

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>2</b>
2.1. Uvod u sveprisutno računarstvo . . . . .	2
2.2. Razvoj . . . . .	2
2.3. Zahtjevi . . . . .	2
2.4. Primjena . . . . .	3
<b>3. Preporučiteljski sustavi</b>	<b>5</b>
3.1. Uvod u preporučiteljske sustave . . . . .	5
3.2. Filtriranje neovisno o korisniku . . . . .	7
3.3. Filtriranje ovisno o korisniku . . . . .	9
3.3.1. Filtriranje zasnovano na sadržaju . . . . .	10
3.3.2. Filtriranje zasnovano na suradnji . . . . .	11
3.4. Hibridni tehnike . . . . .	13
3.5. Moguća područja primjene . . . . .	14
3.6. Preporučiteljski sustavi u sveprisutnom računarstvu . . . . .	14
<b>4. Izgrađeni model preporučiteljskog sustava</b>	<b>16</b>
4.1. Preporučiteljski sustavi s izraženom prostornom i vremenskom komponentom . . . . .	16
4.2. Naš model . . . . .	16
4.3. Prostorna komponenta . . . . .	17
4.4. Vremenska komponenta . . . . .	18
4.5. Modeliranje korisnika . . . . .	20
<b>5. Ispitivanje modela</b>	<b>22</b>
5.1. Ispitni scenarij . . . . .	22
5.2. Simulacija primjene . . . . .	24

5.3. Provođenje testiranja . . . . .	25
5.4. Rezultati testiranja . . . . .	25
<b>6. Zaključak</b>	<b>26</b>
<b>Literatura</b>	<b>27</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 [2] odnosno 30 milijardi [5]. 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 [9], 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.

Internet stvari samo je jedna od mogućih manifestacija sveprisutnog računarstva.

S druge strane, preporučiteljski sustavi filtriranje sadržaja multimedija, internetske trgovine i ono manje primjetno, pametni prostori, arhitektura preporučitelji svjesni konteksta vrijeme i prostor važni dionici konteksta

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.

Dodano u uvod

Motiv ovog diplomskog rada jest manjak dostupnih algoritama za ovakvu specifičnu vrstu preporučitelja. U ovom radu dat će se teorijska podloga 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. 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 [8].

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 [7].

### 2.2. Razvoj

U općem slučaju mogu se razlikovati tri velika vala u računalnoj eri. 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

### 2.3. Zahtjevi

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



**Slika 2.1:** Tri vala u računarstvu (preuzeto u edukacijske svrhe, ne valja slika)

- 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

Udri brigu na veselje u ovom poglavlju natuč za sva vremena xD



Pametni gradovi, pametni prostori, internet stvari.

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

Primjene u urbanom računarstvo:

- 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 primjeniti na činjenicu da svi imaju isti cilj u šopingu)

## 3. Preporučiteljski sustavi

### 3.1. 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:

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

gdje je  $R$  rezultat, tj. predikcija (engl. *prediction, recommendation*), preferencije korisnika  $U$  (engl. *user*) koji je zatražio preporuku, tj. filtriranje sadržaja, a  $I$  predmet nad kojim se vrši predikcija preferencije (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 postaje:

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

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 raznorazne komponente koje mogu utjecati na preporuku, npr:

- Vrijeme predikcije
- Mjesto na kojem se traži predikcija
- Društvo drugih korisnika u datom trenutku
- Možda još nešto tu navesti?

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, 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*.

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. 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

## 3.2. 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:

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

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.

Razmjerno jednostavna predožba ovog modela jest rejting (engl. *Rating*). Uzmimo za primjer 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 prisanom recenzijom. Također, neka svaki korisnik ima mogućnost ocijeniti uslugu bročanom ocjenom iz intervala od jedan do pet. Model preporučitelja u tom slučaju bio bi 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*. Elemente ovog preporučitelja prikazane relacijama 3.4 i 3.5 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 3.5 prije zaokruživanja prosjek je pomnožen faktorom 10 radi eliminacije decimala. Moguće je odabrati neki drugi prikaz rezultata, primjerice u postocima.

Sam način ocjenjivanja ne mora nužno biti eksplicitna dodjela ocjene. Moguće je primjerice koristiti sustav glasovanja (engl. *Vote up/down*) (engl. *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.

Nedostatak konteksta za preporuke 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:

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

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  $Y$ .

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

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

Naivni preporučitelj preporuku može dati korištenjem intuitivno izvedive formule:

$$R = \frac{X \cap Y}{X} \quad (3.8)$$

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

### 3.3. Filtriranje ovisno o korisniku

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

### 3.3.1. 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 dovoći 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:

$$\mathbf{X}_i = [w_{1,i}, w_{2,i}, w_{3,i}, \dots, w_{N,i}]^T \quad (3.9)$$

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 dva predmeta moguće je tada izraziti kosinusom kuta između vektora njihovih karakteristika.

$$\cos(\mathbf{V}_1, \mathbf{V}_2) = \frac{\mathbf{V}_1 * \mathbf{V}_2}{\|\mathbf{V}_1\| \times \|\mathbf{V}_2\|} \quad (3.10)$$

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 (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.

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?

### 3.3.2. 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 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 gdje je bliskost dva korisnika dobro definirana nekom metrikom, npr. Pearsonovim koeficijentom korelacije (3.11)

$$R = \frac{\sum_{i \in I} [(r_{k,i} - \bar{r}_k) * (r_{u,i} - \bar{r}_u)]}{\sqrt{\sum_{i \in I} (r_{k,i} - \bar{r}_k)^2} * \sqrt{\sum_{i \in I} (r_{u,i} - \bar{r}_u)^2}} \quad (3.11)$$

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 možemo predstaviti kao vektore u  $m$  - dimenzionalnom prostoru pa se težina može izraziti preko skalarnog produkta kao kosinus kuta između ta dva vektora:

$$w_{k,u} = \cos(\vec{r}_k, \vec{r}_u) = \frac{\vec{r}_k * \vec{r}_u}{\|\vec{r}_k\| * \|\vec{r}_u\|} = \frac{\sum_{i=1}^m r_{k,i} * r_{u,i}}{\sqrt{\sum_{i=1}^m r_{k,i}^2} * \sqrt{\sum_{i=1}^m r_{u,i}^2}} \quad (3.12)$$

Mana pristupa 3.12 jest nedostatak negativnih ocjena.



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

$$N = K \setminus \{k\} \quad (3.13)$$

S obzirom na relaciju 3.13 i metriku za ocjenu bliskosti (3.11) ili 3.12 dan je algoritam za filtriranje na relaciji više korisnika: 1

---

**Algorithm 1** Korisnik-korisnik filtriranje

---

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

**Izlaz:** Predikcija  $p$  ocjene korisnika  $K$  za predmet  $P$ .

$topN := 20$

$n := \text{length}(N)$

$w := \text{initVector}()$

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

$w_i := \text{calculatePearson}(k, N_i)$

**end for**

$\text{sort}(w)$

$suma := 0; tezine := 0$

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

$suma := suma + w_i * N_i$

$tezine := tezine + w_i$

**end for**

$p := suma / tezine$

**return**  $p$

---

Algoritam se ugrubo svodi na:

1. Dodjelu težine svim susjedima korisnika  $k$ .
2. Odabir  $n$  susjeda koji imaju najveće težine s obzirom na korisnika  $k$  gdje je broj tih susjeda ( $topN$ ) određen empirijski.
3. Izračun predviđanja ocjene korisnika  $k$  iz težinskog prosjeka ocjena odabranih korisnika

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. U algoritmu 1 se može vidjeti da *for* petlja ima  $N$  prolazaka, a za svaki

taj prolazak računa se Pearsonov koeficijent korelacije dan formulom 3.11 ( $I$  puta). Složenost tada iznosi:

$$O(N * I) \quad (3.14)$$

Iako ocjene 3.14 pokazuje je ovisnost linearna, ipak se vidi da povećanjem broja korisnika ili predmeta dolazi do značajnog povećanja potrebnog broja operacija za izračun preporuke. Valja dodati i da je algoritam 1 pojednostavljena verzija algoritma, pa ni ocjena 3.14 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 [3]. Kao i kod prethodno opisanog pristupa i algoritma 1, metrika za ocjenu sličnosti između predmeta je Pearsonov koeficijent korelacije, u ovom slučaju definiran kao:

$$R = \frac{\sum_{u \in U} [(r_{u,i} - \bar{r}_i) * (r_{u,j} - \bar{r}_j)]}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_i)^2} * \sqrt{\sum_{u \in U} (r_{u,j} - \bar{r}_j)^2}} \quad (3.15)$$

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. Predviđanje ocjene korisnika  $k$  za predmet  $i$  računa se korištenjem težinskog prosjeka:

$$p_{k,i} = \frac{\sum_{j \in N} r_{k,j} * w_{i,j}}{\sum_{j \in N} \|w_{i,j}\|} \quad (3.16)$$

gdje je  $N$  susjedstvo predmeta ocjenjenih od strane korisnika  $k$  najslićnijih predmetu  $i$ . Jedan od najpoznatijih preporučitelja na svijetu, internetska trgovina *Amazon.com* koristi patentirani hibridni algoritam baziran na ovoj tehnici.

## 3.4. Hibridni tehnike

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.

### **3.5. Moguća područja primjene**

### **3.6. 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 sveprisutnom računarstvu 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 [4].

Tri najvažnija aspekta konteksta su [6]:

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 [1]. 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

Priča o postojećim preporučiteljima i kako nemaju prostorno vremenskih komponenti

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.

### 4.2. Naš model

Znači cijeli model treba opisati, od ocjenjivanja do apsolutno svega

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

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

Izražen prostorno vremenski kontekst (općenito) dvije razine reinforcementa. preporučitelj neće raditi online, tj. akcije iz trenutnog sessiona ne utječu na preporuke u tom istom sessionu. tek kad se session commita (po izlasku iz sustava) profili se zbrajaju i preporuke postaju aktualne koje je pravo vrijeme za reinforcement dva profila -> privremeni i trajni profil pa zbroj nakon kupnje netko svaka dva tjedna kupuje jabuke pa mu se svaka dva tjedna kupuju jabuke (recurring)

Tu bi u principu trebalo ubaciti the algoritam \* fade out efekt akcije(u konzumu) I kolaboracija i sadržaj - kod sadržaja, gdje je udaljenost manja od nekog broja (i fizička udaljenost) - račun udaljenosti je apstrahiran

Možda sekcija kontekst pa onda napisati nešto generički o kontekstu pa onda podsekcije prostor i vrijeme. Sekcija za svaku komponentu, korisnika i predmete, fino opisati od čega se sastoje itd.

Kakve su ocjene, koje su skale

1

---

**Algorithm 2** Algoritam kontekstualizirane preporuke u vremenu i prostoru

---

**Ulaz:**

**Izlaz:**

**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**

$p := suma / tezine$

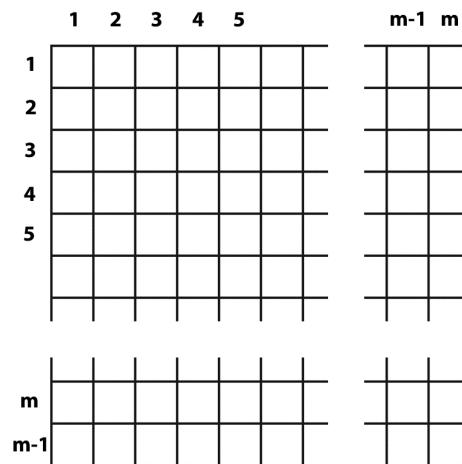
**return**  $p$

---

### 4.3. Prostorna komponenta

S obzirom na kompleksnost postupka modeliranja prostora, prostorna komponenta konteksta modelirat će se pojednostavljeno, 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. 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 putove 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 algoritam* ili *A\* algoritam*.



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

## 4.4. Vremenska komponenta

Vremenska komponenta koja se uzima u obzir u modelu preporučitelja također je za pravo komponenta konteksta u kojem se nalazi korisnik.

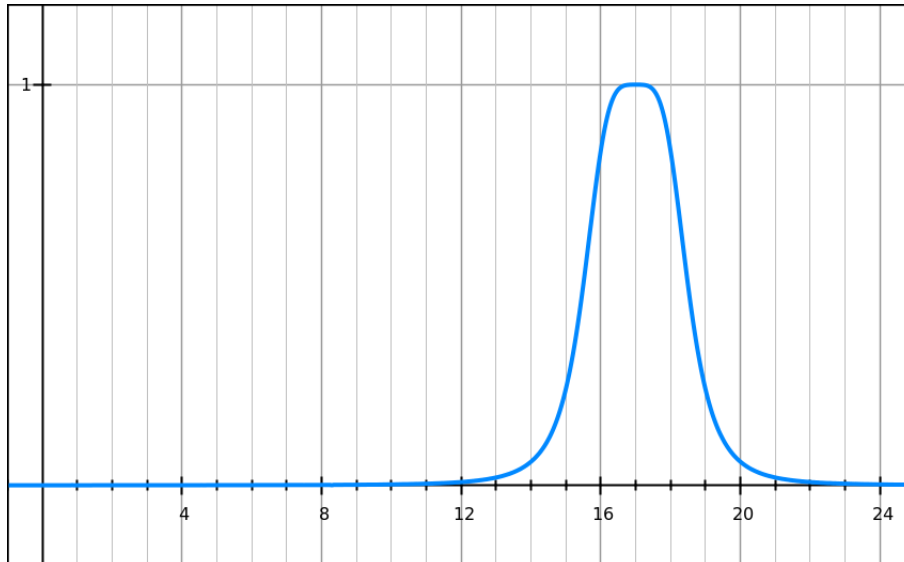
uuuu, korisnik može za određeni proizvod imati funkciju koja opisuje kakve on ima vremenske preferencije s obzirom na taj proizvod. ili funkcija može opisivati samo one trenutke kada se stavlja korisnik u situaciju da traži preporuk

Jedna takva funkcija može biti primjerice tzv. *Gaussova krivulja* (slika 4.2). 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.

$$f(x; a, b, c) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}} \quad (4.3)$$

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 4.2:** 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 (puno kompleksnije situacije). Znači, funkcija može biti na više različitih skala i može odražavati preferenciju za predmet ili preferenciju za traženje preporuke općenito

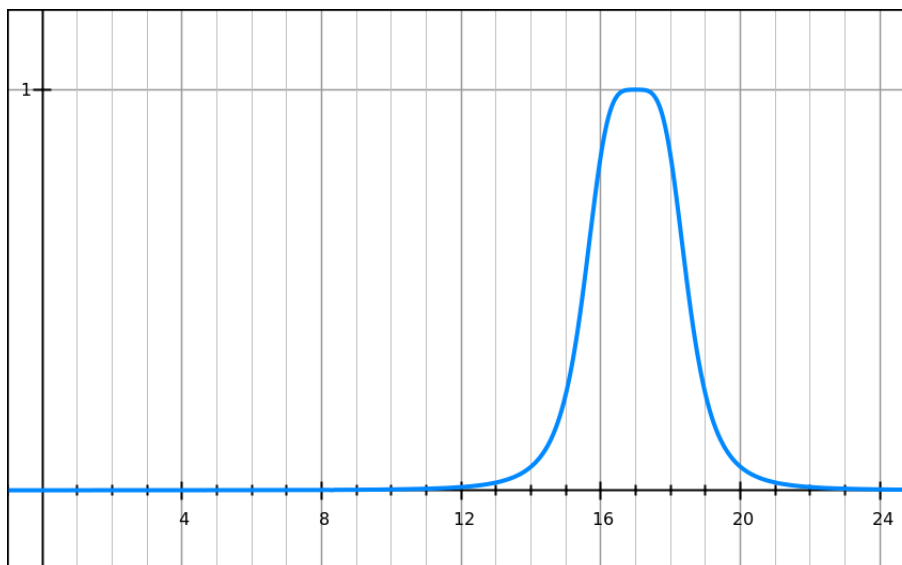
S obzirom na matematičku analizu koja stoji iza računa za pronalazak funkcije, one se mogu zadati i diskretno, npr. kroz vektor uzoraka funkcije. Ta diskretnu funkciju može se koristiti kao takva ili se može procesom interpolacije naći njezin eksplicitni zapis. Pogodne metode interpolacije mogu npr. biti interpolacija polinomom višeg stupnja ili interpolacija splajnovima (engl. *Spline*).

u funkcija može biti kontinuirana, a može biti i uzorkovana i pohranjena u nekom polju. y os funkcije tada je vrijednost koju korisnik pridaje nekom proizvodu u neko doba dana. ne mora biti ni doba dana, može biti doba tjedna. diskretizirane funkcije mogu se interpolirati polinomima višeg stupnja ili splajnovima

Vremenska komponenta omogućava i praćenje obrazaca ponašanja korisnika, pa time i predviđanje ponavljanja ponašanja (engl. *Recurring*).

Vremenska komponenta podrazumjeva i tzv. *efekt blijedenja* (engl. *Fade-out effect*). Ako korisnik kod kojeg je uočen neki obrazac akcija prestane provoditi tu ak-





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

ciju, 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:

$$g(x) = e^{-x} \quad (4.4)$$

Tada će ukupan rezultat vremenske komponente konteksta biti:

$$T \leftarrow f(x) * g(x) \quad (4.5)$$

gdje je  $f(x)$  funkcija koja opisuje vremensku komponentu konteksta nekog korisnika za neki predmet, a  $g(x)$  funkcija koja guši tu komponentu. primjer. Ovdje također postoji mogućnost da se funkcija  $g(x)$  zada diskretno, tj. po uzorcima.

akcije

## 4.5. Modeliranje korisnika

multiplus profil

1. multiplus profil, znamo tko što voli, tj. imamo profil korisnika
2. dopuna profila? kroz šetnju, kroz zadržavanje, kada kupi?
3. cold start profil?

Profil:

1. Proizvodi, skupine proizvoda, redovi? -> stablo?
2. vektor preferencija

## 5. Ispitivanje modela

### 5.1. 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.

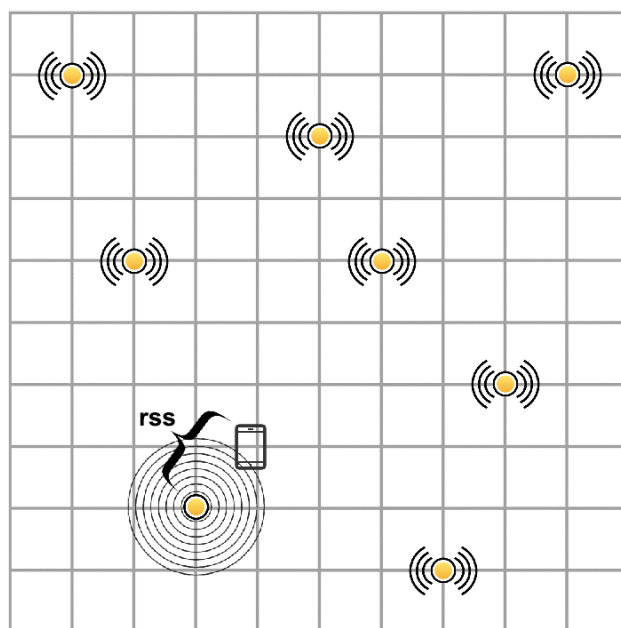
Ispitni scenarij podrazumjeva generički srednje velik dućan u kojem se može naći mješovita roba, dakle, i prehrana i neprehrana. Radi jednostavnosti, a bez značajnijeg utjecaja na krajnji ishod, tlocrt dućana modeliran je mrežastom strukturom prikazanom na slici 4.1. U dućanu se na svakom mjestu mogu 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 možemo aproksimirati lokaciju 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 5.1 prikazuje slučaj kada je mobilni uređaj u dometu jedne radio oznake. U



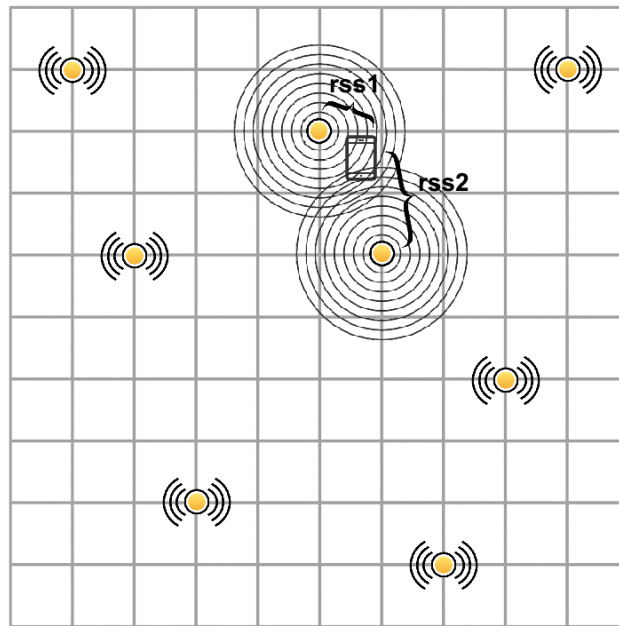
**Slika 5.1:** Grafički prikaz radio oznake i mobilnog telefona

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 5.2), 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 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 poznavanjem 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.



**Slika 5.2:** Grafički prikaz višestrukog prijama signala s radio oznaka

- 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.

## 5.2. Simulacija primjene

Opis načina testiranja Opisati inpute i outpute, konfiguraciju itd.

Ulazi i izlazi

1. logovi (id + rss)
2. multiplus id
3. lokacija beaconsa s id-em
4. multiplus profil (kontekstualizirani), vremenski
5. tendencija kretanja

1. lista s obzirom na lokaciju i vrijeme
2. s obzirom na očekivani profil kupnje

Dnevnički zapisi (engl. *Log*) s pametnog uređaja pišu se i čitaju u predefiniranom formatu

$$timestamp; beaconId1 : rss1, beaconId2 : rss2, beaconId3 : rss3 \quad (5.1)$$

timestamp -> popis beacons + rss magnetometar i akcelerometar shadow preporuke (može u globalni model) -> implicitna preporuka u kojoj korisnika preko akcija vodimo do onoga što njemu treba beacon id -> proizvod i lokaciju

out -> za trenutnu lokaciju -> u kojem smjeru dalje i koji proizvodi idući (topN)

### 5.3. Provođenje testiranja

Opis testnog scenarija u smislu kako smo polijepili beacons po feru, hodali, zaustavljali se itd.

### 5.4. Rezultati testiranja

Prikazati rezultate, jelte

## **6. Zaključak**

Zaključak.

# LITERATURA

- [1] Foursquare. *About Us*, Lipanj 2015. URL <https://foursquare.com/about/>.
- [2] Gartner. *Forecast: The Internet of Things, Worldwide, 2013.*, December 2013. URL <http://www.gartner.com/newsroom/id/2636073>.
- [3] B. ; York J. Linden, G. ; Smith. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 2003.
- [4] 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>.
- [5] 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/>.
- [6] 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.
- [7] N. Vukotić, D. ; Tanković. Alati za razvoj aplikacija bez kodiranja. *Razvoj poslovnih i informatičkih sustava CASE*, 2011.
- [8] 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>.
- [9] 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 ubiquitous computing**

### **Abstract**

Abstract.

**Keywords:** Keywords.