

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

1. Uvod	1
2. Sveprisutno računarstvo	2
2.1. Uvod u sveprisutno računarstvo	2
2.2. Razvoj	2
2.3. Zahtjevi	2
2.4. Primjena	3
3. Preporučiteljski sustavi	5
3.1. Uvod u preporučiteljske sustave	5
3.2. Razvoj preporučiteljskih sustava	7
3.3. Filtriranje neovisno o korisniku	8
3.4. Filtriranje ovisno o korisniku	11
3.4.1. Filtriranje zasnovano na sadržaju	11
3.4.2. Filtriranje zasnovano na suradnji	13
3.5. Redukcija dimenzija	18
3.6. Hibridni tehnike	19
3.7. Moguća područja primjene	20
3.8. Preporučiteljski sustavi u sveprisutnom računarstvu	20
4. Izgrađeni model preporučiteljskog sustava	21
4.1. Preporučiteljski sustavi s izraženom prostornom i vremenskom komponentom	21
4.2. Naš model	21
4.3. Prostorna komponenta	22
4.4. Vremenska komponenta	24
4.5. Modeliranje korisnika	26

5. Ispitivanje modela	29
5.1. Ispitni scenarij	29
5.2. Simulacija primjene	31
5.3. Provođenje testiranja	32
5.4. Rezultati testiranja	32
6. Zaključak	33
Literatura	34
A. Primjer dnevničke datoteke	37

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 [8]. 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 [14], 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 tehnologije koje nestaju, u smislu da korisnici izgube pojam o korištenju te tehnologije [13].

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 [12].

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, podrazred sustava za filtriranje podataka kojima je svrha predviđanje ocjene (ili preferencije) kojom bi neki korisnik ocijenio neki predmet u sustavu [10]. 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 five search results on StackOverflow. Each result includes an up/down arrow, a score, a post icon, a title, a snippet, and metadata like 'submitted' time and 'comments'.

- 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.

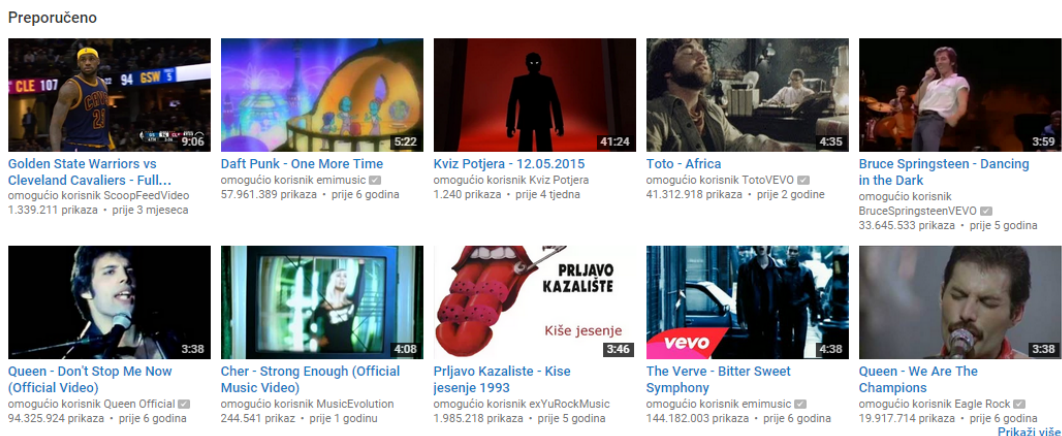
3.4.1. Filtriranje zasnovano na sadržaju

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 [10]. 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:

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

Vektore je poželjno normalizirati, TFIDF

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

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

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:  $\text{idfPolje} := \text{initPolje}(\text{width}(I))$ 
5:  $\text{userPolje} := \text{initPolje}(\text{width}(I))$ 
6: for ( $i := 0; i < \text{height}(I); \text{inc}(i)$ ) do
7:    $\text{suma} := 0$ 
8:   for ( $j := 0; j < \text{width}(I); \text{inc}(j)$ ) do
9:     if ( $i_j \neq 0$ ) then
10:       $\text{inc}(\text{suma});$ 
11:     end if
12:   end for
13:   for ( $j := 0; j < \text{width}(I); \text{inc}(j)$ ) do
14:      $i_j := 1/\sqrt{\text{suma}};$ 
15:   end for
16: end for
17:  $\text{count} := 0;$ 
18: for ( $i := 0; i < \text{width}(I); \text{inc}(i)$ ) do
19:   for ( $j := 0; j < \text{height}(I); \text{inc}(j)$ ) do
20:      $\text{userPolje}_i := \text{userPolje}_i + I_{i,j} * k_j;$ 
21:     if ( $I_{i,j} \neq 0$ ) then
22:        $\text{inc}(\text{count});$ 
23:     end if
24:   end for
25:    $\text{idfPolje}_i := 1/\text{count};$ 
26:    $\text{count} := 0;$ 
27: end for
28:  $L_i := L_i + i_j * \text{idf}_j * \text{userProfile}_j$ 
29: for ( $i := 0; i < \text{height}(I); \text{inc}(i)$ ) do
30:   for ( $j := 0; j < \text{width}(I); \text{inc}(j)$ ) do
31:      $L_i := L_i + I_j * \text{userProfile}_j * \text{idfPolje}_j;$ 
32:   end for
33: end for
34: return  $L$ 
```

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: 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:

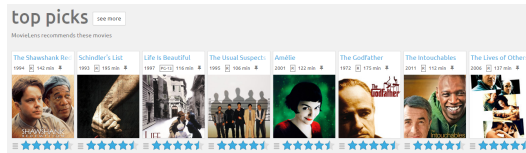
1. Račun korelacije između svih korisnika u sustavu korištenjem 3.12

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  $\text{sort}(L);$ 
```

2. Odabir n susjeda koji imaju najveće težine s obzirom na korisnika k gdje je broj tih susjeda ($topN$) 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“ [9]. Taj preporučitelj na temelju korisničkih ocjena i 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.



Slika 3.7: Primjer preporuke sustava „MovieLens“

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. 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

SVD itd

3.6. 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.

3.7. Moguća područja primjene

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 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 [7].

Tri najvažnija aspekta konteksta su [11]:

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

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.

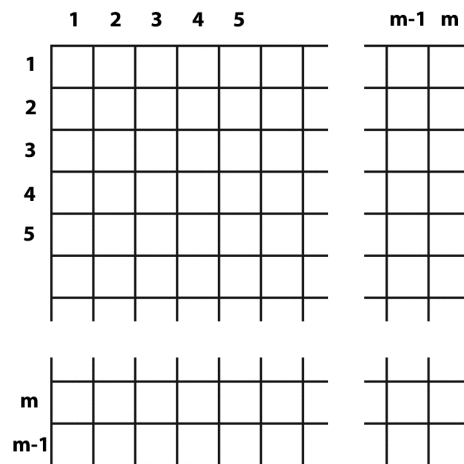
Kakve su ocjene, koje su skale

1. Prostorni filter: sve što je trenutno u okolini korisnika manje od nekog graničnog „radiusa“, uzmi u obzir, ostalo odbaci. Potrebno između ostalog da smanjimo broj računanja. Što je veća blizina, to veća težina u rezultatu. Tu ide definicija udaljenosti, ili graf pa Dijkstra/Bellman-Ford/A*, ili kvadratna mreža pa Manhattan distance
2. Vremenski filter: Uzeti u obzir dio dana i dio tjedna, što smo vremenski bliže vremenu u kojem korisnik kupuje te predmete, to je veća težina/utjecaj. Svaki korisnik ima funkcije (kontinuirane ili diskretne) kojima se opisuje vremenska preferencija za neke predmete
3. U ovoj fazi postoji lista s poredanim prostorno vremenskim „udaljenostima“. Ako sadrži manje od nekog broja predmeta, ponovi od 1. ali olabavi kriterije
4. U ovoj fazi primjenjujemo prilagođeni kolaborativni i content filter. Može i SVD dekompozicija(skupo) pa matrična algebra
5. Imamo listu, sortira se prema lokaciji i vodimo korisnika.

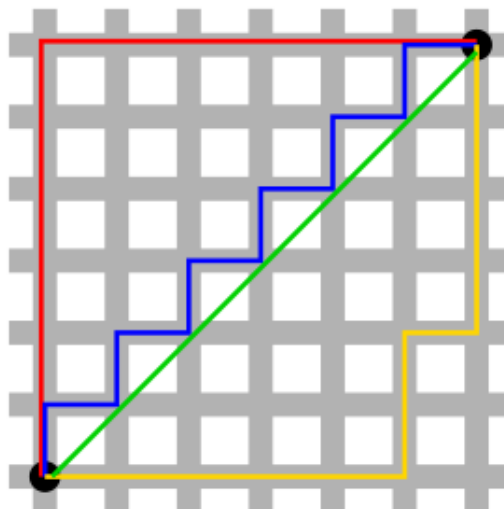
4.3. Prostorna komponenta

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 *ključ* → *vrijednost* gdje je *ključ* neko polje kvadratne mreže a *vrijednost* lista svih *susjednih* polja do kojih je moguće doći iz polja *ključ*. 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



Slika 4.2: Ilustracija Manhattanske udaljenosti

4.4. Vremenska komponenta

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

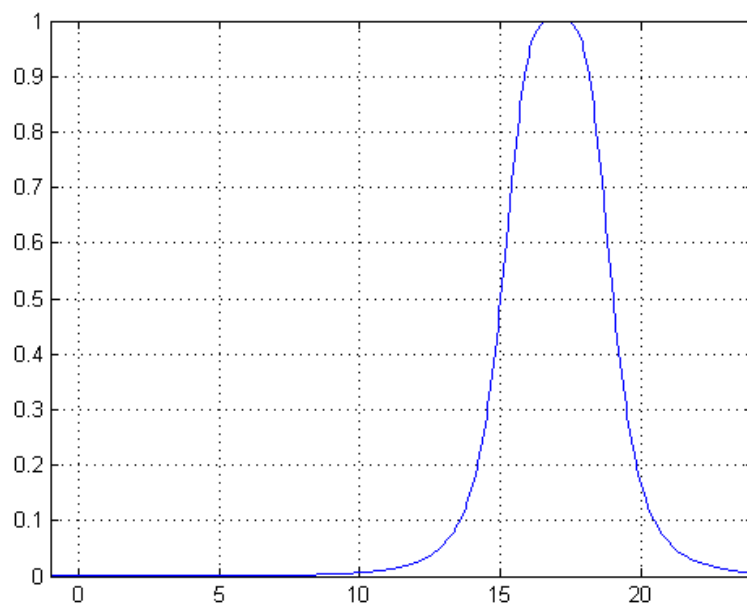
uuuu, korisnik može za određeni proizvod imati funkciju koja opisuje kakve on ima vremenske preferencije s obzirom na taj proizvod. ili funkcija može opisivati samo one trenutke kada se stavlja korisnik u situaciju da traži preporuk

Jedna takva funkcija može biti primjerice tzv. *Gaussova krivulja* (slika 4.3). 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.

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

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

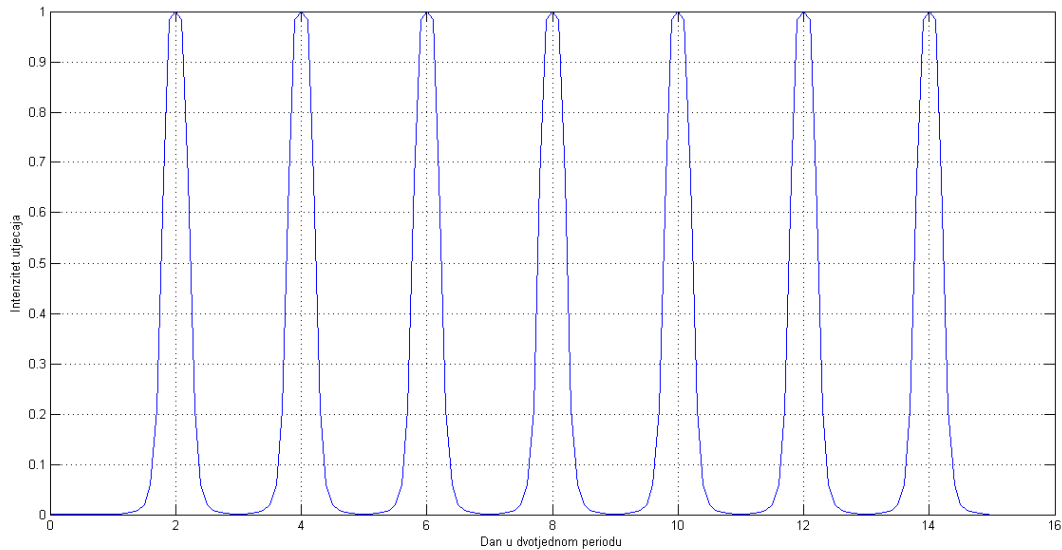
- a - širina vrha
- b - nagib porasta i pada, oblik vrha
- c - pomak na x osi



Slika 4.3: 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 (puno kompleksnije situacije). Znači, funkcija može biti na više

različitih skala i može odražavati preferenciju za predmet ili preferenciju za traženje preporuke općenito



Slika 4.4: Preferencija korisnika za predmet na dnevnoj skali

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 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:

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

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.

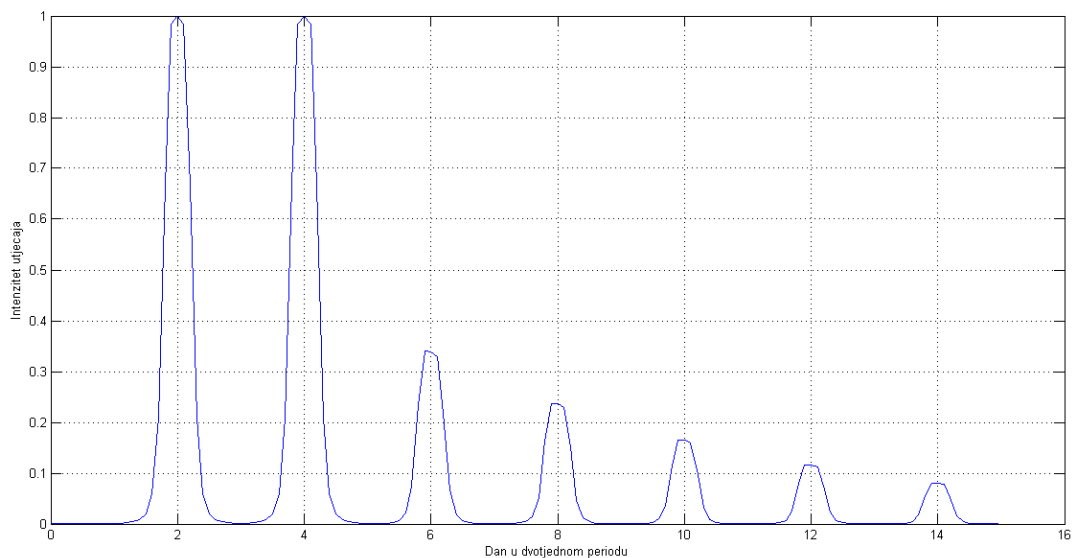
Na slici 4.5 vidljiv je primjer *efekta blijedenja* na tipično periodičko ponašanje korisnika. Prema formuli 4.5 vrijedi:

$$f(x) = \frac{1}{1 + \left| \frac{x-2*k}{0.23} \right|^5} + 2 * k + 1 \quad (4.6)$$

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

$$g(x) = e^{-0.18*x} \quad (4.7)$$

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



Slika 4.5: Preferencija korisnika za predmet na dnevnoj skali

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

Algorithm 3 Algoritam kontekstualizirane preporuke u vremenu i prostoru

Ulaz: Korisnik u , Prostor l , Vrijeme t

Izlaz: Vektor kretanja v

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

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.

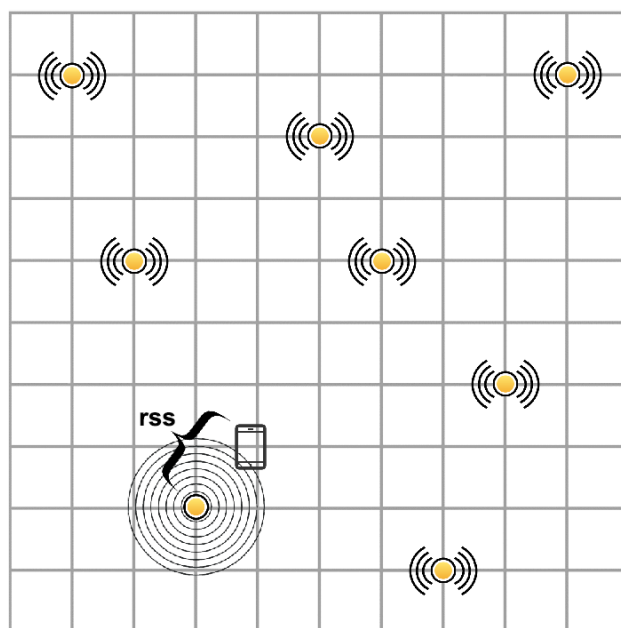
Ispitni scenarij podrazumjeva generički srednje velik dućan u kojem se može naći mješovita roba, dakle, i prehrana i neprehrana. Radi jednostavnosti, a bez značajnijeg utjecaja na krajnji ishod, tlocrt dućana modeliran je mrežastom strukturom prikazanom na slici 4.1. U dućanu se na svakom mjestu mogu 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



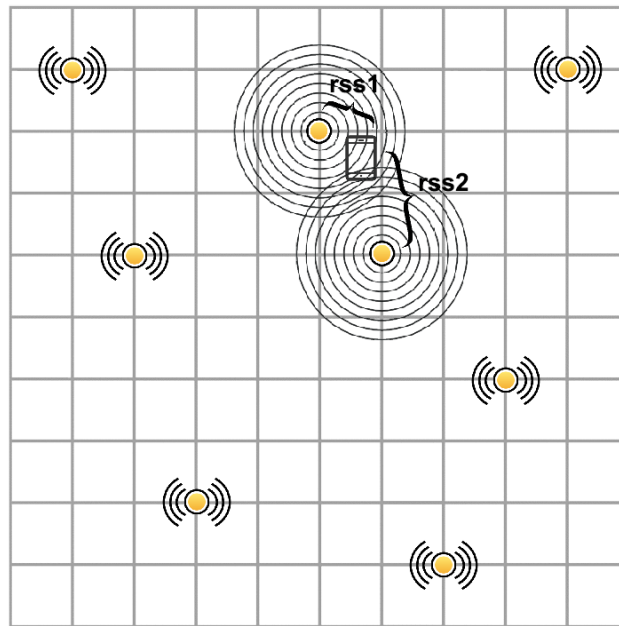
Slika 5.1: Grafički prikaz radio oznake i mobilnog telefona

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

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:

$$timestamp; beaconId\$rss\#accelerometer\#magnetometer\backslash n \quad (5.1)$$

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 μT

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] 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] 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>.
- [8] 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/>.

- [9] GroupLens Research. *About MovieLens*, July 2015. URL <https://movielens.org/info/pages/about/>.
- [10] 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.
- [11] 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.
- [12] N. Vukotić, D. ; Tanković. Alati za razvoj aplikacija bez kodiranja. *Razvoj poslovnih i informatičkih sustava CASE*, 2011.
- [13] 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>.
- [14] 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.

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