Praktiskais darbs - Eirovīzija

Pēteris Račinskis pr20015

05/01/22

Piebilde: praktiskā darba nosacījumos nav norādīts sagaidāmais atskaites apjoms. Kursa aprakstā tas tiek saukts par kursa darbu, kas liecina, ka tiek gaidīts aptuveni 15-30lpp. garš dokuments ar atbilstošu ieguldītā darba un informācijas apjomu. Atvainojos, ja iesūtītā atskaite ir ievērojami plānāka, nekā tiek gaidīts - taču kaut kāds darbs ir padarīts, vairākas dienas laika ir iztērētas, personīgais sesijas grafiks liek virzīties tālāk pie citiem uzdevumiem, un, ja pastāv kaut neliela iespēja uzlabot gala vērtējumu, uzskatu, ka ir vērts iesūtīt pat potenciāli nepietiekamu rezultātu.

1. Ievads

1.1. Uzdevums

Datizraces uzdevumi var būt dažādi. Viens no veidiem, kā tie var radikāli atšķirties pēc būtības, ir pirms pētījuma sākšanas pastāvošā skaidrība par rezultāta formu. Ja kādreiz sākam ar datu kopu, par ko nekas nav zināms, un mēģinām gūt vispārīgus priekštatus par starp tās elementiem pastāvošajām sakarībām, citreiz jau no paša sākuma ir laba izpratne par to, ko vēlamies sasniegt, kādi ir datu kopu veidojošie mehānismi un kādus skaitļus varētu redzēt iegūto aprakstošo modeļu parametros. Šis uzdevums visai pārliecinoši pieder otrajai kategorijai. Prasīts atbildēt uz ļoti specifisku jautājumu. Tāpēc tā vietā, lai sāktu ar ļoti vispārīgu datizraces metožu lietojumu, varam pielāgot vai izstrādāt darba rīkus tieši viena jautājuma atbildēšanai. Turklāt jautājums uzdots par datu kopas ģenerējošo procesu - eirovīzijas dziesmu konkursu - nevis par kādu konkrētu, specifiskā veidā strukturētu korpusu, kas ļauj patstāvīgi izvēlēties maksimāli piemērotu informācijas avotu, ierobežot definīcijas apgabalu pēc saviem ieskatiem, u.t.t.

Intuitīvi uzreiz rodas priekštats, kas domāts ar terminiem "kaimiņu būšana" un "objektīvāks novērtējums". Taču ar to nepietiek, lai iegūtu kaut kādu šo fenomenu skaitlisku izteiksmi. Nepieciešams definēt "objektīvu novērtējumu" un attiecīgi - novirzes no tāda. Viens veids, kā to darīt, varētu būt ieviest latentu dziesmu "popularitātes" mēru. Tādu var modelēt, iztēlojoties "demokrātisku" visu skatītāju balsošanu par, viņuprāt, labāko konkursa kārtas dalībnieku dziesmu:

$$i \in \{1, 2, ..., K\} = [K] \tag{1}$$

$$N_i - \text{balsis par dziesmu}; N - \text{balsis kopā} \tag{2}$$

$$q_i = \frac{N_i}{N}; \sum_{i \in [K]} q_i = 1 \tag{3}$$

$$s_i \in [10] \cup 0 - \text{punktu skaits (score)}; s_i \sim P(s_i = x \mid q_i, K) \tag{4}$$

$$q_i \leq q_j \Rightarrow s_i \leq s_j \tag{5}$$

$$s_i, s_j \neq 0 \Rightarrow s_i \neq s_j \tag{6}$$

kur $P(x \mid q_i, K)$ ir sadalījums, kas apraksta katra iespējamā diskrētā novērtējuma (punktu skaita) varbūtību, pieņemot, ka dziesmas "demokrātiskā" balsojuma varbūtība ir q_i. Šķiet, ka šis sadalījums varētu kaut kādas formas binomiālais, (vai arī kaut kas krietni sarežģītāks), taču tā precīzā analītiskā forma nav svarīga tālākiem aprēķiniem. Svarīgi piebilst, ka eirovīzijas vērtējumu sistēmā parasti punkti pieder kopai $[8] \cup \{0, 10, 12\}$, un šīm skaitliskām vērtībām ir nozīme, rēķinot gala rezultātu (punkti tiek skaitīti kopā), taču

katras dalībvalsts vērtējuma piešķiršanas procesā šiem skaitļiem ir tikai ordināla nozīme, t.i., $10 < 12$; $\forall i \in [8] : i < 10$. Tāpēc var pieņemt, ka $s_i \in [10]$ un vajadzības gadījumā izmantot pārveidojumu $\hat{s}_i = f(s_i); f(9) = 10; f(10) = 12; f(x \neq 9, 10) = x$.

Tad sagaidāmo punktu skaitu, ja balsojums notiek tikai vienreiz un sakrīt ar "objektīvo" novērtējumu (vienā valstī, visās kopā), var izteikt kā:

$$E[s_i^1] = \sum_{s \in [10]} P(s \mid q_i, K) * s \tag{7}$$

bet, ja balsojums tiek atkārots vairākas reizes un tiek skaitīta to svērto variantu summa (K' ir balsojošo dalībvalstu skaits, kas daudzreiz ir tāds pats, kā uzstājošos dalībvalstu skaits, bet ne vienmēr, jo nesenākos konkursos ieviesta pusfinālu sistēma un visas valstis var balsot finālā):

$$E[\hat{s}_i^{K'}] = \sum_{j \in [K']} \sum_{s \in [10]} P(s \mid q_i, K) * f(s) \tag{8}$$

Ja interesē nevis dziesmas rezultāts konkursa uzvarētāja noteikšanai, bet tās vidējā ordinālā pozīcija katras balsotājvalsts vērtējumā, pārveidojumu $f(x)$ var (varētu pat teikt, ka nepieciešams) atmest:

$$E[s_i^{K'}] = \sum_{j \in [K']} \sum_{s \in [10]} P(s \mid q_i, K) * s = K' * E[s_i^1] \tag{9}$$

Kā redzams, rezultāts nav atkarīgs no katras dalībvalsts un ir "objektīvs". Ieviest nobīdes nacionālajos balsojumos varētu ar svariem:

$$E[s_i^{K'}] = \sum_{j \in [K']} E[s_i^1] * w_{ji}, \sum_{i \in [K]} w_{ji} = 1 \Rightarrow E[s_i^{K'}] = E[s_i^1] \tag{10}$$

un tad "neobjektivitāti" varētu potenciāli labot, lai atjaunotu sagaidāmo vērtību, reizinot svarus ar korekcijas koeficientiem, kas iegūti, dalot svarus ar vienmērīgam sadalījumam (pār vērtējamām dalībvalstīm, nevis balsojošajām) atbilstošajiem:

$$c_{ji} = \frac{w_K^0}{w_{ji}} = \frac{\frac{1}{K}}{w_{ji}} = \frac{1}{K * w_{ji}} \tag{11}$$

un visu kopā apkopot korekciju matricā:

$$C = \begin{bmatrix} c_{11} & \dots & c_{1K'} \\ \vdots & \ddots & \vdots \\ c_{K1} & \dots & c_{KK'} \end{bmatrix} \tag{12}$$

Lai šo korekciju matricu pielietotu rezultātu labošanai, punktu matricu (bez $f(s)$ pārveidojuma) pa elementiem reizina ar korekciju matricu:

$$S = \begin{bmatrix} s_{11} & \dots & s_{1K'} \\ \vdots & \ddots & \vdots \\ K1 & \dots & s_{KK'} \end{bmatrix} \tag{13}$$

$$S' = C \circ S \tag{14}$$

Kas tad īsti ir iegūts, un kā no tā aprēķināt konkursa rezultātu? Jāatceras, ka s_i ir punktu skaits, kas iegūts, pēc slēptā mainīgā q_i kārtojot konkursa dalībniekus un piešķirot punktus 10 labākajiem. Ir izdarīts pieņēmums, ka katra valsts vispirms ieguvusi

šos punktu skaitus no vienādiem varbūtību sadalījumiem, tad pareizinājusi ar svariem. Koriģējot, atgriezta pēdējā darbība, un iegūti punktu skaiti, kādi tie būtu pirms šīs fiktīvās svēršanas operācijas, pēc fiktīvas demokrātiskas balsošans un punktu piešķiršanas.

Protams, ka realitātē process ir citāds: svari netiek pielietoti punktiem, tā vietā jau punktu skaitu sadalījums ir kropļots - precīzākus rezultātus varētu iegūt, rēķinot korekcijas saņemto balsu skaitiem, ja tie būtu zināmi (un nebūtu žūrijas komponentes, kas visu šo "demokrātijas" modeli padara par vienkārši nederīgu). Turklāt visas vērtības, kas nav bijušas augstākajā desmitniekā no datu kopas vienkārši ir izgrieztas (vienādas ar 0). Tāpēc jau uzreiz var pateikt, ka no matemātiska viedokļa, ar šādu matricu nav iespējams atjaunot slēpto q_i sadalījumu. Taču no statistikas kursa pagājušajā semestrī zināms, ka t.s. "rangu" metodes, kas strādā ar kārtas skaitļiem, nevis skaitliskām vērtībām tiešā veidā, parasti uzvedas vismaz virpsusēji līdzīgi nepārtrauktajām, un bieži vien ir algoritmiski vienkāršākas (pat ja kaut ko par tām pierādīt mēdz būt grūtāk), tāpēc var pastāvēt zināma cerība, ka pat matemātiski nekorektas un nepilnīgas, no nepilnīgiem datu korpusiem iegūtas korekcijas, varētu darboties vismaz pareizajā virzienā.

Attiecīgi tiek izvirzīts sekojošs korekcijas modelis: tā kā iegūtas ir "atjaunotās" rangu sagaidāmās vērtības, tās drīkst vienkārši pārkārtot - piešķirt kārtas skaitļus no mazākās uz lielāko - un izdarīt korekciju $f(s)$, lai svērtu kārtas skaitļus summās starp balsojošajām valstīm.

$$S'' = S' \text{ rangos 1-10 (pārējie = 0), pa kolonnām} \tag{15}$$

$$\hat{S}'' = f(S'') \tag{16}$$

$$\hat{s}_i = \sum_{j \in [K']} \hat{s}_{ij} \tag{17}$$

Jāmin, ka šādi nav iespējams izšķirt tieši "kaimiņu būšanu" - ko varbūt gribētos rakstu-rot kā tīri etniskas tuvības vai ģeopolitisku interešu sakritības motivētu nobīdi balosjumu rezultātos. No citiem faktoriem, kas arī atšķiras valstu starpā - kulturālas noslieces, demogrāfiskie sadalījumi, konkursa popularitāte, u.t.t. - radušās nobīdes skaitliski izskatītos tāpat.

1.2. Datu kopas

Kā jau minēts iepriekš, uzdevums ir par fenomenu, nevis tā radītu konkrētu datu kopu. Tāpēc iespējams ne tikai brīvi pēc saviem ieskatiem pārveidot vienu datu kopu, bet apzināti meklēt un izvēlēties jau maksimāli atbilstoši noformētu. Nav arī dots stingrs uzstādījums, ka obligāti jāstrādā ar visu konkursa vēsturi. Laika gaitā ir notikušas daudzas noteikumu un organizatoriskas izmaiņas, kas var apgrūtināt dažādu periodu rezultātu salīdzināšanu.

Par datu kopu izvēlēta *Eurovision Song Contest Dataset* (pieejama *GitHub* repozitorijā), kur jau atrodams korpuss *votes.csv*. Tajā katrā rindā dots notikuma gads, atbilstošā stadija (fināls; pusfināls; pirmais vai otrais pusfināls gados, kad ir divi), vērtējošā valsts un punktus saņemošā valsts.

Pietiek vien atvērt *Wikipedia* rakstu par konkursa balsošanas kārtību, lai kristu nelielā panikā. Garākais (un, Latvijas iedzīvotājiem, interesantākais) posms ar samērā noturīgu balsošanas kārtību ir 1980-2015, tāpēc tālāk tieši ar to arī pārsvarā strādāts.

2. Metodes

2.1. Modelis

Sadaļā 1.1. aprakstīts, kā varētu izskatīties korekcijas matrica, un radīta aptuvena nojausma par procesu, kas šādu matricu varētu ģenerēt, taču tā nav konstruktīva - svari w_{ji} *a priori* nav zināmi, tos nepieciešams noteikt empīriski. Līdz šim arī aplūkots tikai gadījums ar vienu q_i "kvalitātes vērtību" sadalījumu. Dažādās konkursa kārtās šie sadalījumi var radikāli atšķirties. Vienam konkursa etapam korekcijas varētu vienkārši algebriski izrēķināt, taču, ja vēlamies noteikt korekcijas matricu C noteikt globāli, jāveido modeļa šablons un jāapmāca.

Var krietni palauzīt galvu, domājot par veidiem, kā to izdarīt. Pirmā ideja, kas varētu rasties, varētu būt vienkārši ņemt punktus tiešā veidā no datu kopas un apmācīt klasifikatoru formā (vērtējošā valsts, vērtējamā valsts) $\rightarrow$ vērtējums (kur vērtējums vai nu skaitliks novērtējums, vai diskrētu vērtību vektors, kur 1 apzīmē konkrētu punktu skaitu). Problēma ar šādu pieeju ir tāda, ka dažādu valstu "kvalitātes" vērtību sadalījumi var ievērojami atšķirties, t.i., dažām valstīm varbūt kopumā veicas labāk vai sliktāk, nekā citām, neatkarīgi no nobīdēm tieši nacionālās balsošanas rezultātā. Vērtējumu paredzēšanas uzdevumam tas netraucē, taču pēc tam grūti spriest, kā iegūtos rezultātus izmantot korekcijām. Turklāt gan ieejas vērtību kopa ir gana liela, salīdzinot ar pieejamo datu korpusa apjomu. Bieži vien pieejami pavisam nedaudzi novērtējumi no vienas valsts uz otru.

Apsverot šo un citus semestra gaitā iepazītos tipveida rīkus datizraces uzdevumu risinājumam, īsti piemērots tieši šim uzdevumam nelikās neviens. Beigās tomēr tika nolemts īstenot savu, lielā mērā intuitīvi vadītu pieeju, cerībā, ka vismaz "kaimiņu būšanas" principu izdosies atrast. Tā vietā, lai pakāpeniski optimizētu šablonu, iterējot pār datu kopas elementiem, datu kopu var mēģināt reducēt uz formu, kur rezultāts atrodams algebriski. Šajā konkrētajā gadījumā tas darīts, nosakot vidējās empīriskās vērtības piešķirto punktu skaitam katram valstu pārim katrā virzienā, pārveidojot tās par divdimensionālu varbūtību sadalījumu un pielīdzinot rezultātu vienmērīgajam sadalījumam.

To dara, atrodot divus korekcijas koeficientus: r_i, kas vienādo rindu varbūtību summas (savā ziņā kompensējot q_i) un c_{ij}, kas vienādo varbūtības koriģētās matricas kolonnās un reizē ir arī korekcijas matricas vērtības.

$$p_i = \sum_{j \in [K']} p_{ij} \tag{18}$$

$$p_i * r_i = \frac{1}{K} \tag{19}$$

$$r_i = \frac{1}{p_i * K} \tag{20}$$

$$p'_{ij} = p_{ij} * r_i \tag{21}$$

$$p'_{ij} * c_{ij} = \frac{\sum_{i \in [K]} p'_{ij}}{K} \tag{22}$$

$$c_{ij} = \frac{\sum_{i \in [K]} p'_{ij}}{p'_{ij} K} \tag{23}$$

$$w_{ij} = \frac{1}{c_{ij} K} \tag{24}$$

Par šī algoritma matemātisko pareizību pārliecības nav nekādas - pirmais kompensācijas solis koriģē q_i, balstoties uz stipri kropļotiem vērtējumiem, un svari tad tiek rēķināti no šī nekorektā starprezultāta. Taču vismaz virspusēji šķiet, ka iegūie rezultāti varētu būt noderīgi.

Iespējams, ka varētu šo procesu pilnveidot, ieviešot iterāciju un cerot uz konverģenci, atkārtoti veicot korekcijas aprēķinu koriģētiem datiem, lai gan šādas funkcionalitātes ieviešanai laika nav pieticis. Galvenais šķērslis ir šaubas par to, kā pareizi kombinēt soļos iegūtās korekcijas matricas - vai tas vispār ir pamatojams. Taču zināms, ka līdzīgas metodes izmanto citur datizracē, lai cīnītos ar t.s. *distribution shift* jeb atšķirību starp modeļa ģenerētiem sadalījumiem un reāliem.

2.2. Datu priekšapstrāde

Pirms iespējams sākt aprēķinus, datu korpusu jāpārveido tiem padevīgā formā. Kopu ar kortežiem formā (t, i, j, s) nepieciešams izteikt kā tensoru ar elementiem s_{ijt}, kur t - konkursa etaps. Ja konkursa etapu dalībnieki un balsotāji vienmēr būtu tie paši, šis būtu triviāls uzdevums. Taču sarežģījumus rada fakts, ka neeksistējošas vērtības un nulles vērtības nav viens un tas pats. Ne katra dalībvalsts piedalās katrā etapā un ne katra dalībvalsts, kas balso, arī piedalās konkursā. Tāpēc neeksistējošas vērtības nepieciešams marķēt atsevišķi. Izvēlētais risinājums ir tās marķēt ar -1, un pēc tam datu apstrādē īpaši apstrādāt šādi marķētus ierakstus, kur nepieciešams. Vērtībām 10,12 tiek ieviesta jau iepriekš aprakstītā korekcija uz 9,10.

Datu priekšapstrāde veikta repozitorijā pieejamajā *Python* skriptā *datagen.py*. Principā priekšapstrādes solī iespējams arī veikt vidējo vērtību rēķināšanu, kas droši vien arī būtu ātrāk nekā faktiski realizētajā tensora ģenerēšanas pieejā, taču tīri praktiski apsvērumu vadībā tika izlemts visas netriviālās datu manipulācijas atstāt atsevišķi: darba autors daudz labāk pazīstams ar n-dimensionālu homogēnu datu bloku skaitļošanas un lineārās algebras bibliotēku *numpy* nekā ar 2-dimensionālu heterogēnu datu korpusu apstrādes bibliotēku *pandas*. Izstrādes procesā vieglāk iteratīvi izmainīt visu procesu, strādājot ar jau gatavu datu tensoru. Lai nebūtu katru reizi jāveic tensora veidošana, kas aizņem gandrīz 40 sekundes, starprezultāts tiek saglabāts failā. Datu pārveidošanu ir iespējams paātrināt daudzkārtīgi, taču tad jāizmanto cita darbību secība, ko grūtāk atkļūdot. Reizi dažās stundās zaudēt 40s šķiet mazāks zaudējums, nekā pavadīt vairākas stundas, pārrakstot jau pietiekami labi strādājošu kodu.

Šī ir arī laba vieta tekstā, kur norādīt, ka šādam datu formātam matrica ar vērtējošajām un vērtejamām valstīm vienmēr ir kvadrātiska, t.i., $K = K'$. Iztrūkstošās šķautnes ir atbilstoši marķētas ar vērtībām ārpus parastā definīcijas apgabala..

2.3. Modeļa aprēķins

Modeļa aprēķins tiek veikts otrā *Python* skriptā - *model.py*. Pirms iespējams aprēķināt vidējās vērtības pār etapiem, nepieciešams noteikt katra balsojuma virziena (varētu teikt, orientēta grafa šķautnes) kopskaitu. To veic funkcija *coincidence_count*, kas atgriež matricu ar vērtībām formā n_{ij} - reižu skaits, kad j-tā valsts balsojusi konkursā, kurā uzstājas i-tā. Piešķirto punktu summu atrod funkcija *coincidence_total*. Vidēji j-tās valsts i-tajai piešķirto novērtējumu tad atrod *edge_average*.

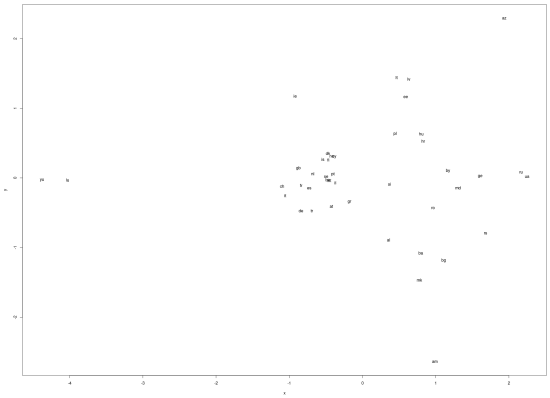
Papildus vidējās vērtības aprēķinam tiek noteikts arī kuras valstis jāizmet no kopējās matricas pēc sliekšņa kritērija - ja maksimālais n_{ij} ir mazāks par slieksni, no matricas tiek izmestas i-tās rindas un i-tās kolonnas. Tādējādi tiek apkarota traucējoša parādība - valstīm, kas piedalījušās konkursā tikai dažas reizes, korekcijas koeficienti var būt krietni

lielāki vai mazāki, nekā visām pārējām, jo lielo skaitļu likumam nav pieticis elementu, lai izlīdzinātu iegūto punktu skaitus un tuvotos sagaidāmajai vērtībai.

Nākamais solis ir iegūto vidējo vērtību pārveidošana par varbūtību sadalījumu. Sākumā tas netika darīts, un rezultāts bija grūtības ar izsekošanu vērtību semantiskajai nozīmei un pareizu algoritma realizāciju. Strādājot ar varbūtībām, daudzas potenciālas programmatiskas kļūdas var novērst, vienkārši sekojot līdzi tam, ka rindu, kolonnu vai kopējā varbūtību summa matricā ir vienāda ar 1 (atkarībā no veicamajām darbībām). Papildus tiek veikts arī matemātiski pilnīgi nepamatots taču praktiski pašsaprotams solis - visām vidējo vērtējumu matricas vērtībām tiek pieskaitīta konstante α, kas ir funkcijas parametrs. Tas tiek darīts, jo citādi nogrieztām vērtībām tiek aprēķināts svars 0 un korekcijas koeficients $+\infty$, kas korekcijas mērķiem neder. To var uztvert kā zaudētās informācijas ekstrapolāciju - valstis, kas ne reizi nav saņēmušas rangu 1-10, tik un tā gandrīz noteikti kādas balsis saņemtu, ja tiktu veikta izlase no balsis veidojošā sadalījuma. Visu augstāk minēto veic funkcija *normalized_score*.

Visbeidzot, 2.1. sadaļā aprakstīto korekcijas matricas aprēķinu veic funkcija *corrections*

3. Rezultāti



Att. 1: Koka atjaunošana pēc Prūfera koda.

Izmantojot formulu ciklomātiskā skaitļa aprēķinam

$$r(G) = m - n + k \tag{25}$$

var pierādīt dažādu modifikāciju izraisītās izmaiņas.

3.1. Virsotnes pievienošana uz malas

Formāli doto modifikāciju $f(V[G], E[G], e) = (V', E') = G'$ var izteikt kā

$$e = \{u, v\} \in E[G] \tag{26}$$

$$f(V, E, e) = (V \cup w, (E \setminus \{u, v\}) \cup \{\{u, w\}, \{w, v\}\}) \tag{27}$$

no kā izriet, ka

$$|V'| = |V| + 1 = n + 1 \tag{28}$$

$$|E'| = |E| - 1 + 2 = m + 1 \tag{29}$$

$$u, v \in K_i \text{ komponentē} \rightarrow u, w, v \in K'_i \tag{30}$$

tātad

$$r(G') = (m + 1) - (n + 1) + k = m - n + k = r(G) \tag{31}$$

3.2. Virsotnes ar pakāpi = 2 aizstāšana ar šķautni

Turpinot pēc analoģijas

3.3. Virsotnes ar pakāpi = 1 izgriešana

Paša definētā modifikācija - visvienkāršākā. Nogriež lapu.

4. Secinājumi

$$w \in V[G] : \deg(v) = 1 \rightarrow \{u, w\} \in E[G] \tag{32}$$

$$f(V, E, v) = (V \setminus w, E \setminus \{u, w\}) \tag{33}$$

$$|V'| = |V| - 1 = n - 1 \tag{34}$$

$$|E'| = |E| - 1 = m - 1 \tag{35}$$

$$u, w \in K_i \text{ komponentē} \rightarrow u \in K_i' \tag{36}$$

tātad

$$r(G') = (m - 1) - (n - 1) + k = m - n + k = r(G) \tag{37}$$

$$w \in V[G] : \deg(v) = 2 \rightarrow \{u, w\}, \{w, v\} \in E[G] \tag{38}$$

$$f(V, E, w) = (V \setminus w, (E \setminus \{\{u, w\}, \{w, v\}\}) \cup \{u, v\}) \tag{39}$$

$$|V'| = |V| - 1 = n - 1 \tag{40}$$

$$|E'| = |E| - 2 + 1 = m - 1 \tag{41}$$

$$u, w, v \in K_i \text{ komponentē} \rightarrow u, v \in K_i' \tag{42}$$

tātad

$$r(G') = (m - 1) - (n - 1) + k = m - n + k = r(G) \tag{43}$$

$$C = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$