

[Near duplicates](#)

[SimHash](#)

[Finding Similar Items](#)

[MinHash](#)

[LSH](#)

[Finding Frequent Itemsets](#)

[Toivonen](#)

[PCY](#)

[Data Streams](#)

[Estimating moments \(Alon-Matias-Szegedy\)](#)

[Filtering streams \(Bloom filter\)](#)

[Counting 1's \(DGIM\)](#)

[Advertising](#)

[BALANCE](#)

[Link Analysis \(svi ovakvi zadaci su u zbirci\)](#)

[Clustering](#)

[CURE](#)

[BFR](#)

[Community Detection in Social Network Graphs](#)

[Recommendation Systems](#)

[Collaborative Filtering](#)

[Content-based](#)

Near duplicates

SimHash

(LJIR 2017)

- a) Opisati rad sustava za brzo pronalaženje sličnih sažetaka iz permutacijske tablice.
- b) Duljina sažetka $f = 64$, a broj sažetaka je $|F| = 2^{34}$. Sažeci su podijeljeni na blokove duljine 13, 13, 13, 13, 12 bitova, signifikatni dio je 2 bloka. Koliki je broj permutacijskih tablica i koliko će se sažetaka uspoređivati Hammingom?

(3 boda)

Rješenje

- a) Riječ je o Fast Queries

- Build t sorted tables of fingerprints: T_1, T_2, \dots, T_t
- Each table T_i also contains
 - p_i – number of significant bits
 - Π_i – random permutation
- Every fingerprint in T_i is permuted with permutation Π_i

For given Q and k

- Read each table (in parallel)
 - 1. Get fingerprints in T_i whose significant p_i bits match the significant p_i bits of $\Pi_i(Q)$
 - T'_i
 - $O(p_i)$ steps (binary search)
 - 2. For each fingerprint in T'_i , check if it's Hamming distance is at most k bits from $\Pi_i(Q)$

b)

$f=64, k=2, |F| = 8B (2^{34})$

- Split f into 5 blocks ($4 \times 13 + 1 \times 12$ bits)
 - Select 2 out of 5 blocks $5C2 = 10$ ways => **10 permutacijskih tablica**
 - Arrange those blocks as significant bits
 - p = sum of those bits
 - 32 or 33
 - On average query returns $2^{34-32} = 4$ fingerprints => **4 sažetka uspoređena Hammingom**
- Ovo bi trebalo biti $p = 25$ ili 26 bitova (kombinacije bloka su 2×13 i $13 + 12$) pa je $2^{34-25} = 512$ bit

Finding Similar Items

MinHash

(MI 2018)

Pretpostavimo da su u nekom sustavu entiteti predstavljeni skupovima duljine 6 redaka. Od ukupno 6 redaka u skupu, 4 retka imaju vrijednost 0, a ostala dva retka imaju vrijednost 1. Nadalje, za generiranje sažetka skupa koristi se algoritam *MinHash*. Primjetite da ukupno postoji $6! = 720$ permutacija 6 redaka. Prilikom stvaranja sažetka odabire se indeks prvog reda u permutiranom poretku koji ima vrijednost 1

- a) Koliko će od ukupno 720 permutacija imati vrijednost MinHash sažetka 6?
- b) Koliko će od ukupno 720 permutacija imati vrijednost MinHash sažetka 5?
- c) Koliko će od ukupno 720 permutacija imati vrijednost MinHash sažetka 4?

(3 boda)

Rješenje

- a) 0 jer je u najboljem fd slučaju 0 0 0 0 1 1 i nikad se neće dogoditi da je prva jedinica na šestom mjestu jer postoje dvije jedinice
- b) 4! Jer je prva jedinica na 5. mjestu, druga na 6. Mjestu i preostaju četiri nule koje se moraju rasporediti - - - 1 1
- c) $2 * 4!$ Prva jedinica je na 4. mjestu, preostaje da druga može biti na 5. ili na 6. mjestu, znaci ---11- i ---1-1

(MI 2016)

- a) Napišite formulu za *Jaccardovu* sličnost dvaju skupova.
- b) Kratko i jasno objasnite što radi funkcija sažimanja $h_{\pi} = \min_{\pi} (D)$ koja se koristi u *MinHash* algoritmu.
- c) Dokažite da gore spomenuta funkcija sažimanja h_{π} zadržava sličnost dokumenta, tj. da vrijedi $p[h_{\pi}(A) = h_{\pi}(B)] \sim \text{Jaccard_Sim}(A, B)$, gdje su A i B dva dokumenta predložena skupovima podataka.

(3 boda)

Rješenje

- a) Jaccardova sličnost dvaju skupova:

$$\text{sim}(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

- b)

Create a **random permutation** π of 0/1 matrix rows

Define a hash function $h_{\pi}(D)$

The **index of the first** (in the permuted order π) row in which column **D** has a value **1**

$$h_{\pi}(D) = \min_{\pi} \pi(D)$$

To create a signature (hash) of the document

Generate a number (for instance 100) of **independent permutations**

For each permutation, **compute** the **hash function** and **represent** each document with the obtained signature.

c)

- **Claim:** $p[h_{\pi}(D_1) = h_{\pi}(D_2)] = Jaccard_Sim(D_1, D_2)$
- **Proof: There are 3 row types:**
 - $[1, 1]$ **x type**, suppose we have **X** rows of type x
 - $[1, 0]$ and $[0, 1]$ **y type**, suppose we have **Y** rows of this type
 - $[0, 0]$ **z type**, suppose we have **Z** rows of z type
- **What is $Jaccard_Sim(D_1, D_2)$ equal to?**
 - It is $X/(X + Y)$
- **On the other hand, what is $p[h_{\pi}(D_1) = h_{\pi}(D_2)]$ equal to?**
 - We **rearrange** rows of D_1 and D_2 using π
 - Then, we **search** the first row that **contains a 1 in either doc**
 - What is the probability that we encounter **x** row type?
 - It is $X/(X + Y)$

(MI 2016) Zadana su tri teksta A, B i C: A = *padaju zvijezde*, B = *ratovi zvijezda* i C = *kisa pada u maju*. Za svaki tekst izračunati sve *shingleove* duljine $s = 3$ te prikazati tekstove u tablici kao stupce nula i jedinica gdje su redci tablice *shingleovi* (**Napomena:** *praznine ignorirati tj. izbaciti*). Jedinica u tablici označava da je neki *shingle* prisutan u tekstu dok nula označava da *shingle* nije prisutan u tekstu. Nakon toga, na skupovne reprezentacije tekstova potrebno je primijeniti algoritam *MinHash* korištenjem sljedeće četiri funkcije $f_1(r) = r + 1 \bmod N$, $f_2(r) = r + 3 \bmod N$, $f_3(r) = r + 6 \bmod N$ i $f_4(r) = r + 11 \bmod N$, gdje je r redni broj retka u tablici, a N jest ukupan broj redaka odnosno *shingleova*. Primjenom četiri funkcija hashiranja izgraditi sažetke tekstova te izračunati sličnost za sve parove tekstova. Konačno izračunati prosječnu vrijednost apsolutnog odstupanja MAE (engl. *Mean Absolute Error*) sličnosti sažetaka u odnosu na sličnost tekstova. Sličnost sažetaka računati prema formuli $Sim_{HA,HB} = K/M$, gdje je K broj redaka u kojima se sažeci H_A i H_B poklapaju, a M je ukupan broj redaka, tj. broj hash funkcija koje se koriste. Za stvarnu sličnost tekstova koristiti *Jaccardovu* sličnost skupova koji su pridruženi tekstovima. **(3 boda)**

Rješenje

$$N = 27$$

$$f_1(r) = r + 1 \bmod N$$

$$f_2(r) = r + 3 \bmod N$$

$$f_3(r) = r + 6 \bmod N$$

$$f_4(r) = r + 11 \bmod N$$

		A	B	C	f1	f2	f3	f4
0	pad	1	0	1	1	3	6	11
1	ada	1	0	1	2	4	7	12
2	daj	1	0	0	3	5	8	13
3	aju	1	0	1	4	6	9	14
4	juz	1	0	0	5	7	10	15
5	uzv	1	0	0	6	8	11	16
6	zvi	1	1	0	7	9	12	17
7	vij	1	1	0	8	10	13	18
8	ije	1	1	0	9	11	14	19
9	jez	1	1	0	10	12	15	20
10	ezd	1	1	0	11	13	16	21
11	zde	1	0	0	12	14	17	22
12	rat	0	1	0	13	15	18	23
13	ato	0	1	0	14	16	19	24
14	tov	0	1	0	15	17	20	25
15	ovi	0	1	0	16	18	21	26
16	viz	0	1	0	17	19	22	0
17	izv	0	1	0	18	20	23	1
18	zda	0	1	0	19	21	24	2
19	kis	0	0	1	20	22	25	3
20	isa	0	0	1	21	23	26	4
21	sap	0	0	1	22	24	0	5
22	apa	0	0	1	23	25	1	6

23	d a u	0	0	1	24	26	2	7
24	a u m	0	0	1	25	0	3	8
25	u m a	0	0	1	26	1	4	9
26	m a j	0	0	1	0	2	5	10

$$\begin{aligned}
|A \cap B| &= 5 & |A \cup B| &= 19 \\
|A \cap C| &= 3 & |A \cup C| &= 20 \\
|B \cap C| &= 0 & |B \cup C| &= 23
\end{aligned}$$

$$\begin{aligned}
sim(A, B) &= \frac{5}{19} \\
sim(A, C) &= \frac{3}{20} \\
sim(B, C) &= 0
\end{aligned}$$

	A	B	C
f1	1	7	0
f2	3	9	0
f3	6	12	0
f4	11	0	3

$$sim_{H_A, H_B} = 0$$

$$sim_{H_A, H_C} = 0$$

$$sim_{H_B, H_C} = 0$$

Na kraju možemo izračunati prosječnu apsolutno pogrešku *MAE* (eng. *Mean Absolute Error*) sličnosti sažetaka u odnosu na sličnost dokumenata korištenjem sljedeće formule:

$$MAE = \sum_{A,B} \frac{|Sim_{H_A, H_B} - JaccSim_{S_A, S_B}|}{P}, \quad (2.25)$$

$$MAE = 0.1577$$

malo su mi sumnjivo ispale ove nule.. može netko provjeriti?

LSH

(MI 2016) U sustavu postoji $N = 10^9$ različitih tekstualnih datoteka među kojima postoji određeni broj duplikata. Za identifikaciju duplikata koristi se metoda generiranje sažetaka pomoću neke funkcije sažimanja (npr. *SimHash*). Zbog velikog broja dokumenata/sažetaka nemoguća je izravna usporedba svih parova dokumenata/sažetaka pa se koristi tehnika *sažimanja osjetljivog na lokalne promjene* (engl. *Locality Sensitive Hashing*) *LSH*. Veličina

sažetka iznosi 256 bita, koristi se $b = 16$ pojaseva svaki veličine $r = 16$. Cilj je identificirati sve parove kandidata za izravnu usporedbu kojima se sličnost sažetaka poklapa do uključivo 95%, tj. sažeci dokumenata se poklapaju u barem 95% bita.

- Koliko iznosi vjerojatnost da par dokumenata koji ima sličnost 95% ne postane kandidat za izravnu usporedbu - lažno negativni *FN* (eng. *False Negative*) par?
- Pretpostavimo da postoji neki par dokumenata kojem se sažeci poklapaju 15%, kolika je vjerojatnost da takav lažno pozitivan *FP* (eng. *False Positive*) par dokumenata postane kandidat za izravnu usporedbu?
- Skicirati kako izgleda idealna funkcija sažimanja neosjetljivog na lokalne promjene te pokraj nje skicirati funkciju koja se dobije korištenjem metode *LSH*. Na grafu označiti područja koja predstavljaju lažno negativne i lažno pozitivne kandidate. Skicirati što će se dogoditi s grafom funkcije ako se koriste $b = 32$ pojasa svaki veličine $r = 8$ bita.

(3 boda)

Rješenje

a)

Lažno negativni kandidati parovi će se pojaviti ako imamo dva sažetka H_1 i H_2 koji se poklapaju u 95% bita ali se u svakom pojasu razlikuju u barem jednom bitu pa se onda neće niti u jednom pojasu hashirati u isti pretinac (eng. bucket).

Dakle, vjerojatnost da su dva sažetka jednaka u jednom bitu iznosi $s = 0.95$. **Vjerojatnost da su dva sažetka jednaka u cijelom pojasu** iznosi:

$$P[H_1 = H_2 \text{ u točno jednom pojasu}] = s^r$$

gdje je r broj bita u pojasu.

Nadalje, **vjerojatnost da su dva sažetka različita u točno jednom pojasu** iznosi:

$$P[H_1 \neq H_2 \text{ u točno jednom pojasu}] = 1 - s^r$$

Sada možemo izračunati **vjerojatnost da su dva sažetka koja imaju sličnost s različita u svim pojasevima**:

$$P[H_1 \neq H_2 \text{ u svim pojasevima}] = (1 - s^r)^b$$

, gdje je r broj bita u pojasu, b je broj pojaseva.

Uvrštavanjem zadanih vrijednosti u prethodno dobivenu formulu dobijamo vjerojatnost pojave lažno negativnih **95%** sličnih kandidata:

$$P[H_1 \neq H_2 \text{ u svim pojasevima}] = (1 - s^r)^b = (1 - 0.95^{16})^{16} = 9.32 \cdot 10^{-5} \checkmark$$

b)

U slučaju a) izračunali smo vjerojatnost da se par sažetaka koji ima sličnost s razlikuje u svim pojasevima. **Negiranjem** te relacije možemo izvesti **vjerojatnost da se par sažetaka poklapa u barem jednom pojasu** čime *de facto* postaje kandidat za usporedbu:

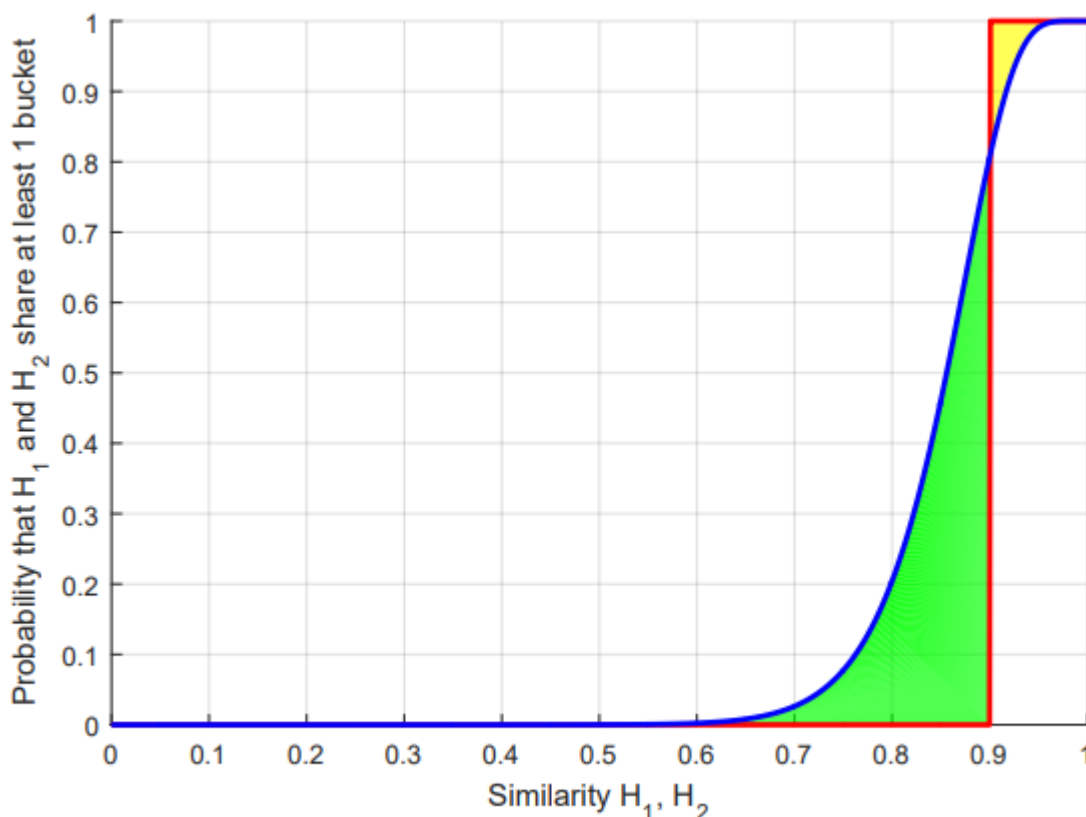
$$P[H_1 = H_2 \text{ u bar jednom pojasu}] = 1 - (1 - s^r)^b$$

, gdje je r broj bita u pojasu, b je broj pojaseva.

Uvrštavanjem zadanih vrijednosti u prethodno dobivenu formulu dobijamo vjerojatnost pojave lažno pozitivnih **15%** sličnih kandidata:

$$P[H_1 = H_2 \text{ u bar jednom pojasu}] = 1 - (1 - s^r)^b = 1 - (1 - 0.15^{16})^{16} = 1.04 \cdot 10^{-12} \checkmark$$

c)



Slika 2.1: Sažimanje osjetljivo na lokalnost graf idealne i stvarne funkcije sažimanja

Što je veći b , imamo manje false-negatives i obrnuto pa se graf tako i mijenja. Za više detalja pogledajte na kraju prezentacije (slajd 37 i 38) koji to ilustrira.

(LJIR 2017) Sažetci duljine 12 bita, LSH, koriste se pojasevi veličine 2, 4 i 6 bita.

- Vjerojatnost da par dokumenata koji ima sličnost 85% ne postane kandidat za usporedbu?
- Vjerojatnost da par dokumenta koji ima sličnost 15% postane kandidat za usporedbu?
- Zadaci a) i b) ako se koriste 3 pojasa duljine 4 bita
- Skicirati idealnu i stvarnu funkciju sažimanja, što se događa s vjerojatnosti iz c) ako sustav koristi 4 pojasa duljine 3 bita?

(3 boda)

Rješenje

- Slično prethodnom zadatku, ali za svaki pojas zasebno (jer nemamo iste pojaseve pa ne možemo jednostavno staviti vjerojatnost p^b), uvrštavanjem zadanih vrijednosti u prethodno dobivenu formulu dobijamo vjerojatnost pojave lažno negativnih **85%** sličnih kandidata:

$$P[H_1 \neq H_2 \text{ u svim pojasevima}] = (1 - s^r)^b = (1 - 0.85^2)(1 - 0.85^4)(1 - 0.85^6) = 0.083 \checkmark$$

- b) Slično prethodnom zadatku, uvrštavanjem zadanih vrijednosti u prethodno dobivenu formulu dobijamo vjerojatnost pojave lažno pozitivnih **15%** sličnih kandidata:

$$P[H_1 = H_2 \text{ u bar jednom pojasu}] = 1 - (1 - s^r)^b = 1 - (1 - 0.15^2)(1 - 0.15^4)(1 - 0.15^6) = 0.023 \checkmark$$

- c) $B = 3$, $r = 4$ (3 pojasa duljine 4 bita)

$$P[H_1 \neq H_2 \text{ u svim pojasevima}] = (1 - s^r)^b = (1 - 0.85^4)^3 = 0.109 \checkmark$$

$$P[H_1 = H_2 \text{ u bar jednom pojasu}] = 1 - (1 - s^r)^b = 1 - (1 - 0.15^4)^3 = 1.518 \cdot 10^{-3} \checkmark$$

- d) Isto kao u prethodnom zadatku

Finding Frequent Itemsets

(MI 2016) Za zadani skup košara odredite sve česte podskupove podataka. Pretpostavite da je prag potpore (engl. *support*) $1/3$.

$$B1 = \{a,b,c,d,e\} \quad B4 = \{a,b,d,e\}$$

$$B2 = \{a,c,d,e\} \quad B5 = \{a,d\}$$

$$B3 = \{b,c,e\} \quad B6 = \{b,c,e\}$$

(3 boda)

Rješenje

$$n = 6$$

$$\text{ceil}(s \cdot n) = 2$$

Česti podskupovi kardinaliteta 2:

$\{a,b\}$ (2)

$\{a,c\}$ (2)

$\{a,d\}$ (4)

$\{a,e\}$ (3)

$\{b,c\}$ (3)

$\{b,d\}$ (2)

$\{b,e\}$ (4)

$\{c,d\}$ (2)

$\{c,e\}$ (4)

$\{d,e\}$ (3)

Česti podskupovi kardinaliteta 3:

$\{a,b,d\}$ (2)

$\{a,b,e\}$ (2)

$\{a,c,e\}$ (2)

$\{a,d,e\}$ (3)

$\{a,c,d\}$ (2)

$\{b,c,e\}$ (3)

$\{b,d,e\}$ (2)

$\{c,d,e\}$ (2)

Česti podskupovi kardinaliteta 4:

$\{a,b,d,e\}$ (3)

$\{a,c,d,e\}$ (2)

(LJIR 2017)

$$B1 = \{a,b,c,d\} \quad B4 = \{a,b\}$$

$$B2 = \{a,c\} \quad B5 = \{a,d\}$$

$$B3 = \{b,c\} \quad B6 = \{a,b,c\}$$

$$B7 = \{a,b,d\}$$

- a) Odrediti česte podskupove za support = 3
- b) Odrediti interesantnost $a \rightarrow c$
- c) Na koji je način najprimjerenije pohranjivati podatke za zadani skup košara, trokutastom matricom ili raspršenim adresiranjem?

(3 boda)

Rješenje

Pretpostavljam da se mislilo da je support 0.3 jer bi inace broj kosara bio 21 da bi se smatralo da je predmet cest.

a)

$0.3 * 7 \sim 2$ kosare

Česti podskupovi kardinaliteta 2:

$\{a,b\}$ (4)

$\{a,c\}$ (3)

$\{a,d\}$ (3)

$\{b,c\}$ (3)

$\{b,d\}$ (2)

$\{c,d\}$ (1)

Česti podskupovi kardinaliteta 3:

$\{a,b,c\}$ (2)

$\{a,b,d\}$ (2)

b)

Formula za interesantnost je

$$I(I \rightarrow j) = c(I \rightarrow j) - Fr[j]$$

gdje je $c(I \rightarrow j)$ pouzdanost pravila asocijacije, a $Fr[j]$ udio ponavljanja predmeta j u skupu kosara.

Pouzdanost pravila asocijacije racunamo

$$c(I \rightarrow j) = s(I \cup j) / s(I),$$

odnosno omjerom broja ponavljanja skupa $\{I,j\}$ s brojem ponavljanja predmeta I.

Udio ponavljanja $Fr[j]$ dobijemo omjerom brojem ponavljanja predmeta j s ukupnim brojem kosara.

Za $I(a \rightarrow c)$:

$$c(a \rightarrow c) = s(\{a,c\})/s(a) = 3/6 = 1/2$$

$$Fr[c] = 4/7$$

$$I(a \rightarrow c) = 1/2 - 4/7 = -1/14$$

c) Raspršeno adresiranje je bolje (PCY algoritam) od trokutaste matrice (A-Priori). Matrica nije gusto popunjena i prazna mjesta (mjesta na kojima su nečesti parovi) nepotrebno zauzimaju mem.prostor. Raspršeno adresiranje stvara bit-vektor (nešto slično Bloom filteru) koji ima postavljene jedinice na mjestima koja odgovaraju hashiranim čestim parovima, i nule za nečeste parove (zbog mogućnosti kolizije može doći do FP-ova, ali ne i do FN-ova)
-> **Ispravak - raspršeno je bolje za brojanje trojki i većih k-torki, matrica je bolje za parove (prezentacija 04, slajd 48)**

Toivonen

(MI 2018) Opišite Toivonenov algoritam. Definirajte pojam negativne granice (engl. negative border). Konstruirajte negativnu granicu za zadane košare uz prag potpore (engl. *support threshold*) $s = 3$.


$$B1 = \{a,b,c\} \quad B4 = \{a,b,c\}$$

$$B2 = \{a,b\} \quad B5 = \{a,c\}$$

$$B3 = \{a\} \quad B6 = \{b\}$$

(3 boda)

Rješenje

 Ola Toivonen igra sutra protiv Švicarske - to sam jedino našao u literaturi
https://www.fer.unizg.hr/download/repository/AVSP_04_Finding_Frequent_Itemsets.pdf
(slide 57/66)

PCY

(MI 2018) Odredite stanje struktura podataka nakon izvođenja prvog i drugog koraka algoritma PCY (Park, Chen, Yu) za zadani skup košara. Pretpostaviti da se elementi u košarama indeksiraju abecednim redoslijedom od početnog indeksa 0. Funkcija sažimanja

(engl. *hash function*) zadana je formulom $(i + j) \% 4$, pri čemu su i i j indeksi elementa u košari. Prag potpore (engl. *support threshold*) iznosi 4.

$B1 = \{a,b,c,d\}$ $B4 = \{a,c,d\}$

$B2 = \{a,c,d\}$ $B5 = \{b,c,d\}$

$B3 = \{a,b,d\}$ $B6 = \{c,d\}$

(3 boda)

Rješenje



(MI 2016) Prikažite stanje struktura podataka nakon izvođenja prvog, međukoraka i drugog koraka algoritma PCY za zadani skup košara. Pretpostavite da algoritam redom obilazi košare B1 - B6 te predmete unutar košara s lijeva na desno. Tim redoslijedom algoritam dodjeljuje indekse predmetima. Prag potpore (engl. *support*) iznosi $2/3$. Pretpostavite da algoritam u prvom koraku koristi funkciju sažimanja $h(i, j) = (i * 4 + j) \bmod 5$, pri čemu su i i j indeksi predmeta i vrijedi $0 \leq i < j < N$, pri čemu je N ukupan broj različitih predmeta.

$B1 = \{a,b,c,d\}$ $B4 = \{b,c,d\}$

$B2 = \{a,c,d\}$ $B5 = \{a,d\}$

$B3 = \{a,b,c\}$ $B6 = \{b,c\}$

(3 boda)

Rješenje



Data Streams

(MI 2018) Senzor za mjerenje temperature zraka šalje odgovarajuća mjerenja s preciznošću $0.1\text{ }^{\circ}\text{C}$. Napišite (u pseudokodu ili jeziku po želji) algoritam koji za svaku pristiglu temperaturu odgovara na sljedeći upit: *Koliko je puta u prethodnih 10^6 mjerenja izmjerena ista temperatura kao sada?* Algoritam mora biti egzaktni (ne samo približno točan), smije koristiti poznate strukture podataka i mora biti vremenski i memorijski što efikasniji. Odredite vremensku složenost algoritma po pojedinom upitu.

(3 boda)

Rješenje

Stream Data model -> standing query (execute permanently) -> temperature senzor

Average of n recent readings

- First n readings
 - $\text{avg} = \text{sum}/n$
- Next readings
 - x – new element, y – oldest element
 - $\text{new_avg} = \text{avg} + (x - y)/n$

(MI 2018) Analiziramo tok podataka koji se sastoji od e-mail adresa. Uniformne *hash* funkcije h_1, h_2, \dots, h_{10} preslikavaju adrese u 32-bitne brojeve. Funkciju h_1 koristimo za uzorkovanje: radi ubrzanja analize, u uzorak ulaze samo adrese x za koje je $h_1(x) < 10^8$. Funkcije h_2, h_3, \dots, h_9 koristimo za Bloomov filter u koji ubacujemo adrese uzorka. Funkciju h_{10} koristimo za Flajolet-Martinovu procjenu broja različitih elemenata, pri čemu je r = maksimalan broj nula na kraju nekog $h_{10}(x)$ iz uzorka.

- a) Procijenite broj različitih adresa u uzorku ako je $r = 25$.
- b) Procijenite broj različitih adresa u uzorku ako je popunjenost (gustoća) Bloomovog filtera 10%.
- c) Na temelju prethodnih dviju procjena p_a i p_b procijenite broj različitih adresa u cijelom toku podataka.

(3 boda)

Rješenje



Estimating moments (Alon-Matias-Szegedy)

(MI 2016)

- a) Matematički izrazite drugi moment toka podataka.

- b) Dokažite da izraz za procjenu drugog momenta toka podataka prema Alon-Matias-Szegedyju konvergira ka izrazu iz a) dijela zadatka. Detaljno objasnite svaki korak dokaza!

(3 boda)

Rješenje

a)

$$\sum_a m_a^2$$

a je element skupa, m_a je broj pojavljivanja elementa a u toku podataka, npr. za ulazni tok 1 1 2 3 4 m_1 je 2 dok je za ostale 1

b) Knjiga, poglavlje 4.5.3

$$E(n(2X.value - 1)) = \frac{1}{n} \sum_{i=1}^n n(2c(i) - 1)$$

Pokratimo n-ove

$$E(n(2X.value - 1)) = \sum_{i=1}^n (2c(i) - 1)$$

Grupiramo sve indekse na kojima se javlja isti element - npr. ako imamo sljedeći tok:

Indeks	0	1	2	3	4	5	6	7	8	9
Tok	1	5	5	6	1	8	7	5	2	1

Onda se element '1' javlja na indeksima 0, 4 i 9.

Kada se odabere zadnji indeks na kojem se javlja (indeks = 9) jedinica se od tog mjesta do kraja toka javlja

$$(2*c - 1) = (2*1 - 1) = 1$$

Kada se odabere predzadnji indeks na kojem se javlja (indeks = 4) jedinica se od tog mjesta do kraja toka javlja

$$(2*c - 1) = (2*2 - 1) = 3$$

Kada se odabere predpredzadnji indeks na kojem se javlja (indeks = 0) jedinica se od tog mjesta do kraja toka javlja

$$(2*c - 1) = (2*3 - 1) = 5$$

Itđ.

Uocavamo da se u sumi javljaju samo neparni brojevi, dakle ona će iznositi:

$1+3+5+7+\dots+(2\cdot m_a-1) = m_a^2$, gdje je m_a broj pojavljivanja elementa a u toku.

Ovo gore vrijedi za neki element a , kako bi dobili ukupno očekivanje, treba sumirati po svima:

$$\sum_a (m_a)^2;$$

Filtering streams (Bloom filter)

(MI 2016) Bloomov filter od $3 \cdot 10^6$ bita je popunjen s $5 \cdot 10^5$ elemenata. Odredite pogrešku (izraženu kao razlika vjerojatnosti pojavljivanja lažno pozitivnog rezultata) pri korištenju 10 funkcija sažimanja u odnosu na optimalni broj funkcija sažimanja. **(3 boda)**

Rješenje

 $p(fp) = (1 - e^{-(mk/n)})^k$ $Kopt = 4$, $Pfk1 - Pfkopt = 0.123 - 0.056 = 0.067$

(LJIR 2017) Dvije inačice Bloomovog filtera. Inačica A je kanonska inačica (normalna). Inačica B ima dvije grupe funkcija - $f_i(x)$, domena $[1, n]$ i $g_i(x)$, domena $[1, m]$. Prvo se za sve f_i postavlja bitovi na 1, a potom za sve g_i na 0. Sve funkcije su uniformne na $[1, m - 1]$. Prvo za inačicu A pa za inačicu B:

- Vjerojatnost postavljanja nekog bita na 1 prilikom unosa novog elementa.
- Vjerojatnost da se pojavi false positive
- Vjerojatnost da se pojavi false negative

(3 boda)

Rješenje



- 1 / (number of bits in the bloom filter)
 - 1 / n
 - 1 / m

Counting 1's (DGIM)

(ZI X) Ulaz u algoritam DGIM je binarni niz 11101110101011.

- Skicirati sadržaj pretinaca za svaki dolazak novog bita u ulaznom nizu za veličine prozora $N = 5$ i $N = 10$.
- Procijeniti koliko ima jedinica u zadnjih N bitova ($k = N$) za obe zadane vrijednosti parametra N .

(4 boda)

Rješenje

a)

	N = 5	N = 10
--	-------	--------

0	[1]	[1]
1	[1][1]	[1][1]
2	[11][1]	[11][1]
3	[11][1]0	[11][1]0
4	[11][1]0[1]	[11][1]0[1]
5	[11][101][1]	[11][101][1]
6	11[101][1][1]	[11][101][1][1]
7	11[101][1][1]0	[11][101][1][1]0
8	11[101][11]0[1]	[11101][11]0[1]
9	11101[11]0[1]0	[11101][11]0[1]0
10	11101[11]0[1]0[1]	[11101][11]0[1]0[1]
11	11101[11]0[1]0[1]0 <i>Mislim da tu treba obrisati ovaj najstariji pretinac</i>	[11101][11]0[1]0[1]0
12	1110110[101]0[1]	[11101][11]0[101]0[1]
13	1110110[101]0[1][1]	[11101][11]0[101]0[1][1]

b)

$$N = 5 \Rightarrow 1 + 1 + 2/2 = 3$$

$$N = 10 \Rightarrow 1 + 1 + 2 + 2 + 4/2 = 8$$

Pls pojašnjenje kako se računa procjena jedinica u zadnjih N bitova

Zbrojis velicine svih novijih pretinaca (znaci svi osim ovog zadnjeg kojeg gledas), a od tog zadnjeg dodas pola velicine. (Npr. za ovaj primjer i N=10 gledas 2 pretinca velicine 1, 2 pretinca velicine 2 i zadnji velicine 4 pa je procjena $1 + 1 + 2 + 2 + 4/2$ jer uzimas pola tog zadnjeg)...i pod zadnji mislim na najstariji pretinac

Veličine pretinaca ili broj jedinica u pretincu?

Kod algoritma Counting 1's velicina pretinca je definirana brojem jedinica u pretincu tako da zapravo gledas broj jedinica u pretincu.

Advertising

BALANCE

(MI 2018) Formalno definirati *Adwords* problem. Opisati *BALANCE* algoritam (navesti pseudokod). **(3 boda)**

Rješenje

- How to match ads to search queries?
Very similar to the general problem of bipartite graph matching
- Search engine gets a set of queries as input values
 - Advertisers bid on search keywords
 - Upon answering the query, the search engine picks a subset of ads to display
Usually more than one ad is shown
- Goal is to maximize the profits from advertising

Statement of the Adwords problem:

1. A set of advertisers' bids on search queries
2. A click-through rate for each ad-query(keyword) pair
3. A budget for each advertiser (e.g. for 1 month)
4. A limit on the number of ads that can be displayed per search query



(MI 2018) Tri oglašivača A, B i C natječu se za prikaz oglasa. Svi imaju isti budžet koji iznosi 3. Svaki oglašivač ponudio je 1 za prikaz oglasa. Oglašivači A i B natječu se za oglase tipa X i Y, dok se oglašivač C natječe za oglase tipa X. U sustav oglašavanja dolazi sljedeći niz korisničkih zahtjeva: Y Y X X X Y Y Y. Pretpostavite da sustav oglašavanja koristi *BALANCE* algoritam. Opisati rad algoritma nad navedenim nizom korisničkih zahtjeva. Odrediti kompetitivni omjer algoritma (engl. *competitive ratio*) za navedeni ulaz. **(3 boda)**

Rješenje

Tiebreaker: A -> B -> C

Y	Y	X	X	X	Y	Y	Y	Y
A	B	C	A	B	A	B	-	-

Optimalan raspored daje profit 9, a u ovom slučaju je profit 7. Kompetitivni omjer je 7/9.

Uoči: način na koji rješavamo izjednačenja utječe na kompetitivni omjer; da smo koristili prioritet C->A->B, profit bi bio 8, odnosno omjer=8/9).

(ZI 2016) U sustavu oglašavanja na webu primjenjuje se *generalized balance* algoritam. Tri oglašivača se natječu za oglasni prostor:

- Oglašivač O_1 želi objaviti oglas koji se do sada prikazao 50 puta, od čega su korisnici odabrali oglas 40 puta, za oglas želi platiti 11 kuna, a do sada je potrošio 40 kuna od ukupnog budžeta koji iznosi 100 kuna.
- Oglašivač O_2 želi objaviti oglas koji se do sada prikazao 120 puta, od čega su korisnici odabrali oglas 72 puta, za oglas želi platiti 12 kuna, a do sada je potrošio 240 kuna od ukupnog budžeta koji iznosi 300 kuna.
- Oglašivač O_3 želi objaviti oglas koji se do sada prikazao 100 puta, od čega su korisnici odabrali oglas 55 puta, za oglas želi platiti 12 kuna, a do sada je potrošio 570 kuna od ukupnog budžeta koji iznosi 600 kuna.

U sustav dolaze tri uzastopna korisnička upita koja odgovaraju oglasnom profilu za koji se natječu sva tri oglašivača. Odredite redoslijed prikazivanja oglasa. **(4 boda)**

Rješenje

Generalized balance algoritam

	Prikazao	Odabrali	Cijena (x_i)	Potrošio (m_i)	Budget (b_i)	CTR
O1	50	40	11	40	100	$80 = 40/50 = 80\%$
O2	120	72	12	240	300	60
O3	100	55	12	570	600	55

Dolaze tri oglasa za koji se svi natječu.
Odrediti redoslijed prikazivanja oglasa

Oglas		$\psi(x_1)$	$\psi(x_2)$	$\psi(x_3)$
Prikazao		O_1	O_1	O_1
O_1		3.96	x`3.40	2.78
O_2		1.30	1.30	1.30
O_3		0.32	0.32	0.32

Kako je 2.78 veće od 3.2?

Meni tu ispada 0.32

$$\psi_i(q) = c_i \cdot x_i \left(1 - e^{-\left(1 - \frac{m_i}{b_i}\right)}\right)$$

, gdje je c_i = CTR.

?

(ZI 2017) U sustavu oglašavanja nalaze se 2 oglašivača:

- Oglašivač A ima početni budžet 5 i nudi jedan novac za oglase X i Y.
- Oglašivač B ima početni budžet 6 i nudi dva novca, ali samo za oglase Y.

Pretpostavite da sustav dodjeljuje oglasni prostor primjenom algoritma BALANCE i da se u izjednačenom slučaju oglasi dodjeljuju prema prioritetu $A \rightarrow B$ - dakle, A ima veći prioritet.

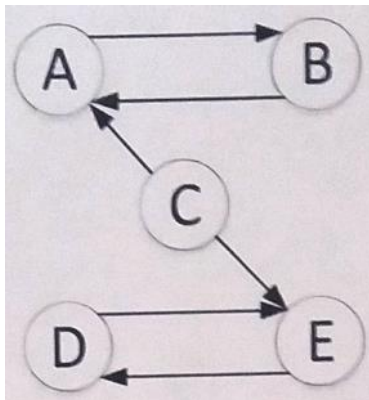
- Kojim će se redoslijedom oglasi prikazivati korisnicima?
- Odrediti *competitive ratio*.
- Navedite općeniti *competitive ratio* za 2 oglašivača s **istim** ograničenim budžetom.

(5 bodova)

? fali lista upita

Link Analysis (svi ovakvi zadaci su u zbirci)

(MI 2018) Za zadani graf na slici napisati jednadžbu toka ranga/utjecaja (eng. *rank*) za sve čvorove u grafu (eng. *Flow Equation Formulation*).



- Primjeni li se metoda uzastopnog potenciranja s ciljem izračunavanja vektora ranga \mathbf{r} na zadani graf, hoće li dobiveni rezultati biti vjerodostojni? Detaljno objasniti odgovor.
- Modificirati zadani graf uvođenjem teleportirajućih poveznica (eng. *teleports*). Napisati vrijednost *Google* matrice \mathbf{A} ako vjerojatnost da će slučajni šetač (eng. *random walker*) slijediti poveznicu iz grafa iznosi $\beta = 0.8$.

(3 boda)

Rješenje

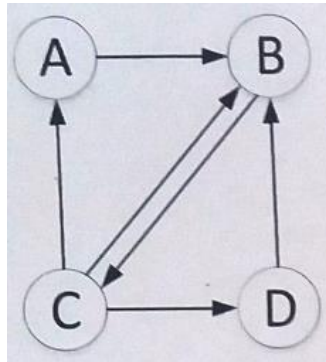
- Prvo treba reći da nije zajamčeno da će metoda uzastopnog potenciranja uopće konvergirati! Međutim za zadani primjer grafa možemo doći do rješenja analitičkim putem. **<rješenje ubacim kasnije>**
Očito je da dobiveno rješenje ne daje stvarni rang/utjecaj čvorova u mreži. Naime, u zadanom grafu postoje dvije paukove zamke (engl. spider trap) koju čine čvorovi A i B te E i D.
Slučajni šetač jednom kada odluči otići u čvor A ili E, on ostaje trajno zarobljen u grupi čvorova koji čine A i B, tj. D i E. Na taj način čvorovi A i B, tj. D i E, preuzimaju sav utjecaj u zadanom grafu i marginaliziraju ostale čvorove. Rješenje u ovakvim situacijama jest uvođenje teleportirajućih poveznica.

b)

0.04	0.84	0.44	0.04	0.04
------	------	------	------	------

0.84	0.04	0.04	0.04	0.04
0.04	0.04	0.04	0.04	0.04
0.04	0.04	0.04	0.04	0.84
0.04	0.04	0.44	0.84	0.04

(MI 2018) Za zadani graf na slici napisati jednadžbu toka ranga/utjecaja (eng. *rank*) za sve čvorove u grafu (eng. *Flow Equation Formulation*).



Analitičkim postupkom riješiti sustav jednadžbi.

Napisati jednadžbu u matričnom obliku (eng. *Matrix Formulation*).

Metodom uzastopnog potenciranja izračunati vrijednost rang vektora r za prve 3 iteracije algoritma.

Rang vektor u početku treba inicijalizirati tako da svi čvorovi dobiju jednak rang/utjecaj.

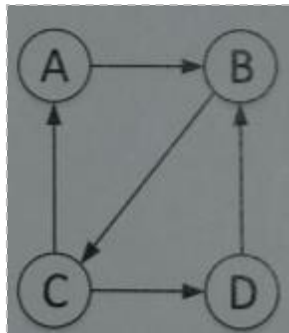
Izračunati prosječnu apsolutnu kvadratnu pogrešku *RMSE* (eng. *Root Mean Square Error*)

metode uzastopnog potenciranja kroz tri iteracije u odnosu na egzaktno analitičko rješenje. **(3 boda)**

Rješenje



(MI 2016) Za zadani graf na slici napisati jednadžbu toka ranga/utjecaja (eng. *rank*) za sve čvorove u grafu (eng. *Flow Equation Formulation*).



Analitičkim postupkom riješiti sustav jednadžbi.

Napisati jednadžbu u matričnom obliku (eng. *Matrix Formulation*).

Metodom uzastopnog potenciranja izračunati vrijednost rang vektora r za prve tri iteracije algoritma.

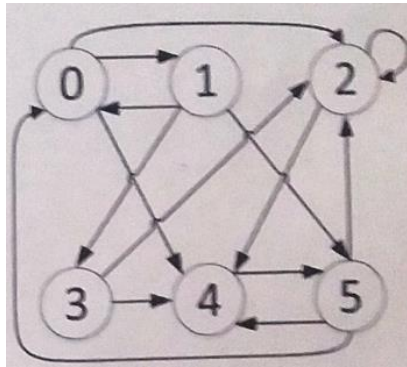
Rang vektor u početku treba inicijalizirati tako da svi čvorovi dobiju jednak rang/utjecaj.

Izračunati prosječnu apsolutnu kvadratnu pogrešku *RMSE* (eng. *Root Mean Square Error*) metode uzastopnog potenciranja kroz tri iteracije u odnosu na egzaktno analitičko rješenje. (3 boda)

Rješenje

(MI 2018) Za zadani graf na slici skicirati strukture podataka r_{t+1} , M i r_t koje se koriste za izračunavanje ranga (utjecaja) čvora u grafu pod pretpostavkom da:

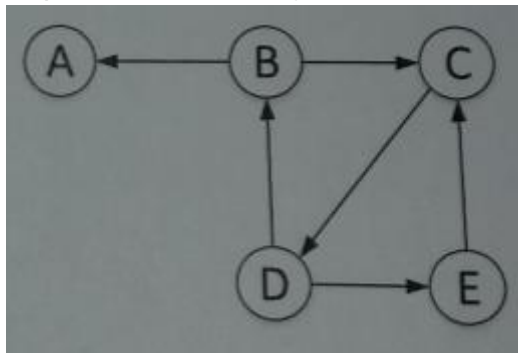
- Rang vektor r može stati u radnu memoriju, a matrica M ne može stati u radnu memoriju.
- Niti rang vektor r , niti matrica M ne mogu stati u radnu memoriju. U radnu memoriju može stati najviše $k = 2$ elemenata rang vektora. Skicirati prilagođeni zapis matrice M koji omogućuje manji broj iteracija kroz čitavu matricu prilikom računanja rang vektora.



(3 boda)

Rješenje

(MI 2016) Za zadani graf na slici napisati jednadžbu toka ranga/utjecaja (eng. *rank*) za sve čvorove u grafu (eng. *Flow Equation Formulation*).



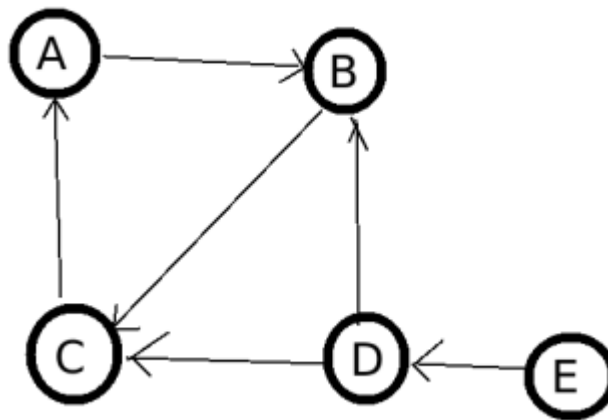
- Primjeni li se metoda uzastopnog potenciranja s ciljem izračunavanja vektora ranga r na zadani graf, hoće li dobiveni rezultati biti vjerodostojni? Detaljno objasniti odgovor.
- Modificirati zadani graf uvođenjem teleportirajućih poveznica (eng. *teleports*). Napisati vrijednost Google matrice A ako vjerojatnost da će slučajni šetač (eng. *random walker*) slijediti poveznicu iz grafa iznosi $\beta = 0.8$.

(3 boda)

Rješenje



(ZI 2017) Za zadani graf na slici napisati jednadžbu toka ranga/utjecaja (eng. *rank*) za sve čvorove u grafu (eng. *Flow Equation Formulation*).



- Primjeni li se metoda uzastopnog potenciranja s ciljem izračunavanja vektora ranga r na zadani graf, hoće li dobiveni rezultati biti vjerodostojni? Detaljno objasniti odgovor.
- Modificirati zadani graf uvođenjem teleportirajućih poveznica (eng. *teleports*). Napisati vrijednost Google matrice A ako vjerojatnost da će slučajni šetač (eng. *random walker*) slijediti poveznicu iz grafa iznosi $\beta = 0.9$.

(3 boda)

Rješenje

c) Vidi prijašnje zadatke

d)
$$\begin{bmatrix} 0.02 & 0.02 & 0.92 & 0.02 & 0.02 \\ 0.92 & 0.02 & 0.02 & 0.47 & 0.02 \\ 0.02 & 0.92 & 0.02 & 0.47 & 0.02 \\ 0.02 & 0.02 & 0.02 & 0.02 & 0.92 \\ 0.02 & 0.02 & 0.02 & 0.02 & 0.02 \end{bmatrix}$$

Clustering

CURE

(ZI 2018) (ZI 2016) Točke (1,5), (2,4), (2,8), (7,7) algoritam *CURE* (engl. *Clustering using representatives*) dodijelio je u jednu grupu.

Odredite udaljenost točke (6,4) od navedene grupe uz pretpostavku da grupa ima dvije reprezentativne točke koje se odabiru po principu najveće udaljenosti, a faktor približavanja reprezentativnih točaka iznosi 10%. **(4 boda)**

Rješenje

Algoritam, slajd 52

1. pass Odredi RT

$$d(T_1(1,5), T_4(7,7)) = \sqrt{(7-1)^2 + (7-5)^2} = 6.325 \checkmark$$

$$d(T_1(1,5), T_3(2,8)) = \sqrt{(2-1)^2 + (8-5)^2} = 3.162$$

$$d(T_1(1,5), T_2(2,4)) = \sqrt{(2-1)^2 + (4-5)^2} = 1.414$$

$$d(T_2(2,4), T_4(7,7)) = \sqrt{(7-2)^2 + (7-4)^2} = 5.831$$

$$d(T_2(2,4), T_3(2,8)) = \sqrt{(2-2)^2 + (8-4)^2} = 4$$

$$d(T_3(2,8), T_4(7,7)) = \sqrt{(7-2)^2 + (7-8)^2} = 5.099$$

- a. Representative points **T1** i **T4** (imaju najveću međusobnu udaljenost)
i. Pomakni RT za 10% prema centroidu

$$T'_1 = ((C_x - T_{1x}) \cdot 0.1 + T_{1x}, (C_y - T_{1y}) \cdot 0.1 + T_{1y}) = (1.2, 5.1)$$

$$T'_4 = ((C_x - T_{4x}) \cdot 0.1 + T_{4x}, (C_y - T_{4y}) \cdot 0.1 + T_{4y}) = (6.6, 6.9)$$

- b. Nema se što *mergati* (**zašto?**) jer ne piše u zadatku (inače bi se trebale mergeati neke grupe ako su im reprezentativne točke dovoljno blizu)

2. Pass

- a. Udaljenost od RT

$$d(T'_1(1.2, 5.1), T(6, 4)) = \sqrt{(6-1.2)^2 + (4-5.1)^2} = 4.924$$

$$d(T'_4(6.6, 6.9), T(6, 4)) = \sqrt{(6-6.6)^2 + (4-6.9)^2} = 2.961 \checkmark$$

(LJIR 2017) Točke (2, 4), (3, 3), (3, 8), (5, 3), (5, 6), (7, 2), (6, 8), (9, 6). Reprezentativne točke su (3, 3) i (9, 6), pomak je 20%. Kojoj grupi pripada točka (5, 6)? **(3 boda)**

Rješenje

$$\text{Centroid } \left(\frac{SUM_x}{N}, \frac{SUM_y}{N} \right) = X(5, 5)$$

Odakle ovi brojevi što se množe s 0.2, točnije 2,2,4,1? Razlika koordinate točke i koordinate centroida, npr: $R1=(T2x+0.2*(Xx-T2x), T2y+0.2*(Xy-T2y))$

$$R_1: (3 + 0.2 \cdot 2, 3 + 0.2 \cdot 2) = (3.4, 3.4)$$

$$R_2: (9 - 0.2 \cdot 4, 6 - 0.2 \cdot 1) = (8.2, 5.8)$$

$$d_1 = \sqrt{(3.4 - 5)^2 + (3.4 - 6)^2} = \sqrt{2.56 + 6.76} = \sqrt{9.32} = 3.05 \checkmark$$

$$d_2 = \sqrt{(8.2 - 5)^2 + (5.8 - 6)^2} = \sqrt{10.24 + 0.04} = \sqrt{10.28} = 3.2$$

Točka (5,6) će završiti u klasteru s reprezentativnom točkom (3.4,3.4)

BFR

(ZI 2017) Skup podataka (1,1), (1,3), (1,5), (2,2), (2,4) i (3,3) grupiran je primjenom algoritma BFR u jednu grupu C1. Nakon toga, u skup je dodana točka (4,4).

- Navedite stanje struktura podataka algoritma prije dodavanja točke.
- Definirajte i objasnite Mahalanobisovu udaljenost
- Izračunajte MD((4,4)) i navedite stanje strukture podataka nakon dodavanja točke.

(3 boda)

Rješenje

a) prije dodavanja točke

	N	SUM_x	SUM_y	$SUMSQ_x$	$SUMSQ_y$	v_x	v_y
C1	6	10	18	20	64	0.555	1.666

Note: varijanca ne spada u strukturu podataka!

Varijanca:

$$v_i = \left(\frac{SUMSQ_i}{N} \right) - \left(\frac{SUM_i}{N} \right)^2$$

b) def. Mahalanobisova udaljenost se računa po formuli:

$$\sqrt{\sum_{i=1}^d \left(\frac{x_i - c_i}{\sigma_i} \right)^2}$$

σ_i - standardna devijacija: $\sigma_i = \sqrt{v_i}$.

c_i - centroid $\left(\frac{SUM_x}{N}, \frac{SUM_y}{N} \right) = (1.666, 3)$.

d - broj dimenzija = 2

c)

Trebamo izračunati MD točke (4,4), uvrstimo u formulu iz b):

$$= \sqrt{\left(\frac{4-1.666}{\sqrt{0.555}}\right)^2 + \left(\frac{4-3}{\sqrt{1.666}}\right)^2} =$$
$$= \sqrt{9.815 + 0.6} = 3.227 < 3\sqrt{d} = 3\sqrt{2} \text{ pokriva 99\% točaka}$$

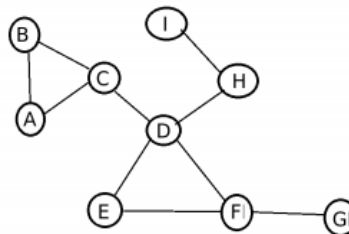
Nakon dodavanja točke:

	N	SUM_x	SUM_y	$SUMSQ_x$	$SUMSQ_y$
C1	7	14	22	36	80

Community Detection in Social Network Graphs

(ZI 2017) Zadan je graf društvene mreže na slici.

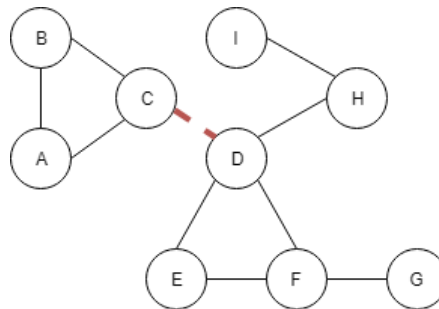
- Navedite izlaz girvan algoritma nakon 3 koraka.
- Nacrtajte AGM (Affiliation Graph Model) za mrežu nakon provođenja algoritma u zadatku a).
- Navedite definiciju društvene mreže.



(3 boda)

RJEŠENJE:

a) izlaz Girvan-Newman algoritma nakon 3 koraka



1. korak

Jel može netko objasniti na koji način se dođe do tih brojeva na vezama?

Gledaš bridove koji su mostovi... npr. brid DH je sigurno dio najkraćeg puta od bilo kojeg od čvorova (D, E, F, G) do čvorova (I, H). Pomnožiš $4 * 2$ (4 čvora u prvoj skupini, 2 u drugoj) i dobiješ 8.. To je edge betweenness brida DH.

Brid IH je dio najkraćeg puta od svih čvorova do čvora I. Sad imaš (H, D, E, F, G) - njih 5 - i (I) - 1. $5 * 1 = 5$.

Za brid GF je isto kao i za I.

itd.

Bridove DE i DF nema baš puno smisla gledati jer će njihov betweenness biti podijeljen (dio putova ide kroz DE, dio kroz DF).

Zasto u tome di racunas IH ne uzimas u obzir ostale cvorove (a,b,c) ? gledam sliku drugog koraka - u njemu su vrhovi A, B i C već odvojeni. Ako gledaš prvi korak, tad je betweenness brida IH 8 ($8 * 1$), brida HD 14 ($7 * 2$), a brida CD 18 ($6 * 3$)

Dakle ovi bridovi tipa ED (3) ili DF(6) kako si do njih dosao? *koji bridovi su mostovi?

[https://en.wikipedia.org/wiki/Bridge_\(graph_theory\)](https://en.wikipedia.org/wiki/Bridge_(graph_theory))

Mostovi u ovom slučaju su IH, DH i FG jer kad bi maknuo neki od tih bridova, dobio bi novu komponentu povezanosti (odnosno novi cluster u ovom kontekstu).. Ako makneš DE, ne dobiješ novu komponentu jer je E i dalje s D povezan preko F.

Nema smisla računati betweenness za bridove koji nisu mostovi jer će betweenness mostova uvijek biti veći.

Nije bitno koje su to vrijednosti (3 i 6 u ovom slučaju), just ignore it.

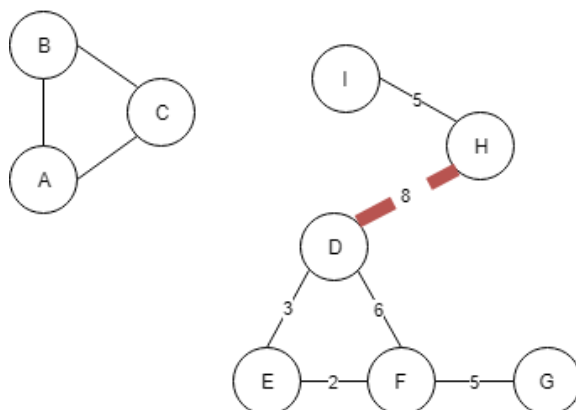
Kul :D tnx

Ako stvaarno želiš znati:

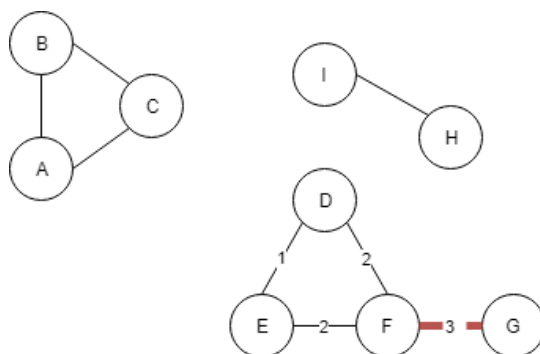
Brid ED sudjeluje u 3 najkraća puta: od E do D, od E do H i od E do I. Ako želiš od E doći do F, ići ćeš preko EF (samo 1 korak); ako želiš od E doći do G, ići ćeš preko EF i FG (2 koraka) - nema smisla ići preko ED pa ED ne sudjeluje u tim najkraćim putevima.

Slično za DF: to je najkraći put za od F do D, H i I te od G do D, H i I. Njegov betweenness je 6.

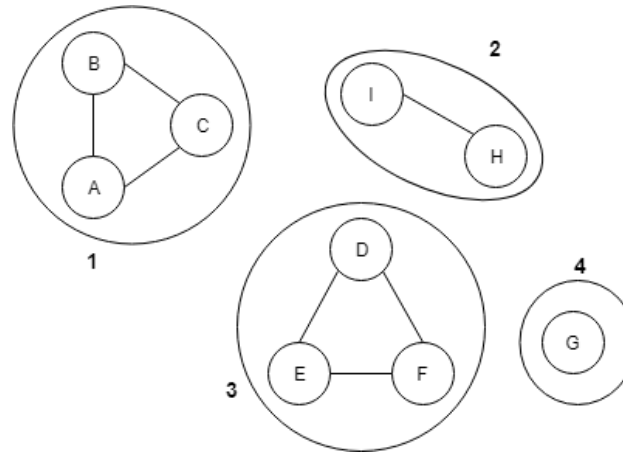
Rule of thumb: odvojiš graf na dvije komponente (npr. E, F, G i D, H, I). Pomnožiš brojeve komponenti ($3 \times 3 = 9$) i to je suma betweennessa za sve bridove koji povezuju te komponente (u ovom slučaju su to bridovi DE (3) i DF (6))



2. korak

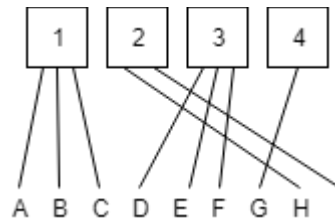


3. korak



PRO TIP: nakon n -tog koraka algoritma Girvan-Newman, mreža će biti podijeljena na $(n+1)$ zajednicu.

b) AGM



c) def. društvena mreža

- 1) Collection of entities (usually people, but not necessary)
- 2) At least one relationship exists between entities (friendship, follower, ...)
 - a) unidirectional/bidirectional
 - b) binary/weighted
- 3) Assumption of non-randomness (locality)
 - a) If entity A is related to both B and C, there is higher than average probability that B and C are related

HINT: dodatni [linž](#) [k](#)

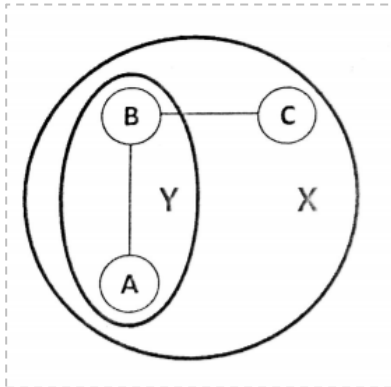
potvrda netko ???

Naišao sam na [ovaj link](#) na kojem je objašnjeno, ali ne vidim taj postupak ovdje. Postoji možda neko jednostavnije objašnjenje?

Za gornji graf se najkraći put može naći BFSom. Za izračunat *betweenness* jednog brida e , prođeš n povrh 2 parova čvorova (s, t) i zbrojiš broj najkraćih puteva između (s, t) koji prolaze kroz brid e podijeljeno sa brojem najkraćih puteva između (s, t) .

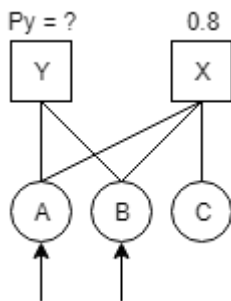
(ZI 2018) (ZI 2016) Graf društvene mreže zadan na slici podijeljen je na dvije zajednice X i Y. Nacrtajte AGM (engl. Affiliation graph model) za zadanu mrežu.

Izračunajte vjerojatnost da su entiteti u zajednici Y povezani, ako je poznato da vjerojatnost entiteta u zajednici X iznosi $p_x = 0.8$. **(4 boda)** Asistent dodao: $P(AGM) = 0.01$



Rješenje

AGM (Affiliation graph model)



$$P(AGM) = 0.01$$

$$p_x = 0.8$$

$$p_y = ?$$

Čvor u i v su povezani s vjerojatnošću p_c ako vrijedi

$$P(u, v) = 1 - \prod_{c \in M_u \cap M_v} (1 - p_c)$$

, gdje je $P(u, v)$ ukupna vjerojatnost.

$$P(a, b) = 1 - (1 - p_x)(1 - p_y) \text{ povezani}$$

$$P(a, c) = 1 - (1 - p_x) = p_x \text{ nisu povezani} \Rightarrow 1 - p_x$$

$$P(b, c) = 1 - (1 - p_x) = p_x \text{ povezani}$$

$$P(AGM) = [1 - (1 - p_x)(1 - p_y)] * (1 - p_x) * p_x$$

$$0.01 = [1 - 0.2(1 - p_y)] * 0.2 * 0.8$$

$$0.01 = 0.16 [1 - 0.2 + 0.2p_y]$$

$$0.0625 = 0.8 + 0.2p_y$$

$p_y = -3.6875$ //kako vjerojatnost nečeg može bit negativna, nema veze, možda su krivo namjestili brojeke u zadatku??? Ispravi :)

//svejedno ispadne minus. Mislim da je problem sto je ovo ipak nested model, a ne overlapping pa ne bi trebala vrijediti ista pravila, ne znam.

Da nije možda $P(\text{AGM}) = P(\text{AGM} \mid p_y) * p_y$?

Meni se formule čine ok osim ove za $P(\text{AGM})$, meni je ovako

$$P(\text{AGM}) = [1-(1-p_x)(1-p_y)] * [1-(1-(1-p_x))] * [1-(1-p_x)]$$

a to se dobije kad se uvrštava u formulu ovako

$$P(\text{AGM}) = P_{ab} * (1-P_{ac}) * P_{bc}$$

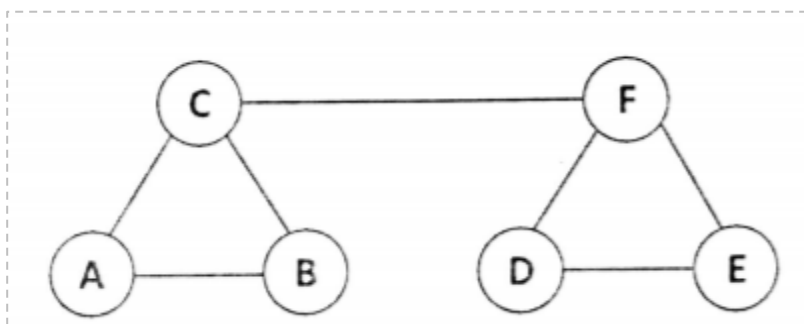
Dobijem $P_y = 0.25$

Referenca : zadatak 22 u zbirci, vrlo sličan

Tvoj navedeni $P(\text{AGM})$ izraz je drugaciji od gornjeg samo u tome sto nisi nista skracivao/la, i nisi odmah unosio/la vrijednosti od p_x . Ako ti se da, slikaj kako dodes do 0.25

Ja sam dobila 0.786, još netko tako? Kako?

(ZI 2016) Odredite zadovoljava li graf na slici svojstva društvene mreže. Obrazložite odgovor. (4 boda)



Rješenje

Prosječna povezanost bilo koja dva čvora u mreži je:

$$p = \frac{\frac{v - 2}{n(n - 1)} - 2}{2}$$

, gdje je v broj veza, a n broj čvorova.

U našem slučaju je broj veza $v = 7$, a broj čvorova $n = 6$ (sa slike), pa je povezanost jednaka $5/13$, odnosno **$p = 0.38462$** .

Lokalnost

Čvor A - 1 pozitivan (A-C-B)

Čvor B - 1 pozitivan (A-B-C)

Čvor C - 1 pozitivan (A-B-C)
- 2 negativna (A-C-F), (B-C-F)

Druga strana je simetrična, stoga imamo **3+** i **2-** s jedne i druge strane, što je ukupno **6+** i **4-**.

Vjerojatnost da su dva čvora povezana s obzirom na lokalnost je:

$$\frac{\textit{pozitivni}}{\textit{pozitivni} + \textit{negativni}}$$

Kad uvrstimo broj pozitivnih = 6 i broj negativnih = 4, dobijemo vrijednost **0.6**.

S obzirom da je vjerojatnost povezanosti čvorova s obzirom na svojstvo lokalnosti (0.6) **veće** od prosječne povezanosti čvorova (0.3846), ovaj graf **zadovoljava** svojstvo lokalnosti i time zadovoljava svojstvo grafa društvene mreže.

Recommendation Systems

Collaborative Filtering

(ZI 2018) (ZI 2017) U tablici ispod zadana je matrica ocjena korisnika za pojedine filmove (eng. *user-item matrix*). Prazna polja u matrici predstavljaju ocjene koje nedostaju. Korištenjem algoritama suradničkog filtriranja (eng. *Collaborative Filtering*) potrebno je izračunati preporuku-ocjenu za korisnika U_4 i film M_5 .

- Koristi se *Item-Item* pristup suradničkog filtriranja? Ovo mi ispada 1.333 (uz pretpostavku da uzmemo i M3 (0.22361) i M4(0.1118) kao slične
Ispravno ti je, provjerio sam i preko petog labosa
- Koristi se *User-User* pristup suradničkog filtriranja? Uzmem U1 (0.577) i U3 (0.477) i dobijem 2.905?
Meni je isto toliko ispalo :D

x	U1	U2	U3	U4	U5
M1	2	1	1		3
M2		3	3		4
M3			3	1	2
M4	4	2	5	2	2
M5	2	5	4	?	1

(3 boda)

Rješenje



Item-Item matrica, normalizirana

	U1	U2	U3	U4	U5
M1	$1/4$	$-3/4$	$-3/4$		$5/4$
M2		$-1/3$	$-1/3$		$2/3$
M3			1	-1	0
M4	1	-1	2	-1	-1
M5	-1	2	1	?	-2

User-user matrica, normalizirana

	U1	U2	U3	U4	U5
M1	$-2/3$	$-7/4$	$-11/5$		$3/5$
M2		$1/4$	$-1/5$		$8/5$
M3			$-1/5$	$-1/2$	$-2/5$
M4	$4/3$	$-3/4$	$9/5$	$1/2$	$-2/5$
M5	$-2/3$	$9/4$	$4/5$?	$-7/5$

(ZI 2016) U tablici ispod zadana je matrica ocjena korisnika za pojedine filmove (eng. *user-item matrix*). Prazna polja u matrici predstavljaju ocjene koje nedostaju. Korištenjem algoritama suradničkog filtriranja (eng. *Collaborative Filtering*) potrebno je izračunati preporuku-ocjenu za korisnika U_1 i film M_1 .

- Ako se koristi *Item-Item* pristup suradničkog filtriranja? **3.6455**
- Ako se koristi *User-User* pristup suradničkog filtriranja? **2**

x	U1	U2	U3	U4	U5	U6
M1	?	1		2	2	
M2	4	1	2	5	3	3
M3	1		5	5	2	4
M4	3		3		5	
M5	4	3		2	4	
M6	1	4	4	1	1	5

U oba slučaja kardinalni broj skupa sličnih filmova/korisnika koje sustav preporuke razmatra iznosi najviše 3.

Nadalje, sustav ne razmatra korisnike i filmove koji nisu slični ($sim(A,B) \leq 0.0$).

Tako npr. može biti i samo jedan (ili čak nijedan) sličan entitet, jer svi entiteti imaju negativnu sličnost s obzirom na entitet za kojeg se procjenjuje preporuka.

U slučaju da nema sličnih entiteta, napisati da sustav ne može izračunati preporuku. Za računanje sličnosti koristi se *PCC* (eng. *Pearson Correlation Coefficient*).

Dakle, potrebno je od pojedinih ocjena oduzeti prosjek filma (za *Item-Item* pristup), odnosno prosjek korisnika (za *User-User* pristup) te nad normaliziranim ocjenama izračunati cosine mjeru sličnosti.

- c. Navesti prednosti i nedostatke suradničkog filtriranja u odnosu na ostale načine preporučivanja?

(5 bodova)

Rješenje

a)

Sličnost računamo formulom:

$$sim(M_A, M_B) = \frac{\sum_i r_{i_A} \cdot r_{i_B}}{\sqrt{\sum_i r_{i_A}^2 \cdot \sum_i r_{i_B}^2}}$$

, gdje je u našem slučaju:

$$\sum_i r_{i_{M_1}}^2 = \frac{2}{3}$$

HINT: izračunaj prvo brojnik $\sum_i r_{i_A} \cdot r_{i_B}$ ako je ≤ 0 , redak se preskače (nazivnik sigurno ≥ 0).

	U_1	U_2	U_3	U_4	U_5	U_6	$sim(M_i, M_1)$
M_1	?	-0.667	0	0.333	0.333	0	1.0
M_2	1	-2	-1	2	0	0	$\frac{2}{\sqrt{\frac{2}{3} \cdot 10}} = 0.7746$
M_3	-2.4	0	1.6	1.6	-1.4	0.6	$\frac{0.06667}{\sqrt{\frac{2}{3} \cdot 13.2}} = 0.02247$
M_4	-0.667	0	-0.667	0	1.333	0	$\frac{0.444}{\sqrt{\frac{2}{3} \cdot \frac{8}{3}}} = 0.3333$
M_5	0.75	-0.25	0	-1.25	0.75	0	0 (brojnik je 0)
M_6	-1.667	1.333	1.333	-1.667	-1.667	2.333	< 0 (brojnik < 0)

Uzimamo 3 filma u obzir: M_2, M_3, M_4 .

Procjenjujemo ocjenu M_1 za korisnika U_1 koristeći ocjene korisnika U_1 za filmove M_2, M_3, M_4 .

PRO TIP: koristimo stvarne (ne normalizirane) ocjene!

$$r'_{M_1, U_1} = \frac{sim(M_2, M_1) \cdot r_{M_2, U_1} + sim(M_3, M_1) \cdot r_{M_3, U_1} + sim(M_4, M_1) \cdot r_{M_4, U_1}}{\sum_{i=2}^4 sim(M_i, M_1)}$$

$$r'_{M_1, U_1} = \frac{0.7746 \cdot 4 + 0.02247 \cdot 1 + 0.3333 \cdot 3}{1.13037}$$

$$r'_{M_1, U_1} = 3.6455$$

b)

Rješavamo analogno *item-item*, ali ovdje gledamo **stupce**.

Tako je u našem slučaju:

$$\sum_i r_{U_1, i}^2 = 9.2$$

	U_1	U_2	U_3	U_4	U_5	U_6
--	-------	-------	-------	-------	-------	-------

M_1	?	-1.25	0	-1	-0.833	0
M_2	1.4	-1.25	-1.5	2	0.1667	-1
M_3	-1.6	0	1.5	2	-0.533	0
M_4	0.4	0	-0.5	0	2.1667	0
M_5	1.4	0.75	0	-1	1.667	1
M_6	-1.6	1.75	0.5	-2	-1.6167	< 0
$sim(U_i, U_1)$	1.0	< 0	< 0	0.2064	0.6167	< 0

Uzimamo 2 korisnika u obzir: U_4, U_5 .

$$r'_{U_1, M_1} = \frac{1.6462}{0.8231}$$

$$r'_{U_1, M_1} = 2$$

c)

Prednosti: radi s različitim tipovima entiteta (nije potrebno domensko znanje)

Nedostaci:

- **sparsity** - utility matrica je često rijetka - teško se pronalazi korisnike koji su ocijenili iste objekte
- **first rater** - ne može predložiti neocijenjene objekte
- **popularity bias** - favorizira popularne objekte; korisnici sa specifičnim ukusom predstavljaju problem
- **cold start** - novi korisnici nisu ocijenili dosta proizvoda (kod user-user), novi proizvodi nemaju dosta ocjena (kod item-item)

(LJIR 2017) Dva sustava za preporučivanje R1 user-item tablica ima posjetitelje - države, a sadrži 1 ako je posjetitelj posjetio državu, inače sadrži 0. R2 user-item tablica ima posjetitelje - države, a sadrži ocjenu države ako ju je korisnik ocijenio.

- Definirati matematičkih izrazom Jaccard, cosine i Pearson mjere sličnosti.
- Koja od te 3 je najprimjerenija za R1 i zašto druge nisu?
- Koja od te 3 je najprimjerenija za R2 i zašto druge nisu?

(3 boda)

Rješenje

a)

Jaccard similarity of two sets is the size of their intersection divided by the size of their union:

$$\text{sim}(S_1, S_2) = |S_1 \cap S_2| / |S_1 \cup S_2|$$

$$\text{sim_cosine}(x, y) = \text{cosine}(r_x, r_y) = \frac{r_x \cdot r_y}{\|r_x\| \cdot \|r_y\|}$$

$$\text{sim_pearson}(x, y) = \frac{\sum_{s \in S_{xy}} (r_{xs} - \bar{r}_x)(r_{ys} - \bar{r}_y)}{\sqrt{\sum_{s \in S_{xy}} (r_{xs} - \bar{r}_x)^2 \sum_{s \in S_{xy}} (r_{ys} - \bar{r}_y)^2}}$$

- b) Jaccardova jer ne uzima u obzir vrijednosti ocjena (koje su u ovom slučaju binarne) i slični će biti oni korisnici koji su posjetili iste države.
- c) Pearson zato što su ocjene brojevi u nekom rasponu i oduzimanjem srednje vrijednosti se neunesene vrijednosti ne tretiraju kao negativne.

Content-based

(ZI 2018) Content-based sustav s računalima A, B i C koji svaki imaju 3 karakteristike: CPU, GB i RAM.

	A	B	C
CPU	2.92	3.02	2.96
GB	570	450	680
RAM	6	4	6

Korisnik U1 ocijeni računala A, B i C redom ocjenama 4, 2 i 5. Potrebno je odrediti profil korisnika iz prosjeka profila računala. Pritom je potrebno normalizirati ocjene korisnika oduzimajući prosječnu vrijednost.

(5 bodova)