

Sustavi za preporučivanje

Josip Šumečki

28. svibnja 2010.

Problem preporučivanja

- ▶ nastao sredinom 90.-ih godina, ali se danas još razvija i proširuje se područje primjene

Problem preporučivanja

- ▶ nastao sredinom 90.-ih godina, ali se danas još razvija i proširuje se područje primjene
- ▶ 2006. godine populariziran zbog „Netflix Prize“

Problem preporučivanja

- ▶ nastao sredinom 90.-ih godina, ali se danas još razvija i proširuje se područje primjene
- ▶ 2006. godine populariziran zbog „Netflix Prize“
- ▶ predviđanje koji predmet će zanimati određenog korisnika

Problem preporuĉivanja

- ▶ nastao sredinom 90.-ih godina, ali se danas još razvija i proširuje se područje primjene
- ▶ 2006. godine populariziran zbog „Netflix Prize“
- ▶ predviđanje koji predmet će zanimati određenog korisnika
- ▶ predviđanje se bazira na podacima o:

Problem preporučenja

- ▶ nastao sredinom 90.-ih godina, ali se danas još razvija i proširuje se područje primjene
- ▶ 2006. godine populariziran zbog „Netflix Prize“
- ▶ predviđanje koji predmet će zanimati određenog korisnika
- ▶ predviđanje se bazira na podacima o:
 - ▶ korisniku kojem se preporučaju predmeti (starost, mjesto prebivanja, zaposlenje, ...)

Problem preporučenja

- ▶ nastao sredinom 90.-ih godina, ali se danas još razvija i proširuje se područje primjene
- ▶ 2006. godine populariziran zbog „Netflix Prize“
- ▶ predviđanje koji predmet će zanimati određenog korisnika
- ▶ predviđanje se bazira na podacima o:
 - ▶ korisniku kojem se preporučaju predmeti (starost, mjesto prebivanja, zaposlenje, ...)
 - ▶ povratne informacije korisnika o nekim drugim predmetima (ocjene, broj pregleda, ...)

Problem preporučenja

- ▶ nastao sredinom 90.-ih godina, ali se danas još razvija i proširuje se područje primjene
- ▶ 2006. godine populariziran zbog „Netflix Prize“
- ▶ predviđanje koji predmet će zanimati određenog korisnika
- ▶ predviđanje se bazira na podacima o:
 - ▶ korisniku kojem se preporučaju predmeti (starost, mjesto prebivanja, zaposlenje, ...)
 - ▶ povratne informacije korisnika o nekim drugim predmetima (ocjene, broj pregleda, ...)
 - ▶ podacima vezanim uz korisnike koji imaju sličan ukus

Problem preporučenja

- ▶ nastao sredinom 90.-ih godina, ali se danas još razvija i proširuje se područje primjene
- ▶ 2006. godine populariziran zbog „Netflix Prize“
- ▶ predviđanje koji predmet će zanimati određenog korisnika
- ▶ predviđanje se bazira na podacima o:
 - ▶ korisniku kojem se preporučaju predmeti (starost, mjesto prebivanja, zaposlenje, ...)
 - ▶ povratne informacije korisnika o nekim drugim predmetima (ocjene, broj pregleda, ...)
 - ▶ podacima vezanim uz korisnike koji imaju sličan ukus
 - ▶ predmetima koji su slični uz predmet koji se korisniku sviđa

Formalna definicija problema

U konačan skup kojeg nazivamo **skup korisnika** (*users*)

Formalna definicija problema

U konačan skup kojeg nazivamo **skup korisnika** (*users*)

I konačan skup kojeg nazivamo **skup predmeta** (*items*)

Formalna definicija problema

- U konačan skup kojeg nazivamo **skup korisnika** (*users*)
- I konačan skup kojeg nazivamo **skup predmeta** (*items*)
- T totalno uređen skup (npr. $[0, 1]$, $\{0, 1, \dots, 10\}$, \mathbb{R}^+)

Formalna definicija problema

- U konačan skup kojeg nazivamo **skup korisnika** (*users*)
- I konačan skup kojeg nazivamo **skup predmeta** (*items*)
- T totalno uređen skup (npr. $[0, 1]$, $\{0, 1, \dots, 10\}$, \mathbb{R}^+)
- $r : U \times I \rightarrow T$ funkcija koja pridružuje vrijednost korisnost nekog predmeta određenom korisniku. (*rank*)

Formalna definicija problema

- U konačan skup kojeg nazivamo **skup korisnika** (*users*)
- I konačan skup kojeg nazivamo **skup predmeta** (*items*)
- T totalno uređen skup (npr. $[0, 1]$, $\{0, 1, \dots, 10\}$, \mathbb{R}^+)
- $r : U \times I \rightarrow T$ funkcija koja pridružuje vrijednost korisnost nekog predmeta određenom korisniku. (*rank*)
 - ▶ nije nužno udaljenost vezana uz normu, iako se može zahtijevati

Formalna definicija problema

- U konačan skup kojeg nazivamo **skup korisnika** (*users*)
- I konačan skup kojeg nazivamo **skup predmeta** (*items*)
- T totalno uređen skup (npr. $[0, 1]$, $\{0, 1, \dots, 10\}$, \mathbb{R}^+)
- $r : U \times I \rightarrow T$ funkcija koja pridružuje vrijednost korisnost nekog predmeta određenom korisniku. (*rank*)
 - ▶ nije nužno udaljenost vezana uz normu, iako se može zahtijevati
 - ▶ ideja je da se pridružuje intuitivno veća korisnost predmeta sa većim elementom iz T

Formalna definicija problema

- U konačan skup kojeg nazivamo **skup korisnika** (*users*)
- I konačan skup kojeg nazivamo **skup predmeta** (*items*)
- T totalno uređen skup (npr. $[0, 1]$, $\{0, 1, \dots, 10\}$, \mathbb{R}^+)
- $r : U \times I \rightarrow T$ funkcija koja pridružuje vrijednost korisnost nekog predmeta određenom korisniku. (*rank*)
 - ▶ nije nužno udaljenost vezana uz normu, iako se može zahtijevati
 - ▶ ideja je da se pridružuje intuitivno veća korisnost predmeta sa većim elementom iz T
 - ▶ obično se najprije definira na predmetima koje je korisnik ocjenio i dalje se proširuje na ostale predmete

Formalna definicija problema

U konačan skup kojeg nazivamo **skup korisnika** (*users*)

I konačan skup kojeg nazivamo **skup predmeta** (*items*)

T totalno uređen skup (npr. $[0, 1]$, $\{0, 1, \dots, 10\}$, \mathbb{R}^+)

$r : U \times I \rightarrow T$ funkcija koja pridružuje vrijednost korisnost nekog predmeta određenom korisniku. (*rank*)

- ▶ nije nužno udaljenost vezana uz normu, iako se može zahtijevati
- ▶ ideja je da se pridružuje intuitivno veća korisnost predmeta sa većim elementom iz T
- ▶ obično se najprije definira na predmetima koje je korisnik ocjenio i dalje se proširuje na ostale predmete

Problem je tada formalno definiran kao nalaženje i'_u da vrijedi

$$(\forall u \in U) \quad i'_u := \operatorname{argmax}_{i \in I} r(u, i).$$

Rješavanje problema

Algoritmi kojima se problem rješava:

- ▶ **kNN** je osnovni algoritam za rješavanje, a većina ostalih ga proširuje na razne načine

Rješavanje problema

Algoritmi kojima se problem rješava:

- ▶ **kNN** je osnovni algoritam za rješavanje, a većina ostalih ga proširuje na razne načine
- ▶ **Collaborative Filtering (CF)** je najrazvijeniji algoritam za rješavanje ovog problema. Traže se predmeti koje su visoko ocijenili ljudi koji imaju sličan ukus kao tražena osoba. (baziran na kNN-u)

Rješavanje problema

Algoritmi kojima se problem rješava:

- ▶ **kNN** je osnovni algoritam za rješavanje, a većina ostalih ga proširuje na razne načine
- ▶ **Collaborative Filtering (CF)** je najrazvijeniji algoritam za rješavanje ovog problema. Traže se predmeti koje su visoko ocjenili ljudi koji imaju sličan ukus kao tražena osoba. (baziran na kNN-u)
- ▶ **Content-based recommendation (CB)** je algoritam koji predlaže predmete slične onima koje je korisnik dobro ocjenio ili onom predmetu kojeg korisnik trenutno pregledava (baziran na kNN-u)

- ▶ **tagommenderi**, tj. algoritmi koji svakom itemu pridružuju tagove i prate koliko korisnik preferira iteme vezane uz svaki tag (npr. stereomood.com)

- ▶ **tagommenderi**, tj. algoritmi koji svakom itemu pridružuju tagove i prate koliko korisnik preferira iteme vezane uz svaki tag (npr. stereomood.com)
- ▶ **hibridni algoritmi** koji kombiniraju gore navedene algoritme te ih upotpunjuju raznim drugim metodama (neuralne mreže, stabla odlučivanja, particioniranje grafova i sl.)

- ▶ cilj CB algoritma je na temelju prethodno dobro ocijenjenih predmeta nekog korisnika pronaći nekoliko sličnih predmeta koje korisnik još nije ocijenio

- ▶ cilj CB algoritma je na temelju prethodno dobro ocijenjenih predmeta nekog korisnika pronaći nekoliko sličnih predmeta koje korisnik još nije ocijenio
- ▶ pokazalo se dobrim da probleme vezane uz text mining uz TF-IDF funkciju za traženje sličnosti (preporučivanje blogova, znanstvenih radova i sl.)

- ▶ cilj CB algoritma je na temelju prethodno dobro ocijenjenih predmeta nekog korisnika pronaći nekoliko sličnih predmeta koje korisnik još nije ocijenio
- ▶ pokazalo se dobrim da probleme vezane uz text mining uz TF-IDF funkciju za traženje sličnosti (preporučivanje blogova, znanstvenih radova i sl.)
- ▶ nedostaci:

- ▶ cilj CB algoritma je na temelju prethodno dobro ocijenjenih predmeta nekog korisnika pronaći nekoliko sličnih predmeta koje korisnik još nije ocijenio
- ▶ pokazalo se dobrim da probleme vezane uz text mining uz TF-IDF funkciju za traženje sličnosti (preporučivanje blogova, znanstvenih radova i sl.)
- ▶ nedostaci:
 - ▶ teško povezati sa multimedijalnim sadržajem

- ▶ cilj CB algoritma je na temelju prethodno dobro ocijenjenih predmeta nekog korisnika pronaći nekoliko sličnih predmeta koje korisnik još nije ocijenio
- ▶ pokazalo se dobrim da probleme vezane uz text mining uz TF-IDF funkciju za traženje sličnosti (preporučivanje blogova, znanstvenih radova i sl.)
- ▶ nedostaci:
 - ▶ teško povezati sa multimedijalnim sadržajem
 - ▶ teško je bilo što preporučiti novom korisniku

- ▶ cilj CB algoritma je na temelju prethodno dobro ocijenjenih predmeta nekog korisnika pronaći nekoliko sličnih predmeta koje korisnik još nije ocijenio
- ▶ pokazalo se dobrim da probleme vezane uz text mining uz TF-IDF funkciju za traženje sličnosti (preporučivanje blogova, znanstvenih radova i sl.)
- ▶ nedostaci:
 - ▶ teško povezati sa multimedijalnim sadržajem
 - ▶ teško je bilo što preporučiti novom korisniku
 - ▶ ako smo dobro ocijenili neki predmet, moguće da će nam sustav preporučiti neki identični, samo sa drugog izvora

- ▶ cilj CB algoritma je na temelju prethodno dobro ocijenjenih predmeta nekog korisnika pronaći nekoliko sličnih predmeta koje korisnik još nije ocijenio
- ▶ pokazalo se dobrim da probleme vezane uz text mining uz TF-IDF funkciju za traženje sličnosti (preporučivanje blogova, znanstvenih radova i sl.)
- ▶ nedostaci:
 - ▶ teško povezati sa multimedijalnim sadržajem
 - ▶ teško je bilo što preporučiti novom korisniku
 - ▶ ako smo dobro ocijenili neki predmet, moguće da će nam sustav preporučiti neki identični, samo sa drugog izvora
 - ▶ korisniku nikad neće biti preporučeni predmeti koji bi mu se možda sviđali, ali ih nikad nije ocjenio (ili vidio) ... rješava se ubacivanjem šuma (randomizacija)

- ▶ cilj CB algoritma je na temelju prethodno dobro ocijenjenih predmeta nekog korisnika pronaći nekoliko sličnih predmeta koje korisnik još nije ocijenio
- ▶ pokazalo se dobrim da probleme vezane uz text mining uz TF-IDF funkciju za traženje sličnosti (preporučivanje blogova, znanstvenih radova i sl.)
- ▶ nedostaci:
 - ▶ teško povezati sa multimedijalnim sadržajem
 - ▶ teško je bilo što preporučiti novom korisniku
 - ▶ ako smo dobro ocijenili neki predmet, moguće da će nam sustav preporučiti neki identični, samo sa drugog izvora
 - ▶ korisniku nikad neće biti preporučeni predmeti koji bi mu se možda sviđali, ali ih nikad nije ocjenio (ili vidio) ... rješava se ubacivanjem šuma (randomizacija)
 - ▶ ne prepoznaje koji članak je kvalitetniji među onima koji imaju iste ključne riječi

Primjer CB filteringa na problemu text mininga

- ▶ svaki tekst d_j prikazati kao vektor težina

$CI(d_j) = (w_{1,j}, \dots, w_{k,j})$, gdje su

$$w_{i,j} = TF(i,j) \times IDF(i)$$

za ključne riječi sa indeksima $j \in \{1, \dots, k\}$

Primjer CB filteringa na problemu text mininga

- ▶ svaki tekst d_j prikazati kao vektor težina

$CI(d_j) = (w_{1,j}, \dots, w_{k,j})$, gdje su

$$w_{i,j} = TF(i,j) \times IDF(i)$$

za ključne riječi sa indeksima $j \in \{1, \dots, k\}$

- ▶ svakog korisnika c prikazati kao vektor težina

$CU(c) = (w_{c,1}, \dots, w_{c,k})$, gdje su $w_{c,i}$ težine koje opisuju važnost ključne riječi sa indeksom i korisniku c

Primjer CB filteringa na problemu text mininga

- ▶ svaki tekst d_j prikazati kao vektor težina

$CI(d_j) = (w_{1,j}, \dots, w_{k,j})$, gdje su

$$w_{i,j} = TF(i,j) \times IDF(i)$$

za ključne riječi sa indeksima $j \in \{1, \dots, k\}$

- ▶ svakog korisnika c prikazati kao vektor težina

$CU(c) = (w_{c,1}, \dots, w_{c,k})$, gdje su $w_{c,i}$ težine koje opisuju važnost ključne riječi sa indeksom i korisniku c

- ▶ za predviđanje koliko će se nekom korisniku c svidjeti predmet s se koristi funkcija

$$u(c, s) = score(CU(c), CI(s))$$

Collaborative filtering

CF algoritmi se najčešće dijele na sljedeće podvrste:

- ▶ **Memory(heuristic)-based** CF koji traži nekoliko najsličnijih korisnika i vraća srednju vrijednost njihovi preporuka za neki predmet

Collaborative filtering

CF algoritmi se najčešće dijele na sljedeće podvrste:

- ▶ **Memory(heuristic)-based** CF koji traži nekoliko najsličnijih korisnika i vraća srednju vrijednost njihovi preporuka za neki predmet
 - ▶ **user-based** CF koji koristi već izračunate tabele sa međusobnim sličnostima korisnika

Collaborative filtering

CF algoritmi se najčešće dijele na sljedeće podvrste:

- ▶ **Memory(heuristic)-based** CF koji traži nekoliko najsličnijih korisnika i vraća srednju vrijednost njihovi preporuka za neki predmet
 - ▶ **user-based** CF koji koristi već izračunate tabele sa međusobnim sličnostima korisnika
 - ▶ **item-based** CF koji koristi već izračunate tabele sa međusobnim sličnostima predmeta

Collaborative filtering

CF algoritmi se najčešće dijele na sljedeće podvrste:

- ▶ **Memory(heuristic)-based** CF koji traži nekoliko najsličnijih korisnika i vraća srednju vrijednost njihovi preporuka za neki predmet
 - ▶ **user-based** CF koji koristi već izračunate tabele sa međusobnim sličnostima korisnika
 - ▶ **item-based** CF koji koristi već izračunate tabele sa međusobnim sličnostima predmeta
- ▶ **Model-based** CF koji se temelji na učenju modela (raznoliki načini implementacije)

Collaborative filtering

CF algoritmi se najčešće dijele na sljedeće podvrste:

- ▶ **Memory(heuristic)-based** CF koji traži nekoliko najsličnijih korisnika i vraća srednju vrijednost njihovi preporuka za neki predmet
 - ▶ **user-based** CF koji koristi već izračunate tabele sa međusobnim sličnostima korisnika
 - ▶ **item-based** CF koji koristi već izračunate tabele sa međusobnim sličnostima predmeta
- ▶ **Model-based** CF koji se temelji na učenju modela (raznoliki načini implementacije)
- ▶ **Kombinirani** CF koji koristi prednosti obje gore navedene vrste CF-a

Usporedba podataka u CF-u

- ▶ odabir funkcije za usporedbu bilo koja dva predmeta i/ili korisnika. U obzir dolaze:

Usporedba podataka u CF-u

- ▶ odabir funkcije za usporedbu bilo koja dva predmeta i/ili korisnika. U obzir dolaze:
 - ▶ sve norme, ali promijenjene da vraćaju veću vrijednost za sličnije podatke;

Usporedba podataka u CF-u

- ▶ odabir funkcije za usporedbu bilo koja dva predmeta i/ili korisnika. U obzir dolaze:
 - ▶ sve norme, ali promijenjene da vraćaju veću vrijednost za sličnije podatke;
 - ▶ Pearsonov koeficijent korelacije;

Usporedba podataka u CF-u

- ▶ odabir funkcije za usporedbu bilo koja dva predmeta i/ili korisnika. U obzir dolaze:
 - ▶ sve norme, ali promijenjene da vraćaju veću vrijednost za sličnije podatke;
 - ▶ Pearsonov koeficijent korelacije;
 - ▶ cosine similarity measure;

Usporedba podataka u CF-u

- ▶ odabir funkcije za usporedbu bilo koja dva predmeta i/ili korisnika. U obzir dolaze:
 - ▶ sve norme, ali promijenjene da vraćaju veću vrijednost za sličnije podatke;
 - ▶ Pearsonov koeficijent korelacije;
 - ▶ cosine similarity measure;
 - ▶ udaljenost vezana uz „slope 1“ pravac;

Usporedba podataka u CF-u

- ▶ odabir funkcije za usporedbu bilo koja dva predmeta i/ili korisnika. U obzir dolaze:
 - ▶ sve norme, ali promijenjene da vraćaju veću vrijednost za sličnije podatke;
 - ▶ Pearsonov koeficijent korelacije;
 - ▶ cosine similarity measure;
 - ▶ udaljenost vezana uz „slope 1“ pravac;
 - ▶ mnoge druge funkcije (Jaccard, Tanimoto, ...).

Usporedba podataka u CF-u

- ▶ odabir funkcije za usporedbu bilo koja dva predmeta i/ili korisnika. U obzir dolaze:
 - ▶ sve norme, ali promijenjene da vraćaju veću vrijednost za sličnije podatke;
 - ▶ Pearsonov koeficijent korelacije;
 - ▶ cosine similarity measure;
 - ▶ udaljenost vezana uz „slope 1“ pravac;
 - ▶ mnoge druge funkcije (Jaccard, Tanimoto, ...).
- ▶ Često se primjenjuju i uz linearnu regresiju (Pearsonov koeficijent, slope 1).

Matrica user vs. item

Prvi korak kod stvaranja item-based CF algoritma jest stvoriti matricu (tabelu) koja svakom paru (korisnik, predmet) pridružuje ocjenu koju je korisnik dodijelio predmetu, odnosno specifičnu oznaku ako vrijednost nije dodijeljena (obično 0)

Svojstva user vs. item matrice:

- ▶ **sparsity**: matrica je veoma rijetka (većinom iznad 99%)

Matrica user vs. item

Prvi korak kod stvaranja item-based CF algoritma jest stvoriti matricu (tabelu) koja svakom paru (korisnik, predmet) pridružuje ocjenu koju je korisnik dodijelio predmetu, odnosno specifičnu oznaku ako vrijednost nije dodijeljena (obično 0)

Svojstva user vs. item matrice:

- ▶ **sparsity**: matrica je veoma rijetka (većinom iznad 99%)
- ▶ korisnici načešće ocjenjuju ono što vole

Matrica user vs. item

Prvi korak kod stvaranja item-based CF algoritma jest stvoriti matricu (tabelu) koja svakom paru (korisnik, predmet) pridružuje ocjenu koju je korisnik dodijelio predmetu, odnosno specifičnu oznaku ako vrijednost nije dodijeljena (obično 0)

Svojstva user vs. item matrice:

- ▶ **sparsity**: matrica je veoma rijetka (većinom iznad 99%)
- ▶ korisnici načešće ocjenjuju ono što vole
- ▶ korisnici daju često najvišu ocjenu ako im se nešto sviđa

Matrica user vs. item

Prvi korak kod stvaranja item-based CF algoritma jest stvoriti matricu (tabelu) koja svakom paru (korisnik, predmet) pridružuje ocjenu koju je korisnik dodijelio predmetu, odnosno specifičnu oznaku ako vrijednost nije dodijeljena (obično 0)

Svojstva user vs. item matrice:

- ▶ **sparsity**: matrica je veoma rijetka (većinom iznad 99%)
- ▶ korisnici načešće ocjenjuju ono što vole
- ▶ korisnici daju često najvišu ocjenu ako im se nešto sviđa
- ▶ ukus nekog korisnika se mijenja kroz vrijeme

Matrica user vs. item

Prvi korak kod stvaranja item-based CF algoritma jest stvoriti matricu (tabelu) koja svakom paru (korisnik, predmet) pridružuje ocjenu koju je korisnik dodijelio predmetu, odnosno specifičnu oznaku ako vrijednost nije dodijeljena (obično 0)

Svojstva user vs. item matrice:

- ▶ **sparsity**: matrica je veoma rijetka (većinom iznad 99%)
- ▶ korisnici načešće ocjenjuju ono što vole
- ▶ korisnici daju često najvišu ocjenu ako im se nešto sviđa
- ▶ ukus nekog korisnika se mijenja kroz vrijeme
- ▶ neki predmete ovise i o kontekstu u kojem su ocjenjivani

Matrica user vs. item

Prvi korak kod stvaranja item-based CF algoritma jest stvoriti matricu (tabelu) koja svakom paru (korisnik, predmet) pridružuje ocjenu koju je korisnik dodijelio predmetu, odnosno specifičnu oznaku ako vrijednost nije dodijeljena (obično 0)

Svojstva user vs. item matrice:

- ▶ **sparsity**: matrica je veoma rijetka (većinom iznad 99%)
- ▶ korisnici načešće ocjenjuju ono što vole
- ▶ korisnici daju često najvišu ocjenu ako im se nešto sviđa
- ▶ ukus nekog korisnika se mijenja kroz vrijeme
- ▶ neki predmete ovise i o kontekstu u kojem su ocjenjivani
- ▶ dimenzije matrice

Matrica item vs. item

- ▶ item-based CF koristi pretpostavku da se sličnosti između svakog para predmeta rijede mijenja, pa se stvara matrica koja svakom predmetu pridružuje sličnost sa svakim drugim predmetom

Matrica item vs. item

- ▶ item-based CF koristi pretpostavku da se sličnosti između svakog para predmeta rijede mijenja, pa se stvara matrica koja svakom predmetu pridružuje sličnost sa svakim drugim predmetom
- ▶ stvaranje ove matrice traje veoma dugo, ali se obavlja veoma rijetko i posao je vrlo lako paralelizirati

Matrica item vs. item

- ▶ item-based CF koristi pretpostavku da se sličnosti između svakog para predmeta rijede mijenja, pa se stvara matrica koja svakom predmetu pridružuje sličnost sa svakim drugim predmetom
- ▶ stvaranje ove matrice traje veoma dugo, ali se obavlja veoma rijetko i posao je vrlo lako paralelizirati
- ▶ za stvaranje matrice se koristi pretpostavka da su dva itema slična, ako ih je mnogo ljudi slično ocijenilo

Matrica item vs. item

- ▶ item-based CF koristi pretpostavku da se sličnosti između svakog para predmeta rijede mijenja, pa se stvara matrica koja svakom predmetu pridružuje sličnost sa svakim drugim predmetom
- ▶ stvaranje ove matrice traje veoma dugo, ali se obavlja veoma rijetko i posao je vrlo lako paralelizirati
- ▶ za stvaranje matrice se koristi pretpostavka da su dva itema slična, ako ih je mnogo ljudi slično ocijenilo
- ▶ za većinu implementacija nije potrebna cijela matrica, nego samo lista najslbličnijih predmeta povezana sa svakim predmetom

Svojstva item vs. item matrice:

- ▶ matrica se može promatrati kao matrica potpuno povezanog grafa, gdje se „slabe“ veze mogu eliminirati i vršiti razne matrične operacije (razvijeno područje u teoriji grafova)
- ▶ ovakva matrica je ustvari matrica korelacije (razvijeno područje u statistiku)

Ovakvi pristupi ovoj matrici daje nam mogućnost za razvoj metoda koje će veoma brzo vršiti testove vezane uz ovu matricu.

Algoritam item-based CF

Sljedeći algoritam koristi za svaki predmet listu njemu sličnih predmeta (M_s).

Preporučanje n predmeta korisniku u (koje on još nije ocijenio):

1. pronađi sve predmete koje je korisnik u ocijenio

Algoritam item-based CF

Sljedeći algoritam koristi za svaki predmet listu njemu sličnih predmeta (M_s).

Preporučanje n predmeta korisniku u (koje on još nije ocijenio):

1. pronađi sve predmete koje je korisnik u ocijenio
2. za svaki ocijenjen predmet pronađi njemu slične predmete (iz tabele M_s)

Algoritam item-based CF

Sljedeći algoritam koristi za svaki predmet listu njemu sličnih predmeta (M_s).

Preporučanje n predmeta korisniku u (koje on još nije ocijenio):

1. pronađi sve predmete koje je korisnik u ocijenio
2. za svaki ocijenjen predmet pronađi njemu slične predmete (iz tabele M_s)
3. za sve moguće dobivene predmete nađi n predmeta koji će doseći najveći score koji je baziran na:

Algoritam item-based CF

Sljedeći algoritam koristi za svaki predmet listu njemu sličnih predmeta (M_s).

Preporučanje n predmeta korisniku u (koje on još nije ocijenio):

1. pronađi sve predmete koje je korisnik u ocijenio
2. za svaki ocijenjen predmet pronađi njemu slične predmete (iz tabele M_s)
3. za sve moguće dobivene predmete nađi n predmeta koji će doseći najveći score koji je baziran na:
 - ▶ ocjeni koju smo mi dodijelili predmetu kojem je ovaj sličan

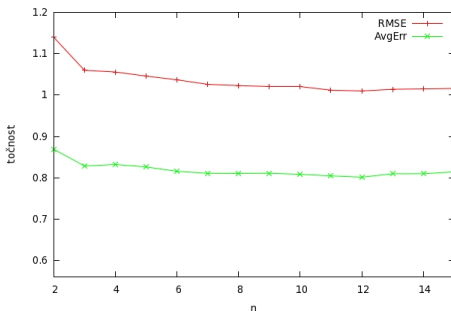
Algoritam item-based CF

Sljedeći algoritam koristi za svaki predmet listu njemu sličnih predmeta (M_s).

Preporučanje n predmeta korisniku u (koje on još nije ocijenio):

1. pronađi sve predmete koje je korisnik u ocijenio
2. za svaki ocijenjen predmet pronađi njemu slične predmete (iz tabele M_s)
3. za sve moguće dobivene predmete nađi n predmeta koji će doseći najveći score koji je baziran na:
 - ▶ ocjeni koju smo mi dodijelili predmetu kojem je ovaj sličan
 - ▶ koeficijentu koji nam govori koliko su ta dva predmeta slična

Testiranje item-based CF algoritma



Slika: Promatran je MovieLens skup podataka koristenjem skupova podataka za treniranje i testiranje. Minimum se postize za $n = 12$ i iznosi $RMSE = 1.01$, $AvgErr = 0.8011$

Nedostaci CF metode

- ▶ teško je bilo što preporučiti novom korisniku

Nedostaci CF metode

- ▶ teško je bilo što preporučiti novom korisniku
- ▶ teško da će novi predmet biti bilo kome preporučen

Nedostaci CF metode

- ▶ teško je bilo što preporučiti novom korisniku
- ▶ teško da će novi predmet biti bilo kome preporučen
- ▶ rjetkost user vs. item matrice (rješava se singular-value dekompozicijom)

Nedostaci CF metode

- ▶ teško je bilo što preporučiti novom korisniku
- ▶ teško da će novi predmet biti bilo kome preporučen
- ▶ rjetkost user vs. item matrice (rješava se singular-value dekompozicijom)
- ▶ korisnici češće ocjenjuju draže predmete, a zatim one koje izrazito ne vole pa nemamo ocjene uobičajenih predmeta i često je ocjena ustvari mjera koliko netko želi ocijeniti određeni predmet, a ne kako ga je ocjenio

Nedostaci CF metode

- ▶ teško je bilo što preporučiti novom korisniku
- ▶ teško da će novi predmet biti bilo kome preporučen
- ▶ rjetkost user vs. item matrice (rješava se singular-value dekompozicijom)
- ▶ korisnici češće ocjenjuju draže predmete, a zatim one koje izrazito ne vole pa nemamo ocjene uobičajenih predmeta i često je ocjena ustvari mjera koliko netko želi ocjeniti određeni predmet, a ne kako ga je ocjenio
- ▶ ne uzima se u obzir kontekst u kojem je korisnik ocjenio određen predmet

Nedostaci CF metode

- ▶ teško je bilo što preporučiti novom korisniku
- ▶ teško da će novi predmet biti bilo kome preporučen
- ▶ rjetkost user vs. item matrice (rješava se singular-value dekompozicijom)
- ▶ korisnici češće ocjenjuju draže predmete, a zatim one koje izrazito ne vole pa nemamo ocjene uobičajenih predmeta i često je ocjena ustvari mjera koliko netko želi ocjeniti određeni predmet, a ne kako ga je ocjenio
- ▶ ne uzima se u obzir kontekst u kojem je korisnik ocjenio određen predmet
- ▶ bez obzira što se neki dijelovi mogu koristiti već preračunati, problem je i dalje veoma složen

Partitioniranje podataka

- ▶ ideja je podijeliti korisnike i/ili predmete u disjunktne skupove tako da se najsličniji nalaze u istoj grupi

Partitioniranje podataka

- ▶ ideja je podijeliti korisnike i/ili predmete u disjunktne skupove tako da se najsličniji nalaze u istoj grupi
- ▶ algoritmi koji se koriste su k -means clustering, Gibbs sampler i sl.

Particioniranje podataka

- ▶ ideja je podijeliti korisnike i/ili predmete u disjunktne skupove tako da se najsličniji nalaze u istoj grupi
- ▶ algoritmi koji se koriste su k -means clustering, Gibbs sampler i sl.
- ▶ ovakvim se pristupom daleko smanjuje složenost problema

Particioniranje podataka

- ▶ ideja je podijeliti korisnike i/ili predmete u disjunktne skupove tako da se najsličniji nalaze u istoj grupi
- ▶ algoritmi koji se koriste su k -means clustering, Gibbs sampler i sl.
- ▶ ovakvim se pristupom daleko smanjuje složenost problema
- ▶ jednostavnije verzije koje koriste particioniranje jednostavno svim korisnicima iz istog skupa dodijeljuju iste preporuke

Content-boosted Collaborative Filtering (CBCF)

- ▶ CBCF je hibridna metoda kojom se poboljšavaju rezultati CF algoritma kombiniranjem sa CB filteringom

Content-boosted Collaborative Filtering (CBCF)

- ▶ CBCF je hibridna metoda kojom se poboljšavaju rezultati CF algoritma kombiniranjem sa CB filteringom
- ▶ efikasno rješava problem rjetke matrice

Content-boosted Collaborative Filtering (CBCF)

- ▶ CBCF je hibridna metoda kojom se poboljšavaju rezultati CF algoritma kombiniranjem sa CB filteringom
- ▶ efikasno rješava problem rjetke matrice
- ▶ Algoritam za preporuku n predmeta korisniku u :

Content-boosted Collaborative Filtering (CBCF)

- ▶ CBCF je hibridna metoda kojom se poboljšavaju rezultati CF algoritma kombiniranjem sa CB filteringom
- ▶ efikasno rješava problem rjetke matrice
- ▶ Algoritam za preporuku n predmeta korisniku u :
 1. matricu M (user vs. item) nadopuniti sa vrijednostima gdje nisu prisutne pomoću neke CB metode

Content-boosted Collaborative Filtering (CBCF)

- ▶ CBCF je hibridna metoda kojom se poboljšavaju rezultati CF algoritma kombiniranjem sa CB filteringom
- ▶ efikasno rješava problem rjetke matrice
- ▶ Algoritam za preporuku n predmeta korisniku u :
 1. matricu M (user vs. item) nadopuniti sa vrijednostima gdje nisu prisutne pomoću neke CB metode
 2. pronaći nekoliko korisnika sličnih korisniku u pomoću nadopunjene matrice M (npr. Pearsonovim koeficijentom)

Content-boosted Collaborative Filtering (CBCF)

- ▶ CBCF je hibridna metoda kojom se poboljšavaju rezultati CF algoritma kombiniranjem sa CB filteringom
- ▶ efikasno rješava problem rjetke matrice
- ▶ Algoritam za preporuku n predmeta korisniku u :
 1. matricu M (user vs. item) nadopuniti sa vrijednostima gdje nisu prisutne pomoću neke CB metode
 2. pronaći nekoliko korisnika sličnih korisniku u pomoću nadopunjene matrice M (npr. Pearsonovim koeficijentom)
 3. izračunati ocjenu za korisnika u pomoću ocjena dobivenih sličnih korisnika i njihovih koeficijenata sličnosti promatranih kao težine

Tagommenders

- ▶ ideja je da je osobi koja gradi sustav za preporuke i računalu teško povezati sadržaj predmeta i korisnika

Tagommenders

- ▶ ideja je da je osobi koja gradi sustav za preporuke i računalu teško povezati sadržaj predmeta i korisnika
- ▶ svakom korisniku se daje mogućnost da bilo kojem predmetu pridružuje oznake (tagove)

Tagommenders

- ▶ ideja je da je osobi koja gradi sustav za preporuke i računalu teško povezati sadržaj predmeta i korisnika
- ▶ svakom korisniku se daje mogućnost da bilo kojem predmetu pridružuje oznake (tagove)
- ▶ korisniku se preporučaju predmeti iz skupova vezanim uz tagove prema kojima ima najveću povezanost

Tagommenders

- ▶ ideja je da je osobi koja gradi sustav za preporuke i računalu teško povezati sadržaj predmeta i korisnika
- ▶ svakom korisniku se daje mogućnost da bilo kojem predmetu pridružuje oznake (tagove)
- ▶ korisniku se preporučaju predmeti iz skupova vezanim uz tagove prema kojima ima najveću povezanost
- ▶ svaki puta kada korisnik odabere neki predmet, povećava mu se povezanost sa svim tagovima vezanim uz taj predmet

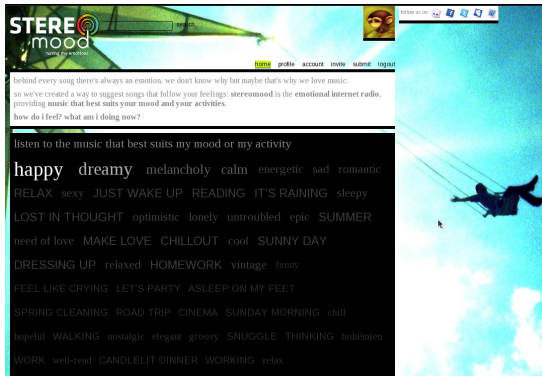
Tagommenders

- ▶ ideja je da je osobi koja gradi sustav za preporuke i računalu teško povezati sadržaj predmeta i korisnika
- ▶ svakom korisniku se daje mogućnost da bilo kojem predmetu pridružuje oznake (tagove)
- ▶ korisniku se preporučaju predmeti iz skupova vezanim uz tagove prema kojima ima najveću povezanost
- ▶ svaki puta kada korisnik odabere neki predmet, povećava mu se povezanost sa svim tagovima vezanim uz taj predmet
- ▶ redoslijed predmeta (važnost) unutar taga se tada može određivati nekim drugim sustavom za preporučivanje

Tagommenders

- ▶ ideja je da je osobi koja gradi sustav za preporuke i računalu teško povezati sadržaj predmeta i korisnika
- ▶ svakom korisniku se daje mogućnost da bilo kojem predmetu pridružuje oznake (tagove)
- ▶ korisniku se preporučaju predmeti iz skupova vezanim uz tagove prema kojima ima najveću povezanost
- ▶ svaki puta kada korisnik odabere neki predmet, povećava mu se povezanost sa svim tagovima vezanim uz taj predmet
- ▶ redoslijed predmeta (važnost) unutar taga se tada može određivati nekim drugim sustavom za preporučivanje
- ▶ primjena u blog portalima, sustavima za multimediju i sl.

stereomood.com



Slika: stereomood.com kao primjer tagommendera

Literatura

- ▶ G. Adomavicius i A. Tuzhilin, „Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions “, IEEE, 2005
- ▶ C. Basu, H. Hirsh i W. Cohen, „Recommendation as Classification: Using Social and Content-Based Information in Recommendation “
- ▶ R. M. Bell, Y. Koren i C. Volinsky, „The BellKor solution to the Netflix Prize “, AT&T Labs, 2007
- ▶ J. S. Breese, D. Heckerman i C. Kadie, „Empirical Analysis of Predictive Algorithms for Collaborative Filtering “, Microsoft Research, 1998

- ▶ S. Hammarling, „The Singular Value Decomposition in Multivariate Statistics “, Manchester Institute for Mathematical Sciences, 1985
- ▶ J. L. Herlocker, J.h A. Konstan i J. Riedl, „Explaining Collaborative Filtering Recommendations “, University of Minnesota
- ▶ J. Konstan, J. Riedl i A. Jameson, „AI Techniques for Personalized Recommendation “, IJCAI, 2003
- ▶ N. Littlestone i M. K. Warmuth, „The Weighted Majority Algorithm “, UCSC, 1992
- ▶ P. Melville, R. J. Mooney i R. Nagarajan, „Content-Boosted Collaborative Filtering for Improved Recommendations “, University of Texas, 2002

- ▶ B. Sarwar, G. Karypis, J. Konstan i J. Riedl, „Item-Based Collaborative Filtering Recommendation Algorithms “, University of Minnesota
- ▶ T. Segaran, „Programming Collective Intelligence“, O'Reilly, 2007
- ▶ S. Sen, J. Vig i J. Riedl, „Tagommenders: Connecting Users to Items through Tags “
- ▶ L. H. Ungar i D. P. Foster, „Clustering Methods for Collaborative Filtering “, CIS, 1998
- ▶ S. Vucetic i Z. Obradovic, „Collaborative Filtering Using a Regression-Based Approach “, Center for Information Science and Technology, Temple University, Philadelphia, PA, USA, 2004