

# Priporočilni sistemi

Tilen Venko (63140280)

4. januar 2017

## 1 Uvod

Cilj naloge je bil, da implementiramo tri metode za priporočilne sisteme. In sicer preprost priporočilni sistem, tak, ki meri podobnost s kosinusno raydaljo in priporočilni sistem, ki temelji na latentnem modelu.

## 2 Podatki

Podatke smo dobili že pripravljene, razdeljene na učno in testno množico iz spletne strani Movielens 100k. Učna množica zavzema 943 uporabnikov in 1662 različnih filmov.

## 3 Metode

### 3.1 Preprost priporočilni sistem

Ta metoda deluje tako, da izračunamo povprečne ocene vseh filmov, kar sem storil v metodi *average\_movie()* in povprečje vseh ocen uporabnikov, da ugotovimo kako strog ocenjevalec je uporabnik. To sem implementiral v funkciji *average\_user()*. Nato pa iteriramo čez vse filme in uporabnike ter dopolnjujemo ocene za tiste filme, katerim uporabnik ni dal ocene, po formuli  $(\text{povprečje\_uporabnik} + \text{povprečje\_film}) / 2$ , kar sem implementiral v funkciji *user\_missing()*.

### 3.2 Priporočilni sistem, ki temelji na podobnosti med filmi

Ta nači deluje tako, da za vsak film, kateremu uporabnik ni podal ocene, vzamemo njegov vektor vseh ocen in izračunamo kosinusne razdalje med vektorjem ocen tega filma in ocenami vseh ostalih filmov, ki jih je ta uporabnik ocenil, uteženo z oceno uporabnika, na koncu pa seštevek uteženih raydalj delimo s seštevkom neuteženih razdalj. V funkciji *cos()* sem implementiral izračun kosinusne razdalje, v funkciji *user\_missing()* pa vso ostalo logiko, da se kosinusne razdalje izračunajo za vse uporabnike in vse filme, katerim niso podali ocene.

### 3.3 Priporočilni sistem, ki temelji na latentnem modelu

Pri tej metodi si najprej zgeneriramo dve matriki P in Q, ki imata naključne vrednosti med -0.01 in 0.01. P je dimenzij [št. uporabnikov][k], Q pa [k][št. filmov], k si določimo samo privzeto

je 1. Naš cilj je da bi lahko s množenjem  $P$  in  $Q$  matrike določili oceno katerega koli filma v matriki tipa [št. uporabnikov][št. filmov]. to dosežemo tako da izračunamo ciljno funkcijo  $e_{ui}$ , nato pa  $P$  in  $Q$  matriko popravimo glede na gradientni spust parcialnega odvoda  $e_{ui}$ .

## 4 Rezultati

S preprostim priporočilnim sistemom dobim rmse enak 0.1295. Za priporočilne sisteme s podobnostmi med filmi nisem uspel izračunati rmseja, saj se je program izvajal predolgo časa. Priporočilni sistem na podlagi latentnega modela nam da pri  $k=1$  rmse 0.01007, za  $k=2$  0.00993,  $k=5$  0.00987 in za  $k=10$  je rmse enak 0.00987. S tem lahko vidimo, da stopnja razcepa sicer vpliva na rezultate vendar ne drastično in da se od stopnje  $k=5$  praktično ne pozna več, saj ni sprememb na petih decimalkah. Naša želja pa je tudi čim manjši  $k$ , saj je tako računski čas manjši.

## 5 Izjava o izdelavi domače naloge

Domačo nalogo in pripadajoče programe sem izdelal sam.