Nenadzorovano modeliranje

Primož Pečar (63150213)

10. april 2017

1 Uvod

V drugi domači nalogi sem se ukvarjal z iskanjem osamelcev in gručenjem sorodnih primerov na podatkovni množici Movie Lens.

2 Podatki

Večina podatkov je bilo kopiranih iz prve domače naloge, saj sem tam imel že vse pripravljeno. Kar je na novo je xlsx datoteka, ki je bila posebaj pripravljena za uporabo z Orange3.

3 Metode

Prva naloga in podnaloge so popolnoma rešene v Pythonu, pri drugi nalogi pa so podatki pripravljeni za uporabo z Orange3, saj sem tam lahko hitreje testiral različne metode hierarhičnega gručenja, potreben je bil le pravilen format xlsx datoteke.

4 Naloge podrobno

V tem odseku bom opisal vsako od nalog bolj podrobno. Govoril bom o ugotovitvah, do katerih sem prišel med nalogo in zanimivostih, ki sem jih srečal.

4.1 Naloga 1

V prvi nalogi je bilo potrebno ugotoviti, ocene katerih filmov so najbolj enotne. Zanimalo nas je, če obstajajo filmi, ki so dobili podobne ocene med vsemi uporabniki ali pa filmi z razpršenimi ocenami.

4.1.1 Ustrezna naključna spremenljivka

Potrebno je bilo ugotoviti, s katerimi podatki si lahko pomagamo pri ugotavljanju povezav med filmi. Spremenljivka, ki sem jo uporabil so bili ratingi. Za vsak film posebaj sem izračunal standardni odklon in nato vse filme predstavil v histogramu.

4.1.2 Porazdelitev

Porazdelitev se je spreminjala na podlagi parametrov filmov. Kot v prvi nalogi sem upošteval filme, ki imajo 30 ali več ocen, tako se znebim filmov, ki imajo po eno ali celo nič ocen in robnih primerov, kot so naprimer neznani filmi. Porazdelitev, ki sem jo dobil je bila normalna. Saj je 68% primerov v prvem odklonu (na sredini), potem 95% v drugem odklonu in 99.7% v tretjem. V primeru, da sem upošteval vse filme, pa sem dobil Beta porazdelitev.

Slika 1: Histogram standardnih odklonov filmov.

4.1.3 Ocena parametrov

Za oceno parametrov sem uporabil sledeče formule:

$\mu = E[X_i] = \bar{X}$ (povprečje vzorca)

$\sigma^2 = \frac{n-1}{n} E[(X_i - \bar{X})^2] = \frac{n-1}{n} var[x]$ (popravljena varianca vzorca)

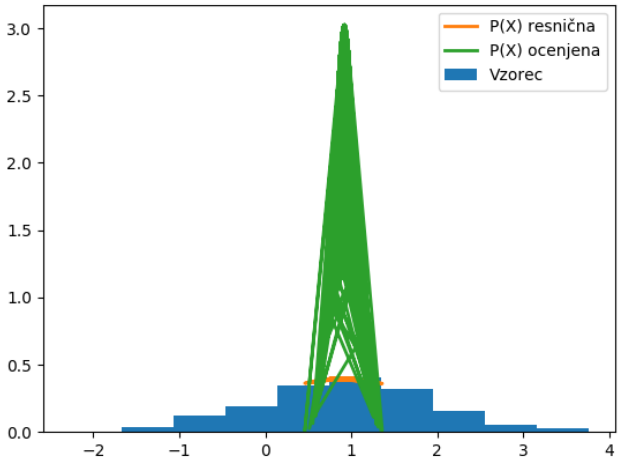
V kodo se prevede sledeče, kjer je mySTD seznam vseh standardnih odklonov filmov:

```
mu_fit = np.mean(mySTD)
n = len(mySTD)
sigma2_fit = (n-1)/n * np.var(mySTD)
print("##########OCENA###########")
print(mu_fit,sigma2_fit)
```

Dobil sem kar dobro oceno, predvidevam zaradi tega ker sem uporabil le filme, ki imajo 30 ali več ocen. (Prva številka je povprečje vzorca, druga pa popravljena varianca vzorca.)
0.919678380176 0.017390364507

4.1.4 Izbira porazdelitve

Kot omenjeno, porazdelitev je zelo verjetno normalna, saj ima zvonasto obliko, po kateri je znana. Dejstvo, da izvajamo test na celotni populaciji, tudi kaže na to da je normalna. Skratka normalna porazdelitev se uporablja takrat ko je porazdelitev podatkov normalno(pričakovano). Na vajah pa smo spoznali še studentovo, beta, normalno oz. gaussovo porazdelitev. Razvedrilo za bralca, poskušal sem izrisati ocenjeno in resnično porazdelitev, vendar je rezultat bil sledeč.



Slika 2: Skoraj normalna porazdelitev.

4.1.5 Zgornjih 5%

Tukaj pa sem gledal primere, ki so v tretjem odklonu, torej vse filme, ki so manjši od 0.6 in večji od 1.2. Zanimajo nas filmi od 1.2 naprej, torej filmi, z največjim standardnim odklonom. Izpis vseh je v Python kodi, tukaj sem izbral 7 filmov iz vsake skupine.

Tabela 1: Padajočih 5%

Ime filma	Standardni odklon
Roger & Me (1989)	0.582847222138
Blood Diamond (2006)	0.511732330507
Deliverance (1972)	0.596212000886
Harry Potter and the Deathly Hallows: Part 2 (2011)	0.596449993843
Scent of a Woman (1992)	0.553022000768
Muppet Movie	0.459813626841
Delicatessen (1991)	0.596284794

Tabela 2: Naraščajočih 5%

Ime filma	Standardni odklon
Mad Max: Fury Road (2015)	1.3426874444
Star Wars: Episode II - Attack of the Clones (2002)	1.25118920231
Space Jam (1996)	1.22784363825
Saw (2004)	1.36009641532
Blair Witch Project	1.31123664377
Showgirls (1995)	1.259737582
Scary Movie (2000)	1.29193712462

4.2 Naloga 2

V drugi nalogi je bilo potrebno poiskati 100 najbolj gledanih filmov. Vzel sem vse filme in jih sortiral padajoče po ogledih in vzel vrhnjih 100. Potrebno je bilo narediti vektor, po imenu filma, atributi so bili pa vsi uporabniki. Tako smo dobili matriko, katera je imela ratinge vseh uporabnikov za vsak film. Če uporabnik ni ocenil določenega filma, sem na to mesto zapisal povprečno oceno za ta film.

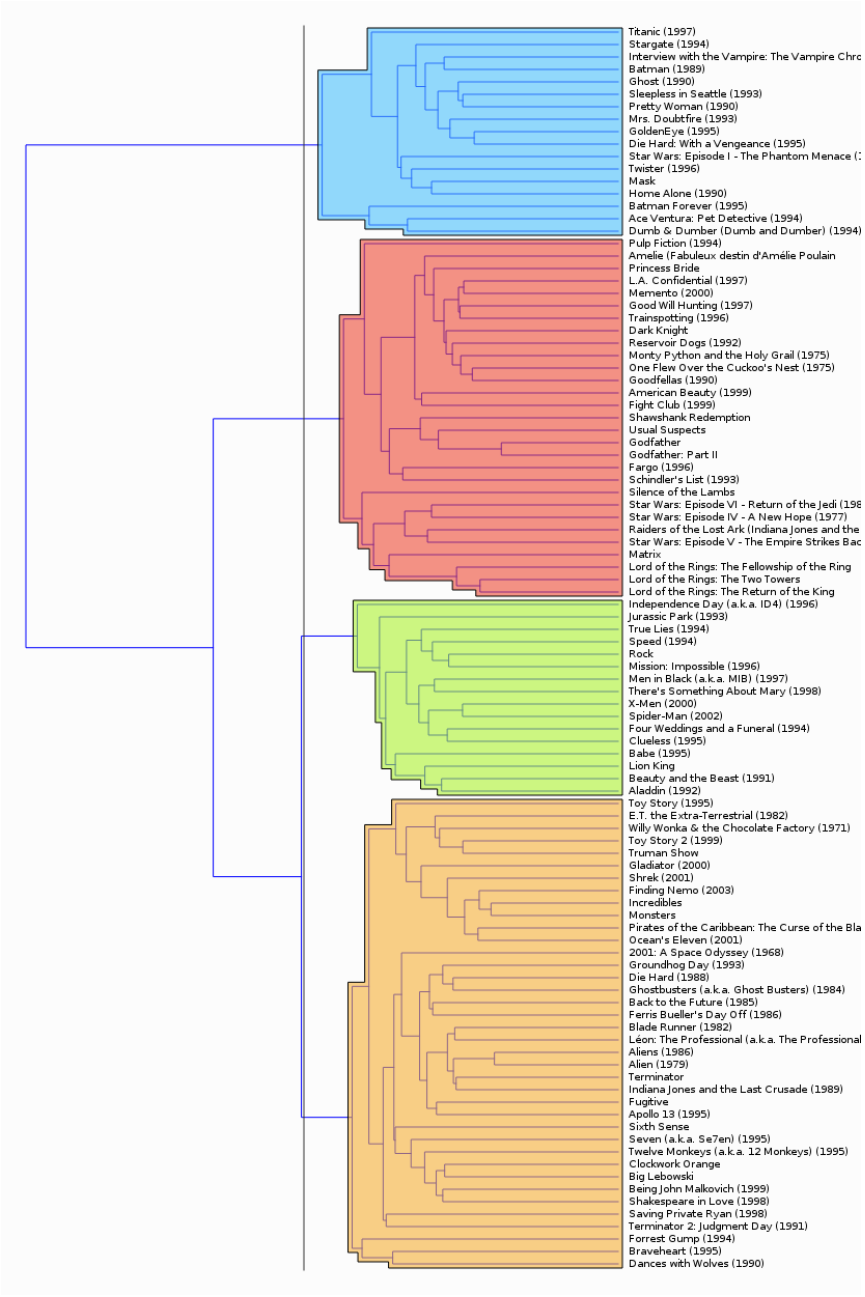
4.2.1 Algoritem in mere podobnosti

Poskusil sem zelo veliko stvari, ker sem delal v Orange3 je bilo vse bolj ali manj vizualno poskušanje. Nekako v grobem se filmi najbolje delijo na 3 podskupine. To sem dobil s pomočjo k-means algoritma in Silhouette plot-a. Potem sem poskušal še različne metode za razdaljo, vendar se je izkazalo da je Manhattenska razdalja in average linkage prineseta najboljše rezultate. Bolj podrobno bo opisano v nadalnjih odstavkih.

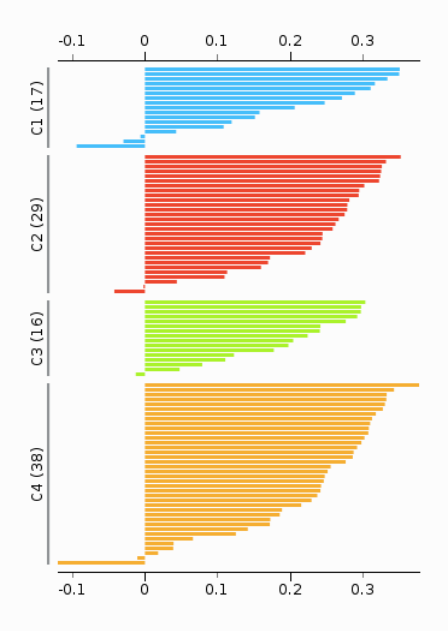
4.2.2 Rezultati za različne mere

Poskusil sem kar nekaj kombinacij, tukaj jih bom predstavil.

Dokaj dober rezultat sem dobil z evklidsko mero za razdaljo in complete linkage.



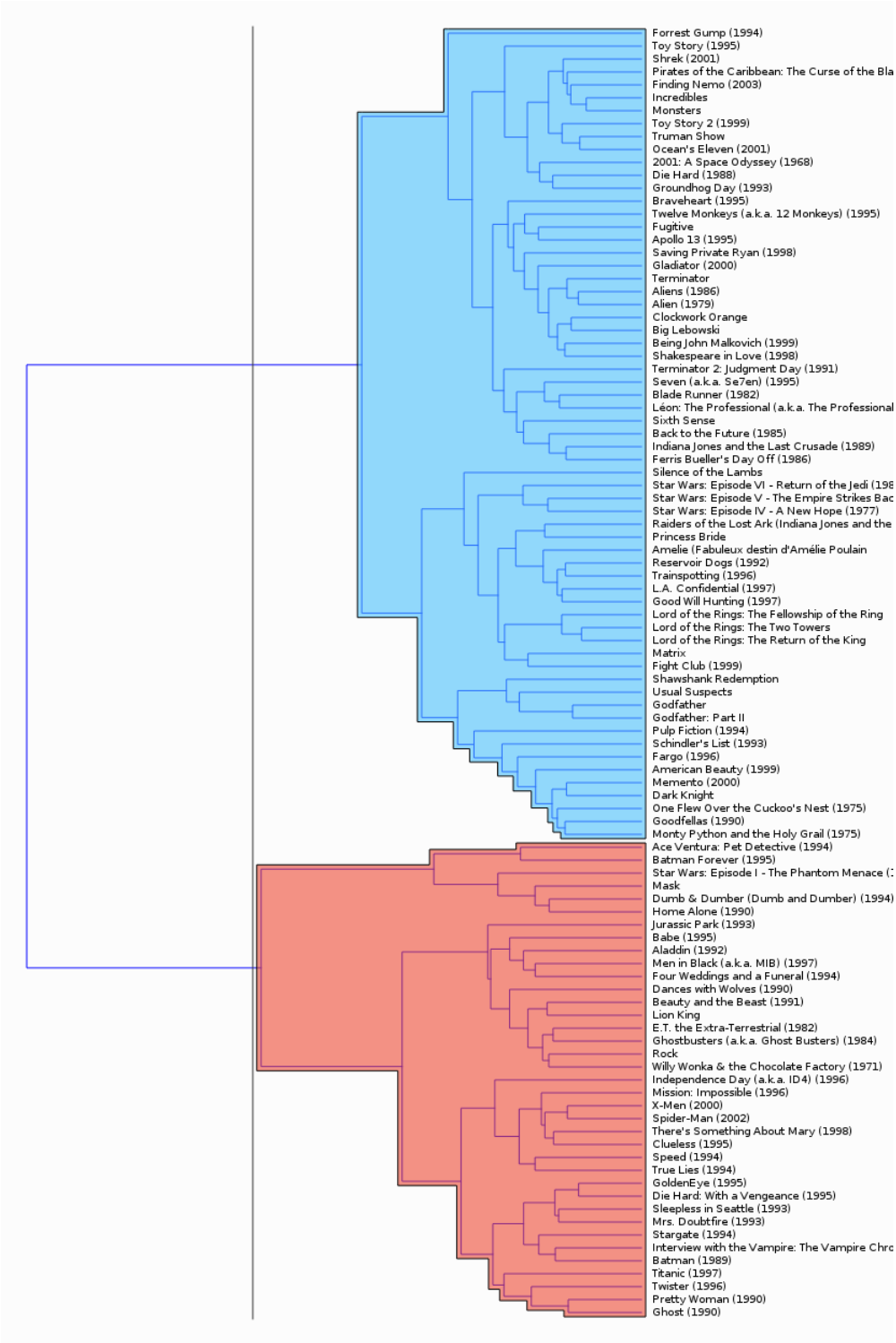
Slika 3: Dendrogram za evklidsko in complete.



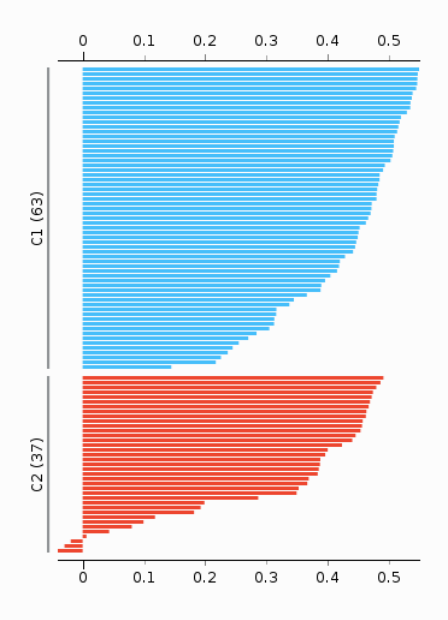
Slika 4: Silhuette plot za ekv. in compl.

Dobro je videti, da je razdalja med različnimi clusterji zelo velika, vendar tesno povezani pa ševedno držijo skupaj. Če bi imel le 2 skupine bi bili rezultati še boljši, vendar bi v clusterju z največ filmi imeli kar dosti primerov, ki ne spadajo v isto skupino.

Edina kombinacija še vredna omembe pa je Manhattenska razdalja pri average ali pa compleate linkage-u. Kar se je spremenilo pri average in complete so robni primeri pri rdečem so šli v modrega in obratno. Tukaj se izkaže, da če imamo dve skupine imamo najboljše rezultate.

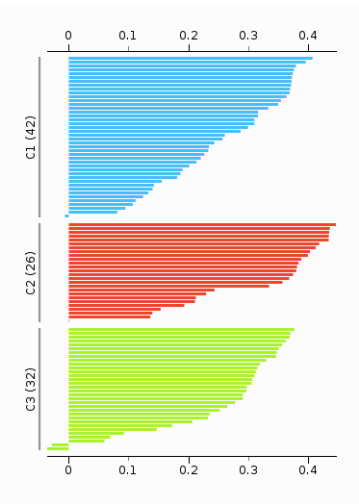


Slika 5: Dendrogram za manhattensko in complete.



Slika 6: Silhuette za manhattensko in complete.

Kot primer prilagam skico silhuettnega diagrama, ki ga izriše knn, pri katerem imamo silhuette score 0.122 (za 3) in 0.252 (za 2).



Slika 7: Silhuette za evklidsko in complete.

4.2.3 Koliko skupin je med izbranimi, ocene za razvrščanje

Kot prej omenjeno, sem čez podatke pognal knn, kateri je šel čez 20-krat po 300 iteracij. Izkaže se, da je skupin med 2 in 4, če bi moral izbrati nek obseg. Od 4 naprej pa se razbijejo na majhne podskupine, katere imajo veliko primerov, ki ne spadajo v ta cluster. Če bi želeli najbolj optimalno, bi imeli 3 skupine. Od tu naprej pade izven clusterjev prb. 13 in naprej filmov (ti se večajo s št. skupin).

Mera, ki pa se tudi uporablja je skupna deljena informacija, problem pri tej meri je to, da moramo vedeti kam spada določen film.

4.2.4 Ustrezne vizualizacije

Glej figure 3 in 4 za evklidsko razdaljo. Za manhattensko pa 5 in 6. Silhuette, ki ga izriše knn pa je 7.

4.2.5 Smiselnost rezultatov

Pričakoval sem, da se rezultati razdelijo v gruče na podlagi žanra, torej ocene komedij bodo podobne drugim komedijam medtem, ko bodo ocene dram podobne drugim dramam. Vendar ker smo delali na ocenah, katere niso imele nobene povezave z žanri je pričakovano, da jih bo grupiral drugače, vendar se je izkazalo, da so bila pričakovanja izpolnjena. Kar sem še opazil je to, da so bistvene razlike med zelo dobrimi filmi, in nadpovprečnimi filmi, saj so Shawshank redemption, Godfather, Pulp fiction vedno v isti gruči, vendar se potem povežejo v večjo gručo. Na koncu vidimo, da so filmi kot so Starwars ali pa razni otroški filmi (oz PG-13, kot so Shrek, Finding Nemo, Incredibles), vedno tesno povezani, kar pomeni, da filmi iz enakih žanrov padejo v iste gruče, kar potrdi vprašanje, ki sem si ga zastavil na začetku naloge.

5 Izjava o izdelavi domače naloge

Domačo nalogo in pripadajoče programe sem izdelal sam.

Priloge

A Slike in programska koda

Slike se nahajajo v mapi slike, programska koda v source, LaTex datoteke pa v tex.