – Korisnik

Suradničko filtriranje na relaciji između korisnika koji traži preporuku i ostalih korisnika sustava svodi se na predikciju ocjene korisnika koji traži preporuku za neki predmet na osnovu ocjena njemu bliskih ljudi. 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:

$$N = K \setminus \{k\} \quad (3.11)$$

Bliskost, odnosno udaljenost dvaju korisnika dobro je definirana nekom metrikom,

npr. Pearsonovim koeficijentom korelacije (3.12):

$$R = \frac{\sum_{i \in I} [(r_{k,i} - \bar{r}_k) * (r_{u,i} - \bar{r}_u)]}{\sqrt{\sum_{i \in I} (r_{k,i} - \bar{r}_k)^2} * \sqrt{\sum_{i \in I} (r_{u,i} - \bar{r}_u)^2}} \quad (3.12)$$

gdje je  $w_{q,u}$  težina između aktivnog korisnika  $k$  i korisnika  $u$ ,  $i$  skup predmeta koje su ocijenila oba korisnika,  $r_{u,i}$  ocjena koju je korisnik  $u$  dodijelio predmetu  $i$ , a  $\bar{r}_u$  srednja vrijednost svih ocjena koje je dodijelio korisnik  $u$ . Alternativno, ocjene korisnika mogu se predstaviti kao vektore u  $m$  - dimenzionalnom prostoru pa se udaljenost, tj. bliskost može izraziti preko skalarnog produkta kao kosinus kuta između ta dva vektora:

$$w_{k,u} = \cos(\vec{r}_k, \vec{r}_u) = \frac{\vec{r}_k * \vec{r}_u}{\|\vec{r}_k\| * \|\vec{r}_u\|} = \frac{\sum_{i=1}^m r_{k,i} * r_{u,i}}{\sqrt{\sum_{i=1}^m r_{k,i}^2} * \sqrt{\sum_{i=1}^m r_{u,i}^2}} \quad (3.13)$$

Mana pristupa 3.13 jest nedostatak negativnih ocjena.

Iako prema 3.11 u susjedstvo spadaju svi korisnici sustava, u praksi se uzima najbližih  $n$  određenih pomoću 3.12 ili 3.13. S obzirom na tu činjenicu i navedene relacije, slijedi relacija za izračun predviđanja:

$$p_{u,i} = \bar{r}_u + \frac{\sum_{j=1}^n (r_j - \bar{r}_j) * w_j}{\sum_{j=1}^n w_j} \quad (3.14)$$

gdje je  $p_{u,i}$  predviđanje ocjene korisnika  $u$  za predmet  $i$ ,  $r_j$  ocjena koju je  $j$  susjed korisnika  $u$  dao predmetu  $i$ ,  $\bar{r}_j$  prosjek ocjena koje dodjeljuje korisnik  $j$ ,  $w_j$  težina između korisnika  $u$  i susjeda  $j$ , a  $\bar{r}_u$  prosjek ocjena koje dodjeljuje korisnik  $u$ .  $\bar{r}_u$  i  $\bar{r}_j$  su zapravo normalizacijski članovi jer različiti korisnici mogu imati individualne kriterije i skale.

Algoritam za filtriranje na relaciji korisnik-korisnik slijedi:

---

**Algorithm 2** Korisnik-korisnik filtriranje

---

```
1: Ulaz:  $k$  - korisnik za kojeg se traži predikcija.  $I$  - matrica ocjena korisnika gdje  
   su stupci predmeti, a retci korisnici.  $N$  - veličina susjedstva  
2: Izlaz: Lista  $L$  najboljih predmeta za korisnika  $k$ .  
3:  $L := \text{initList}(\text{width}(I));$   
4:  $I^T := \text{transpose}(I);$   
5:  $\text{correlMatrix} := \text{calculatePearsonElementwise}(N, N^T);$   
6:  $k_a := \text{calculateAverageForUser}();$   
7:  $A := \text{calculateAveragesForNeighborhood}();$   
8:  $\text{topNForUser} := \text{sort}(\text{row}(\text{correlMatrix}_k)).\text{takeFirstN};$   
9: for ( $i := 0; i < \text{width}(I); \text{inc}(i)$ ) do  
10:    $\text{sum} := 0;$   
11:    $\text{sumWeight} := 0;$   
12:   for ( $j := 0; j \leq N; \text{inc}(j)$ ) do  
13:      $\text{sum} := \text{sum} + ((I_{i,j} - A_j) * \text{correlMatrix}_{k,j})$   
14:      $\text{sumWeight} := \text{sumWeight} + \text{correlMatrix}_{k,j};$   
15:   end for  
16:    $L_i := k_a + (\text{sum} / \text{sumWeight});$   
17: end for  
18: return  $\text{sort}(L);$ 
```

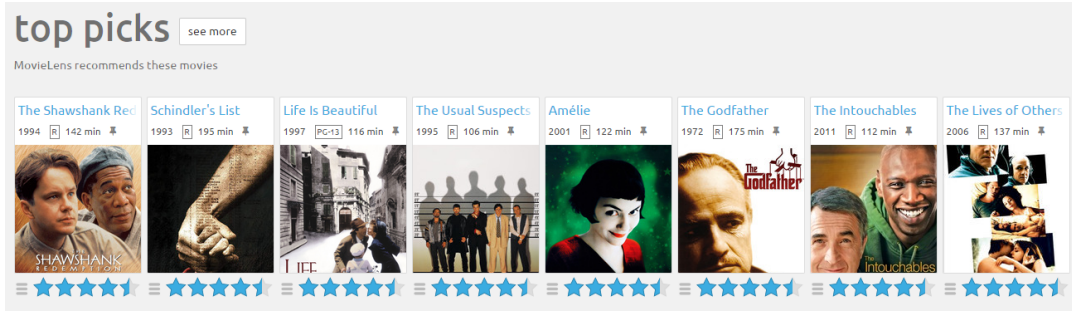
---

Indeks  $j$  u algoritmu 2 se odnosi na onaj redak, odnosno stupac matrice po referenci na promatranog susjeda, a ne slijedno od nultog elementa matrice.

Algoritam se ugrubo svodi na:

1. Račun korelacije između svih korisnika u sustavu korištenjem 3.12
2. Odabir  $n$  susjeda koji imaju najveće težine s obzirom na korisnika  $k$  gdje je broj tih susjeda ( $\text{topN}$ ) određen empirijski.
3. Izračun prosječne ocjene kojom svaki susjed  $n \in N$  ocjenjuje predmete
4. Izračun prosječne ocjene kojom ocjenjuje korisnik  $k$
5. Izračun predviđanja korištenjem normalizacijske formule 3.14

Primjer korištenja ovog pristupa je „MovieLens“ sustav razvijen od strane istraživačke skupine „GroupLens“ [10]. Taj preporučitelj na temelju korisničkih ocjena i



Slika 3.7: Primjer preporuke sustava „MovieLens“

odabira traži slične korisnike sustava te krajnjem korisniku daje listu filmova koji su dovoljno dobro ocjenjeni od strane tih sličnih korisnika. Primjer je vidljiv na slici 3.7.

Ovaj pristup je relativno efikasan i jednostavan za implementaciju, ali druge strane, računalno postaje vrlo zahtjevan ako se primjeni na velike baze podataka korisnika i predmeta. Računanje transponirane matrice u općem slučaju je složenosti  $O((n * (n - 1))/2)$ . Računanje Pearsonovog koeficijenta korelacije između svih korisnika iznosi  $O(n^3/2)$  jer je složenost računa  $O(n)$ , a potrebno je izvesti račun za gornji trokut kvadratne matrice širine  $n$ . Na kraju složenost računa normaliziranog prosjeka iznosi  $O(n * s)$  gdje je  $s$  veličina susjedstva. Naposljetku algoritam 2 daje složenost:

$$O\left(\frac{n * (n - 1)}{2} + \frac{n^3}{2} + (n * s)\right) = O(n^3) \quad (3.15)$$

Ocjena 3.15 ima kvadratnu ovisnost i s povećanjem broja korisnika dolazi do nemalog povećanja broja operacija za izračun preporuke. Valja dodati i da je algoritam 2 pojednostavljena verzija algoritma, pa ni ocjena 3.15 ne može biti uzeta kao konačna.

## Predmet – Predmet

S obzirom na nedostatke prethodne tehnike u performansama, inženjeri poznate internetske trgovine *Amazon.com* predložili su sličan pristup ali iz druge perspektive gdje se ne traže sličnosti između korisnika, nego između predmeta [6]. Kao i kod prethodno opisanog pristupa i algoritma 2, metrika za ocjenu sličnosti između predmeta je Pearsonov koeficijent korelacije, u ovom slučaju definiran kao:

$$R = \frac{\sum_{u \in U} [(r_{u,i} - \bar{r}_i) * (r_{u,j} - \bar{r}_j)]}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_i)^2} * \sqrt{\sum_{u \in U} (r_{u,j} - \bar{r}_j)^2}} \quad (3.16)$$

gdje je  $U$  skup svih korisnika koji su ocjenili predmete  $i$  i  $j$ ,  $r_{u,i}$  ocjena korisnika  $u$  za predmet  $i$ , a  $\bar{r}_i$  srednja ocjena  $i$ -tog predmeta za sve korisnike. Radi jednostavnosti,



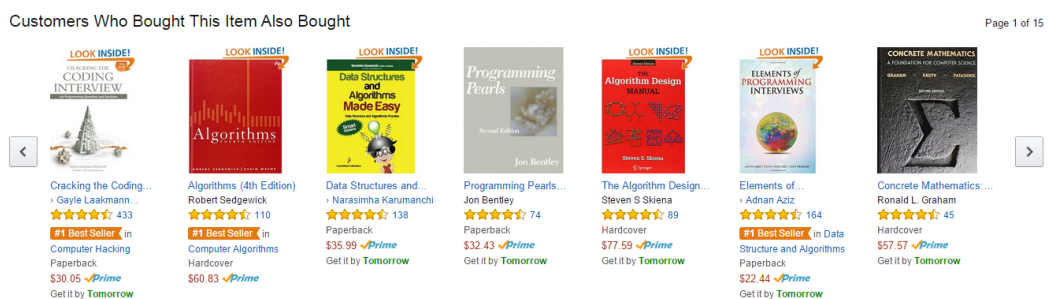
kod ovog algoritma matrica ocjena korisnika za predmete normalizira se oduzimanje prosječne ocjene kojom je svaki korisnik ocjenio predmete.

Slično kao i kod suradnje na razini Korisnik-korisnik, predviđanje ocjene korisnika  $k$  za predmet  $i$  računa se korištenjem težinskog prosjeka:

$$p_{k,i} = \frac{\sum_{j \in N} r_{k,j} * w_{i,j}}{\sum_{j \in N} \|w_{i,j}\|} \quad (3.17)$$

gdje je  $N$  susjedstvo predmeta ocjenjenih od strane korisnika  $k$  najslbličnijih predmetu  $i$ .

Jedan od najpoznatijih preporučitelja na svijetu, internetska trgovina *Amazon.com* koristi patentirani hibridni algoritam baziran na ovoj tehnici (slike 3.8 i 3.9).



Slika 3.8: Primjer preporuke sustava „Amazon.com“



Slika 3.9: Primjer preporuke sustava „Amazon.com“

### 3.5. Redukcija dimenzija

Skup algoritama redukcije dimenzija bazira se na činjenici da je u općem slučaju baza podataka korisnika i predmeta rijetko popunjena (engl. (.)sparse). Za većinu korisnika neće biti dostupna većina ocjena, tj. u matrici ocjena korisnika za predmete, većina elemenata će vjerojatno biti 0. S druge strane, filtriranje zasnovano na suradnji

i suradničko filtriranje u naprednijim slučajevima mogu biti naivni kada je u pitanju sinonimija između pojmova.

Neka je dan primjer sustava za prodaju knjiga. Ako je korisnik kupio više knjiga istog autora, može se pretpostaviti da njegovom profilu odgovara kupljeni autor. S druge strane, u općem slučaju i ukoliko se isključi mogućnost slučajnosti, knjiga o tom autoru vjerojatno mu neće biti preporučena jer suradničko filtriranje neće moći naći vezu između knjige o tom autoru i kupljenih knjiga, a sadržajno neće biti kompatibilne. Iz tog razloga algoritam redukcije dimenzija teži u izdvajanje kompleksnih karakteristika u zasebne podatkovne strukture koje ne ovise niti o korisniku, niti o predmetu.

Matematički gledano, algoritam se bazira na *SVD* dekompoziciji matrice korisnika i predmeta (engl. *Singular value decomposition*). Rezultat te dekompozicije je:

$$R = U * \sigma * V^* \quad (3.18)$$

gdje je  $U$  matrica dimenzija  $m \times r$ ,  $S$  dimenzija  $r \times r$ , a  $V^*$  dimenzija  $r \times n$ . Matrica  $S$  je dijagonalna matrica koja sadrži svojstvene vrijednosti, a te svojstvene vrijednosti su težine dimenzija koje doprinose konačnom rezultatu. Ukoliko se vrijednosti sortiraju silazno, može se pokazati da se većina elemenata nalazi u okolini nule [5]. To omogućuje da se odstrane ti elementi pa se matrica  $S$  reducira na dimenzije  $k \times k$ , matrica  $U$  na dimenzije  $m \times k$ , a matrica  $V^*$  na dimenzije  $k \times n$ . Konačni rezultat redukcije jest:

$$R_k = U_k * \sigma_k * V_k^* \quad (3.19)$$

$R_k$  najbliža *rank* –  $k$  matrica originalne matrice  $R$ , a  $k$  predstavlja broj dimenzija kompleksnih karakteristika. Izračun predikcije tada se svodi na skalarni produkt odgovarajućih vektora iz matrica  $U_k$  i  $V_k^*$  te svojstvenih vrijednosti iz dijagonalne matrice  $\sigma_k$ .

Mana ovog algoritma jest skupa i zahtjevna operacija *SVD* dekompozicije relativno velikih ulaznih matrica.

### 3.6. Hibridne tehnike

Hibridne tehnike uglavnom podrazumjevaju kombiniranje više različitih tehnika filtriranja. Razne okolnosti u kojima rade preporučiteljski sustavi dovode do različitih primjenjivosti sustava. Primjerice, suradničko filtriranje niti u teoriji ne može raditi dobro ukoliko nisu zadovoljeni neki preduvjeti, npr. određeni broj korisnika u sustavu. Zbog toga se kombinacijom više različitih preporučiteljskih tehnika nastoji proširiti

spektar okolnosti u kojima preporučiteljski sustav može raditi. Nadalje, razlog hibridizacije preporučitelja može biti, ne samo nemogućnost preporuke, nego traženje kvalitetnije preporuke u nekim okolnostima. Također, razlog mogu biti i performanse u različitim okolnostima, pa jedna tehnika može dati bolje rezultate u nekoj okolini, a neka druga tehnika bolje rezultate u nekoj drugoj okolini. Neki od poznatijih hibridnih preporučiteljskih tehnika su: [11]

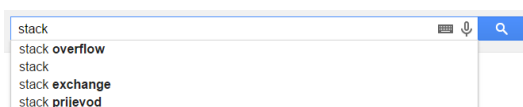
1. Težinski - ocjena predmeta preporuke računa se iz svih dostupnih preporučiteljskih tehnika u sustavu. Najjednostavniji primjer jest linearna kombinacija preporuka. Neki sustavi koji koriste ovu tehniku kombiniraju preporuke zasnovane na suradnji i preporuke zasnovane na sadržaju uz kalibraciju težina u ovisnosti o povratnoj informaciji korisnika.
2. Izmjenični - kod ove tehnike sustav koristi neki predefinirani kriterij za izmjenu tehnike preporuke, primjerice, neka je dan sustav koji inicijalno za svakog korisnika koristi preporuku zasnovanu na suradnji. Ako sustav tom tehnikom ne može stvoriti preporuku određenog nivoa pouzdanosti, prebacuje se na preporuku zasnovanu na sadržaju.
3. Miješani - ova tehnika računa preporuke iz više elementarnih preporučitelja i prikazuje ih simultano.
4. Kombinacija obilježja - jedan od načina za spajanje suradničkog i sadržajnog preporučitelja jest interpretirati informacije suradničkog preporučitelja kao dodatne podatke koji čine ulaz sadržajnog preporučitelja.
5. Kaskada - ova tehnika predstavlja svojevrsnu dvorazinsku preporuku. Prvo jedan preporučitelj relativno grubo rangira potencijalne kandidate, a nakon toga drugi preporučitelj radi precizniju razdiobu. Tehnika je posebno pogodna za one slučajeve kada jedan preporučitelj ne može dovoljno precizno generirati predviđanja za preporuke predmeta.
6. Pojačavanje obilježja - tehnika kod koje je generirano predviđanje ili klasifikacija predmeta jednog preporučitelja pripojena slijedećoj preporučiteljskoj tehnici u nizu.
7. Meta razina - izlaz jednog preporučitelja je ulaz drugog preporučitelja.

Konkretna primjena preporučiteljskih sustava u heterogenim okolinama, danas se svode u velikoj mjeri na različite hibride osnovnih tehnika. Velike industrije trgovine

i zabavnih sadržaja u praksi imaju autorska algoritamska rješenja hibridnih preporučitelja.

### 3.7. Moguća područja primjene

Prikazane tehnike preporuke danas se aktivno koriste u gotovo svim sferama internet-skog života. U velikoj mjeri preporučitelji nisu eksplicitni, nego rade u pozadini, pa se na prvi pogled ne bi moglo reći da je korišten preporučitelj. Na prvom mjestu svakako jest svakodnevna upotreba internetskih tražilica. Internetske tražilice danas, čak i



Slika 3.10: Preporuka internetske tražilice Google

za neregistrirane korisnike vrlo precizno mogu prikupiti implicitne podatke i ponuditi personalizirane preporuke. Pretraživanje internetskih trgovina i *e-commerce* sustava uz današnju ponudu predmeta i potražnju korisnika praktički je nezamislivo bez nekog oblika filtra koji će ponuditi korisniku interesantne predmete i navesti ga na druge bliške predmete. Ovo čini preporučitelje interesnom sferom i za marketing s obzirom da često upravo o njima ovisi prodaja, a time i zarada.

Internetski servisi za razmjenu i pregledavanje multimedijalnih sadržaja (*Youtube*, *Last.fm*, *Netflix*, *Pandora*) naveliko koriste preporučitelje kako bi povećali kvalitetu usluge a time i zadovoljstvo korisnika, pogotovo ako se radi o plaćenim korisničkim računima. Također, koriste ih često i za pogađanje ukusa korisnika. Usluga *Last.fm* nudi opciju „Scrobblera“ koja u pozadini prikuplja informacije i stvara liste slušanja bez eksplicitne pomoći korisnika.

Preporučitelje koriste i *IpTV* usluge (engl. *Internet protocol television*) za lakše snalaženje u digitalnim videotekama i pojednostavljenje usluga multimedije na zahtjev (engl. *multimedia - on - demand*).

Društvene mreže i servisi za upoznavanje koriste preporučiteljske sustave za davanje preporuka na razini korisnik - korisnik. Takve usluge tretiraju objekte drugih korisnika kao predmete i na sličan način traže *top – N* listu za aktualnog korisnika.

Preporučitelje je moguće koristiti i u pametnim prostorima te u *Internet of things* okolini [7]. U tom kontekstu prostorne i vremenske karakteristike, te kontekst u kojem se korisnici i predmeti nalaze potrebno je na poseban način modelirati i uzeti u obzir.

### 3.8. Preporučiteljski sustavi u sveprisutnom računarstvu

Primjena preporučiteljskih sustava u sveprisutnom računarstvu logična je posljedica ne samo činjenice da je sveprisutno računarstvo računarstvo kao takvo pa u njemu postoji interes za preporukom, nego i činjenice da se u njemu generiraju velike količine podataka pa se preporučiteljski algoritmi mogu primjeniti kao filtri podataka. O podacima sveprisutnog računarstva praktički je nemoguće govoriti bez konteksta u kojem se nalaze. Taj kontekst može se definirati kao bilo koja informacija o okolnostima, predmetima ili uvjetima koje okružuju korisnika, a smatra se relevantnom za interakciju između korisnika i sveprisutnog računalnog okruženja [8].

Tri najvažnija aspekta konteksta su [12]:

1. Gdje se korisnik sustava nalazi
2. U društvu kojih je drugih korisnika sustava
3. Koji resursi se nalaze u blizini

Koje su specifičnosti? Nisam ni sam ziher, algoritam vjerojatno mora biti prilagođen filtriranju ogromne količine nestrukturiranih podataka, kontekst vremena i prostora je najbitniji, predmeti preporuke su iz puno više različitih domena nego npr na amazonu, pa ih je teže opisati.

Primjer preporučiteljskog sustava koji se djelomično nalazi u sveprisutnom računarstvu jest algoritam koji koristi *Foursquare*. *Foursquare* je servis koji preporučuje lokale na temelju preferencija korisnika, njegovih ocjena sličnih lokacija te ocjena njegovih najbližih prijatelja i stručnjaka koje prati [3]. Kod ovog preporučitelja vidljiv je prostorni kontekst u smislu preporučivanja

specifičnosti, koji postoje, mobilni uređaji, foursquare itd.

## 4. Izgrađeni model preporučiteljskog sustava

### 4.1. Preporučiteljski sustavi s izraženom prostornom i vremenskom komponentom

Osnovni problem s kojim se konvencionalni preporučiteljski sustavi susreću (a time i popularniji radni okviri i biblioteke koji ih implementiraju) jest u većoj mjeri izostanak podrške za kontekstualizaciju. U kontekstu sveprisutnog računarstva preporučitelj kao minimalan skup konteksta u kojem se sustav nalazi treba imati podršku za prostornu i vremensku komponentu. Model takvog preporučitelja može se prikazati relacijom 4.1

$$R \leftarrow U \times I \times C \quad (4.1)$$

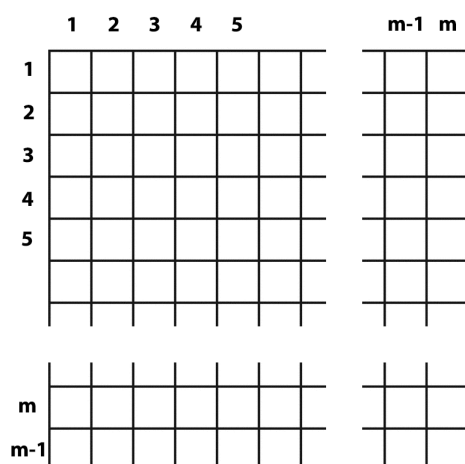
gdje je  $C$  kontekst preporuke, tj. kontekst u kojem se nalazi sustav. Minimalan skup sastavnica konteksta tada je:

$$C \leftarrow T \times S \quad (4.2)$$

gdje je  $T$  vremenska komponenta, a  $S$  prostorna komponenta

### 4.2. Prostorna komponenta

S obzirom na kompleksnost postupka modeliranja prostora, prostorna komponenta konteksta modelirat će se poednostavljeno, i to kvadratnom mrežom (slika 4.1). Iako se na prvi pogled tako čini, to nije ograničenje jer modeliranje prostora u kojem će korisnici tražiti preporuku najčešće nije slobodno, tj. korisnik se u takvim prostorima kreće putem koji mu je dostupan. Primjer je dućan u kojem se korisnik može kretati između polica, a očito je i razumno da se ne može kretati *preko* polica.

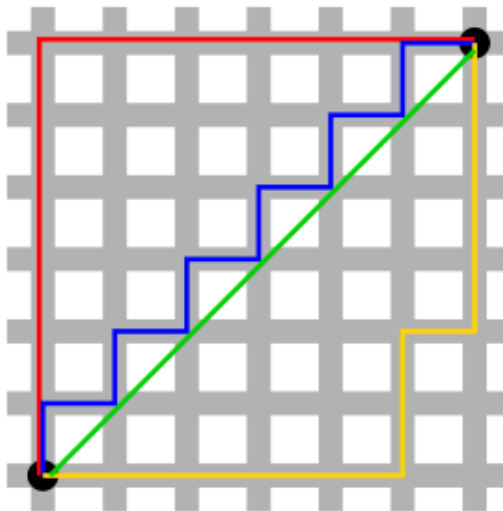


**Slika 4.1:** Tlocrt prostora modeliran kvadratnom mrežom

Dostupne puteve za korisnika može se čuvati u npr. riječnicima (engl. *Dictionary*), tj. parovima  $kljuc \rightarrow vrijednost$  gdje je  $kljuc$  neko polje kvadratne mreže a  $vrijednost$  lista svih *susjednih* polja do kojih je moguće doći iz polja  $kljuc$ . Minorna mana ovog pristupa jest dvostruko spremanje putova jer ako se iz polja  $p_1$  može doći u polje  $p_2$ , onda se iz polja  $p_2$  može doći u polje  $p_1$ . Dvostruko spremanje može se izbjeći nauštrb vremenu pretraživanja dvostrukih putanja, a za relativno malu memorijsku uštedu, čak i kod modeliranja vrlo velikih prostora.

Ako se kvadratnu mrežu i dostupne puteve kretanja prikaže kao neusmjereni težinski graf, onda se taj prostor može predstaviti *matricom susjedstva*, a kako je takva matrica simetrična, spremati se može samo npr. gornja trokutasta matrica, čime se postiže svojevrsna rijetka popunjenost matrice (engl. *Sparse Matrix*). Prednost predstavljanja mreže i prostora grafom jest i postojanje nebrojeno mnogo pouzdanih i efikasnih algoritama za traženje najkraćeg puta između dva vrha, odnosno dva polja. Neki od njih su npr. *Dijkstrin algoritam*, *Bellman-Ford algoriam* ili *A\* algoritam*.

Kvadratna mreža ne mora se nužno prevoditi u graf. U tom slučaju fizička udaljenost između dva polja može se modelirati *Manhattanskom udaljenosti* (slika 4.2). Radi se o vrsti udaljenosti koja od standardne Euklidske udaljenosti odudara po činjenici da je udaljenost dviju točaka suma apsolutnih razlika njihovih Kartezijevih koordinata.



**Slika 4.2:** Ilustracija Manhattanske udaljenosti

### 4.3. Vremenska komponenta

Vremenska komponenta koja se uzima u obzir u modelu preporučitelja također je komponenta konteksta u kojem se nalazi korisnik. S obzirom na ponavljajuće navike korisnika i prirodno uklapanje, ta komponenta modelirat će se funkcijom koja preslikava vrijeme na potrebu korisnika za nekim predmetom:

$$f : \text{vrijeme} \rightarrow \text{predikcija} \quad (4.3)$$

Domena funkcije jest vrijeme, a period u teoriji može biti proizvoljan. Dakle, svaka funkcija prikazivat će želju ili potrebu korisnika za nekim predmetom u periodu konzistentnom za sve predmete u sustavu. Rezultat funkcije jest predikcija u intervalu  $[0, 1]$

Jedna takva funkcija može biti primjerice tzv. *Gaussova krivulja* (slika 4.3). Ona je pogodna zbog činjenice da prosječan korisnik prosječnu akciju u kojoj će tražiti preporuku u principu traži u jednom kraćem vremenskom intervalu. Drugim riječima, pojedini predmeti preporuke za nekog korisnika u vremenski svjesnom preporučiteljskom sustavu bit će zanimljivi samo jedan kraći interval.

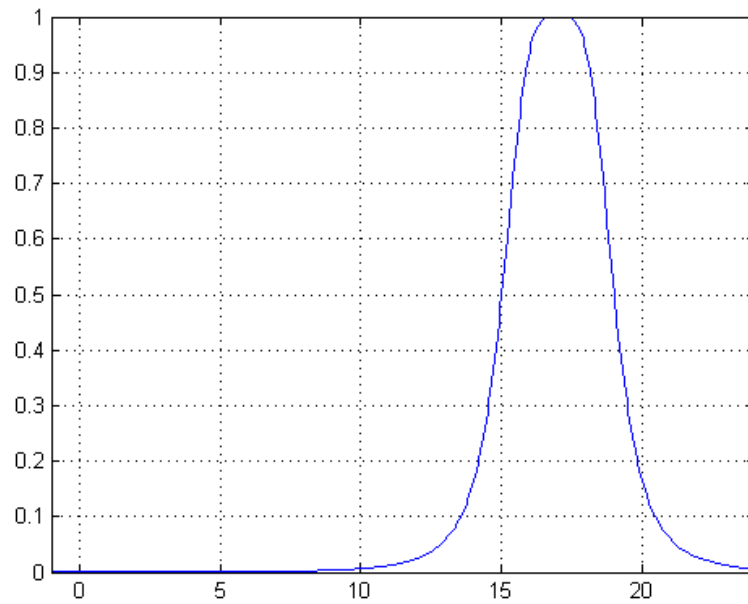
$$f(x; a, b, c) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}} \quad (4.4)$$

gdje je  $x$  varijabla, a  $a$ ,  $b$  i  $c$  parametri kojima se regulira oblik krivulje:

- $a$  - širina vrha
- $b$  - nagib porasta i pada, oblik vrha

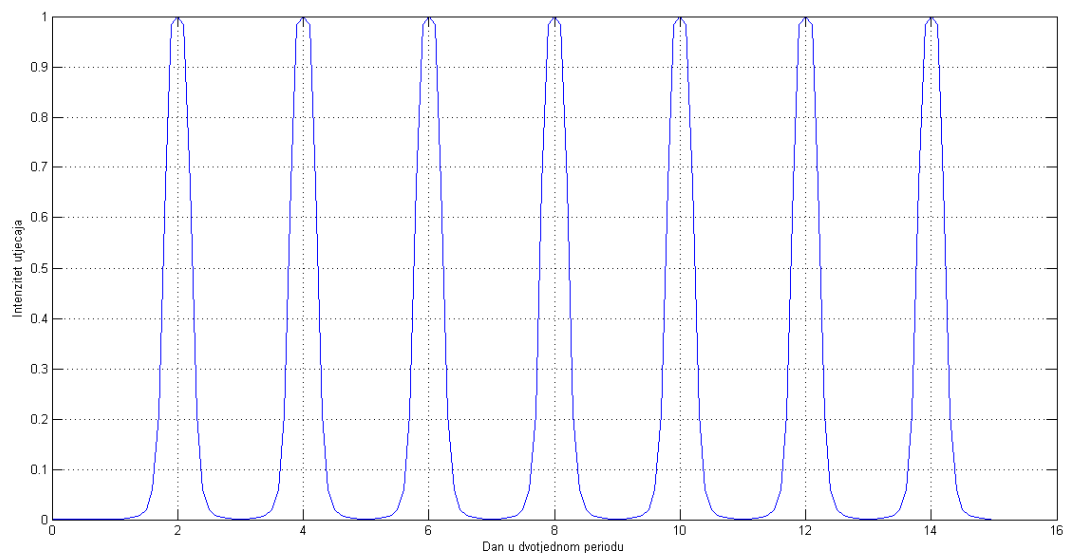


–  $c$  - pomak na  $x$  osi



**Slika 4.3:** Preferencija korisnika za predmet na dnevnoj skali

S druge strane, skala ne mora biti dnevna, može biti i tjedna te se funkciju može zadati po dijelovima. Primjer jedne takve funkcije nalazi se na slici 4.4.



**Slika 4.4:** Preferencija korisnika za predmet na dnevnoj skali

S obzirom na matematičku analizu koja stoji iza računa za pronalazak funkcijete

kompleksnost i težinu izračuna, one se mogu zadati i diskretno, npr. kroz vektor uzoraka funkcije. Ta diskretna funkcija može se koristiti kao takva ili se može procesom interpolacije naći njezin eksplicitni zapis. Pogodne metode interpolacije mogu npr. biti interpolacija polinomom višeg stupnja ili interpolacija splajnovima (engl. *Spline*).

Vremenska komponenta omogućava i praćenje obrazaca ponašanja korisnika, pa time i predviđanje ponavljanja ponašanja (engl. *Recurring*). Također, vremenska komponenta podrazumjeva i tzv. *efekt blijedenja* (engl. *Fade-out effect*). Ako korisnik kod kojeg je uočen neki obrazac akcija prestane provoditi tu akciju, preporučitelj će prestati primati pobudu za održavanje te akcije. Međutim, neće je odmah zanemariti nego će se vremenska komponenta, tj. funkcija koja opisuje tu komponentu, lagano gušiti nekom drugom funkcijom, primjerice recipročnom eksponencijalnom funkcijom:

$$g(x) = e^{-x} \quad (4.5)$$

Tada će ukupan rezultat vremenske komponente konteksta biti:

$$T \leftarrow f(x) * g(x) \quad (4.6)$$

gdje je  $f(x)$  funkcija koja opisuje vremensku komponentu konteksta nekog korisnika za neki predmet, a  $g(x)$  funkcija koja guši tu komponentu. primjer.

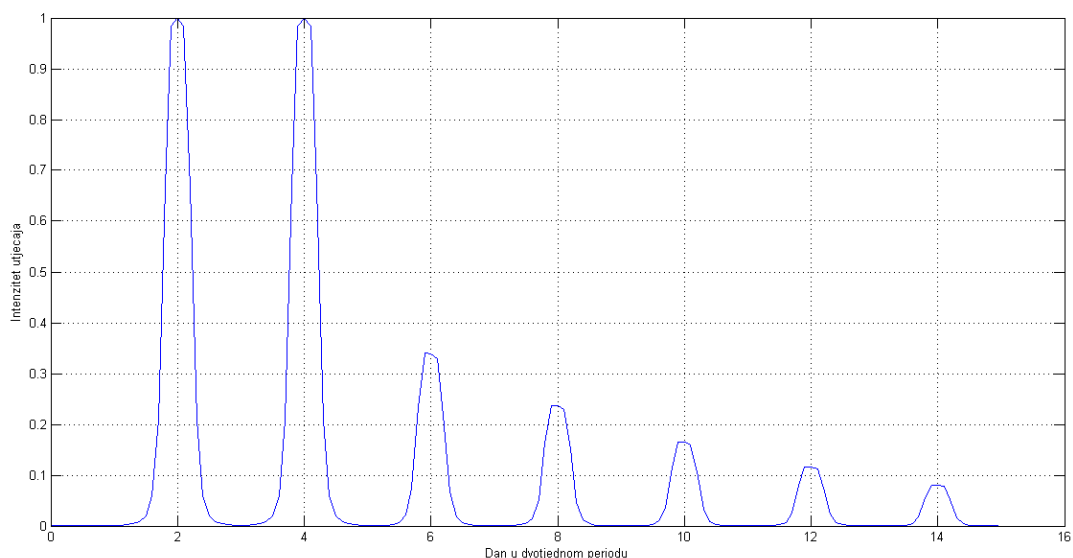
Na slici 4.5 vidljiv je primjer *efekta blijedenja* na tipično periodičko ponašanje korisnika. Prema formuli 4.6 vrijedi:

$$f(x) = \frac{1}{1 + \left| \frac{x-2*k}{0.23} \right|^5} + 2 * k + 1 \quad (4.7)$$

gdje je  $k \in \{0, 1, 2, 3, 4, 5, 6, 7\}$

$$g(x) = e^{-0.18*x} \quad (4.8)$$

Ovdje također postoji mogućnost da se funkcija  $g(x)$  zada diskretno, tj. po uzorcima.



**Slika 4.5:** Preferencija korisnika za predmet na dnevnoj skali

Kod naprednijih modeliranja vremenske komponente, te kod kompleksnijih prilika ponašanja korisnika, vremensku komponentu teoretski je moguće simulirati neuron-skom mrežom. Mana ovog postupka svakako bi bilo potrebno vrijeme za treniranje mreže navikama korisnika u vremenskom kontekstu.

## 4.4. Modeliranje korisnika

Korisnika se u sustavu modelira korisničkim profilom. Takav profil sadrži elementarne podatke o korisniku:

- Jedinstveni identifikator korisnika u sustavu
- Ime i prezime korisnika
- Vektor preferencija korisnika

Vektor preferencija korisnika predstavlja preferencije korisnika za predefinirane karakteristike. Vektor kao takav može se predstaviti fizičkim vektorom ili riječnikom gdje je ključ naziv karakteristike a vrijednost mjera preferencije. Potonji pristup razmjerno je bolji jer minimizira mogućnost krivog korištenja vektora preferencija. Kroz rad sustava profil korisnika se dopunjuje na temelju povratnih informacija korisnika sustava prema preporučitelju. Primjerice, moguće je detektirati interes korisnika za nekim predmetom pa na takvu vrstu pobude reagirati pojačavanjem karakteristika u profilu korisnika koje odgovaraju tom predmetu. Moguće je i pojačavati karakteristike

na više različitih načina u ovisnosti o jačini pobude. Ovakva vrsta korisničkog profila donekle rješava problem hladnog početka (engl. *Cold start*), jer se može pretpostaviti da su u inicijalnom trenutku vektori preferencija dostupni u datotekama. Zbog toga će sigurno biti moguće izvesti sadržajno temeljenu preporuku, dok će do suradničke preporuke moći doći tek nakon ponovnog treniranja algoritma.

## 4.5. Modeliranje predmeta

Model predmeta analogno je sličan modelu korisnika. Svaki objekt tipa predmet sadrži elementarne podatke o predmetu:

- Jedinstveni identifikator predmeta u sustavu
- Naziv predmeta
- Identifikator (oznaka) lokacije predmeta u prostoru
- Vektor karakteristika predmeta

Vektor karakteristika predmeta posjeduje sadržajni opis predmeta. U vektoru se na unaprijed dogovorenim indeksima nalaze brojčane mjere primjenjivosti karakteristike na predmet. U modelu se može koristiti i binarna primjenjivost koja prikazuje da je neka karakteristika primjenjiva, odnosno nije primjenjiva na predmet. Alternativno, vektor karakteristika može se prikazati riječnikom, gdje je ključ u riječniku naziv karakteristike, a vrijednost mjera primjenjivosti karakteristike na predmet. Prilikom unosa predmeta u sustav, vektor karakteristika potrebno je normalizirati dijeljenjem svakog elementa vektora drugim korijenom zbroja svih vrijednosti vektora:

$$w_i = \frac{w_i}{\sqrt{\sum_{j=0}^N w_j}} \quad (4.9)$$

gdje je  $N$  duljina vektora karakteristika.

## 4.6. Izgrađeni model

Izgrađeni model preporučiteljskog sustava kombinira mogućnosti svih navedenih algoritama za filtriranje. Zbog mogućnosti filtriranja po vremenu i prostoru, može se reći da je algoritam svjestan konteksta te je kao takav primjenjiv u sveprisutnoj okolini. Algoritam se sastoji od nekoliko koraka:

1. Faza prostornog filtra

2. Faza vremenskog filtra
3. Opcionalna faza odluke u kojoj, ukoliko nema dovoljno predmeta na listi preporuke, algoritam može početi od prvog koraka ali sa spuštenim kriterijima
4. Faza filtriranja na suradničkoj ili sadržajnoj razini
5. Faza prezentacije rezultata

Algoritam započinje filtriranjem po prostoru. Sve što je trenutno u okolini korisnika manje od nekog graničnog radiusa uzima se u obzir, a ostalo se implicitno odbacuje jer je rezultat preporuke 0. Ovaj filter se prvi izvodi jer u prostornom kontekstu nema smisla filtrirati prema drugim kriterijima, ako se predmetu fizički ne može pristupiti. Također, filter je koristan jer u početku eliminira velik broj nedostupnih predmeta te tako smanjuje opseg izračuna. Algoritam prema kojem se filtrira jest neki od poznatijih algoritama za pronalazak najkraćeg puta ukoliko je prostor na vrhu apstrakcije modeliran grafom, npr. *Dijkstra algoritam*, *Bellman - Ford algoritam* ili *A\* algoritam*, odnosno algoritam za izračun *Manhattanske udaljenosti* ukoliko je na vrhu apstrakcije kvadratna mreža.

Vremensko filtriranje izvodi se skaliranjem vremenskog trenutka u trenutku izvođenja algoritma na interval tjedna. Granulacija do koje se uzima trenutno vrijeme je sekunda, a relacija za konverziju je:

$$t = d + \frac{s}{24} + \frac{m}{1440} \quad (4.10)$$

gdje je  $t$  vrijeme primjenjivo u sustavu,  $d$  redni broj dana u tjednu počevši od ponedjeljka kao 1. dana,  $s$  trenutni sati od ukupno 24 sata, a  $m$  trenutne minute. Nakon skaliranja, evaluira se funkcija kojom je opisana preferencija korisnika za neki predmet. Funkcija je takva da joj je rezultat skaliran na interval  $[0, 1]$ . Efekt blijeđenja (engl. *Fade out*) izvodi se množenjem opisne funkcije funkcijom gušenja koja je u općem slučaju recipročna eksponencijalna funkcija. Moguće je primijeniti proizvoljnu funkciju koja najvjernije opisuje individualnu potrebu svakog korisnika.

Nakon sveprisutno kontekstualiziranog filtriranja, slijedi filtriranje zasnovano na sadržaju ili filtriranje zasnovano na suradnji i relaciji između više korisnika. Krajnji korisnik kroz konfiguraciju može birati koji filter će koristiti, ili može koristiti oba. U trenutku početka rada sustava, teško je ostvariti veze između korisnika pa je preporučljivije koristiti sadržajni filter. Nakon određenog vremena, moguće je prebacivanje na suradnički filter koji se u općem slučaju pokazuje kao stabilnije rješenje [11]. Ove vrste filtriranja koriste podatke o preferencijama korisnika, karakteristikama predmeta i

međusobnim sličnostima između korisnika. Skup ocjena kojima korisnik može ocijeniti predmet je:

$$R \in [0, 5] \quad (4.11)$$

Opća granulacija ocjene jest 0.5 iako sustavski nije ograničeno, tj. sustav će ispravno raditi sa svim granulacijama.

Razvijeni model sustava odgovara na povratne informacije korisnika. Analizom korisničkih radnji i kretanja dolazi do pobude na koju je odziv sustava pojačavanje neke od karakteristika u podatkovnim strukturama. Odzivi mogu biti različiti i ovise o jačini pobude, primjerice, detekcija zastajkivanja korisnika kraj određenog predmeta nosi manju pobudu na karakteristike tog predmeta od detekcije kupnje tog istog proizvoda. Zbroj svih odziva stvara privremeni profil korisnika koji će postati aktivan tek nakon zbrajanja sa stvarnim profilom, tako da trenutne akcije ne mogu utjecati na predikcije u trenutnoj sesiji korištenja sustava.

S obzirom na sve komponente sustava, slijedi algoritam za kontekstualiziranu preporuku u vremenu i prostoru (3):

---

**Algorithm 3** Algoritam kontekstualizirane preporuke u vremenu i prostoru

---

```
1: Ulaz: Korisnik  $u$ , Lista Predmeta  $i$ , Prostor  $s$ , Lokacija korisnika  $l$ , Ocjene  $r$ ,  
   Korelacije  $c$ , Veličina liste  $n$ , Konfiguracija  $k$   
2: Izlaz: Predikcija  $p$ , Slijedeća lokacija  $l_i$   
   {Prostorni filter}  
3:  $ListaLokacija := ProstorniFilter(s, l)$ ;  
4:  $sort(ListaLokacija)$ ;  
5:  $NajvecaUdaljenost := ListaLokacija[ListaLokacija.length - 1]$   
6: for ( $Lokacija_j \in ListaLokacija$ ) do  
7:   for ( $Predmetp_i \in Lokacija$ ) do  
8:      $PostaviTezinu(p_i, 1 - (p_i.udaljenost)/NajvecaUdaljenost)$   
9:   end for  
10: end for  
   {Vremenski filter}  
11:  $VremenskaLista := InitLista()$ ;  
12:  $Vrijeme := SkalirajNaTjedan(TrenutnoVrijeme)$ ;  
13: for ( $Item_j \in i$ ) do  
14:    $f := EvaluirajFunkciju(u.funkcija(Vrijeme))$ ;  
15:    $VremenskaLista.add((i_j, f))$ ;  
16: end for  
   {Podatkovni filtri}  
17:  $SuradnickaLista := InitLista()$ ;  
18:  $SadrzajnaLista := InitLista()$ ;  
19: if  $k.FilterType == COLLABORATIVE$  then  
20:    $SuradnickaLista := SuradnickiFiltar(u, r, c)$ ;  
21: else if  $k.FilterType == CONTENT$  then  
22:    $SadrzajnaLista := SadrzajnaLista(u, i, r)$ ;  
23: else if  $k.FilterType == BOTH$  then  
24:    $SuradnickaLista := SuradnickiFiltar(u, r, c)$ ;  
25:    $SadrzajnaLista := SadrzajnaLista(u, i, r)$ ;  
26: end if
```

---

---

---

27:

*Rezultat := LinearnaKombinacija(ListaLokacija, VremenskaLista,*  
*SuradnickaLista, SadrzajnaLista).kombiniraj(k);*

28: *sort*(*Rezultat*);

29: *SlijedeciCvor := NadjiDostupan*(*Rezultat*);

30: **return** (*Rezultat, SlijedeciCvor*)

---



## 5. Implementacija i ispitivanje modela

### 5.1. Arhitektura implementacije

Programsko rješenje opisanog modela preporučiteljskog sustava izvedeno je u programskom jeziku *Java* verzije 8, a sastoji se od više interoperabilnih modula koje je prema potrebi moguće nadograđivati. Dijagram razreda sustava vidljiv je na slici 5.1.

Glavnina funkcionalnosti sadržana je u paketu *filter*. Taj paket u prvom redu sadrži sučelja *SpaceFilter*, *TimeFilter* te *DataFilter* implementacijom kojih se ostvaruje mogućnost filtriranja po prostoru, vremenu i generički po podacima respektivno. Osim sučelja, paket sadržava implementacije prostornog filtera, vremenskog filtera, suradničkog filtriranja na korisničkoj razini te filtriranja zasnovanog na sadržaju. Prostorno filtriranje implementirano je upotrebom standardnog Dijkstrinog algoritma za pronalaženje najkraćeg puta u grafu. Vremensko filtriranje izvedeno je evaluacijom funkcije vremena u trenutku traženja preporuke. Konačni algoritam implementiran je u razredu *TheAlgorithm*, a kao takav je hibrid filtriranja po prostoru, vremenu i podacima.

U paketu *data* nalaze se razredi koji enkapsuliraju podatke koje koriste algoritmi za filtriranje. Razred *Correlations* enkapsulira tablicu Pearsonovih koeficijenata korelacije između svaka dva korisnika u sustavu, dok razred *Ratings* enkapsulira tablicu ocjena svih korisnika za sve predmete. Potonji razred posjeduje metode potrebne za sustav pobuda i pojačavanja obilježja (engl. *Reinforcement*). Anotacija (engl. *Annotation*) *Link* označava razrede *ItemBeaconLink*, *ItemPredictionLink* i *UserCorrelationLink*. Navedeni razredi su jednostavni nositelji podataka između preporučiteljskih algoritama i klijentskog koda. Spremnici podataka također su i razredi *Matrix* i *Vector* u paketu *commons* koji modeliraju proizvoljnu matricu, odnosno vektor.

Paket *functions* sadrži sučelja za funkcije koje se koriste u vremenskom filtru. Sučelje *Function* predstavlja osnovu za implementaciju funkcije. Naslijeđuju ga sučelja *DiscreteFunction* i sučelje *Continuous function* koja je potrebno implementirati da bi se ostvarila diskretna, odnosno kontinuirana funkcija pogodna za korištenje u vremenskom filtru. Također, paket sadrži konkretnu implementaciju Gaussove zvonolike

funkcije konfigurabilne pomoću parametara  $a$ ,  $b$  i  $c$  (4.7).

Model grafa koji opisuje prostor nalazi se u paketu *space*. Graf je modeliran kroz vrhove (razred *Node*) koji sadrže mape oblika  $vrh \rightarrow udaljenost$  i ostale podatke nužne za ostvarivanje grafa. Razred *Graph* sadrži listu svih vrhova u grafu te metodu za učitavanje grafa iz CSV datoteke u kojoj je pohranjena matrica susjedstava. Sučelje *SPAlgorithm* modelira algoritam za traženje najkraćeg puta, a konkretna implementacija jednog takvog algoritma (Dijkstrinog) nalazi se u razredu *Dijkstra*.

Korisnici i predmeti modelirani su kroz razrede *User* odnosno *Item* paketa *model*, a sadržavaju podatke i metapodatke korisnicima i predmetima koji su nužni za ispravan rad sustava. Za korisnika je to npr. vremenska mapa u kojoj su pohranjeni parovi ključeva i vrijednosti u obliku  $itemId \rightarrow function$ , gdje je *itemId* jedinstveni identifikator predmeta, a *function* funkcija kojom je opisana vremenska preferencija korisnika za predmetom s identifikatorom *itemId*. Za predmet su to jedinstveni identifikator, identifikator oznake te mapa karakteristika u obliku  $karakteristika \rightarrow preferencija$ , gdje je *karakteristika* naziv, tj. identifikator karakteristike, a *preferencija* brojčana mjera preferencije. Razred također zadržava statični skup imena svih karakteristika u sustavu. Paket *model* također sadržava pomoćne razrede *ItemBuilder* i *UserBuilder* koji korištenjem oblikovnog obrasca *Graditelj* stvaraju nove objekte tipa *Item* odnosno *User*.

Analiza dnevnčkih zapisa proizašlih s mobilnog uređaja vrši se kroz razred *RecordAnalyzer* i metodu *analyze*. Pomoćne strukture za analizu nalaze se u paketu *io*, a služe za modeliranje zapisa s mobilnog uređaja, konkretno, trodimenzionalne točke, vremenske oznake te samog zapisa. Razred *FileInput* s metodom *parseFile* vrši ulaznu obradu dnevnčkih zapisa (tzv. *parsiranje*).



## 5.2. Ispitni scenarij

Osmišljavanje ispitnog scenarija pokazalo se natprosječno zahtjevnim radi pokušaja da takav scenarij obuhvati i ispita sve segmente kontekstualno svjesnog preporučitelja. S druge strane, takav scenarij je morao biti realno ostvariv, u smislu da ne bude sintetički osmišljen nego da u realnom svijetu postoji takva situacija u kojoj će se osmišljeni preporučiteljski model moći implementirati i koristiti. Na kraju je odabrana prosječna svakodnevna aktivnost većine potencijalnih korisnika ovakvog sustava, a to je odlazak u kupovinu.

Ispitni scenarij podrazumjeva generički srednje velik dućan u kojem se može naći mješovita roba, dakle, i prehrana i neprehrana. Radi jednostavnosti, a bez značajnijeg utjecaja na krajnji ishod, tlocrt dućana modeliran je mrežastom strukturom prikazanom na slici 4.1. U dućanu se na svakom mjestu mogu nalaziti radio oznake (engl. *beacon*) kojima mogu biti obilježeni:

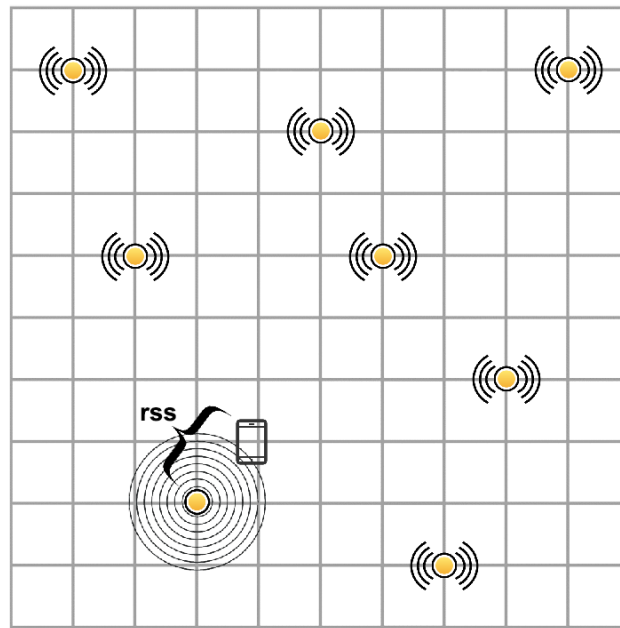
- Neka lokacija u dućanu, npr. red polica
- Skupina proizvoda
- Konkretan proizvod

Ovdje se nameće pitanje je li potrebno radio oznakama obilježiti sve potencijalne predmete preporuke. Sa strane sustava, podrazumjeva se posjedovanje točnih lokacija svake radio oznake, a poznavanjem točnih lokacija te poznavanjem rasporeda u dućanu, interpolacijom se može aproksimirati lokacija korisnika u trenucima kada je on između radio oznaka. Svako radio oznaci pridodani su i neki metapodaci od kojih je najvažniji jedinstveni identifikator u sustavu.

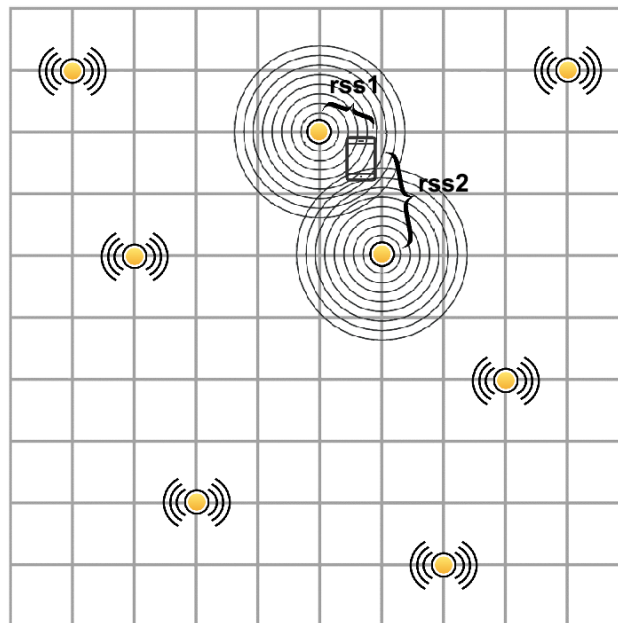
Svaka radio oznaka ima određeni domet signala kojim razaslije (engl. *broadcast*) svoj jedinstveni identifikator. Kada se mobilni uređaj nađe u dometu signala, on prima taj jedinstveni identifikator te može utvrditi *RSS* (engl. *Received signal strength*).

Slika 5.2 prikazuje slučaj kada je mobilni uređaj u dometu jedne radio oznake. U tom slučaju može se smatrati da je korisnik zainteresiran za ono što radio oznaka obilježava. S druge strane, ako se korisnik nađe u dometu dva ili više radio oznaka (slika 5.3), onda se jednostavno uzima signal one radio oznake čiji je *RSS* u tom trenutku jači. Naime, *RSS* opada s udaljenošću pa se s pravom može smatrati da jači *RSS* nužno povlači veću fizičku blizinu korisnika u odnosu na pripadajuću radio oznaku, tj. ono što ta radio oznaka obilježava.

Također, uzima se u obzir i vrijeme zadržavanja korisnika u blizini radio oznake. Ukoliko je korisnik proveo manje vremena od neke empirijski određene vremenske



**Slika 5.2:** Grafički prikaz radio oznake i mobilnog telefona



**Slika 5.3:** Grafički prikaz višestrukog prijama signala s radio oznaka

granice  $t_{min}$ , onda se može smatrati da korisnik nije iskazao interes za predmet koji radio oznaka obilježava. U obzir valja uzeti i orijentaciju korisnika. Nerijetko je moguće da korisnik bude u dometu signala radio oznake, a fizički nije usmjeren prema predmetu koji ona označava. Korištenjem magnetometra na pametnom uređaju i poz-

navanjem razmještaja radio oznaka u prostoru, može se jednostavno utvrditi je li korisnik iskazao interes za obilježeni predmet ili je okrenut od njega.

Izvor podataka je dobro enkapsuliran pa se može reći da mora implementirati sučelje s određenom opremom, a ta oprema je:

- Komunikacijski radio, npr. *WLAN* (engl. *Wireless LAN*) – koristi se za svu komunikaciju sa poslužiteljima po pitanju dohvaćanja metapodataka o radio oznakama, traženje preporuke itd.
- Magnetometar – koristi se utvrđivanje je li korisnik okrenut prema radio oznaci ili od nje.
- Akcelerometar – koristi se za utvrđivanje gibanja korisnika. Ako je korisnik u pokretu, zanemaruju se podaci sa ulaza za očitavanje radio oznaka
- Bluetooth – koristi se za komunikaciju s radio oznakama

Takav izvor u praksi može biti bilo koji moderniji mobilni telefon ili tablet.

### 5.3. Simulacija primjene

U praktičnom smislu, ispitni scenarij modeliran je radio oznakom i prolaskom korisnika s mobilnim telefonom. Korisnik hoda normalnim hodom prosječne brzine, a mobilni telefon drži u ruci koja slobodno visi i giba se u skladu s koracima. Radio oznaka razlaže jedinstveni identifikator frekvencijom od  $1\text{ Hz}$ , a mobilni telefon bilježi podatke frekvencijom od  $5\text{ Hz}$ . Različite razine povratne informacije preporučitelju o akcijama korisnika simulirane su različitim ponašanjem u ispitnom scenariju, a vidljive su u tablici 5.1

**Tablica 5.1:** Ispitni podscenariji

Dataset	Način zaustavljanja	Korištenje mobitela
1	Ne postoji	Ne
2	Usporavanje	Ne
3	Zastajkivanje	Ne
4	Zastajkivanje	Kraće podizanje
5	Zastajkivanje	Dulje podizanje
6	Zaustavljanje	Ne
7	Zaustavljanje	Kraće podizanje
8	Zaustavljanje	Dulje podizanje

Tablica 5.2 prikazuje konkretne iznose odgovora na različite pobude od strane korisnika na sustav.

**Tablica 5.2:** Reakcije na interakciju korisnika i predmeta

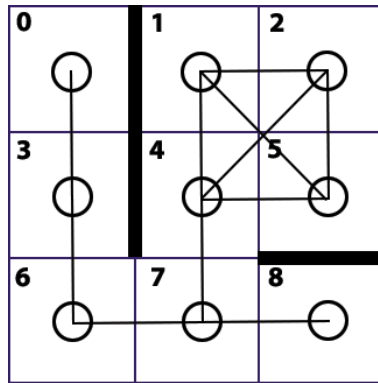
Način zaustavljanja	Pojačanje
Ne postoji	0
Usporavanje	0.025
Zastajkivanje	0.05
Zaustavljanje	0.1

### 5.3.1. Ulazi

Ulazi u sustav ugrubo se mogu podijeliti na ulaze potrebne za postavljanje sustava i ulaze na koje sustav reagira preporukom. Ulazi potrebni za postavljanje su tlocrt, tj. prostorni opis okoline, podaci o ocjenama korisnika sustava za predmete, predmeti sustava sa vektorima svojih karakteristika te podaci o vezama između radio oznaka i predmeta. Uz navedene podatke, ulaz u sustav je i jedinstveni identifikator korisnika za kojeg se traži preporuka. Taj identifikator jest jednostavan podatak cjelobrojnog tipa pa neće biti detaljnije opisivani. Također, ulaz u sustav je i cijeli broj  $N$  kojim se specificira koliko se najboljih predmeta traži za korisnika.

#### Prostorni opis okoline

Za potrebe testiranja koristi se jednostavni generirani tlocrt prikazan slikom 5.4. Razlog relativno jednostavnom tlocrtu jest kombinatorna eksplozija kojoj su veći prostori podložni. Tlocrt se učitava iz CSV datoteke (engl. *Comma separated values*), a u toj datoteci zapisana je matrica susjedstva grafa (5.1) koji odgovara tlocrtu. Zbog jednostavnosti, praktično je tlocrt prikazati grafom zbog velikog broja dostupnih i provjerenih algoritama za izračun udaljenosti između čvorova



**Slika 5.4:** Jednostavan ispitni tlocrt

$$M = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (5.1)$$

U datoteci u kojoj je tlocrt zapisan, elementi stupaca matrice odvojeni su znakom točka - zarez (engl. *Semicolon*), a elementi redaka znakom novog retka. Datoteka se procesira na početku rada sustava, a po potrebi svaki put kada se dogodi promjena rasporeda u prostoru. U tom slučaju, pretpostavlja se da će u datoteci biti zapisan novi tlocrt prostora.

### **Veza između radio oznaka i predmeta**

Veza između radio oznaka i predmeta razmjerno je jednostavna. Također se učitava iz CSV datoteke, a format joj je:

$$beaconId; itemId \quad (5.2)$$

S obzirom da izmjena radio oznake za neki predmet nije česta radnja, datoteka se procesira jedanput na početku rada sustava. U slučaju promjena, nove podatke moguće je unijeti dinamički u sustav.



## Podaci o predmetima

Podaci o predmetima učitavaju se iz CSV datoteke. Opći izgled retka datoteke je:

$$itemId; itemName; locationId\#CharName1; CharName2 \dots \quad (5.3)$$

Prvo se čita prvi redak datoteke iz kojeg se nakon razdvajanja učitano retka po znaku # prebrojavaju i učitavaju imena karakteristika kojima će se opisivati predmeti. Karakteristika u sustavu može biti neograničeno mnogo. Nakon toga slijedi slijedno čitanje datoteke, redak po redak. Svaki redak započinje cijelim brojem koji je jedinstveni identifikator predmeta u sustavu. Nakon njega, odvojen točka-zarezom slijedi naziv predmeta te cijeli broj koji je jedinstveni identifikator lokacije, tj. ćelije, u sustavu. Nakon znaka # slijedi niz sastavljen od 0 ili 1 odvojen znakom točka-zarez. Znak 0 označava da ta karakteristika nije prisutna u predmetu kojem pripada redak, a znak 1 da je karakteristika prisutna, tj. primjenjiva na predmet.

Nakon inicijalnog čitanja datoteke, nove predmete u sustav moguće je unijeti dinamički. Primjer ove datoteke nalazi se u dodatku.

## Matrica ocjena korisnika

Matrica ocjena korisnika također razlikuje prvi redak od ostalih. Oblik prvog retka datoteke je:

$$; itemId\#itemName; itemId\#itemName \quad (5.4)$$

gdje predmeta može biti nedefinirano mnogo. Općeniti izgled ostalih redaka je:

$$userId; ratingForItem1; ratingForItem2 \dots \quad (5.5)$$

gdje je  $userId$  jedinstveni identifikator korisnika u sustavu, a  $ratingForItem$  ocjena korisnika za predmet kojem odgovara stupac u kojem se ta ocjena nalazi. Ocjene su predstavljene skupom:

$$R \in [0.5, 5.0] \quad (5.6)$$

Općenito, granulacija dodjeljivanja ocjena jest 0.5 iako to nije kriterij za stabilan rad sustava. Ukoliko ocjena korisnika za neki predmet ne postoji, pri obradi retka, na to mjesto umeće se numerička oznaka –100.0 koja se naravno ne koristi kao ocjena nego kao oznaka neodstatka ocjene. Primjer ove datoteke može se naći u dodatku

### Dnevnički zapisi mobilnog uređaja

Krajnji ulaz u sustav, bez ikakve apstrakcije, su dnevnički zapisi (engl. *Log*) s pametnog uređaja. U njih uređaj zapisuje:

- Vremensku oznaku
- Jedinstveni identifikator radio oznake
- *RSS* (engl. *Received signal strength*)
- Podatke s mjerača ubrzanja (engl. *Accelerometer*)
- Podatke s magnetometra (engl. *Magnetometer*)

Dnevnički zapisi se pišu i čitaju u, odnosno iz obične tekstualne datoteke u predefini-  
ranom formatu:

$$timestamp; beaconId\$rss\#accelerometer\#magnetometer\backslash n \quad (5.7)$$

gdje je  $\backslash n$  oznaka za novu liniju (engl. *Line feed*) Podaci *beaconId* i *rss* su, tipski gledano, jednostavni, a *timestamp*, *accelerometer* i *magnetometer* složeni.

- Vremenska oznaka – *sat : minuta : sekunda.stotinka*
- Mjerač ubrzanja – *x; y; z* - mjeri ubrzanje sile na tri fizičke osi uređaja u  $m/s^2$ , uključujući ubrzanje sile teže.
- Magnetometar – *x; y; z* - mjeri ambijentalno geomagnetsko polje na tri fizičke osi uređaja u  $\mu T$

### 5.3.2. Izlazi

Izlazi iz sustava su serijalizirani *Java* interni objekti *ItemPredictionLink* slijedećeg oblika:

$$ItemPredictionLink\{itemId = N, prediction = P\} \quad (5.8)$$

gdje je *itemId* jedinstveni identifikator predmeta u sustavu, a *prediction* predikcija ocjene korisnika za kojeg je preporuka zatražena za predmet s identifikatorom *itemId*. Objekti su smješteni u jednostruko povezanoj listi, a prethodno su sortirani padajućim redoslijedom prema predikciji. Rezultati rada sustava kao objekti se ispisuju na standardni izlaz računala (engl. *STDOUT*).

## 5.4. Rezultati ispitivanja

Ispitivanje rada sustava izvedeno je korištenjem *JUnit4* knjižnice za programski jezik *Java*. Ispitana je svaka komponenta preporučiteljskog algoritma, a rezultati su prikazani u nastavku. Za testiranje se koristio korisnik s jedinstvenim identifikatorom 89. Ispitni skup obuhvaća bazu od 25 korisnika i 100 predmeta. Tablica ocjena je rijetko popunjena kako bi se simulirala realna situacija u kojoj korisnici ne mogu ocijeniti svaki predmet u sustavu. Funkcija za filtriranje po vremenu jest neprigušena Gaussova funkcija (4.7) s parametrima  $a = 0.23$ ,  $b = 2.5$  i  $c = 2$ , a testni tlocrt je prikazan na slici 5.4. Korištena je i jedna radio oznaka kojom je u simulaciji bio označen predmet s jedinstvenim identifikatorom 11.

Zbog jednostavnosti i velike količine potrebnih ispitnih podataka, samo je dio tablica ocjena i liste predmeta s karakteristikama prikazan u dodatku.

### 5.4.1. Ispitivanje prostornog filtera

```
1 ItemPredictionLink { itemId=24, prediction=1.0}
2 ItemPredictionLink { itemId=77, prediction=1.0}
3 ItemPredictionLink { itemId=98, prediction=1.0}
4 ItemPredictionLink { itemId=194, prediction=1.0}
5 ItemPredictionLink { itemId=243, prediction=1.0}
6 ItemPredictionLink { itemId=414, prediction=1.0}
7 ItemPredictionLink { itemId=604, prediction=1.0}
8 ItemPredictionLink { itemId=745, prediction=1.0}
9 ItemPredictionLink { itemId=1572, prediction=1.0}
10 ItemPredictionLink { itemId=1597, prediction=1.0}
11 ItemPredictionLink { itemId=1637, prediction=1.0}
12 ItemPredictionLink { itemId=2502, prediction=1.0}
13 ItemPredictionLink { itemId=7443, prediction=1.0}
14 ItemPredictionLink { itemId=12, prediction=0.8}
15 ItemPredictionLink { itemId=153, prediction=0.8}
16 ItemPredictionLink { itemId=187, prediction=0.8}
17 ItemPredictionLink { itemId=272, prediction=0.8}
18 ItemPredictionLink { itemId=274, prediction=0.8}
19 ItemPredictionLink { itemId=275, prediction=0.8}
20 ItemPredictionLink { itemId=280, prediction=0.8}
```

21 ItemPredictionLink { itemId=558, prediction=0.8}  
22 ItemPredictionLink { itemId=568, prediction=0.8}  
23 ItemPredictionLink { itemId=854, prediction=0.8}  
24 ItemPredictionLink { itemId=180, prediction=0.6}  
25 ItemPredictionLink { itemId=329, prediction=0.6}

#### **5.4.2. Ipsitivanje vremenskog filtera**

1 ItemPredictionLink { itemId=36658, prediction=0.787298935}  
2 ItemPredictionLink { itemId=1637, prediction=0.784401130}  
3 ItemPredictionLink { itemId=105, prediction=0.780483141}  
4 ItemPredictionLink { itemId=9802, prediction=0.775195808}  
5 ItemPredictionLink { itemId=275, prediction=0.773624913}  
6 ItemPredictionLink { itemId=77, prediction=0.772531135}  
7 ItemPredictionLink { itemId=4327, prediction=0.772157584}  
8 ItemPredictionLink { itemId=1597, prediction=0.765785763}  
9 ItemPredictionLink { itemId=114, prediction=0.763866127}  
10 ItemPredictionLink { itemId=36955, prediction=0.76240516}  
11 ItemPredictionLink { itemId=807, prediction=0.76148483}  
12 ItemPredictionLink { itemId=424, prediction=0.75867900}  
13 ItemPredictionLink { itemId=272, prediction=0.75497504}  
14 ItemPredictionLink { itemId=788, prediction=0.75153934}  
15 ItemPredictionLink { itemId=2501, prediction=0.74214122}

#### **5.4.3. Ispitivanje sadržajnog filtera**

1 ItemPredictionLink { itemId=141, prediction=2.396023342}  
2 ItemPredictionLink { itemId=393, prediction=2.396023342}  
3 ItemPredictionLink { itemId=1892, prediction=2.396023342}  
4 ItemPredictionLink { itemId=3049, prediction=2.396023342}  
5 ItemPredictionLink { itemId=11, prediction=2.337201970}  
6 ItemPredictionLink { itemId=745, prediction=2.337201970}  
7 ItemPredictionLink { itemId=63, prediction=2.064130067}  
8 ItemPredictionLink { itemId=161, prediction=2.064130067}  
9 ItemPredictionLink { itemId=601, prediction=2.064130067}  
10 ItemPredictionLink { itemId=812, prediction=2.064130067}  
11 ItemPredictionLink { itemId=8358, prediction=2.064130067}

12 ItemPredictionLink { itemId=107, prediction=1.991899926}  
13 ItemPredictionLink { itemId=238, prediction=1.991899926}  
14 ItemPredictionLink { itemId=955, prediction=1.991899926}  
15 ItemPredictionLink { itemId=9802, prediction=1.991899926}

#### **5.4.4. Ispitivanje suradničkog filtera**

1 ItemPredictionLink { itemId=238, prediction=5.322015033}  
2 ItemPredictionLink { itemId=278, prediction=5.261423692}  
3 ItemPredictionLink { itemId=275, prediction=5.241110580}  
4 ItemPredictionLink { itemId=807, prediction=5.201983914}  
5 ItemPredictionLink { itemId=424, prediction=5.199223002}  
6 ItemPredictionLink { itemId=122, prediction=5.186845906}  
7 ItemPredictionLink { itemId=105, prediction=5.073047679}  
8 ItemPredictionLink { itemId=550, prediction=5.069938928}  
9 ItemPredictionLink { itemId=13, prediction=5.049074129}  
10 ItemPredictionLink { itemId=680, prediction=5.046814493}  
11 ItemPredictionLink { itemId=120, prediction=5.018683300}  
12 ItemPredictionLink { itemId=121, prediction=5.018683300}  
13 ItemPredictionLink { itemId=77, prediction=4.977639850}  
14 ItemPredictionLink { itemId=329, prediction=4.947990506}  
15 ItemPredictionLink { itemId=568, prediction=4.935984896}

#### **5.4.5. Ispitivanje sustava za pojačavanje obilježja**

1 ItemPredictionLink { itemId=77, prediction=3.845381767}  
2 ItemPredictionLink { itemId=194, prediction=3.304305022}  
3 ItemPredictionLink { itemId=275, prediction=3.243722975}  
4 ItemPredictionLink { itemId=2502, prediction=3.189499501}  
5 ItemPredictionLink { itemId=24, prediction=3.137182428}  
6 ItemPredictionLink { itemId=243, prediction=3.001130814}  
7 ItemPredictionLink { itemId=12, prediction=2.861055920}  
8 ItemPredictionLink { itemId=1597, prediction=2.805229099}  
9 ItemPredictionLink { itemId=272, prediction=2.774823738}  
10 ItemPredictionLink { itemId=1637, prediction=2.750916214}  
11 ItemPredictionLink { itemId=568, prediction=2.634207367}  
12 ItemPredictionLink { itemId=604, prediction=2.582231016}

```
13 ItemPredictionLink { itemId=98, prediction=2.580845177}  
14 ItemPredictionLink { itemId=745, prediction=2.519079541}  
15 ItemPredictionLink { itemId=1572, prediction=2.400344565}
```

## **6. Zaključak**

Zaključak.

# LITERATURA

- [1] ASK. *etown.com Enlists Ask Jeeves to Deliver First Consumer Shopping Advisor with Intelligent Decision Technology*, October 1999. URL <http://http://ask.mediaroom.com/index.php?s=32522&item=107749>.
- [2] James Bennett i Stan Lanning. The netflix prize. ... *of KDD cup and ...*, stranice 3–6, 2007. URL <http://su-2010-projekt.googlecode.com/svn-history/r157/trunk/literatura/bennett2007netflix.pdf>.
- [3] Foursquare. *About Us*, Lipanj 2015. URL <https://foursquare.com/about/>.
- [4] Gartner. *Forecast: The Internet of Things, Worldwide, 2013.*, December 2013. URL <http://www.gartner.com/newsroom/id/2636073>.
- [5] Michael Ekstrand Joe Konstan. Introduction to recommender systems. *Coursera*, 2015. doi: 10.1145/329124.329126. URL <https://www.coursera.org/learn/recommender-systems>.
- [6] B. ; York J. Linden, G. ; Smith. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 2003.
- [7] G.a. Munoz-Organero, M. ; Ramiez-Gonzalez, P.J. Munoz-Merino, i C. Delgado Kloos. A Collaborative Recommender System Based on Space-Time Similarities. *IEEE Pervasive Computing*, 9(3):81–87, 2010.
- [8] Anand Ranganathan i Roy H. Campbell. An infrastructure for context-awareness based on first order logic. *Personal Ubiquitous Comput.*, 7(6):353–364, Prosinac 2003. ISSN 1617-4909. doi: 10.1007/s00779-003-0251-x. URL <http://dx.doi.org/10.1007/s00779-003-0251-x>.



- [9] 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/>.
- [10] GroupLens Research. *About MovieLens*, July 2015. URL <https://movielens.org/info/pages/about/>.
- [11] Francesco Ricci, Lior Rokach, Bracha Shapira, i Paul B. Kantor. *Recommender Systems Handbook*. Springer-Verlag New York, Inc., New York, NY, USA, 1st izdanju, 2010. ISBN 0387858199, 9780387858197.
- [12] Bill Schilit, Norman Adams, i Roy Want. Context-aware computing applications. U *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*, stranice 85–90. IEEE, 1994.
- [13] N. Vukotić, D. ; Tanković. Alati za razvoj aplikacija bez kodiranja. *Razvoj poslovnih i informatičkih sustava CASE*, 2011.
- [14] Mark Weiser. The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3(3):3–11, Srpanj 1999. ISSN 1559-1662. doi: 10.1145/329124.329126. URL <http://doi.acm.org/10.1145/329124.329126>.
- [15] Mark Weiser. The computer for the 21st century. *ACM SIGMOBILE Mobile Computing and Communications Review*, 1999.

## **Preporučiteljski sustavi u sveprisutnom računarstvu**

### **Sažetak**

U ovom radu dan je uvod u sveprisutno računarstvo. Analizirani su temeljni algoritmi koje koriste preporučiteljski sustavi, te su analizirani zahtjevi na preporučiteljske sustave korištene u sveprisutnim okruženjima. Osmišljen je model kontekstno svjesnog preporučiteljskog sustava s posebnim naglaskom na prostor i vrijeme. Prikazani su mogući načini modeliranja prostora, vremena, korisnika i predmeta u sustavu, te je prikazan način sinteze filtarskih algoritama u jedinstvenu predikciju. Na kraju, osmišljeni model je implementiran i ispitan na osmišljenom ispitnom scenariju u realnim uvjetima.

**Ključne riječi:** sveprisutno računarstvo, preporučiteljski sustavi, kontekst, svjesnost konteksta, prostor-vrijeme, filtriranje podataka

## **Recommender systems in ubiquitous computing**

### **Abstract**

This thesis shows fundamentals of Ubiquitous computing. Basic algorithms used by Recommender systems have been analysed, as well as requirements for Recommender systems used in ubiquitous environments. Context aware recommender system with special interest in space and time has been designed. Possible ways of modelling space, time, user and item have been analyzed and shown. Also, a way of synthesys of the filtering algorithms into one unique prediction has been shown. Finally, designed model has been implemented and tested on made up scenario in real conditions.

**Keywords:** ubiquitous computing, recommender systems, context, context awareness, spacetime, data filtering

## Dodatak A

### Primjer dnevničke datoteke

- 1 18:27:10.75#74278bda-b644-4520-8f0c-720eaf059935-baba-bebe\$-85#8.5;-6;0#-23.9;35.8;20.2
- 2 18:27:10.127#74278bda-b644-4520-8f0c-720eaf059935-baba-bebe\$-85#7.9;-5.6;0.4#-23.6;35.7;19.7
- 3 18:27:10.275#74278bda-b644-4520-8f0c-720eaf059935-baba-bebe\$-85#6.8;-5.3;-0.1#-23.6;35.1;17.8
- 4 18:27:10.327#74278bda-b644-4520-8f0c-720eaf059935-baba-bebe\$-85#7.3;-5.5;-0.3#-23.9;34.9;17
- 5 18:27:10.475#74278bda-b644-4520-8f0c-720eaf059935-baba-bebe\$-85#9;-6.9;-0.5#-25;34.6;15.1
- 6 18:27:10.527#74278bda-b644-4520-8f0c-720eaf059935-baba-bebe\$-85#8.9;-6.2;-0.9#-25.2;34.6;14.6
- 7 18:27:10.675#74278bda-b644-4520-8f0c-720eaf059935-baba-bebe\$-85#7;-5.8;-0.4#-26.3;34;13.7
- 8 18:27:10.728#74278bda-b644-4520-8f0c-720eaf059935-baba-bebe\$-85#6.5;-5.2;-0.4#-27.2;33.2;14
- 9 18:27:10.875#74278bda-b644-4520-8f0c-720eaf059935-baba-bebe\$-85#7.6;-4.7;-1.8#-30.3;30.6;15.8
- 10 18:27:10.929#74278bda-b644-4520-8f0c-720eaf059935-baba-bebe\$-85#8.1;-4.8;-1.6#-31;30;16.4

## Dodatak B

### Datoteka s podacima o predmetima

```
1 id;Name;location#C1;C2;C3;C4;C5;C6;C7;C8;C9;C10
2 11;Item1;8#1;0;1;0;1;1;0;0;0;1
3 12;Item2;3#0;1;1;1;0;0;0;1;0;0
4 13;Item3;5#0;0;0;1;1;1;0;0;0;0
5 14;Item4;1#0;0;1;1;0;0;1;1;0;0
6 22;Item5;2#0;1;0;0;0;0;0;0;1;1
7 24;Item6;0#1;0;0;1;0;0;0;0;0;0
8 38;Item7;7#0;0;0;0;0;0;0;1;0;1
9 63;Item8;2#0;0;1;1;0;0;1;0;0;1
10 77;Item9;0#0;0;0;0;0;1;0;0;1;0
11 85;Item10;1#0;1;0;0;1;0;1;0;0;0
12 98;Item11;0#0;0;1;0;1;0;0;0;1;0
13 105;Item12;7#1;0;0;0;0;1;1;0;0;0
14 107;Item13;1#0;0;1;1;1;0;0;1;0;0
15 114;Item14;4#0;1;1;1;0;0;0;0;1;0
```

## Dodatak C

### Datoteka s matricom ocjena korisnika

```
1 ;11#Item1;12#Item2;13#Item3;14#Item4;22#Item5;24#Item6
2 1648;;;4;3
3 5136;4.5;5;5;4;5;5
4 918;5;5;4.5;;3;
5 2824;4.5;;5;;4.5;4
6 3867;4;4;4.5;;4;3
7 860;4;4;4.5;;2.5;3
8 3712;;4.5;;4.5;;
9 2968;5;4.5;5;2;5;0.5
10 3525;4;4;4.5;3.5;3;3.5
11 4323;5;5;5;5;4;5
12 3617;;;5;3.5;4;4
13 4360;4;4;4.5;5;4.5;4
14 2756;;5;4.5;3.5;4;4
15 89;4;4.5;5;;;5
```