UNIVERZA V LJUBLJANI

FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Miha Grčar

Empirična evalvacija algoritmov za izbiranje s sodelovanjem

UNIVERZITETNO DIPLOMSKO DELO

Mentor: akad. prof. dr. Ivan Bratko

Somentorica: doc. dr. Dunja Mladenič

Ljubljana, oktober 2006

Zahvala

Zahvaljujem se akad. prof. dr. Ivanu Bratku za mentorstvo ter doc. dr. Dunji Mladenič za somentorstvo pri izdelavi diplomske naloge. Njuni predlogi so občutno dvignili kakovostni nivo tega dela.

Zahvaljujem se Inštitutu Jožef Stefan za dodeljeno štipendijo.

Zahvaljujem se sodelavcem z Odseka za tehnologije znanja na Inštitutu Jožef Stefan, ki ga vodi prof. dr. Nada Lavrač, ker so me vzeli na krov ter mi priskrbeli sodobno strojno opremo in pozitivno delovno okolje. Med njimi bi želel izpostaviti predvsem Marka Grobelnika, ki moje računalniško udejstvovanje spremlja in usmerja že od ranih začetkov, in skupino ožjih sodelavcev, s katerimi sodelujemo v raziskovalno-razvojnih projektih.

Zahvaljujem se študijskim kolegom, ki so mi pomagali pri pripravah na izpite, še posebej Mojci Oman za številne zapiske in napotke.

Nadalje se iz srca zahvaljujem Tanji Brajnik, ki mi je pomagala z lekturo in nasveti glede jezika ter mi ves čas pisanja tega dela stala ob strani.

Nenazadnje gre zahvala tudi mojim staršem, ki so me brezpogojno podpirali tekom celotnega študija.

Slovarček izrazov

Slovenski izraz	**Angleški izraz**
algoritem za napovedovanje naključne ocene	random predictor
algoritem za napovedovanje povprečne ocene	popularity predictor
algoritmi na osnovi modela	model-based methods
algoritmi na osnovi pomnjenja	memory-based methods
dani podatki (v nasprotju s prikritimi podatki)	given data (v nasprotju s hidden data)
eksplicitne ocene	explicit ratings
evalvacijska metrika	evaluation metric
evalvacijski protokol	evaluation protocol
glava proti repu	head versus tail
implicitne ocene	implicit ratings
indikator napake	error bar
informacijsko poizvedovanje	information retrieval
iskanje najbolj verjetnega modela	expectation maximization, EM
iskanje prikrite semantike	latent semantic analysis, LSA
izbiranje s sodelovanjem	collaborative filtering
k najbližjih sosedov	*k* nearest neighbors, *k*-NN
klasifikacijski problem	classification problem
kosinusna podobnost	cosine similarity
linearno jedro	linear kernel
matrika ocen	rating matrix
mešani model	mixture model
metrika F_1	F_1 metric
metrika mejnega obiska	half-life utility metric
metrika preko uporabnikov	per-user metric
metrika preko vseh podatkov	overall metric
metrika ROC	ROC metric
nadzorovano učenje	supervised learning
naključno vzorčenje	random sampling
natančnost	precision
neuravnotežen podatkovni nabor	unbalanced dataset
neuspele zahteve	failed requests
(normalizirana) povprečna absolutna napaka	(normalized) mean absolute error, (N)MAE
parameter mejnega obiska	half-life parameter
Pearsonov korelacijski koeficient	Pearson correlation coefficient
Pearsonova evalvacijska metrika	Pearson evaluation metric
piškotek	cookie
podatkovni nabor	dataset
posredovalni strežnik	proxy server
povprečna natančnost	average precision
požarni zid	firewall
predmetno izbiranje s sodelovanjem	item-based collaborative filtering
predpomnilnik spletnega brskalnika	(Web) browser cache
prehod (v kontekstu omrežij)	gateway
preusmeritve	redirections
priklic	recall
prikrita naključna spremenljivka	latent random variable
prikriti podatki	hidden data

Slovenski izraz	Angleški izraz
privzeta ocena	default rating
problem nadgradljivosti	scalability problem
problem redkosti matrike	matrix sparseness problem
profiliranje uporabikov	user profiling
razgradnja na singularne vrednosti	singular value decomposition, SVD
razporejanje bremena	load balancing
razpršeni grafikon	scatter plot
razsežnost	dimension
redek vektor	sparse vector
redka matrika	sparse matrix
sistemi za priporočanje	recommender systems
soseska uporabnikov	user neighborhood
splet	(World Wide) Web, WWW
spletna personalizacija	Web personalization
(spletni) strežniški dnevnik	Web-server log
srednja povprečna natančnost	mean average precision, MAP
SVM-klasifikator	SVM classifier
SVM-regresija	SVM regression
testni primeri, testni podatkovni nabor	test examples, test set
razvrščanje v skupine, razvrščanje	clustering
učni primeri, podatkovni nabor	training examples, training set
umerjena kosinusna podobnost	adjusted cosine similarity
uporabniški profil	user profile
večrazredna klasifikacija	multiclass classification
verjetnostno iskanje skrite semantike	probabilistic latent semantic analysis, pLSA
vsak proti vsakemu	one versus one
vsak proti vsem	one versus all
vsebinsko izbiranje	content-based filtering
zahteve anonimnih uporabnikov	anonymous requests
zahteve tipa POST	POST requests
značilke	features

Izjava

Izjavljam, da sem diplomsko delo izdelal samostojno pod vodstvom mentorja akad. prof. dr. Ivana Bratka in somentorice doc. dr. Dunje Mladenič. Izkazano pomoč drugih sodelavcev sem v celoti navedel v zahvali.

Ljubljana, oktober 2006

Miha Grčar

Kazalo

Povzetek

To diplomsko delo govori o izbiranju s sodelovanjem (*collaborative filtering*), metodi za priporočanje vsebin, ki je v zadnjih letih deležna precejšnje pozornosti tako z akademske strani kot s strani industrije. Poglavitna razlika med izbiranjem s sodelovanjem in vsebinskim izbiranjem (*content-based filtering*) je, da je izbiranje s sodelovanjem neodvisno od vsebine predmetov in ga je torej mogoče aplicirati tudi na predmete, katerih vsebina ni znana ali pa jo je težko formulirati.

Namen te diplomske naloge je:

- predstaviti algoritme za izbiranje s sodelovanjem, ki jih je mogoče zaslediti v literaturi,
- implementirati platformo za evalvacijo algoritmov za izbiranje s sodelovanjem,
- primerjati algoritme za izbiranje s sodelovanjem z vidika tipičnih evalvacijskih metrik,
- preučiti vpliv redkosti podatkov na kakovost algoritmov za izbiranje s sodelovanjem,
- kritično obravnavati evalvacijske metrike in
- predlagati nove evalvacijske metrike, ki so v primerjavi s standardnimi metrikami primernejše za ocenjevanje kakovosti algoritmov za izbiranje s sodelovanjem.

Diplomsko nalogo sem izdelal v sledečih korakih:

- Zbral in preučil sem literaturo na temo izbiranja s sodelovanjem in jo povzel v prvem delu naloge (poglavje 2).
- Poiskal sem tri domene in pridobil pripadajoče podatkovne nabore. Domene in podatkovne nabore sem opisal v poglavju 3.2. Dva podatkovna nabora (EachMovie in Jester) sta javno dostopna na spletu, tretjega pa sem izluščil iz relativno velike količine strežniških dnevnikov velikega britanskega podjetja. V ta namen sem v programskem jeziku C++ implementiral program za obdelavo strežniških dnevnikov (približno 2.000 vrstic programske kode).
- V programskem jeziku C++ sem implementiral evalvacijsko platformo (približno 4.000 vrstic programske kode); opisal sem jo v poglavju 3.3.
- Algoritme sem ocenil z vidika štirih standardnih evalvacijskih metrik (opisal sem jih v poglavju 4) in v poglavju 5 izpostavil tisto njihovo lastnost, ki po mojem mnenju predstavlja poglavitno pomanjkljivost pri evalvaciji algoritmov za izbiranje s sodelovanjem – enakovredno obravnavanje vsakega uporabnika ne glede na njegovo odstopanje od povprečja.
- Predlagal sem spremembo evalvacijskih metrik (poglavje 6) z obteževanjem uporabnikov glede na njihovo odstopanje od povprečja in s tem ponudil svež pogled na izbiranje s sodelovanjem – pogled, v katerem postane obravnava ekscentričnih uporabnikov sestavni del vsakega dobro zasnovanega algoritma za izbiranje s sodelovanjem.
- Pri končni evalvaciji izbranih algoritmov sem uporabil tako standardne kot modificirane evalvacijske metrike. V poglavju 7 sem opisal eksperimentalno nastavitev in podal svoj pogled na razlike v rezultatih, ki so nastale kot posledica prehoda na modificirane inačice evalvacijskih metrik.

Pred zaključno evalvacijo sem postavil hipotezo, da algoritmi, ki dosežejo najboljši/najslabši rezultat glede na določeno standardno metriko, ne dosežejo nujno najboljšega/najslabšega rezultata glede na pripadajočo modificirano metriko. Evalvacija je v nekaterih primerih potrdila mojo hipotezo.

1. Uvod

Količina spletnih vsebin iz dneva v dan naglo narašča, prav tako narašča število in raznolikost produktov in storitev, ki jih je moč kupiti preko spletnih trgovin in podjetij. S tem pa pridobivajo na veljavi sistemi, ki omogočajo personaliziran dostop do informacij. Na nekaterih področjih računalniških znanosti – predvsem na področjih spletne personalizacije (*Web personalization*), profiliranja uporabnikov (*user profiling*) in sistemov za priporočanje (*recommender systems*) – so se pojavile številne metode za realizacijo takih sistemov. V tem delu predstavljam izbiranje s sodelovanjem (*collaborative filtering*), eno od metod za izbiranje vsebin, ki postaja nepogrešljiv člen v sistemih za priporočanje. Izbiranje s sodelovanjem je v akademskem svetu pomembna tematika, saj je prenos razvitih tehnologij v industrijo več kot očitna.

Tvorjenje kakovostnih priporočil je velikega pomena v spletnih trgovinah, knjigarnah in knjižnicah. Dobro znana spletna knjigarna Amazon.com na primer s pridom uporablja izbiranje s sodelovanjem za priporočila v kategorijah "Priporočamo vam" ("*Recommended for you*") in "Stranke, ki so kupile ta artikel, so kupile tudi" ("*Customers who bought this item also bought*"). Prav tako izbiranje s sodelovanjem uporabljajo nekatere druge spletne knjigarne (kot na primer AlexLit.com in Barnes & Noble[1]), ponudniki novic (kot na primer Findory.com), trgovine (kot na primer eBay.com in Half.com), izposojevalnice in prodajalne filmov in iger (kot na primer Hollywood Video[2], Netflix.com in iTiVi.si), radijske postaje in sistemi za priporočanje glasbe (kot na primer Last.fm, Musicmatch.com in radiolibre.ca), posredniki zasebnih stikov (kot na primer Reciprodate.com) in ponudniki samih sistemov za priporočanje (kot na primer Loomia.com, NetPerceptions.com, Sourcelight.com in StoryCode.com). Izbiranje s sodelovanjem je vključeno celo v nekatere digitalne videorekorderje (to vrsto videorekorderjev izdeluje na primer podjetje TiVo.com). Obstaja tudi veliko nekomercialnih sistemov. Največ jih priporoča filme (kot na primer like-i-like.org, Movies.Monigo.com, Everyone's a Critic[3], FilmAffinity.com in MovieLens.org) in glasbo (kot na primer Gnomoradio.org, Indy.tv, iRATEradio.com, Musicmobs.com in MyStrands.com), zasledimo pa tudi sisteme za priporočanje člankov, spletnih strani, knjig, šal, računalniških iger, restavracij, piva in tako naprej.

Izbiranje s sodelovanjem se opira na predpostavko, da imajo podobni si uporabniki podobno mnenje o istih produktih, dokumentih, skratka – neke vrste predmetih. Z drugimi besedami – če algoritem poišče uporabnike, ki so podobni opazovanemu uporabniku, in preuči njihova mnenja o določenem predmetu, lahko predvidi mnenje (tj. oceno) opazovanega uporabnika o opazovanem predmetu. Tako lahko tvori urejen seznam predmetov, ki so potencialno relevantni za opazovanega uporabnika. V splošnem se izbiranje s sodelovanjem ne ozira na obliko in vsebino predmetov in ga je torej mogoče aplicirati tudi na predmete, katerih vsebina ni znana ali pa jo je težko formulirati (slike, artikli v trgovini, glasba idr.). Izbiranje s sodelovanjem izpelje razmerja med predmeti neposredno iz zbirke profilov uporabnikov, ki do teh predmetov dostopajo ali pa na kakršenkoli drug način izražajo svoj odnos do njih.

[1] http://www.barnesandnoble.com
[2] http://www.hollywoodvideo.com
[3] http://www.everyonesacritic.net

Algoritmi za izbiranje s sodelovanjem se soočajo z dvema poglavitnima problemoma: problemom redkosti matrike ocen (*sparseness problem*) in problemom nadgradljivosti[4] (*scalability problem*). Prvi problem se pojavi takrat, ko so podatki preredki, da bi lahko iz njih gradili zanesljive modele ali tvorili smiselne soseske (glej poglavja 2.2.1, 2.3.1 in 3.1). Drugi problem se navezuje predvsem na realizacijo izbiranja s sodelovanjem z metodo *k* najbližjih sosedov (ta metoda je predstavljena v poglavju 2.2.1) – standardni algoritem kaj kmalu postane prepočasen za praktično uporabo, če število primerov in/ali značilk narašča. Kar nekaj raziskav se v zadnjem času dotika tudi vprašanja kakovosti metrik za evalvacijo izbiranja s sodelovanjem. Pojavlja se predvsem vprašanje, ali predlagane metrike zares zajamejo bistvo izbiranja s sodelovanjem in ali se algoritmi, ki so s strani predlaganih metrik ocenjeni najbolje, za najboljše izkažejo tudi v praksi [12][17]. Nenazadnje se raziskovalci sprašujejo tudi o kakovosti podatkov, ki so na voljo v realnih domenah (glej na primer [10]). Najbolj vprašljiva je kakovost podatkov, ki so zbrani na strani strežnika in zatorej ne vsebujejo eksplicitnih informacij o preferencah uporabnikov (glej poglavje 3.1). V tem diplomskem delu se dotikamo vseh omenjenih problemov in jih v okviru lastne implementacije evalvacijske platforme do neke mere tudi razrešujemo.

[4] Problem obravnave naraščajočega števila primerov in/ali značilk.

2. Opis problema izbiranja s sodelovanjem

2.1 Osnovni pojmi in členitve

Izbiranje s sodelovanjem primerja uporabnike glede na njihove uporabniške profile (*user profiles*), ki vsebujejo mnenja o predmetih iz neke zbirke. "Mnenje" v tem kontekstu ni nič drugega kot dodeljeno število točk oz. ocena po neki ocenjevalni lestvici (npr. šolska ocena z lestvice od 1 do 5). Zbirka uporabniških profilov ima običajno obliko relativno velike matrike ocen, $\mathbf{R} = [\ r_{u,i}\ ]$, v kateri vrstice predstavljajo uporabnike (le-te so torej uporabniški profili), stolpci pa predmete (glej sliko 1). Element matrike ocen $r_{u,i}$ predstavlja oceno, ki jo je uporabnik u dodelil predmetu i. Postopek izbiranja s sodelovanjem lahko v grobem razdelimo v dve fazi: fazo izgradnje modela iz matrike ocen in fazo napovedovanja ocen in priporočanja potencialno zanimivih predmetov. Prva faza je neodvisna od druge in lahko poteka nekaj ur, lahko pa celo več dni.

K problemu izbiranja s sodelovanjem lahko v osnovi pristopimo na dva načina. Prvi način predstavljajo algoritmi na osnovi pomnjenja (*memory-based methods*). Tipičen predstavnik te vrste je metoda k najbližjih sosedov (glej poglavje 2.2.1). Algoritmi, realizirani na podlagi te metode, preskočijo fazo izgradnje modela. Vedno, ko se od njih zahteva, da napovejo mnenje uporabnika o določenem predmetu, neposredno iz podatkov v matriki ocen sklepajo

	Predmet 1	Predmet 2	Predmet 3	Predmet 4	Predmet 5	Predmet 6	Predmet 7	Predmet 8	Predmet 9
Uporabnik A	4		1	2	4				
Uporabnik B	2		5	**4**		5			
Uporabnik C	5		1					5	
Uporabnik D	4			1		1			

Slika 1: Primer matrike ocen štirih uporabnikov za devet predmetov. Krepko natisnjena vrednost 4 v 2. vrstici na primer pomeni, da je uporabnik B ocenil predmet številka 4 z oceno 4. Ker vsi uporabniki niso ocenili vseh predmetov, je v matriki ocen mnogo manjkajočih vrednosti.

o uporabnikovi oceni predmeta. Drugi način predstavljajo algoritmi na osnovi modela (*model-based methods*). Ti iz matrike ocen zgradijo model, ki ga v fazi priporočanja uporabljajo za hitrejše generiranje priporočil in/ali generiranje boljših priporočil. Pri tem "generiranje priporočil" ni nič drugega kot napovedovanje uporabnikovih ocen za predmete,

do katerih uporabnik še ni dostopil – algoritem potem uporabniku lahko priporoči predmete, za katere je predvidel visoke ocene.

2.2 Izbiranje s sodelovanjem na osnovi pomnjenja

Najpogosteje uporabljen algoritem za izbiranje s sodelovanjem na osnovi pomnjenja je realiziran na podlagi metode *k* najbližjih sosedov (*k-nearest neighbors*, *k-NN*). Opisan je v poglavju 2.2.1. Temelji na podobnosti med dvema uporabnikoma, ki jo je mogoče izračunati na različne načine. Najpogosteje uporabljeni načini za računanje podobnosti so opisani v poglavju 2.2.2.

2.2.1 Metoda *k* najbližjih sosedov

Algoritem na podlagi *k* najbližjih sosedov (v nadaljevanju tudi "*k*-NN") najprej pošče skupino *k* uporabnikov, ki so od vseh najbolj podobni opazovanemu uporabniku, in formira sosesko (*neighborhood*). Nato izračuna povprečje njihovih ocen za predmet, ki ga opazovani uporabnik še ni ocenil. Ker niso vsi uporabniki iz soseske podobni opazovanemu uporabniku v enaki meri, povprečenje običajno obtežimo – za uteži uporabljamo stopnje podobnosti med opazovanim uporabnikom in uporabniki iz soseske. Na uporabniški profil (tj. vrstico matrike ocen) lahko gledamo kot na redek vektor števil. Podobnost med dvema uporabnikoma določimo s funkcijo, ki vzame dva redka vektorja, vrne pa vrednost (navadno med –1 in 1), ki nam pove, v kolikšni meri sta si ta dva redka vektorja podobna (običajno –1 pomeni popolno različnost, 1 pa popolno podobnost). Funkcije za računanje podobnosti si bomo ogledali nekoliko kasneje, zaenkrat privzemimo, da tako funkcijo, $\mathrm{sim}(\mathbf{a}, \mathbf{b}) \in [-1, 1]$, že imamo. Formula, ki jo uporabljamo za napovedovanje ocen pri izbiranju s sodelovanjem tipa *k*-NN, je sledeča:

$$p_{u,i} = \overline{v_u} + \kappa_u \sum_{j \in U} \mathrm{sim}(u, j)(v_{j,i} - \overline{v_j}) \ , \ \kappa_u = \frac{1}{\sum_{j \in U} \mathrm{sim}(u, j)} \ , \qquad (1)$$

kjer je $p_{u,i}$ napovedana ocena, $\overline{v_u}$ povprečna vrednost ocen, ki jih je podal uporabnik u, $\mathrm{sim}(u_1, u_2)$ določa utež (podobnost), ki je večja, tem bolj sta si u_1 in u_2 podobna, κ_u je normalizacijski faktor, $v_{j,i}$ predstavlja oceno za predmet i, ki jo je podal uporabnik j, množica U pa vsebuje uporabnike, ki jih je našel algoritem za iskanje najbližjih sosedov. Vsi uporabniki iz U so že ocenili predmet i.[5]

Enačba (1) je bila prvič predstavljena v [19]. Od enostavnega obteženega povprečenja se razlikuje predvsem po tem, da vrne povprečno oceno uporabnika u, če za predmet i ni na voljo nobene ocene.

[5] Za uporabnike in predmete nismo uporabili notacije za vektorje (zapisani so kot u, i in ne kot $\mathbf{u}$, $\mathbf{i}$). Razlog za to je v tem, da na uporabnika/predmet lahko gledamo tudi kot na številko vrstice/stolpca v matriki ocen. Ne glede na to vsakemu uporabniku pripada vektor, ki ga določa vrstica v matriki, in vsakemu predmetu vektor, ki ga določa stolpec v matriki.

2.2.2 Funkcije za računanje podobnosti

Podobnost med dvema uporabnikoma (tj. dvema redkima vektorjema iz matrike ocen) je mogoče izračunati na različne načine. V nadaljevanju navajamo najpogosteje uporabljene funkcije za računanje podobnosti.

***Kosinusna podobnost* (cosine similarity)**. Podobnost je lahko definirana kot kosinus kota med dvema (redkima) vektorjema. Ta mera se na primer uporablja v informacijskem poizvedovanju (*information retrieval*) za določanje podobnosti med dvema dokumentoma, ki sta prav tako predstavljena s pripadajočima redkima vektorjema[6]. Formula za izračun kosinusne podobnosti je sledeča:

$$\mathrm{sim}_{\cos}(u_1,u_2)=\sum_{i\in I}\frac{v_{u_1,i}v_{u_2,i}}{\sqrt{\sum_{k\in I_1}v_{u_1,k}^2}\sqrt{\sum_{k\in I_2}v_{u_2,k}^2}}\ , \tag{2}$$

kjer je $v_{u,i}$ ocena, ki jo je podal uporabnik u za predmet i, množica I vsebuje vse predmete, pri katerih se ocene prekrivajo, I_1 vsebuje vse predmete, ki jih je ocenil uporabnik u_1, I_2 pa vse predmete, ki jih je ocenil uporabnik u_2.

Kosinusna podobnost ima zalogo vrednosti od 0 do 1, če so vse ocene na ocenjevalni lestvici pozitivne, oziroma od –1 do 1, če obsega ocenjevalna lestvica tudi negativne vrednosti[7]. Tudi pozitivno lestvico je mogoče z zamikom prilagoditi tako, da ima kosinusna podobnost zalogo vrednosti od –1 do 1. Zaželjeno je tudi, da ocenjevalna lestvica ne vsebuje ničle. Vektor s samimi ničlami (tj. uporabnik, ki je vse predmete, ki si jih je ogledal, ocenil z 0) namreč nima smeri in posledično kota ni mogoče izračunati. Tako je, na primer, smiselno zamakniti ocenjevalno lestvico z ocenami od 1 do 5 tako, da dobimo lestvico z ocenami –1'5, –0'5, 0'5, 1'5, 2'5 ali –2'5, –1'5, –0'5, 0'5, 1'5. Druga lestvica je bolj kritična do predmetov kot prva.

***Pearsonov korelacijski koeficient* (Pearson correlation coefficient)**. Mera za podobnost lahko temelji na stopnji korelacije med vrednostmi dveh (redkih) vektorjev [19]. Ta mera se uporablja v statistiki za računanje medsebojne odvisnosti dveh naključnih spremenljivk. Ima zalogo vrednosti od –1 do 1, pri čemer –1 predstavlja popolno nekoreliranost, 1 pa popolno koreliranost vrednosti v dveh vektorjih; 0 pomeni, da so vrednosti v dveh vektorjih popolnoma nekorelirane. Formula za izračun korelacijskega koeficienta je sledeča:

$$\mathrm{sim}_{\mathrm{P}}(u_1,u_2)=\frac{\sum_{j\in I}(v_{u_1,j}-\overline{v_{u_1}})(v_{u_2,j}-\overline{v_{u_2}})}{\sqrt{\sum_{j\in I}(v_{u_1,j}-\overline{v_{u_1}})^2\sum_{j\in I}(v_{u_2,j}-\overline{v_{u_2}})^2}}\ , \tag{3}$$

kjer je $v_{u,i}$ ocena, ki jo je podal uporabnik u za predmet i, $\overline{v_u}$ povprečna vrednost ocen, ki jih je podal uporabnik u, množica I pa vsebuje vse predmete, pri katerih se ocene prekrivajo.

***Umerjena kosinusna podobnost* (adjusted cosine similarity)**. Pri predmetnem izbiranju s sodelovanjem (glej poglavje 2.4) računamo podobnost med predmeti in ne podobnost med

[6] Dokument je predstavljen kot vektor frekvenc besed, ki jih vsebuje.
[7] –1 predstavlja popolno različnost, 0 pomeni, da med vektorjema ni razmerja (vektorja sta pravokotna eden na drugega), 1 pa predstavlja popolno podobnost.

uporabniki [21]. Težave nastopijo, ker vsak par istoležnih ocen v dveh vektorjih pripada drugemu uporabniku. Osnovna formula za kosinusno podobnost ne upošteva tega dejstva in tako dovoljuje napako, ki nastane zaradi različne interpretacije ocenjevalne lestvice s strani različnih uporabnikov. Umerjena kosinusna podobnost omili to napako tako, da od vsakega para istoležečih ocen odšteje povprečno oceno pripadajočega uporabnika. Umerjena kosinusna podobnost med dvema predmetoma ima torej obliko:

$$\mathrm{isim}_{\cos}(i_1, i_2) = \frac{\sum_{u \in U} (v_{u,i_1} - \overline{v_u})(v_{u,i_2} - \overline{v_u})}{\sqrt{\sum_{u \in U} (v_{u,i_1} - \overline{v_u})^2 \sum_{u \in U} (v_{u,i_2} - \overline{v_u})^2}}\ , \tag{4}$$

kjer je $v_{u,i}$ ocena, ki jo je podal uporabnik u za predmet i, $\overline{v_u}$ povprečna vrednost ocen, ki jih je podal uporabnik u, množica U pa vsebuje vse uporabnike, ki so ocenili oba predmeta.

***Privzeta ocena* (default rating)**. Osnovni problem funkcij za računanje podobnosti predstavlja dejstvo, da delujejo na prekrivajočih se vrednostih. Zaradi velike redkosti podatkov je število prekrivajočih se vrednosti največkrat majhno. Če se uporabnik A prekriva z uporabnikom B pri predmetih 1, 2 in 7 in če se uporabnik B prekriva z uporabnikom C pri predmetih 4, 8 in 12, uporabnik A pa z uporabnikom C nima skupnega nobenega predmeta, potem funkcije za računanje podobnosti v svoji osnovni obliki ne znajo izračunati podobnosti med uporabnikoma A in C, čeprav podobnost izhaja iz zakona tranzitivnosti, ki pravi, da če je A podoben B in je B podoben C, potem je A podoben C. Funkcije za računanje podobnosti v svoji osnovni obliki torej ne zaznavajo tranzitivnih relacij. Da se izognemo tej pomanjkljivosti, uvedemo modificirane formule, pri katerih upoštevamo vse predmete, ki jih je ocenil vsaj eden od obeh uporabnikov (vzamemo torej unijo namesto preseka predmetov), manjkajoče vrednosti pa nadomestimo z neko privzeto oceno d. To je največkrat nevtralna ocena z ocenjevalne lestvice, lahko pa jo določimo tudi kot povprečje ostalih ocen, ki jih je podal uporabnik, ali s primerjavo (tekstovnih) vsebin predmetov. Slednja možnost je opisana v [18], v tem delu pa ni obravnavana.

2.3 Izbiranje s sodelovanjem na osnovi modela

Za razliko od algoritmov na osnovi pomnjenja, algoritmi na osnovi modela najprej zgradijo model iz matrike ocen. Model omogoča hitrejše generiranje priporočil in/ali generiranje boljših priporočil. V nadaljevanju opisujemo nekaj metod za zmanjšanje razsežnosti izvorne matrike ocen in razlagamo izbiranje s sodelovanjem kot klasifikacijski problem (primeren za uporabo klasifikacijskih pristopov strojnega učenja). V obeh primerih (tj. pri manjšanju razsežnosti matrike ocen in izgradnji modela s pomočjo strojnega učenja) gre za izgradnjo modela, čeprav v dveh popolnoma ločenih fazah. Metode za zmanjšanje razsežnosti matrike ocen uporabimo v fazi predobdelave vhodnih podatkov, medtem ko strojno učenje uporabimo za izgradnjo modela, ki povzema znanje, potrebno za napovedovanje ocen in tvorjenje priporočil.

2.3.1 Metode za zmanjšanje razsežnosti izvorne matrike ocen

Prvotna matrika ocen lahko vsebuje milijone uporabnikov in milijone predmetov. Posledično ima prvotni vektorski prostor, v katerem "živijo" uporabniki ali predmeti, zelo veliko

razsežnosti. Iz tega dejstva izhaja potreba po zmanjšanju števila razsežnosti prvotnega vektorskega prostora. Zmanjšanje je mogoče doseči z izločitvijo uporabnikov (primerov, *examples*) in/ali predmetov (značilk, *features*), ki so manj relevantni za napovedovanje ocen. Raziskave so pokazale, da ustrezna izbira primerov in/ali značilk in tudi druge tehnike za zmanjšanje razsežnosti ne le rešujejo problem nadgradljivosti, ampak tudi prispevajo k večji natančnosti pri napovedovanju ocen [14][22][23], hkrati pa zgostijo matriko ocen, kar do neke mere rešuje problem redkosti matrike ocen.

Najenostavnejša rešitev za zmanjšanje razsežnosti je izločitev uporabnikov, ki niso podali dovolj ocen, in/ali predmetov, ki jih ni ocenilo zadostno število uporabnikov. Med preostalimi uporabniki lahko naključno izberemo n uporabnikov in s tem omejimo prostor za iskanje najbližjih sosedov. Slednja metoda se v literaturi pojavlja pod imenom "naključno vzorčenje" (*random sampling*). Ti relativno enostavni pristopi običajno niso dovolj učinkoviti, zato se poslužujemo tudi bolj kompleksnih postopkov – nekateri so opisani v nadaljevanju.

Iskanje skrite semantike (*latent semantic analysis, LSA*) [8]. Metoda LSA temelji na postopku razgradnje matrike ocen na singularne vrednosti (*singular value decomposition*, *SVD*). Postopek izhaja iz linearne algebre in omogoča razgradnjo prvotne matrike $\mathbf{M}$ v trojico $\mathbf{M} = \mathbf{U\Sigma V}^T$. Diagonalna matrika $\mathbf{\Sigma}$ vsebuje singularne vrednosti matrike $\mathbf{M}$. Če diagonalno matriko predelamo tako, da postavimo vse razen K največjih singularnih vrednosti na 0 in s tem dobimo matriko $\mathbf{\Sigma'}$, lahko aproksimiramo matriko $\mathbf{M}$ kot $\mathbf{M'} = \mathbf{U\Sigma'V}^T$. S tem pretvorimo prvotno velikorazsežnostno matriko v K-razsežnostni (manjrazsežnostni) približek. Sedaj lahko poiščemo sosesko določenega uporabnika tako, da vektor, ki predstavlja uporabnika, preslikamo v dobljeni K-razsežnostni prostor in v njem poiščemo uporabnike, ki so mu najbolj podobni. Iskanje v manjrazsežnostnem prostoru je vsekakor hitrejše. Poleg tega LSA zmanjša redkost matrike in izpostavi tranzitivne relacije med uporabniki. Rezultat slednjega je večja natančnost pri napovedovanju ocen.

Verjetnostno iskanje skrite semantike (*probabilistic LSA, pLSA*). Na osnovi metode LSA je bila razvita metoda pLSA [13]. pLSA ima svoje korenine v informacijskem poizvedovanju (*information retrieval*), a jo je mogoče s pridom uporabiti tudi pri izbiranju s sodelovanjem [14]. V statističnem modelu dogodek "oseba u je izbrala predmet i" (v primeru implicitnih[8] binarnih ocen) običajno predstavimo s parom (u,i), ki se med našim opazovanjem "pojavi" z določeno verjetnostjo, tj. $\mathrm{P}(u,i)$. Pri izbiranju s sodelovanjem nas zanima pogojna verjetnost, da se pojavi predmet i, če smo osredotočeni na določenega uporabnika, tj. $\mathrm{P}(i \mid u)$. Slednja oblika je primerna zato, ker nas zanimajo interesi določenega uporabnika.

Osnovna ideja je uvedba prikrite naključne spremenljivke (*latent random variable*) z, ki za vsak možen dogodek (u,i) zavzame neko stanje. Privzamemo, da je pojavitev uporabnika neodvisna od pojavitve predmeta: $\mathrm{P}(u,i) = \mathrm{P}(u)\mathrm{P}(i)$. V odvisnosti od z ima slednja enačba obliko: $\mathrm{P}(u,i) = \Sigma_z \mathrm{P}(z)\mathrm{P}(u \mid z)\mathrm{P}(i \mid z)$. $\mathrm{P}(i \mid u)$ torej lahko zapišemo v naslednji obliki:

$$\mathrm{P}(i \mid u) = \frac{\mathrm{P}(u,i)}{\mathrm{P}(u)} = \sum_z \frac{\mathrm{P}(z)\,\mathrm{P}(u \mid z)\,\mathrm{P}(i \mid z)}{\mathrm{P}(u)} = \sum_z \frac{\mathrm{P}(u,z)\,\mathrm{P}(i \mid z)}{\mathrm{P}(u)} = \sum_z \mathrm{P}(z \mid u)\,\mathrm{P}(i \mid z)\ . \qquad (5)$$

Zmanjšanje razsežnosti dosežemo tako, da število stanj, ki jih lahko zavzame z, omejimo na vrednost, ki je (precej) manjša od števila vseh možnih parov (u,i). Označimo število uporabnikov z N_u, število predmetov z N_i in število različnih stanj z z N_z, kjer je $N_z \ll N_u, N_i$.

[8] Pojem implicitnih ocen razlagamo v poglavju 3.1.

Verjetnosti P(i | u) lahko opišemo z $S_1 = N_i \times N_u$ neodvisnimi parametri. Po drugi strani pa lahko povzamemo verjetnosti P(i | z) in P(z | u) z $S_2 = N_i \times N_z + N_u \times N_z$ neodvisnimi parametri. Zmanjšanje razsežnosti je očitno iz dejstva, da $S_2 < S_1$, če je le N_z dovolj majhno število. Tak model s prikrito spremenljivko združuje uporabnike v skupine medsebojno podobnih uporabnikov in predmete v skupine medsebojno podobnih predmetov. pLSA tako kot nekateri algoritmi za razvrščanje v skupine (glej poglavje 2.4) dovoljuje delno pripadnost uporabnika/predmeta neki skupini (vsako stanje spremenljivke z tvori eno skupino).

Verjetnosti P(z | u) in P(i | z) v enačbi (5) lahko določimo s postopkom iskanja najbolj verjetnega modela (*expectation maximization*, *EM*), pri čemer lahko uporabimo poljuben mešani model (*mixture model*). Postopek pLSA lahko razširimo tako, da podpira tudi ocenjevalne lestvice, ki niso dvojiške. Slednje dosežemo tako, da enostavno dodamo tretjo spremenljivko v naš opazovani par (u,i) in tako tvorimo trojico (u,i,r), v kateri r predstavlja oceno, ki jo je uporabnik u dodelil predmetu i.

Povezavo med pLSA in LSA (SVD) je mogoče nazorno prikazati z razvrstitvijo vseh vrednosti P(u,i) v matriko $\mathbf{M_p}$, ki jo z uporabo metode SVD razgradimo v tri matrike, $\mathbf{M_p} = \mathbf{U_p}\mathbf{\Sigma_p}\mathbf{V_p}^T$. Elementi teh matrik so namreč $u_{i,k} = P(u_i \mid z_k)$, $\sigma_{k,k} = P(z_k)$, $v_{j,k} = P(i_j \mid z_k)$ [2].

2.3.2 Izbiranje s sodelovanjem kot klasifikacijski problem

Izbiranje s sodelovanjem lahko razložimo kot klasifikacijski problem, pri čemer so klasifikacijski razredi vrednosti z ocenjevalne lestvice [3]. Praktično vsak algoritem za nadzorovano učenje (*supervised learning*) je mogoče uporabiti za klasifikacijo (tj. napovedovanje ocen). Za vsakega uporabnika zgradimo po en klasifikator. Za učne primere nam služijo vektorji, ki predstavljajo predmete, ki jih je uporabnik že ocenil – za oznake (*labels*) uporabimo uporabnikove ocene. Problem se pojavi, če izbrani algoritem za učenje ne zna ravnati z manjkajočimi vrednostmi v redkih vektorjih. V [3] predlagajo, da v takih primerih predstavimo enega uporabnika z več primeri (z enim primerom za vsako vrednost z ocenjevalne lestvice). Če razpolagamo z ocenjevalno lestvico od 1 do 5, predstavimo uporabnika A s petimi primeri – A_1, A_2, ..., A_5. Primer A_3 na primer vsebuje vrednost 1 za vse predmete, ki jih je uporabnik ocenil s 3 in vrednost 0 za vse ostale predmete. Na ta način napolnimo matriko ocen s konkretnimi vrednostmi in uporabimo dobljene dvojiške vektorje za učenje. Ko zgradimo model, napovemo neko manjkajočo oceno tako, da preprosto klasificiramo vektor, ki predstavlja pripadajoči predmet, v enega od razredov, ki predstavljajo vrednosti z ocenjevalne lestvice. Če je ocenjevalna lestvica zvezna in ne diskretna, namesto klasifikacijskega uporabimo regresijski algoritem ali pa se poslužimo diskretizacije podatkov [15]. Pri našem delu smo uporabili klasifikacijsko in regresijsko različico modela (glej poglavje 7).

2.4 Nekateri drugi pristopi k izbiranju s sodelovanjem

V tem poglavju so na kratko predstavljeni nekateri drugi pristopi k izbiranju s sodelovanjem.

Predmetno izbiranje s sodelovanjem (*item-based collaborative filtering*). Pristopi, ki so bili predstavljeni doslej, kot osrednjo entiteto obravnavajo uporabnika (iščejo soseske uporabnikov), nekateri raziskovalci pa so se lotili tudi pristopov, ki v središče pozornosti postavijo predmete [21]. Napoved ocene uporabnika u za predmet i se izvrši tako, da

algoritem za napovedovanje izračuna obteženo povprečje ocen *k* predmetov, ki jih je uporabnik *u* že ocenil in so najbolj podobni predmetu *i*. Podobnosti med predmeti je mogoče izračunati s funkcijami za računanje podobnosti med uporabniki (le da tokrat operiramo s stolpci matrike in ne z vrsticami), najučinkovitejša funkcija pa je, glede na [21], umerjena kosinusna podobnost (glej poglavje 2.2.2).

Horting. V tem odstavku je povzet postopek, ki so ga avtorji poimenovali *horting* [1]. Zanj je značilno, da iz podatkov zgradi usmerjen graf, katerega vozlišča predstavljajo uporabnike, povezave pa odražajo podobnosti med njimi. Oceno uporabnika *u* za predmet *i* napove z iskanjem usmerjenih poti v grafu od uporabnika *u* do uporabnikov, ki so predmet *i* že ocenili. Z linearno kombinacijo uteži (tj. podobnosti) na povezavah poiskanih usmerjenih poti in ocen, ki so jih podali uporabniki na koncih teh poti, algoritem nato izračuna napoved za iskano oceno. Ena od lastnosti omenjenega grafa je, da nihče od uporabnikov (vozlišč), ki tvorijo pot od uporabnika *u* do uporabnika, ki je že ocenil predmet *i*, sam še ni ocenil predmeta *i*. Torej opisani postopek upošteva tranzitivnost podobnosti med uporabniki.

Razvrščanje v skupine (*clustering*). Razvrščanje lahko uporabimo za tvorjenje skupin podobnih uporabnikov [4][14][23]. Ker opazovani uporabnik pripada določeni skupini, lahko napovemo njegovo oceno za predmet *i* tako, da povprečimo ocene za predmet *i*, ki so jih podali uporabniki znotraj iste skupine. Nekatere metode dovoljujejo delno pripadnost uporabnika določeni skupini – tako uporabnik pripada vsaki skupini z določeno stopnjo ali verjetnostjo. V takem primeru je napovedana ocena povprečje napovedanih ocen v vseh skupinah, ki jim pripada uporabnik, ocene pa so obtežene z uporabnikovo stopnjo pripadnosti skupini (tj. z verjetnostjo, da uporabnik pripada skupini). Razvrščanje je mogoče uporabiti tudi za manjšanje števila primerov (uporabnikov) za iskanje sosesk z algoritmom najbližjih sosedov (sosede iščemo le znotraj skupine, ki ji pripada uporabnik).

Bayesovske mreže (*Bayesian networks*). Za izbiranje s sodelovanjem se uporabljajo tudi Bayesovske mreže z odločitvenimi drevesi (*decision trees*) v vozliščih [4][6]. Vozlišča ponazarjajo predmete, stanja vozlišč pa ocene. Pogojne verjetnosti na vozliščih so predstavljene z odločitvenimi drevesi, v katerih so vozlišča spet predmeti, povezave predstavljajo ocene za vozlišča (tj. predmete), iz katerih izhajajo, listi pa ocene za predmet, ki ga ponazarja opazovano vozlišče Bayesovske mreže. Algoritem za tvorjenje Bayesovske mreže se pri velikem številu podatkov izvaja več ur ali celo več dni, zato Bayesovske mreže niso primerne za domene, v katerih se mora model posodabljati hitro in pogosto.

3. Eksperimentalni material

3.1 Viri in lastnosti podatkovnih naborov

Podatki v matriki ocen so lahko zbrani eksplicitno (eksplicitne ocene, *explicit ratings*) ali implicitno (implicitne ocene, *implicit ratings*). Zbiranje podatkov na ekspliciten način zahteva uporabnikovo sodelovanje (uporabnik mora eksplicitno podati oceno za predmet), medtem ko so implicitne ocene izpeljane iz opazovanja uporabnika pri interakciji s predmeti. Opazovanje uporabnika (in s tem implicitno zbiranje podatkov) se lahko vrši na strani odjemalca[9] ali pa na strani strežnika. Slednji primer zajema tudi strežniške dnevnike (*Web-server logs*), ki beležijo dostope uzporabnikov do spletnih vsebin. Če je uporabnik dostopil do predmeta, ga je na primer implicitno ocenil z 1 ("ogledal sem si ta predmet"), sicer pa z 0 ("nisem si ogledal tega predmeta").

Zbrane podatke najprej obdelamo (postopek obdelave je odvisen od konkretnega podatkovnega nabora) in uredimo v matriko ocen. Kot je bilo že povedano, v matriki ocen vrstice predstavljajo uporabnike (uporabniške profile), stolpci pa predmete (profile predmetov). Elementi matrike ocen so bodisi eksplicitne ocene bodisi ocene, izpeljane iz uporabnikovih akcij. V matriki ocen je navadno veliko manjkajočih vrednosti, ker uporabnik običajno oceni relativno majhno število predmetov. Da bi poudarili to dejstvo, uporabljamo izraz "redka matrika" (*sparse matrix*), za vrstice (stolpce) redke matrike pa tudi izraz "redek vektor" (*sparse vector*).

Številne manjkajoče vrednosti predstavljajo najbolj pereč problem izbiranja s sodelovanjem – problem redkosti matrike ocen. Manjkajoče vrednosti v redkih vektorjih otežijo računanje podobnosti med uporabniki in/ali predmeti. Zgodi se, da imata dva redka vektorja le malo prekrivajočih se vrednosti ali pa – v skrajnem primeru – prav nobene. Izračun podobnosti je v primeru majhnega števila prekrivajočih se vrednosti zanesljiv le do neke mere – v skrajnem primeru, ko se dva redka vektorja ne prekrivata v prav nobeni vrednosti, na standarden način podobnosti sploh ne moremo izračunati.

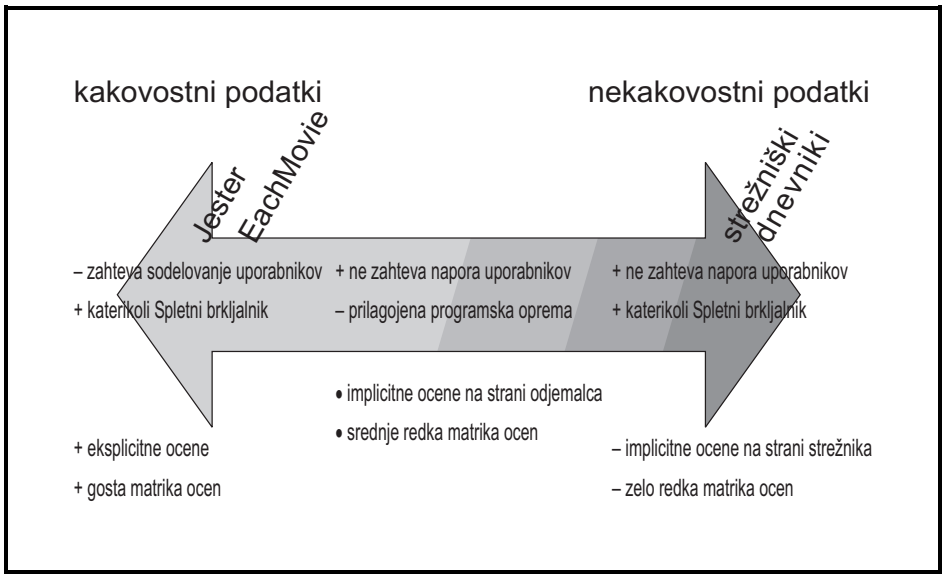
Redkost matrike ocen pa ni edini vzrok za nezadovoljivo delovanje izbiranja s sodelovanjem. Kot je bilo že omenjeno, v primeru implicitnega ocenjevanja konkretne (eksplicitne) ocene običajno izpeljemo iz uporabnikovih interakcij s predmeti. Izpeljava eksplicitnih ocen iz uporabnikovega vedenja pa ni trivialna naloga – v tem postopku pogosto pride do napak. Problem je posebej velik pri podatkih, zapisanih v strežniških dnevnikih, saj strežniški dnevniki vsebujejo dokaj omejene informacije o vedenju uporabnikov. Da bi iz strežniških podatkov ugotovili, koliko časa je uporabnik prebiral nek dokument, moramo izračunati časovno razliko med dvema zaporednima dostopoma tega uporabnika. Že pri tem se pojavijo številni problemi – ne vemo na primer niti tega, ali je uporabnik zares bral dokument ali pa je, denimo, odšel na kosilo, medtem pa je pustil spletni brskalnik odprt. Še večji problem nastane zaradi lokalnih predpomnilnikov (*browser cache*) in posredovalnih strežnikov (*proxy server*) – dostopov uporabnika do teh namreč na strani strežnika ni mogoče zaznati.

Nadalje se v kontekstu implicitnih ocen na strežniški strani pojavlja problem identifikacije uporabnika. Če se uporabnik v sistem ne prijavi z uporabniškim imenom in če onemogoči

[9] Zaradi potrebe po prilagojeni programski opremi se ta možnost ne uporablja pogosto.

piškotke (*cookies*), potem njegove seje ni mogoče izluščiti iz strežniškega dnevnika na enostaven način. V takem primeru namreč edino informacijo o identiteti uporabnika predstavlja njegov IP-naslov, ki pa ni zanesljiv identifikator. Veliko uporabnikov si namreč lahko deli isti IP-naslov (uslužbenci v podjetju za skupnim požarnim zidom (*firewall*)), po drugi strani pa ima lahko isti uporabnik pri vsakem dostopu drugačen IP-naslov celo tekom iste seje (dinamično razporejanje uporabnikov na prehode (*gateways*) z namenom razporejanja bremena (*load balancing*)). Prijava v sistem in/ali uporaba piškotkov sta torej najzanesljivejša mehanizma za sledenje uporabnikom [20].

Iz tega kratkega opisa problemov, ki se pojavljajo pri zbiranju podatkov, lahko povzamemo, da je matrika eksplicitnih ocen zanesljivejši vir podatkov za izbiranje s sodelovanjem kot matrika, zgrajena iz strežniških podatkov. Zakaj bi potem sploh želeli uporabljati strežniške dnevnike kot izvor podatkov za izbiranje s sodelovanjem? Razlog je v tem, da take podatke zelo enostavno zberemo – od uporabnikov ne zahtevajo nobenega napora niti ne zahtevajo uporabe modificiranih odjemalcev. Ta konflikt interesov (kakovost podatkov na eni strani ter enostavnost in cena zbiranja podatkov na drugi) je prikazan na sliki 2.



Slika 2: Lastnosti podatkovnega nabora, ki vplivajo na kakovost izbiranja s sodelovanjem, in položaj treh podatkovnih naborov, ki smo jih uporabili za empirično evalvacijo (Jester, EachMovie in strežniški dnevniki), glede na njihove lastnosti.

3.2 Domene

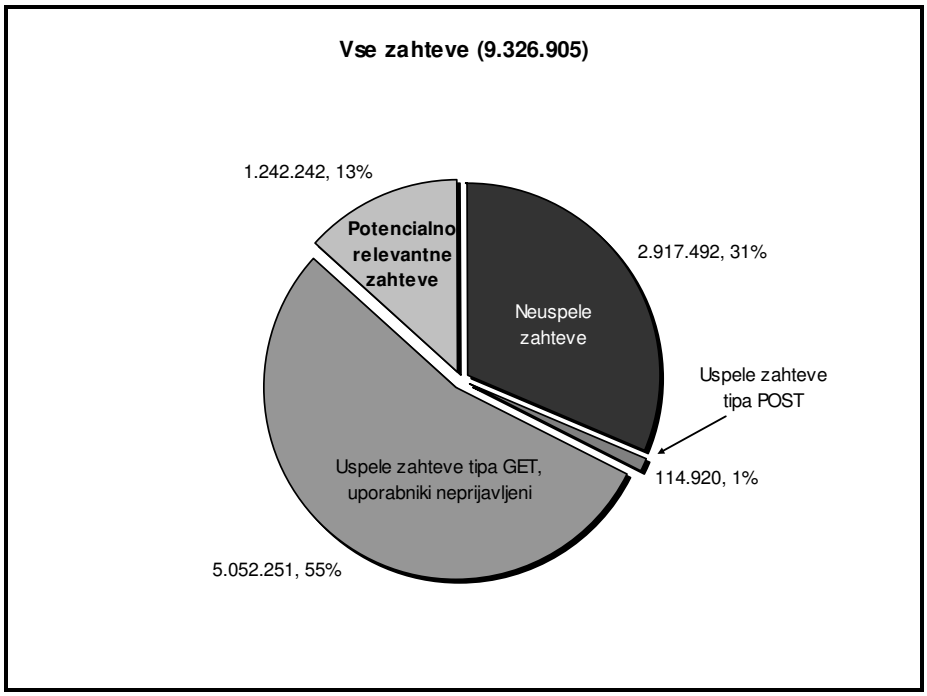
Za izvedbo naše evalvacije smo uporabili podatkovne nabore iz treh različnih domen. Podatkovni nabor iz prve domene, EachMovie[10], se običajno uporablja za evalvacijo izbiranja s sodelovanjem in vsebuje eksplicitne ocene za filme. Spletna storitev, preko katere so uporabniki vnašali svoje ocene za filme, je bila aktivna 18 mesecev. V tem času je 61.131 različnih uporabnikov podalo 2.558.871 ocen za 1.622 filmov. Pri tem je uporabnik povprečno ocenil približno 42 filmov, film pa je bil v povprečju ocenjen s strani približno

[10] V lasti podjetja Digital Equipment Corporation.

1.578 uporabnikov. Redkost matrike, izračunana kot število manjkajočih vrednosti deljeno z velikostjo matrike ocen (tj. zmnožkom števila vrstic s številom stolpcev), je bila v primeru podatkovnega nabora EachMovie 97,42 %.

Podatkovni nabor iz druge domene, Jester [9], se prav tako običajno uporablja za evalvacijo izbiranja s sodelovanjem in vsebuje ocene za šale. Spletni servis za branje šal in posredovanje ocen je bil aktiven 4 leta. V tem času je 73.421 različnih uporabnikov podalo 4.136.360 ocen za 100 šal. Pri tem je uporabnik povprečno ocenil približno 56 šal, šala pa je bila v povprečju ocenjena s strani približno 41.364 uporabnikov. Redkost matrike ocen je bila v primeru podatkovnega nabora Jester 43,66 %.

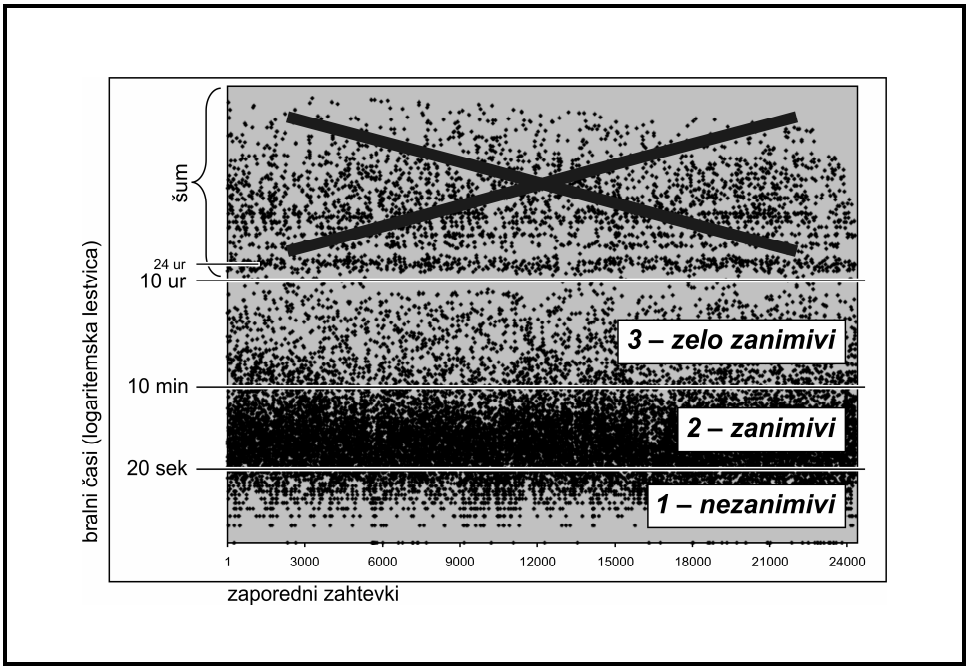
Tretja domena je bila interna digitalna knjižnica velikega britanskega podjetja. Podatkovni nabor so bili pripadajoči strežniški dnevniki – šlo je torej za realen problem z implicitnim ocenjevanjem, podatki pa so bili zbrani na strežniški strani. Dnevniki so vsebovali dostope do digitalne knjižnice v skupnem časovnem obsegu 920 dni. V nasprotju s prvima dvema domenama je bilo potrebno tu podatke najprej intenzivno obdelati. V svoji osnovni obliki so dnevniki vsebovali preko 9,3 milijona dostopov (zahtev). Najprej smo odstranili neuspele zahteve (*failed requests*), preusmeritve (*redirections*), zahteve tipa POST (*POST requests*) in zahteve anonimnih uporabnikov (*anonymous requests*). Ostalo je nekaj preko 1,2 milijona zahtev (kar je predstavljalo le 13 % prvotnega podatkovnega nabora). Delež potencialno relevantnih zahtev je prikazan na sliki 3. Pri tem so preostali podatki še vedno vsebovali zahteve po slikovnih datotekah, nevsebinskih straneh (takšne so na primer strani s kazali) in drugih irelevantnih straneh. Še večji problem je predstavljalo dejstvo, da so v podjetju dostopali do več različnih virov vsebin, a je bil le en vir relevanten za aplikacijo izbiranja s sodelovanjem (dnevniški zapisi pa so pač vsebovali tudi dostope do drugih virov). Posledično je število relevantnih dostopov drastično upadlo. Na koncu je ostalo le 25.088 uporabnih dnevniških zapisov, kar je predstavljalo 0,27 % prvotnega števila zapisov (dostopov).



Slika 3: Delež potencialno relevantnih zahtev v celotni količini zahtev v strežniških dnevnikih.

Naslednji problem se je pojavil pri izpeljevanju eksplicitnih ocen iz dnevniških zapisov. Kot je bilo že omenjeno, taka preslikava ni trivialna naloga. Najenostavnejši način je pripis binarnih vrednosti 1 (“uporabnik je dostopil do predmeta”) in 0 (“uporabnik ni dostopil do predmeta”) vsem predmetom v zbirki, česar so se poslužili tudi v [4]. Slabost tega pristopa je, da ne izraža nikakršnega mnenja uporabnika o predmetu. V [7] so pokazali linearno odvisnost med časom branja dokumenta in eksplicitno oceno, ki jo je uporabnik dodelili istemu dokumentu (ista opažanja so objavili tudi v [16]). V teh raziskavah so uporabniki uporabljali prilagojeno programsko opremo (modificirane spletne brskalnike), zaradi česar so bili zbrani podatki bolj zanesljivi (zbrani so bili namreč na strani odjemalca). Kljub dejstvu, da smo imeli v našem primeru opravka z nekoliko manj zanesljivimi podatki (zbranimi na strežniški strani), smo se pri pretvorbi dnevniških zapisov v matriko ocen odločili upoštevati bralne čase posameznih vsebin (dokumentov).

Bralne čase (izračunane kot časovna razlika med dvema zaporednima dostopoma) so prikazani v razpršenem grafikonu (*scatter plot*) na sliki 4. Os x prikazuje zahteve, urejene po času dostopa do vsebine, os y pa izračunane bralne čase na logaritemski lestvici. Opazimo lahko, da je območje okoli 24 ur zelo gosto posejano s točkami. To pa seveda ne pomeni, da uporabniki berejo neko vsebino toliko časa – to so zgolj zadnji dostopi do interne digitalne knjižnice v dnevu. Uporabnik je naslednji dostop izvedel naslednji dan, kar se v strežniškem dnevniku odraža kot približno 24-urni čas branja. Prav tako je opaziti razredčeno območje med 5 in 15 urami. Bralni časi nad 10 urami torej skoraj zagotovo ne pripadajo koristnim podatkom (so “zunajležniki”, *outliers*, oz. šum). Ta intuicija se potrdi, če izločimo vse dostope, katerih bralni čas je bil določen na podlagi dveh časov z različnima datumoma. S tem žal izgubimo kar nekaj informacij, vendar se izognemu tveganju, da bi te informacije v postopku izbiranja s sodelovanjem napačno obravnavali.



Slika 4: Preslikava implicitnih bralnih časov, vsebovanih v strežniških dnevnikih, na diskretno ocenjevalno lestvico s tremi eksplicitnimi vrednostmi.

Bralne čase smo nato preslikali na ocenjevalno lestvico s tremi vrednostmi (1 = "nezanimiv predmet", 2 = "zanimiv predmet", 3 = "zelo zanimiv predmet"). Nekoliko *ad-hoc* (intuitivno) smo določili dve meji: eno pri 20 sekundah in drugo pri 10 minutah. Ker so predmeti znanstvene publikacije in je 20 sekund komajda dovolj za vpogled v povzetek, smo predmete z bralnimi časi pod 20 sekundami označili za nezanimive. Predmete z bralnimi časi nad 20 sekundami in pod 10 minutami smo označili za zanimive, predmete z bralnimi časi nad 10 minutami pa za zelo zanimive.

Tabela 1 prikazuje primerjavo lastnosti treh evalvacijskih podatkovnih naborov. Očitno je, da nekoliko *ad-hoc* preslikava na diskretno ocenjevalno lestvico ni naš največji problem v kontekstu strežniških dnevnikov. Najbolj zaskrbljujoče dejstvo sledi iz majhnega števila uporabnih zahtev in posledično izredno majhnega povprečnega števila ocen za posamezni predmet – povprečno število ocen za posamezni predmet je namreč le 1,14, kar povzroča izredno slabo prekrivanje. Redkost matrike ocen je posledično zelo velika (99,92 % celic v matriki ocen je praznih). Druga dva podatkovna nabora sta precej bolj obetavna. Najlepše lastnosti ima Jester, ki ima zelo majhno redkost; matrika ocen iz domene EachMovie je bolj redka, a ima kljub temu relativno veliko povprečno število ocen za posamezni predmet. Poleg tega Jester in EachMovie vsebujeta eksplicitne ocene, kar pomeni, da so ti podatki bolj zanesljivi kot podatki iz strežniških dnevnikov (glej tudi sliko 2).

	Ocene		Velikost			Redkost		
	ekspl./ impl.	**lestvica**	**št. uporab.**	**št. predm.**	**št. ocen**	**%[11]**	**povp. št. ocen/upor.**	**povp. št. ocen/predm.**
EachMovie	ekspl.	0–5 (diskr.)	61.131	1.622	2.558.871	97,42	41,86	1.577,60
Jester	ekspl.	–10–+10 (zvezna)	73.421	100	4.136.360	43,66	56,34	41.363,60
Strežniški dnevniki	impl.	1–3 (diskr.[12])	1.398	22.014	25.088	**99,92**	**17,94**	**1,14**

Tabela 1: Primerjava treh podatkovnih naborov. Krepko natisnjene številke predstavljajo najbolj problematične podatke.

3.3 Evalvacijska platforma

Da bi raziskali, kako se algoritmi za izbiranje s sodelovanjem izkažejo v domenah z različnimi karakteristikami in glede na različne evalvacijske metrike, smo implementirali programsko opremo za empirično evalvacijo algoritmov za izbiranje s sodelovanjem. Ta programska oprema (v nadaljevanju "evalvacijska platforma") je sestavljena iz modulov, razvrščenih v cevovod. Cevovod je sestavljen iz naslednjih štirih zaporednih korakov:

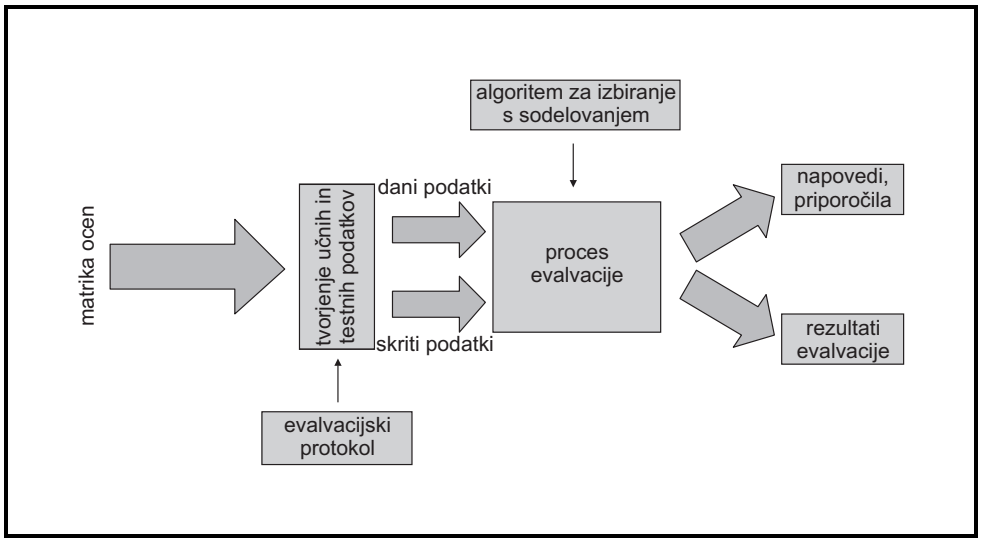
- uvoz matrike ocen (v primeru implicitnih ocen je potrebno podatke najprej obdelati – tj. tvoriti konkretne ocene),
- delitev podatkov na učne (*training set*) in testne podatke (*test set*) – v kontekstu izbiranja s sodelovanjem se uporabljata izraza "dani" (*given data*) in "prikriti podatki" (*hidden data*) – glede na izbrani evalvacijski protokol (*evaluation protocol*),
- aplikacija izbranega algoritma za izbiranje s sodelovanjem in

[11] Izračunano kot število manjkajočih vrednosti deljeno z velikostjo matrike ocen (velikost matrike je izračunana kot število vrstic pomnoženo s številom stolpcev).
[12] Po predobdelavi.

- napovedovanje ocen, tvorjenje urejenih seznamov priporočil, zbiranje evalvacijskih podatkov.

Evalvacijska platforma je prikazana na sliki 5.

Naj na kratko opišemo nekatere manj očitne korake evalvacijskega cevovoda. Delitev podatkov na dane (učne) in prikrite (testne) podatke pogojuje evalvacijski protokol, ki določa količino podatkov, ki jih je potrebno prikriti. V osnovi ločimo dve družini takšnih protokolov: protokoli tipa danih-*n* (*given-n*) in protokoli tipa vse-razen-*n* (*all-but-n*). Protokoli tipa danih-*n* prepustijo *n* ocen vsakega uporabnika v dani (učni) podatkovni nabor – ostale ocene prikrijejo. Protokoli tipa vse-razen-*n*, po drugi strani, prepustijo vse razen *n* ocen vsakega uporabnika v dani (učni) podatkovni nabor – prikrijejo torej le *n* ocen vsakega uporabnika. V tem delu uporabljamo izpeljanko iz ene od teh dveh družin protokolov – vse-razen-*x*-%. Protokol vse-razen-30-% na primer določa razdelitev podatkov, v kateri je 30 % naključno izbranih ocen od vsakega uporabnika prikritih, preostale pa uporabljamo kot naše izključno znanje o uporabnikovem mnenju o predmetih. Dane ocene uporabimo za formiranje sosesk v fazi napovedovanja (pri algoritmih tipa *k* najbližjih sosedov) ali izgradnjo modela v fazi učenja, prikrite ocene pa uporabimo v fazi evalvacije izbranega algoritma za izbiranje s sodelovanjem. Algoritem napove prikrite ocene, napovedi pa nato primerjamo z dejanskimi vrednostmi – iz razlik med dejanskimi in napovedanimi ocenami lahko določeni algoritem za izbiranje s sodelovanjem ocenimo na podlagi izbrane evalvacijske metrike.



Slika 5: Evalvacijska platforma – shematični prikaz.

4. Evalvacijske metrike

V tem poglavju predstavljamo izbrane evalvacijske metrike. Algoritme za izbiranje s sodelovanjem smo želeli oceniti glede na tri vidike kakovosti:

- kakovost napovedovanja ocen,
- kakovost odkrivanja relevantnih predmetov in
- kakovost tvorjenja urejenega seznama relevantnih predmetov.

Za vsako od treh omenjenih kategorij najdemo v literaturi več evalvacijskih metrik. Izbiro smo omejili tako, da smo se oprli na delo [12], ki razkriva linearno soodvisnost nekaterih evalvacijskih metrik. Iz linearne soodvisnosti metrik sledi, da lahko izberemo po eno metriko iz vsake skupine linearno soodvisnih in kljub temu še vedno zajamemo vse aspekte evalvacije.

Naj še omenimo, da so vse evalvacijske metrike lahko povprečene na dva načina: preko vseh podatkov (*overall*) in preko uporabnikov (*per-user*). V prvem primeru si predstavljamo, da imamo enega samega uporabnika, ki je podal vse ocene v matriki ocen. Evalvacijsko oceno nato izračunamo glede na tega namišljenega uporabnika. V drugem primeru evalvacijsko oceno izračunamo za vsakega uporabnika posebej, nato pa te (delne) ocene povprečimo (tj. delimo s številom uporabnikov), da dobimo končni rezultat.

Kot je bilo že omenjeno, so v [12] pokazali linearno soodvisnost nekaterih evalvacijskih metrik. Pokazali so, da je *povprečna absolutna napaka* (*mean absolute error*, *MAE*) v linearni soodvisnosti z metrikami *ROC*, merjenimi preko vseh podatkov, in Pearsonovo evalvacijsko metriko, *metrika mejnega obiska* (metrika za evalvacijo urejenih seznamov; *half-life utility metric*) je v linearni soodvisnosti z metrikami *ROC*, merjenimi preko uporabnikov, *srednja povprečna natančnost* (*mean average precision*, *MAP*) pa je v linearni soodvisnosti z mnogimi drugimi metrikami, ki so merjene preko uporabnikov. Glede na opisano smo se odločili, da v evalvacijo vključimo naslednje metrike:

- povprečno absolutno napako,
- metriko mejnega obiska,
- srednjo povprečno natančnost in
- metriko F_1.

K reprezentativnim metrikam smo dodali še matriko F_1, saj evalvira sposobnost algoritma, da poišče dobre predmete, ne pa tudi sposobnosti, da jih uredi po relevantnosti. Poleg tega se F_1 s pridom uporablja v strojnem učenju (*machine learning*) in pri informacijskem poizvedovanju (*information retrieval*) in je dobro znana ljudjem, ki se ukvarjajo z omenjenima in sorodnimi področji.

V nadaljevanju so opisane evalvacijske metrike, ki smo jih uporabili za empirično evalvacijo.

4.1 Povprečna absolutna napaka

Povprečna absolutna napaka (v nadaljevanju tudi "MAE") se veliko uporablja za evalvacijo izbiranja s sodelovanjem, kljub temu da ne da dobrih primerjalnih rezultatov, kadar nas zanima, kako dobre urejene sezname priporočil tvori algoritem [17]. MAE odgovori zgolj na vprašanje, kako natančno algoritem napove oceno, ki bi jo dal določeni uporabnik določenemu predmetu. Kadar se poslužujemo metrike MAE, nas torej zanima *natančnost napovedi ocene* algoritma za izbiranje s sodelovanjem. MAE ima dve obliki – lahko je

povprečena preko vseh podatkov ali pa preko uporabnikov. MAE, povprečena preko vseh podatkov, je definirana kot:

$$MAE = \frac{\sum_{r \in R_h} | r - r_p |}{| R_h |} , \tag{6}$$

kjer je R_h množica prikritih ocen, r_p pa oceni r pripadajoča napovedana ocena. MAE, povprečena preko uporabnikov, je definirana rahlo drugače, in sicer kot:

$$\mathrm{MAE}(u) = \frac{\sum_{i \in I_u} | r_{u,i} - p_{u,i} |}{| I_u |} , \; MAE' = \frac{\sum_{u \in U} \mathrm{MAE}(u)}{| U |} , \tag{7}$$

kjer je I_u množica predmetov, ki jih je ocenil uporabnik u, $r_{u,i}$ ocena, ki jo je uporabnik u podal predmetu i, $p_{u,i}$ pripadajoča ocena, ki jo je napovedal algoritem, U pa množica vseh uporabnikov.

Da bi lahko primerjali ocene MAE, dobljene v različnih domenah, velikokrat uporabimo normalizacijo komponent, ki nastopajo pri računanju MAE. S tem dosežemo, da so rezultati neodvisni od razpona ocenjevalne lestvice, ki jo uporabljamo v kontekstu določene domene. Tako modificirana metrika MAE se imenuje *normalizirana povprečna absolutna napaka* (*normalized MAE*; v nadaljevanju tudi "NMAE"). Definirana je kot:

$$NMAE = \frac{MAE}{r_{max} - r_{min}} , \tag{8}$$

kjer sta r_{min} in r_{max} najmanjša in največja vrednost z ocenjevalne lestvice. Imenovalec slednje enačbe efektivno predstavlja razpon ocenjevalne lestvice. Na tak način lahko normaliziramo obe obliki povprečne absolutne napake.

4.2 Metrika mejnega obiska

Metriko mejnega obiska (*half-life utility metric*) uporabljamo za evalvacijo urejenih seznamov. Prvič je bila predstavljena v [4]. V primeru metrike mejnega obiska nas ne zanima natančnost napovedi vsake posamezne ocene, temveč sposobnost tvorjenja seznama predmetov z ozirom na uporabnikove interese – torej *natančnost rangiranja* predmetov. Metrika mejnega obiska temelji na intuiciji, da si bo uporabnik vsak naslednji predmet s seznama ogledal z manjšo verjetnostjo, zato bolj kaznuje napake na vrhu seznama. Opira se na predpostavko, da verjetnost, s katero si bo uporabnik ogledal predmet, upada eksponentno od prvega proti zadnjemu predmetu s seznama. Uvaja *parameter mejnega obiska* α (*half-life parameter*), ki predstavlja rang (tj. mesto na seznamu) predmeta, ki si ga bo uporabnik ogledal s 50 % verjetnostjo. Definirana je kot:

$$\mathrm{R}(u) = \sum_{i \in \mathbf{I}} \frac{\max(r_{u,i} - d, 0)}{2^{(j-1)/(\alpha-1)}} , \; R = 100 \frac{\sum_{u \in U} \mathrm{R}(u)}{\sum_{u \in U} \mathrm{R}^{\max}(u)} , \tag{9}$$

kjer je **I** urejen seznam predmetov, ki jih je algoritem priporočil uporabniku u, $r_{u,i}$ ocena, ki jo je uporabnik u podal predmetu i, d nevtralna ocena z ocenjevalne lestvice, j zaporedna številka (tj. rang oz. mesto) predmeta i na seznamu **I**, α že omenjeni parameter mejnega obiska, $R^{max}(u)$ pa največja možna vrednost, ki jo lahko doseže $R(u)$. $R(u)$ doseže maksimum v primeru, ko je seznam **I** urejen glede na uporabnikove dejanske ocene.

4.3 Srednja povprečna natančnost

Srednjo povprečno natančnost (*mean average precision*; v nadaljevanju tudi "MAP") prav tako uporabljmo za evalvacijo urejenih seznamov – tudi v tem primeru nas torej zanima sposobnost rangiranja predmetov. Povprečna natančnost (*average precision*) evalvira natančnost tvorjenja (enega samega) seznama, MAP pa je srednja vrednost povprečnih natančnosti preko vseh tvorjenih seznamov (v praksi je to enako povprečenju preko vseh uporabnikov, saj v postopku evalvacije algoritem za vsakega uporabnika pripravi en urejen seznam priporočil). Da bi lahko izračunali MAP, najprej binariziramo ocene, tako da matrika ocen vsebuje le dve različni vrednosti, ki predmete označujeta zgolj za "zanimive" ali "nezanimive". To največkrat storimo tako, da vse predmete, ocenjene z oceno, ki je večja od nevtralne ocene z ocenjevalne lestvice (primer: ocenjevalna lestvica z ocenami od 1 do 5 ima nevtralno oceno 2,5), proglasimo za zanimive, ostale pa za nezanimive. Povprečno natančnost sedaj izračunamo tako, da na podlagi danega seznama tvorimo vse podsezname, ki za zadnji element vsebujejo zanimiv predmet in vse njegove predhodnike. Za vsak tak podseznam s izračunamo natančnost:

$$\mathrm{P}(s) = \frac{št\_zanimivih\_predmetov\_na\_podseznamu\_s}{dolžina\_podseznama\_s}\ , \tag{10}$$

pri čemer v enačbi nastopa število predmetov, ki so dejansko zanimivi, in ne število predmetov, ki jih je algoritem proglasil za zanimive.

Nadalje izračunamo povprečno natančnost kot povprečje natančnosti preko vseh podseznamov $s \in S_l$ določenega seznama l:

$$\mathrm{AP}(l) = \frac{\sum_{s \in S_l} \mathrm{P}(s)}{št\_zanimivih\_predmetov\_na\_seznamu\_l}\ . \tag{11}$$

Na koncu izračunamo srednjo vrednost povprečnih natančnosti preko vseh seznamov $l \in L$, kar nam da:

$$MAP = \frac{\sum_{l \in L} \mathrm{AP}(l)}{|L|}\ . \tag{12}$$

4.4 Metrika F_1

Zadnja metrika, ki smo jo vključili v evalvacijo, je metrika F_1, ki je predstavnica družine metrik F. Za razliko od obravnavanih metrik za evalvacijo seznamov (metrika mejnega obiska in MAP) metrike F niso občutljive na vrstni red predmetov na seznamu. V tem primeru nas

torej zanima sposobnost algoritma, da zgolj *najde zanimive predmete*. Metrike F združujejo dve drugi metriki – natančnost (*precision*) in priklic (*recall*). Družina metrik F je definirana kot:

$$F_\alpha = \frac{(1+\alpha)\times natančnost \times priklic}{\alpha \times natančnost + priklic} \ , \tag{13}$$

pri čemer parameter α služi za obtežitev ene in druge komponente in pove, kolikokrat bolj želimo poudariti priklic. Metrika $F_{0,5}$ na primer upošteva natančnost dvakrat bolj od priklica, metrika F_2, po drugi strani, pa upošteva priklic dvakrat bolj od natančnosti. V praksi se velikokrat uporablja metrika F_1, ki upošteva obe komponenti v enaki meri. To metriko smo uporabili tudi v tem delu. Iz (13) pri $\alpha = 1$ sledi:

$$F_1 = \frac{2\times natančnost \times priklic}{natančnost + priklic} \ . \tag{14}$$[13]

V primeru izbiranja s sodelovanjem apliciramo metriko F_1 tako, da najprej za nekega uporabnika napovemo vse prikrite ocene (oz. za vsak predmet s prikrito oceno napovemo, ali je zanimiv ali ne). Nato preštejemo, koliko zanimivih predmetov je algoritem proglasil za zanimive (*true positives*, *TP*), koliko zanimivih predmetov je algoritem proglasil za nezanimive (*false negatives*, *FN*) in koliko nezanimivih predmetov je algoritem proglasil za zanimive (*false positives*, *FP*). Iz teh treh vrednosti izračunamo natančnost *P* in priklic *R* kot:

$$P = \begin{cases} TP/(TP+FP) & TP+FP>0 \\ 1 & TP+FP=0 \end{cases}, \tag{15}$$

$$R = \begin{cases} TP/(TP+FN) & TP+FN>0 \\ 1 & TP+FN=0 \end{cases}. \tag{16}$$

Iz tako izračunanih natančnosti in priklica po enačbi (14) izračunamo oceno F_1 glede na vsakega uporabnika $u \in U$. Nazadnje izračunamo še povprečno oceno F_1 preko vseh uporabnikov:

$$F_1 = \frac{\sum_{u\in U} F_1(u)}{|U|} \ . \tag{17}$$

[13] Kadar računamo oceno, podano z enačbo (14), za določenega uporabnika *u*, levo stran enačbe zapišemo kot $F_1(u)$.

5. Ekscentriki pri izbiranju s sodelovanjem

V naših preliminarnih poizkusih smo opazili, da nekateri uporabniki prispevajo večjo napako od ostalih ne glede na izbiro algoritma in metrike. Skupna lastnost teh uporabnikov je, da so podali ocene, ki so izredno neusklajene s povprečnimi ocenami predmetov. Iz slednjega sledi, da so ti uporabniki nepovprečni uporabniki ali – kot smo jih poimenovali mi – "ekscentriki". Nadaljnja obravnava je razkrila, da se kakovost naprednejših algoritmov za izbiranje s sodelovanjem pokaže šele, ko izvedemo evalvacijo tudi preko skupine ekscentričnih uporabnikov. Premoč algoritmov tipa *k*-NN nad preprostim napovedovanjem povprečne ocene na primer postane veliko bolj očitna. V tem poglavju predlagamo definicijo mere za ekscentričnost, s katero v naslednjem poglavju razširjamo standardne evalvacijske metrike (opisane v poglavju 4) z upoštevanjem stopnje ekscentričnosti uporabnikov.

Velika slabost večine – če ne vseh – evalvacijskih metrik, predlaganih v kontekstu izbiranja s sodelovanjem, je, da so povprečene preko vseh uporabnikov, ki so vsi obravnavani enakovredno (tj. k evalvacijski oceni vsi prispevajo v enaki meri). Problemi nastanejo, če je domena taka, da je večina uporabnikov zadovoljna s priporočili za povprečnega uporabnika (tj. kadar je večina uporabnikov povprečnih, neekscentričnih). V takem primeru se napake, storjene pri napovedovanju ocen ekscentrikov, in posledično tudi nezadovoljstvo ekscentričnih uporabnikov, skrijejo v veliki množici povprečnih uporabnikov. Z drugimi besedami: interesi ekscentričnih uporabnikov so "izpovprečeni" zaradi narave evalvacijske metrike. Menimo, da mora biti obravnava ekscentričnih uporabnikov sestavni del vsakega dobro zasnovanega algoritma za izbiranje s sodelovanjem. To prepričanje sloni na intuiciji, da preprosto napovedovanje povprečne ocene v veliki meri zadostuje za povprečne uporabnike, medtem ko potrebujejo ostali uporabniki (bolj ali manj ekscentrični) bolj kompreksno obravnavo – *tu* nastopi izbiranje s sodelovanjem.

Da bi lahko pogledali na izbiranje s sodelovanjem z vidika ekscentričnih uporabnikov, moramo najprej definirati ekscentričnost kot izmerljivo količino. Idealnega povprečnega uporabnika lahko predstavimo z vektorjem značilk (oz. vektorjem predmetov), v katerem vsaka značilka predstavlja povprečno oceno za pripadajoči predmet[14]. Zdaj lahko ekscentričnost nekega uporabnika izračunamo tako, da izračunamo podobnost med njim in idealnim povprečnim uporabnikom – večja podobnost pomeni manjšo ekscentričnost.

V naših poizkusih smo za izmero podobnosti med uporabnikom in idelanim povprečnim uporabnikom uporabili normalizirano povprečno absolutno napako (NMAE), ki je ekvivalentna normalizirani manhattanski razdalji. V našem primeru je torej ekscentričnost definirana kot sledi:

$$\mathrm{ecc}(u) = \left(1 - \frac{\sum_{k \in R} | r_{u,k} - \overline{r_k} |}{| R | (r_{max} - r_{min})} \right)^{\beta} , \tag{18}$$

kjer je R množica vseh indeksov predmetov, ki so ocenjeni v obeh vektorjih, $r_{u,k}$ predstavlja oceno, ki jo je podal uporabnik u za predmet k, $\overline{r_k}$ predstavlja povprečno oceno predmeta k, r_{max} in r_{min} predstavljata maksimalno in minimalno vrednost ocene z ocenjevalne lestvice, β

[14] Z ozirom na pripadajočo matriko ocen.

pa je parameter za ojačanje ekscentričnosti, s katerim lahko poudarimo razlike med ekscentričnimi in povprečnimi uporabniki.

6. Predlagana razširitev evalvacijskih metrik

V tem razdelku so evalvacijske metrike, predstavljene v poglavju 4, razširjene z mero ekscentričnosti. Mero ekscentričnosti v metrike vključimo tako, da pri računanju končne (povprečne) evalvacijske ocene komponente napake, izračunane za posameznega uporabnika, obtežimo z ekscentričnostjo pripadajočega uporabnika.

Metriko MAE, merjeno preko vseh podatkov in predstavljeno z enačbo (6), razširimo kot sledi:

$$MAE_{ecc} = \frac{\sum_{u \in U, \atop r \in R_u^h} \left(| r - r_p | \operatorname{ecc}(u) \right)}{\sum_{u \in U} | R_u^h | \operatorname{ecc}(u)} \ , \tag{19}$$

kjer je U množica vseh uporabnikov, R_u^h množica prikritih ocen, ki jih je podal uporabnik u, r_p pa napovedana ocena, ki pripada dejanski oceni r.

Metrika mejnega obiska, predstavljena z enačbo (9), je razširjena z mero ekscentričnosti – v skladu s svojo osnovno obliko – kot sledi:

$$R_{ecc} = 100 \frac{\sum_{u \in U} \mathrm{R}(u) \operatorname{ecc}(u)}{\sum_{u \in U} \mathrm{R}^{\max}(u) \operatorname{ecc}(u)} \ . \tag{20}$$

$R(u)$ in $R^{max}(u)$ sta definirani v poglavju 4.2.

Srednja povprečna natančnost, merjena preko uporabnikov in predstavljena z enačbo (12), je razširjena kot sledi:

$$MAP_{ecc} = \frac{\sum_{u \in U} \mathrm{AP}(l_u) \operatorname{ecc}(u)}{\sum_{u \in U} \operatorname{ecc}(u)} \ , \tag{21}$$

kjer je l_u seznam predmetov, priporočen uporabniku u v procesu evalvacije. Mera AP(l) je definirana in razložena v poglavju 4.3. Enačba (21) predvideva, da algoritem med evalvacijo za vsakega uporabnika generira natanko en urejen seznam priporočil.

Nazadnje razširjamo še metriko F_1, predstavljeno z enačbo (17). Inačica, razširjena z mero ekscentričnosti, ima naslednjo obliko:

$$F_{1ecc} = \frac{\sum_{u \in U} \mathrm{F}_1(u) \operatorname{ecc}(u)}{\sum_{u \in U} \operatorname{ecc}(u)} \ . \tag{22}$$

Mera $F_1(u)$ je definirana v poglavju 4.4.

7. Empirična evalvacija

7.1 Eksperimentalna nastavitev

Izvedli smo zaporedje poizkusov, da bi ugotovili, kako se natančnost napovedovanja ocen in kakovost tvorjenja urejenih seznamov priporočil razlikujeta pri posameznih kombinacijah algoritma, domene in evalvacijske metrike. Uporabili smo evalvacijsko platformo, predstavljeno v poglavju 3.3, in tri različne podatkovne nabore (EachMovie, Jester in strežniški dnevniki), predstavljene v poglavju 3.2. Pri podatkovnih naborih Jester in EachMovie smo naključno izbrali 10.000 uporabnikov in s tem nekoliko pohitrili proces evalvacije. Ocene smo razdelili na dane in prikrite glede na evalvacijski protokol vse-razen-30-%, ki prikrije 30 % naključno izbranih ocen vsakega uporabnika, preostale ocene pa se uporabijo za formiranje sosesk ali gradnjo modelov (odvisno od uporabljenega algoritma). Pri algoritmih tipa *k*-NN, opisanih v poglavju 2.2, smo *k* nastavili na 120, ker pri tej velikosti soseske tovrstni algoritmi dosegajo najboljše rezultate [9].

Aplicirali smo štiri evalvacijske metrike:
- NMAE,
- metriko mejnega obiska,
- MAP in
- F_1.

Te evalvacijske metrike so podrobneje opisane v poglavju 4. Poleg tega smo uporabili tudi njihove ekscentrične inačice, definirane v poglavju 6, da bi videli razliko v rezultatih. Postavili smo hipotezo, da algoritmi, ki dosežejo najboljši/najslabši rezultat glede na določeno standardno metriko, ne dosežejo nujno najboljšega/najslabšega rezultata glede na pripadajočo ekscentrično metriko.

Pri metriki mejnega obiska, metriki MAP in metriki F_1 smo evalvirali sezname dolžine 15 predmetov. Parameter mejnega obiska smo nastavili na $\alpha = 7{,}5$ (glej poglavje 4.2), parameter za ojačanje ekscentričnosti pa smo pri ekscentričnih inačicah metrik nastavili na $\beta = 8$ (glej poglavje 5).

V evalvacijo smo vključili šest algoritmov za izbiranje s sodelovanjem:
- napovedovanje naključne ocene (*random predictor*),
- napovedovanje povprečne ocene (*popularity predictor*),
- *k* najbližjih sosedov s Pearsonovo formulo za izračun podobnosti (v nadaljevanju tudi "*k*-NN Pearson"),
- *k* najbližjih sosedov s kosinusno formulo za izračun podobnosti (v nadaljevanju tudi "*k*-NN kosinus"),
- SVM-klasifikator (*SVM classifier*) in
- SVM-regresijo (*SVM regression*).

Pri prvih dveh niti ne moremo govoriti o pravem izbiranju s sodelovanjem – to še posebej velja za napovedovanje naključne ocene. Algoritem za napovedovanje naključne ocene se ne ozira na ostale uporabnike in enostavno napove naključno vrednost z ocenjevalne lestvice. Tudi algoritem za napovedovanje povprečne ocene je trivialen – napovedana ocena $p_{u,i}$ je vedno kar povprečje ocen za predmet *i* izračunano preko vseh uporabnikov.

Kot je bilo že omenjeno, smo v evalvacijo vključili dva SVM-algoritma: SVM-klasifikator in SVM-regresijo. SVM-klasifikator v svoji osnovni (tj. binarni) obliki lahko umesti testni primer v enega od dveh razredov: *pozitivni* ali *negativni* razred. Da bi lahko napovedovali ocene, smo morali uporabiti eno od večrazrednih inačic SVM-klasifikatorja (razredi so v tem kontekstu posamezne ocene z ocenjevalne lestvice). V primeru podatkovnega nabora Jester smo morali za potrebe klasifikacije ocenjevalno lestvico diskretizirati[15] – ocenjevalni interval smo zato vzorčili z razmakom $^1/_3$.

Čeprav v delu [3] za učne/testne primere uporabljajo predmete, lahko algoritem preoblikujemo tako, da kot primere za strojno učenje obravnava uporabnike. Med tema dvema alternativama je priporočljivo izbirati glede na lastnosti podatkovnega nabora. Če je matrika ocen redkejša "horizontalno" (tj. povprečno število ocen na uporabnika je manjše od povprečnega števila ocen na predmet), potem za primere vzamemo uporabnike, in obratno. Intuitivno nam to da večje število učnih primerov za gradnjo modelov, ki so posledično bolj zanesljivi. Naj še omenimo, da slednja intuicija ne velja za algoritme tipa *k*-NN, kjer bi s takim razmišljanjem zmanjšali prekrivanje med primeri, kar bi se odražalo v slabših rezultatih. Glede na navedeno smo primere tvorili iz uporabnikov, ko smo SVM-algoritma aplicirali na EachMovie in Jester, in iz predmetov, ko smo ju aplicirali na strežniške dnevnike.

Večrazredna SVM-klasifikacija zahteva združitev več binarnih SVM-klasifikatorjev. Metodo, ki smo jo uporabili v ta namen, bomo razložili na primeru. Denimo, da imamo opravka z ocenjevalno lestvico z vrednostmi od 1 do 5. Zgraditi moramo 4 binarne klasifikatorje, da z njimi lahko vršimo klasifikacijo v 5 razredov. Prvi klasifikator razlikuje med primeri, ki sodijo v razred 1 (pozitivni primeri), in primeri, ki sodijo v enega od razredov 2–5 (negativni primeri). Drugi klasifikator razlikuje med primeri, ki sodijo v enega od razredov 1–2 (pozitivni primeri) in primeri, ki sodijo v enega od razredov 3–5 (negativni primeri), in tako naprej po istem principu. Zadnji klasifikator zna razlikovati med primeri, ki sodijo v enega od razredov 1–4 (pozitivni primeri), in primeri, ki sodijo v razred 5 (negativni primeri). Primer v enega od petih razredov klasificiramo tako, da povprašamo vsakega od štirih binarnih klasifikatorjev, ali je primer pozitiven ali negativen. Glede na vsak posamezni odgovor razredi prejemajo glasove (glasovanje, *voting*). Razred, ki prejme največ glasov, je končni rezultat naše večrazredne klasifikacije. Če ima več razredov enako število glasov, izračunamo povprečno vrednost teh razredov. Ta način klasifikacije imenujemo "glava proti repu" (*head versus tail*), da ga ločimo od standardnih večrazrednih SVM-klasifikatorjev tipa "vsak proti vsakemu" (*one versus one*) in "vsak proti vsem" (*one versus all*). V našem primeru smo uporabili binarni SVM-klasifikator, ki je del programske knjižnice Text Garden [11] z linearnim jedrom (*linear kernel*) in nastavitvijo *cost* = 0,1. Model smo zgradili le, če je bilo na voljo vsaj 7 pozitivnih in vsaj 7 negativnih primerov (naši preliminarni eksperimenti so pokazali, da je to primerna količina primerov, da se izognemo izgradnji nezanesljivih modelov).

Drugi algoritem tipa SVM, uporabljen v evalvaciji, je SVM-regresija, ki je po svoji obliki bolj primerna za realizacijo algoritma za izbiranje s sodelovanjem kot SVM-klasifikator. Neposredno lahko obravnava zvezne in torej tudi urejene diskretne vrednosti razredov. To pomeni, da je potrebno zgraditi en sam model; v primeru SVM-klasifikatorja moramo zgraditi več binarnih modelov. Uporabili smo SVM-regresijo, ki je del programske knjižnice LibSvm [5] z linearnim jedrom in nastavitvama $cost = 0{,}1$ in $\varepsilon = 0{,}1$. Podobno kot pri SVM-klasifikaciji smo model zgradili le, če je bilo na voljo vsaj 14 učnih primerov.

[15] Podatki iz domene Jester so bili zbrani na podlagi zvezne ocenjevalne lestvice.

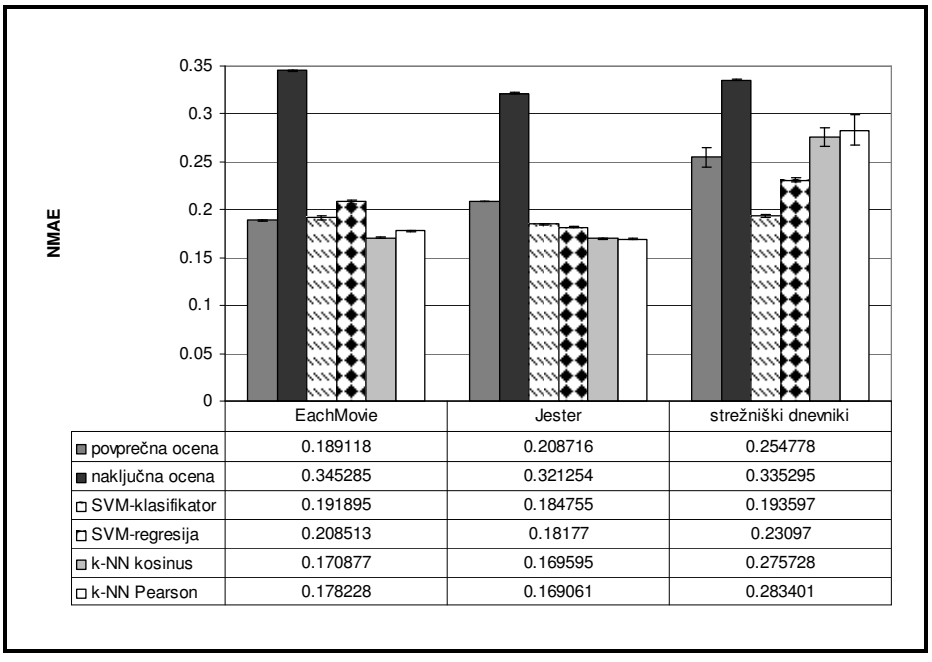
Evalvacijo smo ponovili petkrat za vsako trojico domene, algoritma in metrike. Vsakič smo drugače inicializirali generator naključnih števil in vzeli drugačen vzorec 10.000 uporabnikov iz podatkovnih naborov EachMovie in Jester.

7.2 Rezultati evalvacije

V tem poglavju predstavljamo rezultate eksperimentov, ki so podrobno opisani v poglavju 7.1.

Naj najprej povemo, da v nekaterih primerih izbrani algoritem ni znal napovedati nekaterih ocen, ker dane ocene niso predstavljale zadostne informacije za napoved. Algoritem za računanje povprečne ocene, na primer, ne more napovedati ocene za predmet, za katerega še nihče ni podal ocene. Pri naši evalvaciji smo izključili take primere iz obravnave, ker nas bolj zanima kakovost napovedi, ko je ta mogoča, četudi je delež napovedanih ocen v primerjavi s količino prikritih ocen majhen.

Pri prehodu s standardnih na ekscentrične metrike opazimo povečanje napake pri vseh algoritmih na vseh treh podatkovnih naborih in vseh evalvacijskih metrikah. Bolj zanimivo pa je dejstvo, da se razmerja med napakami posameznih algoritmov spremenijo (v kontekstu istega podatkovnega nabora) – nekateri primeri celo potrjujejo našo hipotezo, da algoritmi, ki dosežejo najboljši/najslabši rezultat glede na določeno standardno metriko, ne dosežejo nujno najboljšega/najslabšega rezultata glede na pripadajočo ekscentrično metriko.

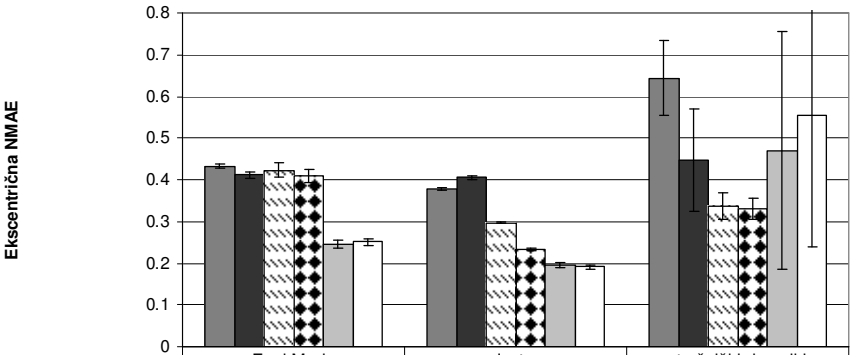


Slika 6: Rezultati evalvacije pri uporabi standardne metrike NMAE. Nižji stolpec predstavlja boljši rezultat. Indikatorji napake predstavljajo standardno deviacijo.

Poglejmo si najprej prehod z metrike NMAE (slika 6) na ekscentrično NMAE (slika 7). V primeru domene EachMovie smo opazili, da je pri uporabi ekscentrične inačice metrike

napaka napovedovanja povprečne ocene celo večja od napake, ki jo povzroči naključno napovedovanje ocene. To pomeni, da je ocene ekscentrikov bolje napovedovati naključno kot s povprečenjem ocen pripadajočega predmeta. Hkrati je napaka obeh SVM-algoritmov večja od napak algoritmov tipa *k*-NN (SVM-algoritma sta z vidika ekscentrikov primerljiva z naključnim in povprečnim napovedovanjem ocen[16]). V primeru domene Jester opazimo podoben preobrat glede algoritma za napovedovanje povprečne ocene, vendar tokrat SVM-algoritma ne dosežeta napake naključnega napovedovanja kot v primeru domene EachMovie – kljub temu pa še vedno ne dosegata kakovosti algoritmov tipa *k*-NN. Podatkovni nabor, izpeljan iz strežniških dnevnikov, ima precej drugačne lastnosti od podatkovnih naborov iz domen Jester in EachMovie (poleg tega, da so podatki zbrani implicitno na strani strežnika in da je matrika ocen izredno redka, je podatkovni nabor tudi izredno neuravnotežen) – iz tega sledijo drugačne spremembe v kakovosti evalviranih algoritmov pri prehodu na ekscentrične metrike. V tem primeru oba SVM-algoritma naredita najmanjšo napako pri napovedovanju, ne glede na inačico evalvacijske metrike NMAE. Oba algoritma tipa *k*-NN (ki sta se odrezala najbolje na obeh javno dostopnih podatkovnih naborih) napovedujeta ocene z napako, ki je primerljiva z napako naključnega napovedovanja (po prehodu na ekscentrično NMAE), medtem ko se napovedovanje povprečne ocene, ki je primerljivo s *k*-NN-algoritmoma glede na standardno NMAE, po prehodu na ekscentrično NMAE odreže najslabše.

Prav tako so zanimive nekatere absolutne in relativne razlike v napakah najboljšega in najslabšega algoritma pri prehodu na ekscentrično NMAE. V primeru domene EachMovie je



	EachMovie	Jester	strežniški dnevniki
povprečna ocena	0.43287	0.377559	0.644077
naključna ocena	0.411979	0.40525	0.447818
SVM-klasifikator	0.423072	0.296644	0.336261
SVM-regresija	0.41014	0.23376	0.330479
k-NN kosinus	0.245287	0.195308	0.469979
k-NN Pearson	0.250623	0.190604	0.554722

Slika 7: Rezultati evalvacije pri uporabi ekscentrične inačice metrike NMAE. Nižji stolpec predstavlja boljši rezultat. Indikatorji napake predstavljajo standardno deviacijo.

[16] To lahko razberemo iz tabele 3 (zgornji prikaz), ki prikazuje signifikantnost razlik med algoritmi pri uporabi ekscentrične inačice metrike NMAE. Iz omenjene tabele lahko na primer razberemo, da razlika (glede na ekscentrično inačico metrike NMAE in domeno EachMovie) med SVM-regresijo in napovedovanjem naključne ocene ni statistično signifikantna z ozirom na test signifikantnosti (*two-tailed paired t-test*) s signifikantnostnim nivojem 0,01, saj celica v tretjem stolpcu druge vrstice vsebuje vrednost, ki je večja od 0,01.

zmagovalni algoritem *k*-NN kosinus – ne glede na inačico metrike. Njegova napaka zraste z 0,17 na 0,25 pri prehodu na ekscentrično NMAE (povečanje napake za 47 %). Najslabši algoritem je pri standardni inačici metrike vsekakor napovedovanje naključne ocene, medtem ko je najslabši algoritem pri ekscentrični NMAE napovedovanje povprečne ocene (napovedovanje povprečne ocene, napovedovanje naključne ocene in oba SVM-algoritma napovedujejo ocene s primerljivo veliko napako). Napaka napovedovanja povprečne ocene zraste z 0,19 na 0,43 pri prehodu na ekscentrično inačico metrike (povečanje napake za 126 %). V primeru domene Jester je zmagovalni algoritem *k*-NN Pearson. Njegova napaka zraste z 0,17 na 0,19 pri prehodu na ekscentrično NMAE (povečanje napake za 12 %). Napovedovanje naključne ocene se ponovno obnese najslabše – tokrat glede na obe inačici evalvacijske metrike. NMAE tega algoritma zraste z 0,32 na 0,41 (povečanje napake za 28 %) ob prehodu na ekscentrično inačico metrike. Podatkovni nabor, izpeljan iz strežniških dnevnikov, ima, kot smo že omenili, precej drugačne lastnosti od preostalih dveh. Najboljši algoritem v tem kontekstu je – glede na NMAE – SVM-klasifikator. Njegova napaka zraste z 0,19 na 0,34 (povečanje napake za 79 %) ob prehodu na ekscentrično inačico metrike. Glede na standardno inačico metrike se najslabše obnese algoritem *k*-NN kosinus, medtem ko je glede na ekscentrično inačico metrike najslabši algoritem napovedovanje povprečne ocene. NMAE napovedovanja povprečne ocene zraste z 0,25 na 0,64 (povečanje napake za 156 %) ob prehodu na ekscentrično inačico metrike.

Pri drugih evalvacijskih metrikah so spremembe v kakovosti algoritmov po prehodu na ekscentrične metrike manj očitne (glej priloge). Pri aplikaciji metrike mejnega obiska in metrike MAP (tj. metrik, ki ocenjuje kakovost urejenenih seznamov priporočil) na javno dostopnih podatkovnih naborih se najbolje izkažeta algoritma tipa *k*-NN, tesno jima sledi SVM-klasifikator (tudi napovedovanje povprečne ocene se izkaže za dobro pri standardnih inačicah obeh metrik, a se napaka tega algoritma občutno poveča ob prehodu na ekscentrične inačice metrik). SVM-regresija in napovedovanje naključne ocene se v tem istem kontekstu izkažeta za najslabša algoritma (pri ekscentrični inačici metrik se jima približa tudi napovedovanje povprečne ocene). Po drugi strani pri aplikaciji teh dveh metrik (in njunih ekscentričnih inačic) na podatkih iz strežniških dnevnikov opazimo dobro delovanje SVM-regresije in napovedovanja povpreče ocene ter slabo delovanje SVM-klasifikatorja, algoritma za napovedovanje naključne ocene in obeh *k*-NN-algoritmov. Slednji rezultat pripisujemo izredno visoki stopnji redkosti matrike ocen, ki ohromi delovanje *k*-NN-algoritmov [10]. Ta opažanja niso popolnoma v skladu z opažanji pri aplikaciji metrike NMAE in njene ekscentrične inačice, kar potrjuje, da NMAE ni dobra metrika, kadar ocenjujemo sposobnost algoritma, da tvori urejene sezname priporočil [17].

Z aplikacijo metrike F_1 v domenah Jester in EachMovie opazimo manj konsistentne rezultate med obema domenama. Pri uporabi standardne inačice metrike se vsi algoritmi (z izjemo napovedovanja naključne ocene) dobro odrežejo. V primeru podatkovnega nabora EachMovie ob prehodu na ekscentrično inačico metrike odpove algoritem *k*-NN kosinus, medtem ko v primeru podatkovnega nabora Jester odpovesta algoritma tipa SVM. V primeru strežniških dnevnikov se najbolje odrežeta napovedovanje povprečne ocene in SVM-regresija. Slednje postane le še bolj očitno po prehodu na ekscentrično inačico metrike.

EachMovie, NMAE

	Naključna ocena	SVM-klasifikator	SVM-regresija	k-NN kosinus	k-NN Pearson
Povprečna ocena	**7,67E–12**	0,034303	**4,08E–05**	**2,33E–07**	**3,93E–06**
Naključna ocena		**7,7E–09**	**1,88E–08**	**7,16E–12**	**3,67E–11**
SVM-klasifikator			**6,36E–07**	**2,88E–05**	**0,000204**
SVM-regresija				**4,23E–06**	**1,28E–05**
k-NN kosinus					**1,33E–07**

Jester, NMAE

	Naključna ocena	SVM-klasifikator	SVM-regresija	k-NN kosinus	k-NN Pearson
Povprečna ocena	**2,91E–11**	**5,93E–07**	**6,77E–07**	**9,88E–09**	**2,12E–08**
Naključna ocena		**1,94E–09**	**2,55E–09**	**2,45E–10**	**3,98E–10**
SVM-klasifikator			**1,68E–05**	**1,52E–06**	**1,32E–06**
SVM-regresija				**9,97E–06**	**8,22E–06**
k-NN kosinus					**0,000978**

Strežniški dnevniki, NMAE

	Naključna ocena	SVM-klasifikator	SVM-regresija	k-NN kosinus	k-NN Pearson
Povprečna ocena	**8,82E–05**	**0,000172**	0,011362	**0,00068**	**0,004395**
Naključna ocena		**8,87E–09**	**8,45E–09**	**0,000142**	**0,001701**
SVM-klasifikator			**9,39E–06**	**3,27E–05**	**0,000152**
SVM-regresija				**0,000497**	**0,001671**
k-NN kosinus					0,168975

Tabela 2: Tabele signifikantnosti razlik v rezultatih pri uporabi standardne metrike NMAE nad vsemi tremi domenami. Krepko natisnjene vrednosti predstavljajo statistično signifikantnost v razliki ocene dveh algoritmov z ozirom na test signifikantnosti (*two-tailed paired t-test*) s signifikantnostnim nivojem 0,01.

EachMovie, ekscentrična NMAE

	Naključna ocena	SVM-klasifikator	SVM-regresija	k-NN kosinus	k-NN Pearson
Povprečna ocena	0,010089	0,193142	0,0136	**1,33E–06**	**1,58E–06**
Naključna ocena		0,357989	0,851347	**2,71E–07**	**4,47E–08**
SVM-klasifikator			0,050354	**6,33E–05**	**7,49E–05**
SVM-regresija				**5,46E–05**	**6,44E–05**
k-NN kosinus					**0,007626**

Jester, ekscentrična NMAE

	Naključna ocena	SVM-klasifikator	SVM-regresija	k-NN kosinus	k-NN Pearson
Povprečna ocena	**0,000205**	**6,22E–08**	**5,84E–08**	**1,37E–06**	**1,01E–06**
Naključna ocena		**1,5E–06**	**5,83E–07**	**2,01E–06**	**1,54E–06**
SVM-klasifikator			**4,82E–07**	**6,67E–06**	**4,18E–06**
SVM-regresija				**0,000194**	**8,49E–05**
k-NN kosinus					**0,000837**

Strežniški dnevniki, ekscentrična NMAE

	Naključna ocena	SVM-klasifikator	SVM-regresija	k-NN kosinus	k-NN Pearson
Povprečna ocena	0,064315	**0,001418**	**0,00135**	0,268078	0,576585
Naključna ocena		0,122137	0,122277	0,805103	0,396378
SVM-klasifikator			0,407524	0,371739	0,194485
SVM-regresija				0,361014	0,195235
k-NN kosinus					0,447779

Tabela 3: Tabele signifikantnosti razlik v rezultatih pri uporabi ekscentrične inačice metrike NMAE nad vsemi tremi domenami. Krepko natisnjene vrednosti predstavljajo statistično signifikantnost v razliki ocene dveh algoritmov z ozirom na test signifikantnosti (*two-tailed paired t-test*) s signifikantnostnim nivojem 0,01.

8. Zaključki

Svetovni splet (pri tem mislimo tudi na spletne trgovine in podobne spletne storitve) predstavlja praktično neomejen vir vsebin, izdelkov in storitev, zato se je pomen izbiranja s sodelovanjem v zadnjih letih močno povečal. Priča smo mnogim raziskavam na tem področju in uspešnim aplikacijam v komercialnih sistemih. Osnovna prednost klasičnega izbiranja s sodelovanjem je neodvisnost od vsebine predmetov, zaradi česar je ta postopek uporaben tudi v primerih, ko vsebine niso na voljo ali pa jih je težko formulirati (slike, artikli v trgovini, glasba idr.).

Pri pregledu literature smo ugotovili, da je kljub perspektivnim raziskavam na mnogih področjih računalniških znanosti (teorija grafov, statistične metode, strojno učenje idr.) metoda *k* najbližjih sosedov še vedno najpogosteje uporabljena metoda za izbiranje s sodelovanjem. Dobro delovanje te metode je potrdila tudi naša empirična evalvacija, ki pa je razkrila tudi njeno slabost – če je matrika ocen preredka, potem metoda *k* najbližjih sosedov odpove [10]. V takem primeru so uspešnejši določeni klasifikacijski in regresijski modeli.

Kar nekaj raziskav se dotika vprašanja kakovosti metrik za evalvacijo izbiranja s sodelovanjem [12][17]. Problemi nastanejo pri domenah, pri katerih je večina uporabnikov zadovoljna s "povprečnimi priporočili" (tj. kadar je večina uporabnikov povprečnih, neekscentričnih). V takem primeru se napake, storjene pri napovedovanju ocen ekscentrikov, in posledično tudi nezadovoljstvo ekscentričnih uporabnikov skrijejo v veliki množici povprečnih uporabnikov. Z drugimi besedami: interesi ekscentričnih uporabnikov so "izpovprečeni" zaradi narave evalvacijske metrike. Menimo, da mora biti obravnava ekscentričnih uporabnikov sestavni del vsakega dobro zasnovanega algoritma za izbiranje s sodelovanjem. To prepričanje sloni na intuiciji, da preprosto napovedovanje povprečne ocene v veliki meri zadostuje za povprečne uporabnike, medtem ko potrebujejo ostali (bolj ali manj ekscentrični) uporabniki bolj kompleksno obravnavo.

Evalvacijske metrike smo spremenili z vključitvijo ekscentričnosti uporabnikov. Postavili smo hipotezo, da algoritmi, ki dosežejo najboljši/najslabši rezultat glede na določeno standardno metriko, ne dosežejo nujno najboljšega/najslabšega rezultata glede na pripadajočo modificirano metriko. Evalvacija je pokazala, da pride pri prehodu s standardnih na modificirane metrike do povečanja napake pri vseh algoritmih na vseh treh domenah in pri vseh evalvacijskih metrikah. Še bolj zanimivo pa je dejstvo, da se razmerja med napakami posameznih algoritmov spremenijo (v kontekstu iste domene) in v nekaterih primerih potrjujejo našo hipotezo.

Glavni rezultati te diplomske naloge so sledeči:

- zbrana in povzeta je literatura s področja izbiranja s sodelovanjem,
- opisan je postopek predobdelave strežniških dnevnikov; izpostavljeni so problemi, ki so se pojavili v tem kontekstu,
- implementirana sta program za predobdelavo strežniških dnevnikov in platforma za evalvacijo algoritmov za izbiranje s sodelovanjem,
- predlagana je sprememba evalvacijskih metrik z vključitvijo ekscentričnosti uporabnikov; predstavljen je svež pogled na izbiranje s sodelovanjem, pri katerem postane ekscentrični uporabnik eno od poglavitnih meril za kakovost algoritma,

- podani so rezultati empirične evalvacije šestih različnih algoritmov (dva od teh sta zgolj "naivni" rešitvi) nad tremi različnimi domenami; pri tem so uporabljene štiri različne evalvacijske metrike in njihove modificirane inačice.

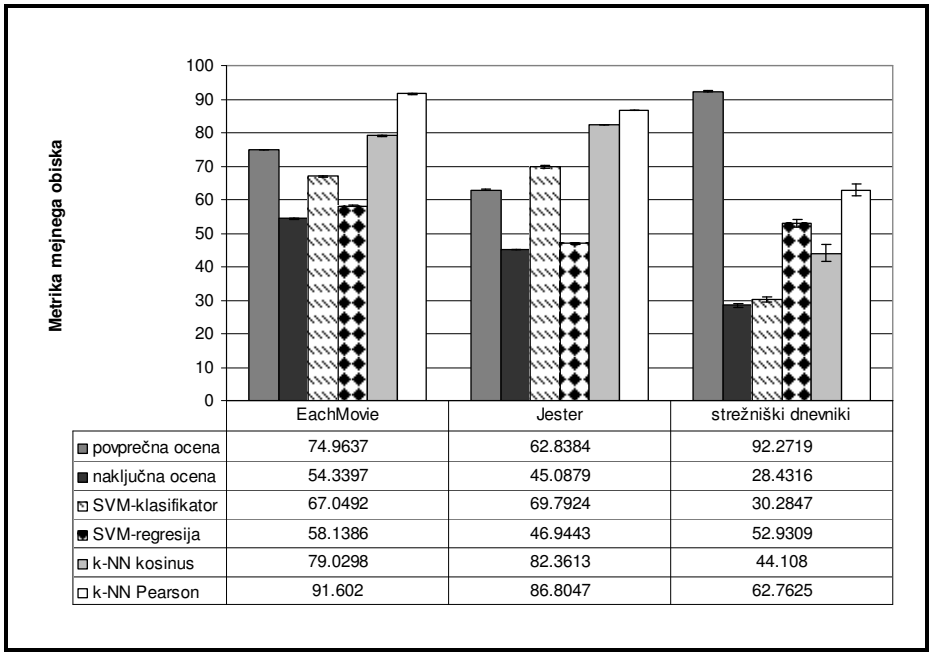
To delo ponuja dober vpogled v izbiranje s sodelovanjem in razkriva probleme, ki se pojavljajo v realnih situacijah, mnoga vprašanja pa pušča še odprta. Pri karakteristiki podatkov se nismo dotaknili problema neuravnoteženega podatkovnega nabora (tak je na primer podatkovni nabor iz domene strežniških dnevnikov). Prav tako kljub izjemno veliki redkosti matrike ocen iz strežniških dnevnikov nismo uporabili metod za zmanjšanje razsežnosti (kot na primer LSI ali razvrščanje v skupine). Odprto ostaja tudi vprašanje, kolikšno izboljšanje rezultatov bi pomenilo upoštevanje tekstovnih vsebin in/ali strukture. Vse tri domene imajo namreč pripadajoče tekstovne vsebine (opis filma, besedilo šale in besedilo znanstvene publikacije), te pa so razvrščene v taksonomije (filmi na primer po žanrih ali režiserjih, šale in znanstvene publikacije pa po tematikah). Pravilna vključitev dodatne informacije v obravnavo namreč intuitivno narekuje izboljšanje rezultata.

Pri naši evalvaciji smo primere, ko algoritem ni znal napovedati ocene zaradi pomanjkanja podatkov, izključili iz obravnave. Kljub temu je delež ocen, ki jih algoritem zna napovedati pri danih podatkih, pomemben kvalitativni faktor algoritma za izbiranje s sodelovanjem. Algoritme bi bilo potrebno primerjati tudi s tega vidika.

V empirično evalvacijo bi lahko vključili še druge metode, za katere avtorji trdijo, da izboljšajo rezultate (predmetno izbiranje s sodelovanjem z umerjeno kosinusno podobnostjo, metodo privzete ocene, *horting* idr.). Prav tako bi bilo smiselno preizkusiti nekatere druge klasifikacijske in regresijske algoritme, uporabiti bolj strokoven pristop pri preslikavi implicitnih ocen na eksplicitno ocenjevalno lestvico (v primeru domene strežniških dnevnikov), preizkusiti še druge mere za ekscentričnost (osnovane na drugih merah za razdaljo/podobnost) in prikazati rezultate evalvacij še v odvisnosti od nekaterih drugih parametrov (kot so parameter mejnega obiska, parameter za ojačanje ekscentričnosti, dolžina seznama priporočil idr.).

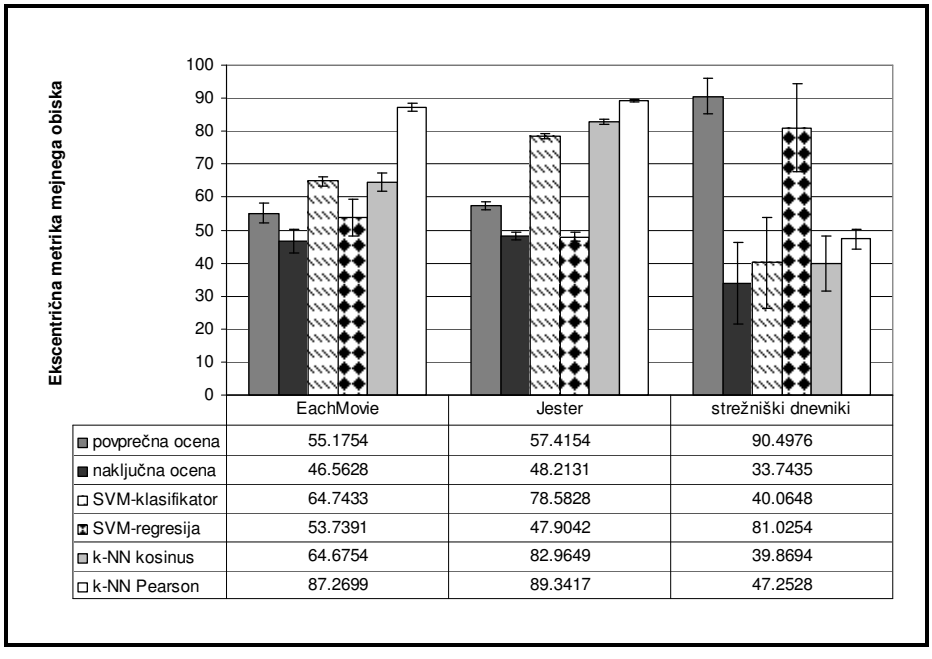
Kot pri vsakem raziskovalnem delu smo pri iskanju odgovorov na nekatera vprašanja odprli tudi mnoga nova vprašanja. Na srečo razvita evalvacijska platforma predstavlja dobro podlago za izvedbo nadaljnjih eksperimentov in pridobitev odgovorov tudi na ta vprašanja.

9. Priloge



	EachMovie	Jester	strežniški dnevniki
povprečna ocena	74.9637	62.8384	92.2719
naključna ocena	54.3397	45.0879	28.4316
SVM-klasifikator	67.0492	69.7924	30.2847
SVM-regresija	58.1386	46.9443	52.9309
k-NN kosinus	79.0298	82.3613	44.108
k-NN Pearson	91.602	86.8047	62.7625

Slika 8: Rezultati evalvacije pri uporabi standardne metrike mejnega obiska. Višji stolpec predstavlja boljši rezultat. Indikatorji napake predstavljajo standardno deviacijo.



	EachMovie	Jester	strežniški dnevniki
povprečna ocena	55.1754	57.4154	90.4976
naključna ocena	46.5628	48.2131	33.7435
SVM-klasifikator	64.7433	78.5828	40.0648
SVM-regresija	53.7391	47.9042	81.0254
k-NN kosinus	64.6754	82.9649	39.8694
k-NN Pearson	87.2699	89.3417	47.2528

Slika 9: Rezultati evalvacije pri uporabi ekscentrične inačice metrike mejnega obiska. Višji stolpec predstavlja boljši rezultat. Indikatorji napake predstavljajo standardno deviacijo.

EachMovie, metrika mejnega obiska

	Naključna ocena	SVM-klasifikator	SVM-regresija	k-NN kosinus	k-NN Pearson
Povprečna ocena	**1,67E–08**	**6,26E–07**	**3,33E–09**	**2,78E–08**	**1,12E–09**
Naključna ocena		**3,28E–07**	**8,27E–06**	**4,17E–09**	**2,69E–10**
SVM-klasifikator			**7,53E–07**	**1,12E–07**	**3,95E–09**
SVM-regresija				**1,08E–09**	**1,53E–10**
k-NN kosinus					**9,04E–10**

Jester, metrika mejnega obiska

	Naključna ocena	SVM-klasifikator	SVM-regresija	k-NN kosinus	k-NN Pearson
Povprečna ocena	**1,74E–08**	**1,07E–06**	**2,2E–08**	**2,6E–08**	**2,77E–09**
Naključna ocena		**4,08E–09**	**1,72E–05**	**7,71E–11**	**5,24E–11**
SVM-klasifikator			**1,86E–08**	**1,81E–07**	**3,66E–08**
SVM-regresija				**8,23E–10**	**3,45E–10**
k-NN kosinus					**1,05E–07**

Strežniški dnevniki, metrika mejnega obiska

	Naključna ocena	SVM-klasifikator	SVM-regresija	k-NN kosinus	k-NN Pearson
Povprečna ocena	**6,3E–09**	**4,13E–09**	**4,37E–07**	**2,7E–06**	**3,57E–06**
Naključna ocena		0,037352	**3,46E–06**	**0,000128**	**4,3E–07**
SVM-klasifikator			**2,79E–06**	**0,000312**	**6,61E–06**
SVM-regresija				**0,000936**	**0,001275**
k-NN kosinus					**0,000113**

Tabela 4: Tabele signifikantnosti razlik v rezultatih pri uporabi standardne metrike mejnega obiska nad vsemi tremi domenami. Krepko natisnjene vrednosti predstavljajo statistično signifikantnost v razliki ocene dveh algoritmov z ozirom na test signifikantnosti (*two-tailed paired t-test*) s signifikantnostnim nivojem 0,01.

EachMovie, ekscentrična metrika mejnega obiska

	Naključna ocena	SVM-klasifikator	SVM-regresija	k-NN kosinus	k-NN Pearson
Povprečna ocena	**0,000872**	**0,001589**	0,594895	**7,82E–06**	**9,65E–06**
Naključna ocena		**0,000538**	0,083593	**4,71E–05**	**1,77E–05**
SVM-klasifikator			**0,003537**	0,958581	**3,2E–06**
SVM-regresija				**0,013972**	**0,000133**
k-NN kosinus					**2,22E–05**

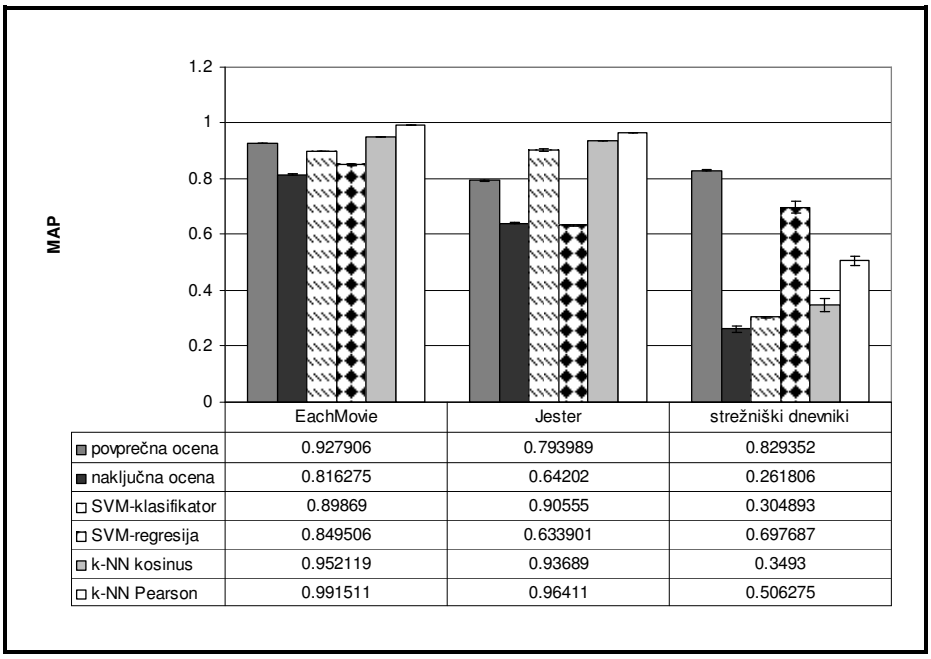
Jester, ekscentrična metrika mejnega obiska

	Naključna ocena	SVM-klasifikator	SVM-regresija	k-NN kosinus	k-NN Pearson
Povprečna ocena	**2,17E–05**	**2,78E–06**	**9,79E–05**	**9,7E–08**	**1,18E–07**
Naključna ocena		**1,67E–07**	0,571984	**2,17E–07**	**4,13E–08**
SVM-klasifikator			**5,49E–08**	**0,000838**	**1,62E–06**
SVM-regresija				**6,94E–07**	**8,18E–08**
k-NN kosinus					**1,23E–05**

Strežniški dnevniki, ekscentrična metrika mejnega obiska

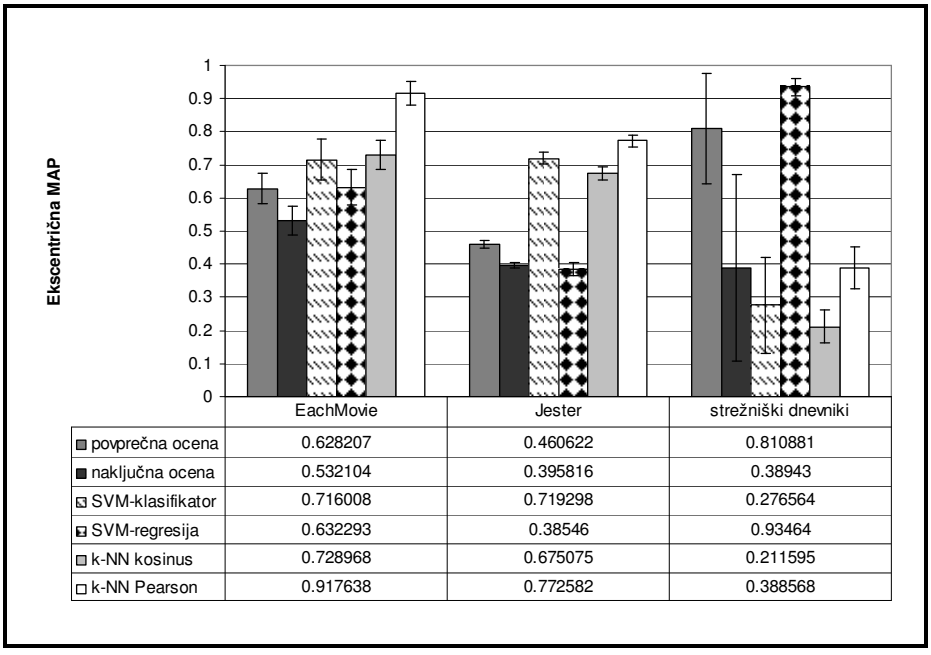
	Naključna ocena	SVM-klasifikator	SVM-regresija	k-NN kosinus	k-NN Pearson
Povprečna ocena	**0,000887**	**0,000216**	0,272523	**0,000483**	**5,95E–05**
Naključna ocena		0,550962	**0,009104**	0,401855	0,103565
SVM-klasifikator			0,016713	0,982327	0,287512
SVM-regresija				**0,001043**	**0,005106**
k-NN kosinus					0,134039

Tabela 5: Tabele signifikantnosti razlik v rezultatih pri uporabi ekscentrične inačice metrike mejnega obiska nad vsemi tremi domenami. Krepko natisnjene vrednosti predstavljajo statistično signifikantnost v razliki ocene dveh algoritmov z ozirom na test signifikantnosti (*two-tailed paired t-test*) s signifikantnostnim nivojem 0,01.



	EachMovie	Jester	strežniški dnevniki
povprečna ocena	0.927906	0.793989	0.829352
naključna ocena	0.816275	0.64202	0.261806
SVM-klasifikator	0.89869	0.90555	0.304893
SVM-regresija	0.849506	0.633901	0.697687
k-NN kosinus	0.952119	0.93689	0.3493
k-NN Pearson	0.991511	0.96411	0.506275

Slika 10: Rezultati evalvacije pri uporabi standardne metrika MAP. Višji stolpec predstavlja boljši rezultat. Indikatorji napake predstavljajo standardno deviacijo.



	EachMovie	Jester	strežniški dnevniki
povprečna ocena	0.628207	0.460622	0.810881
naključna ocena	0.532104	0.395816	0.38943
SVM-klasifikator	0.716008	0.719298	0.276564
SVM-regresija	0.632293	0.38546	0.93464
k-NN kosinus	0.728968	0.675075	0.211595
k-NN Pearson	0.917638	0.772582	0.388568

Slika 11: Rezultati evalvacije pri uporabi ekscentrične inačice metrike MAP. Višji stolpec predstavlja boljši rezultat. Indikatorji napake predstavljajo standardno deviacijo.

EachMovie, MAP

	Naključna ocena	SVM-klasifikator	SVM-regresija	k-NN kosinus	k-NN Pearson
Povprečna ocena	**2,36E–08**	**1,16E–05**	**6,32E–07**	**1,85E–08**	**2,41E–08**
Naključna ocena		**2,24E–07**	**3,01E–05**	**5,54E–09**	**2,98E–09**
SVM-klasifikator			**3,07E–06**	**6,59E–07**	**1,17E–08**
SVM-regresija				**1,64E–07**	**2,51E–08**
k-NN kosinus					**4,91E–08**

Jester, MAP

	Naključna ocena	SVM-klasifikator	SVM-regresija	k-NN kosinus	k-NN Pearson
Povprečna ocena	**3,16E–09**	**1,08E–06**	**2,37E–09**	**5,28E–09**	**8,62E–10**
Naključna ocena		**5,9E–08**	**1,01E–05**	**7,02E–12**	**9,39E–13**
SVM-klasifikator			**5,49E–08**	**0,000289**	**1,78E–05**
SVM-regresija				**8,51E–11**	**8,71E–12**
k-NN kosinus					**1,58E–07**

Strežniški dnevniki, MAP

	Naključna ocena	SVM-klasifikator	SVM-regresija	k-NN kosinus	k-NN Pearson
Povprečna ocena	**1,12E–07**	**2,28E–10**	**8,61E–05**	**7,4E–07**	**2,18E–06**
Naključna ocena		**0,00172**	**4,09E–06**	**0,002677**	**1,89E–05**
SVM-klasifikator			**1,24E–06**	**0,007026**	**1,98E–05**
SVM-regresija				**1,25E–06**	**3,9E–05**
k-NN kosinus					**0,000304**

Tabela 6: Tabele signifikantnosti razlik v rezultatih pri uporabi standardne metrike MAP nad vsemi tremi domenami. Krepko natisnjene vrednosti predstavljajo statistično signifikantnost v razliki ocene dveh algoritmov z ozirom na test signifikantnosti (*two-tailed paired t-test*) s signifikantnostnim nivojem 0,01.

EachMovie, ekscentrična MAP

	Naključna ocena	SVM-klasifikator	SVM-regresija	k-NN kosinus	k-NN Pearson
Povprečna ocena	**0,001781**	**0,001198**	0,840952	**7,52E–06**	**0,000185**
Naključna ocena		**0,001242**	0,028823	**0,000193**	**6,81E–05**
SVM-klasifikator			0,010134	0,279636	**0,001678**
SVM-regresija				**0,004124**	**0,000268**
k-NN kosinus					**0,000651**

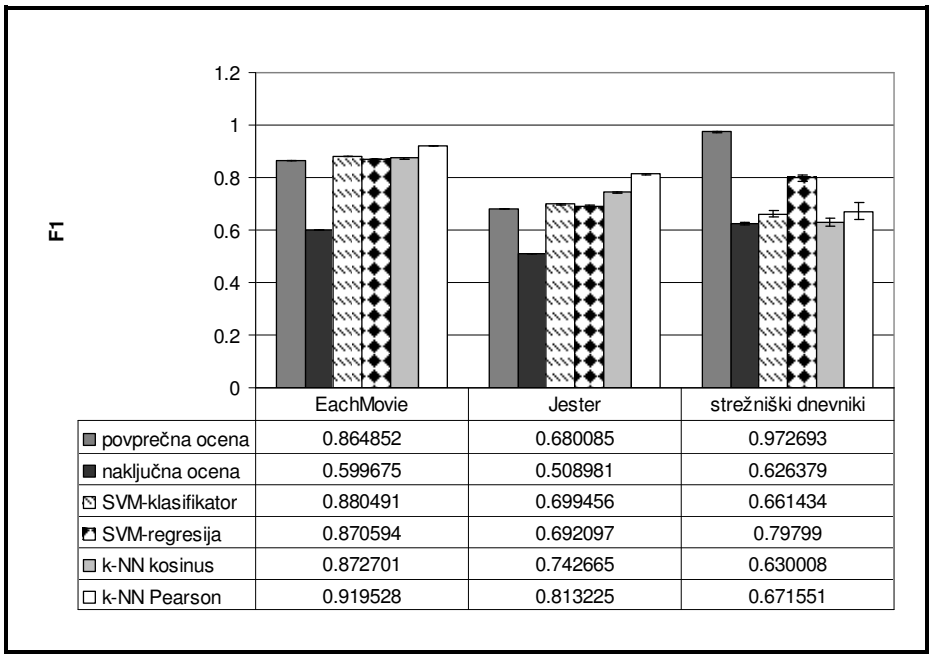
Jester, ekscentrična MAP

	Naključna ocena	SVM-klasifikator	SVM-regresija	k-NN kosinus	k-NN Pearson
Povprečna ocena	**0,000144**	**8,29E–06**	**0,000366**	**2,39E–06**	**1,57E–07**
Naključna ocena		**1,6E–06**	0,28531	**2,15E–06**	**5,48E–07**
SVM-klasifikator			**3,62E–06**	0,0126	**0,009666**
SVM-regresija				**1E–05**	**2,29E–06**
k-NN kosinus					**5,45E–06**

Strežniški dnevniki, ekscentrična MAP

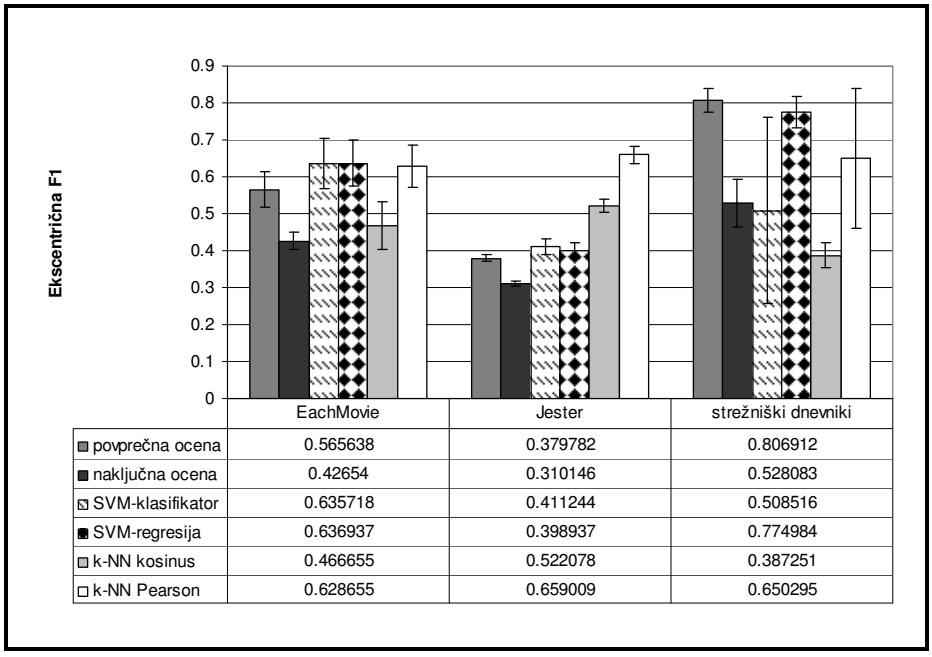
	Naključna ocena	SVM-klasifikator	SVM-regresija	k-NN kosinus	k-NN Pearson
Povprečna ocena	0,053288	0,017647	0,161531	**0,002203**	**0,00574**
Naključna ocena		0,454223	0,013079	0,228659	0,995337
SVM-klasifikator			**0,000761**	0,358829	0,181033
SVM-regresija				**7,1E–06**	**7,53E–05**
k-NN kosinus					**0,00126**

Tabela 7: Tabele signifikantnosti razlik v rezultatih pri uporabi ekscentrične inačice metrike MAP nad vsemi tremi domenami. Krepko natisnjene vrednosti predstavljajo statistično signifikantnost v razliki ocene dveh algoritmov z ozirom na test signifikantnosti (*two-tailed paired t-test*) s signifikantnostnim nivojem 0,01.



	EachMovie	Jester	strežniški dnevniki
povprečna ocena	0.864852	0.680085	0.972693
naključna ocena	0.599675	0.508981	0.626379
SVM-klasifikator	0.880491	0.699456	0.661434
SVM-regresija	0.870594	0.692097	0.79799
k-NN kosinus	0.872701	0.742665	0.630008
k-NN Pearson	0.919528	0.813225	0.671551

Slika 12: Rezultati evalvacije pri uporabi standardne metrika F_1. Višji stolpec predstavlja boljši rezultat. Indikatorji napake predstavljajo standardno deviacijo.



	EachMovie	Jester	strežniški dnevniki
povprečna ocena	0.565638	0.379782	0.806912
naključna ocena	0.42654	0.310146	0.528083
SVM-klasifikator	0.635718	0.411244	0.508516
SVM-regresija	0.636937	0.398937	0.774984
k-NN kosinus	0.466655	0.522078	0.387251
k-NN Pearson	0.628655	0.659009	0.650295

Slika 13: Rezultati evalvacije pri uporabi ekscentrične inačice metrike F_1. Višji stolpec predstavlja boljši rezultat. Indikatorji napake predstavljajo standardno deviacijo.

EachMovie, F_1

	Naključna ocena	SVM-klasifikator	SVM-regresija	k-NN kosinus	k-NN Pearson
Povprečna ocena	**1,06E–11**	**7,39E–05**	**0,005589**	**2,1E–05**	**3,28E–08**
Naključna ocena		**1,06E–09**	**2,11E–09**	**3,45E–11**	**7,35E–11**
SVM-klasifikator			**1,91E–06**	**0,000232**	**1,37E–07**
SVM-regresija				0,049864	**1,52E–07**
k-NN kosinus					**1,59E–09**

Jester, F_1

	Naključna ocena	SVM-klasifikator	SVM-regresija	k-NN kosinus	k-NN Pearson
Povprečna ocena	**6,62E–09**	**0,000327**	**0,002432**	**3,73E–08**	**9,72E–09**
Naključna ocena		**1,66E–08**	**2,6E–08**	**1,92E–09**	**9,12E–10**
SVM-klasifikator			**4,5E–07**	**5,71E–06**	**3,55E–08**
SVM-regresija				**3,74E–06**	**3,28E–08**
k-NN kosinus					**2,98E–08**

Strežniški dnevniki, F_1

	Naključna ocena	SVM-klasifikator	SVM-regresija	k-NN kosinus	k-NN Pearson
Povprečna ocena	**1,49E–08**	**5,17E–07**	**4,5E–06**	**6,77E–07**	**3,73E–05**
Naključna ocena		**0,000585**	**1,07E–05**	0,687024	0,041704
SVM-klasifikator			**0,000147**	0,03808	0,497274
SVM-regresija				**3,9E–05**	**0,003101**
k-NN kosinus					0,052672

Tabela 8: Tabele signifikantnosti razlik v rezultatih pri uporabi standardne metrike F_1 nad vsemi tremi domenami. Krepko natisnjene vrednosti predstavljajo statistično signifikantnost v razliki ocene dveh algoritmov z ozirom na test signifikantnosti (*two-tailed paired t-test*) s signifikantnostnim nivojem 0,01.

EachMovie, ekscentrična F_1

	Naključna ocena	SVM-klasifikator	SVM-regresija	k-NN kosinus	k-NN Pearson
Povprečna ocena	**0,000289**	0,021294	0,013883	**0,001236**	0,013104
Naključna ocena		**0,000566**	**0,000342**	0,112586	**0,000506**
SVM-klasifikator			0,707098	**0,001643**	0,814353
SVM-regresija				**0,00146**	0,775043
k-NN kosinus					**0,000265**

Jester, ekscentrična F_1

	Naključna ocena	SVM-klasifikator	SVM-regresija	k-NN kosinus	k-NN Pearson
Povprečna ocena	**1,83E–05**	**0,007797**	0,051185	**7,69E–05**	**7,11E–06**
Naključna ocena		**0,000143**	**0,000295**	**5,45E–06**	**1,88E–06**
SVM-klasifikator			**6,1E–05**	**0,00013**	**1,5E–06**
SVM-regresija				**8,07E–05**	**9,35E–07**
k-NN kosinus					**1,38E–05**

Strežniški dnevniki, ekscentrična F_1

	Naključna ocena	SVM-klasifikator	SVM-regresija	k-NN kosinus	k-NN Pearson
Povprečna ocena	**0,000584**	0,044311	0,195187	**0,000114**	0,171418
Naključna ocena		0,852082	**0,002267**	**0,008461**	0,296867
SVM-klasifikator			0,069358	0,365512	0,405489
SVM-regresija				**0,000121**	0,238575
k-NN kosinus					0,031099

Tabela 9: Tabele signifikantnosti razlik v rezultatih pri uporabi ekscentrične inačice metrike F_1 nad vsemi tremi domenami. Krepko natisnjene vrednosti predstavljajo statistično signifikantnost v razliki ocene dveh algoritmov z ozirom na test signifikantnosti (*two-tailed paired t-test*) s signifikantnostnim nivojem 0,01.

Viri

[1] C. C. Aggarwal, J. L. Wolf, K.-L. Wu, P. S. Yu, "Horting hatches an egg: A new graph-theoretic approach to collaborative filtering," v zborniku *5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1999.
[2] P. Baldi, P. Frasconi, P. Smyth, *Modeling the Internet and the Web: Probabilistic Methods and Algorithms*, New York: Wiley, 2003.
[3] D. Billsus and M. J. Pazzani, "Learning collaborative information filers," v zborniku *15th International Conference on Machine Learning*, 1998.
[4] J. S. Breese, D. Heckerman, C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," v zborniku *14th Conference on Uncertainty in Artificial Intelligence*, 1998.
[5] C.-C. Chang and C.-J. Lin, *LibSvm: A Library for Support Vector Machines*, 2001.
Programska oprema dostopna na: http://www.csie.ntu.edu.tw/~cjlin/libsvm
[6] D. M. Chickering, D. Heckerman, C. Meek, "A bayesian approach to learning bayesian networks with local structure," v zborniku *13th Conference on Uncertainty in Artificial Intelligence*, 1997.
[7] M. Claypool, P. Le, M. Wased, D. Brown, "Implicit interest indicators," v zborniku *ACM 2001 Intelligent User Interfaces Conference*, 2001.
[8] S. Deerwester, S. T. Dumais, R. Harshman, "Indexing by latent semantic analysis," *Journal of the Society for Information Science*, št. 6, zv. 41, str. 391–407, 1990.
[9] K. Goldberg, T. Roeder, D. Gupta, C. Perkins, "Eigentaste: A constant time collaborative filtering algorithm," *Information Retrieval*, št. 4, str. 133–151, 2001.
[10] M. Grcar, D. Mladenic, B. Fortuna, M. Grobelnik, "Data sparsity issues in the collaborative filtering framework," v zborniku *WebKDD, Lecture Notes in Artificial Intelligence 4198*, 2007 (v postopku objave).
[11] M. Grobelnik, D. Mladenic, *Text Mining Recipes*, Berlin, Heidelberg, New York: Springer-Verlag, 2007 (v postopku objave).
Programska oprema dostopna na: http://www.textmining.net
[12] J. L. Herlocker, J. A. Konstan, L. G. Terveen, J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems*, št. 1, zv. 22, str. 5–53, 2004.
[13] T. Hofmann, "Probabilistic latent semantic analysis," v zborniku *15th Conference on Uncertainty in Artificial Intelligence*, 1999.
[14] T. Hofmann, "Latent semantic models for collaborative filtering," *ACM Transactions on Information Systems*, št. 1, zv. 22, str. 89–115, 2004.
[15] I. Kononenko, *Strojno učenje*, Ljubljana: Založba FE in FRI, 2005.
[16] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, J. Riedl, "Grouplens: Applying collaborative filtering to usenet news," *Communications of the ACM*, št. 3, zv. 40, str. 77–87, 1997.
[17] M. R. McLaughlin, J. L. Herlocker, "A collaborative filtering algorithm and evaluation metric that accurately model the user experience," v zborniku *27th Annual International Conference on Research and Development in Information Retrieval*, 2004.
[18] P. Melville, R. J. Mooney, R. Nagarajan, "Content-boosted collaborative filtering for improved recommendations," v zborniku *18th National Conference on Artificial Intelligence*, 2002.
[19] P. Resnick, N. Iaocvou, M. Suchak, P. Bergstrom, J. Riedl, "Grouplens: An open architecture for collaborative filtering for netnews," v zborniku *ACM 1994 Conference on Computer Supported Cooperative Work*, str. 175–186, 1994.

[20] M. Rosenstein and C. Lochbaum, "What is actually taking place on web sites: E-commerce lessons from web server logs," v zborniku *ACM 2000 Conference on Electronic Commerce*, 2000.
[21] B. Sarwar, G. Karypis, J. Konstan, J. Reidl, "Item-based collaborative filtering recommendation algorithms," v zborniku *10th International Conference on World Wide Web*, 2001.
[22] K. Yu, X. Xu, M. Ester, H.-P. Kriegel, "Selecting relevant instances for efficient and accurate collaborative filtering," v zborniku *10th International Conference on Information and Knowledge Management*, 2001.
[23] C. Zeng, C.-X. Xing, L.-Z. Zhou, "Similarity measure and instance selection for collaborative filtering," v zborniku *12th International World Wide Web Conference*, 2003.