



Podudarnost k -means klastera i Voronoevih ćelija

Diplomski rad

Anđela Perin

Broj indeksa: 123/2021

Smer: Računarske nauke

Računarski fakultet, Union Univerzitet

Mentor: dr. Marko Mladenović

Beograd, Januar 2026.

DEKLARACIJA

Ja, Anđela Perin, izjavljujem da je ovaj diplomski rad rezultat mog samostalnog i originalnog rada. Svi korišćeni izvori literature, podaci i ideje drugih autora su jasno navedeni i korektno citirani. Potvrđujem da ovaj rad nije bio prethodno predat niti korišćen za bilo koji drugi ispit ili akademski stepen. U potpunosti preuzimam odgovornost za njegovu autentičnost i integritet.

Datum: Januar 2026.

Sadržaj

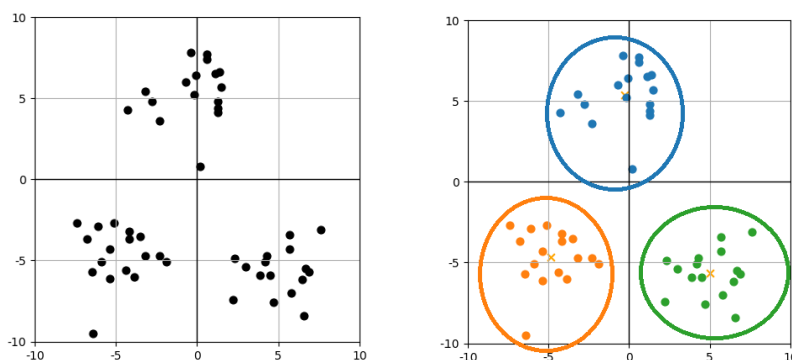
1. Uvod	3
1.1 Istorijski pregled	4
1.1.1 Istorija k -means algoritma.....	4
1.1.2 Istorija Voronojevih dijagrama.....	5
1.2 Značaj k -means algoritma i Voronojevih dijagrama	6
1.3 Ciljevi i struktura rada.....	7
2. Teorijske osnove i postavka problema	8
2.1 Princip rada k -means algoritma	8
2.2 Formalna definicija Voronojevih dijagrama.....	10
2.3 Postavka problema.....	10
2.4 Distanca Minkovskog i Voronojevi dijagrami	11
2.5 Postojeća rešenja.....	13
2.6 Primena k -means algoritma i Voronojevih dijagrama	13
3. Implementacija i analiza problema	15
3.1 Implementacija algoritma	15
3.1.1 Dodeljivanje tačaka najbližem centroidu	15
3.1.2 Generisanje Voronojevog dijagrama	16
3.1.3 Ažuriranje centroida.....	18
3.1.4 Glavna petlja k -means algoritma.....	19
3.1.5 Primer vizuelnog prikaza izvršavanja programa	20
3.2 Eksperimentalna analiza	21
3.2.1 Uvodne napomene i termini.....	21
3.2.2 Cilj analize i slučaj potpune podudarnosti.....	22
3.2.3 Uticaj parametra p na oblik Voronojevih ćelija	23
3.2.4 Primer skupa tačaka sa celobrojnim koordinatama	27
3.2.5 Analiza promena pripadnosti tačaka.....	28
3.2.6 Analiza tačaka u oblastima između Voronojevih ivica.....	30
4. Zaključak	33

1. UVOD

U ovom radu analiziraće se podudarnost između klastera dobijenih k -means algoritmom i Voronojevih dijagrama konstruisanih na osnovu njihovih centroida, pri korišćenju različitih distanci.

Klasterovanje predstavlja jednu od osnovnih tehnika u oblasti analize podataka i mašinskog učenja. Njegov cilj je grupisanje podataka u podskupove (klastera) tako da su elementi istog klastera međusobno što sličniji, a elementi različitih klastera što različitiji prema unapred definisanoj meri sličnosti (videti [1]).

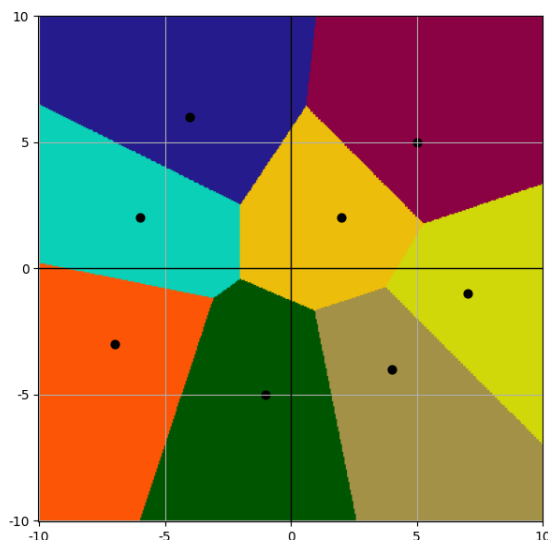
Jedan od najpoznatijih algoritama klasterovanja je k -means, koji deli podatke u k klastera tako da je suma kvadrata distanci između tačaka i njihovog centra minimalna. Ti centri klastera se nazivaju centroidi.



Slika 1.1: K -means klasterovanje

Sa druge strane, Voronojevi dijagrami predstavljaju geometrijsku strukturu koja particioniše ravan u oblasti određene datim skupom generišućih tačaka, zvanih generatori (videti [2]). Svaka oblast Voronojevog dijagrama sadrži sve tačke koje su bliže svojem generatoru nego bilo kojem drugom generatoru iz skupa. Ove oblasti se nazivaju Voronojeve ćelije, dok se granica između dve Voronojeve ćelije naziva Voronojeva ivica. Tačke gde se susreću tri ili više ćelija nazivaju se Voronojeva temena ([3]).

Voronojevi dijagrami nazvani su po ruskom matematičaru Georgiju Voronoju (Георгий Феоодосъевич Вороной) koji je predstavio njihov koncept. Još se nazivaju Voronojeva teselacija, Voronojeva dekompozicija, Voronojevo particionisanje, ili Dirihleova teselacija (po Gustavu Leženu Dirihleu), videti [2].



Slika 1.2: Voronojev dijagram sa 8 tačaka u Euklidskoj metrici

U svakoj iteraciji k -means algoritma svaka tačka se dodeljuje najbližem centru klastera prema Euklidskoj distanci između njih ([4]). Zato se često kaže da k -means u svakoj iteraciji konstruiše Voronojev dijagram nad trenutnim centrima klastera ([1]). Ova veza između klusterskih struktura i geometrijskih podela podataka predstavlja osnovu za dalja formalna istraživanja.

1.1 Istorijski pregled

Istorija k -means algoritma

K -means je veoma star algoritam za klasterovanje, ali se i danas intenzivno koristi zbog svoje jednostavnosti, brzine i dobrih praktičnih rezultata. Nezavisno su ga razvili različiti istraživači u više naučnih oblasti, što govori o njegovoj univerzalnosti. Pogledati [5] i [6].

Do ideje k -means algoritma prvi je došao matematičar Hjužo Štajnhaus (Hugo Steinhaus, 1956) u delu *Sur la Division des Corps Matériels en Parties*. U tom delu Štajnhaus je predstavio problem particionisanja heterogenog čvrstog tela odgovarajućim izborom podela. Takođe je naveo primene u oblastima antropologije i industrije.

Izraz k -means je prvi upotrebio Džejms Mekvin (James MacQueen) 1967. godine u delu *Some methods for classification and analysis of multivariate observations* ([7]).

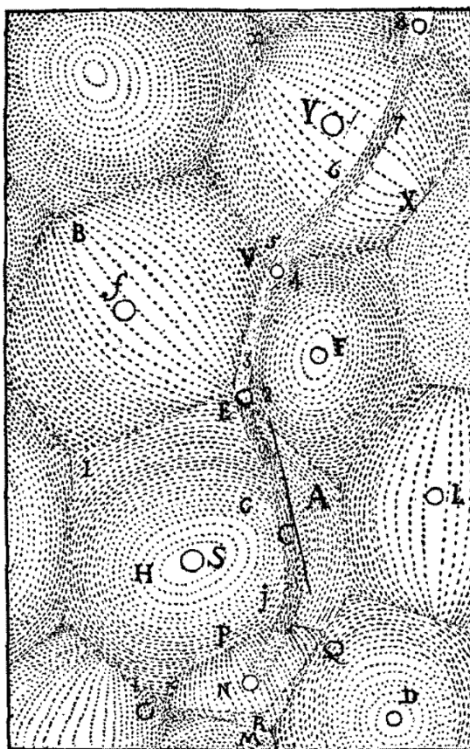
Posebne doprinose dali su i Stjuart Lojd (Stuart Lloyd, 1957), Bol i Hol (Ball and Hall, 1965),

Edvard V. Forgi (Edward W. Forgy, 1965) i Džansi (Jancey, 1966).

Tokom godina, osnovni k -means je unapređivan na mnogo načina. Razvijene su varijante koje omogućavaju fleksibilniju dodelu tačaka klasterima, automatsko određivanje broja klastera, rad sa velikim skupovima podataka, efikasnije pretraživanje najbližih centara, kao i detekciju klastera proizvoljnog oblika.

Istorija Voronojevih dijagrama

Sama ideja particije prostora na oblasti najbližeg uticaja pojavljuje se već kod Renea Dekarta (René Descartes) 1644. godine u delu *Principi filozofije* gde opisuje Sunčev sistem kao skup vrtloga materije koji se okreću oko nepokretnih zvezda ([8]). Na svojim ilustracijama podelio je prostor na konveksne oblasti, pri čemu svaka oblast sadrži materiju koja rotira oko jedne određene zvezde (Slika 1.3). Na taj način, Dekart se može smatrati najranijom pretečom savremenog pojma Voronojevih dijagrama.



Slika 1.3: Sunčev sistem prema Reneu Dekartu

Prvi koji su formalno predstavili pojam Voronojevih dijagrama bili su matematičari Karl Fridrih Gaus (Johann Carl Friedrich Gauss, 1840) i Gustav Ležen Dirihle (Johann Peter Gustav Lejeune Dirichlet, 1850), a kasnije i Georgij Voronoj (1908). Koristili su ga za proučavanje

kvadratnih formi ([8]). U njihovim radovima generatori su bile tačke sa celobrojnim koordinatama na celobrojnoj mreži \mathbb{Z}^2 , dok je bliskost merena standardnom Euklidskom distancom: upravo onako kako se Voronojevi dijagrami najčešće i definišu danas.

Georgij Voronoj je godinama radio na razvoju aritmetičke teorije kvadratnih formi i geometrije brojeva (videti [9]). U martu 1907. godine počeo je da piše rad o teoriji paraleloedra. Voronoj je primetio da je zadatak deljenja n -dimenzionalnog analitičkog prostora na konveksne kongruentne poliedre usko povezan sa aritmetičkom teorijom pozitivnih kvadratnih formi. Taj rad, koji je objavljen 1908–1909. godine, smatra se njegovim najvažnijim i najpoznatijim delom, u kojem je razvio teoriju podele prostora na delove koji se danas nazivaju Voronojevi dijagrami. Rad predstavlja temelj savremenog razumevanja prostorne dekompozicije i imao je veliki uticaj na brojna naučna područja.

1.2 Značaj k -means algoritma i Voronojevih dijagrama

K -means algoritam i Voronojevi dijagrami imaju značajnu ulogu u analizi i organizaciji podataka, posebno kada se želi sagledati struktura skupa podataka i međusobni odnosi elemenata unutar njega. Oni omogućavaju intuitivno razumevanje kako se prostor ili skup podataka može podeliti na delove koji na neki način logično pripadaju pojedinim tačkama, što olakšava dalje analiziranje i razumevanje odnosa među podacima.

Zanimljivo je da se Voronojevi dijagrami ne vezuju samo za apstraktne matematičke koncepte, već se njihovi obrasci mogu uočiti i u prirodi. Slični obrasci pojavljuju se na listovima biljaka, u belom luku, rasporedu pukotina u osušenoj zemlji, oklopu morske kornjače, kao i u šarama na krilima insekata ili koži žirafe (Slika 1.4).



Slika 1.4: Primeri Voronojevih dijagrama u prirodi: krila vilinog konjica ([10]), morska kornjača ([11]), pukotine na suvoj zemlji ([12]), list ([13]), koža žirafe ([14]), beli luk ([13])

1.3 Ciljevi i struktura rada

U ovom radu predstavljene su formalne definicije pojmova i metoda koje se koriste za rešavanje problema autora. Definisan je i sam problem koji će biti obrađen i analiziran. Pored toga, ukratko je prikazana praktična primena k -means algoritma i Voronojevih dijagrama u različitim oblastima, kao i pregled relevantne literature koja se bavi sličnim problemima. Nakon toga, biće prikazana autorova implementacija problema, uz detaljan opis i analizu dobijenih rezultata.

Cilj ovog rada je da podstakne razmišljanje o povezanosti klasterovanja i geometrijskih podela prostora.

2. TEORIJSKE OSNOVE I POSTAVKA PROBLEMA

2.1 Princip rada k -means algoritma

K -means algoritam deli podatke na k klastera (C_1, C_2, \dots, C_k), koji su predstavljeni svojim centrima (centroidima) ili srednjim vrednostima. Centroid svakog klastera izračunava se kao srednja vrednost svih instanci koje pripadaju tom klasteru (videti [4]).

K -means algoritam započinje sa početnim skupom centara klastera, koji se biraju nasumično ili prema nekom određenom pravilu. U svakoj iteraciji, svaka instanca podataka se dodeljuje najbližem centroidu prema Euklidskoj distanci između njih. Zatim se centri ponovo izračunavaju.

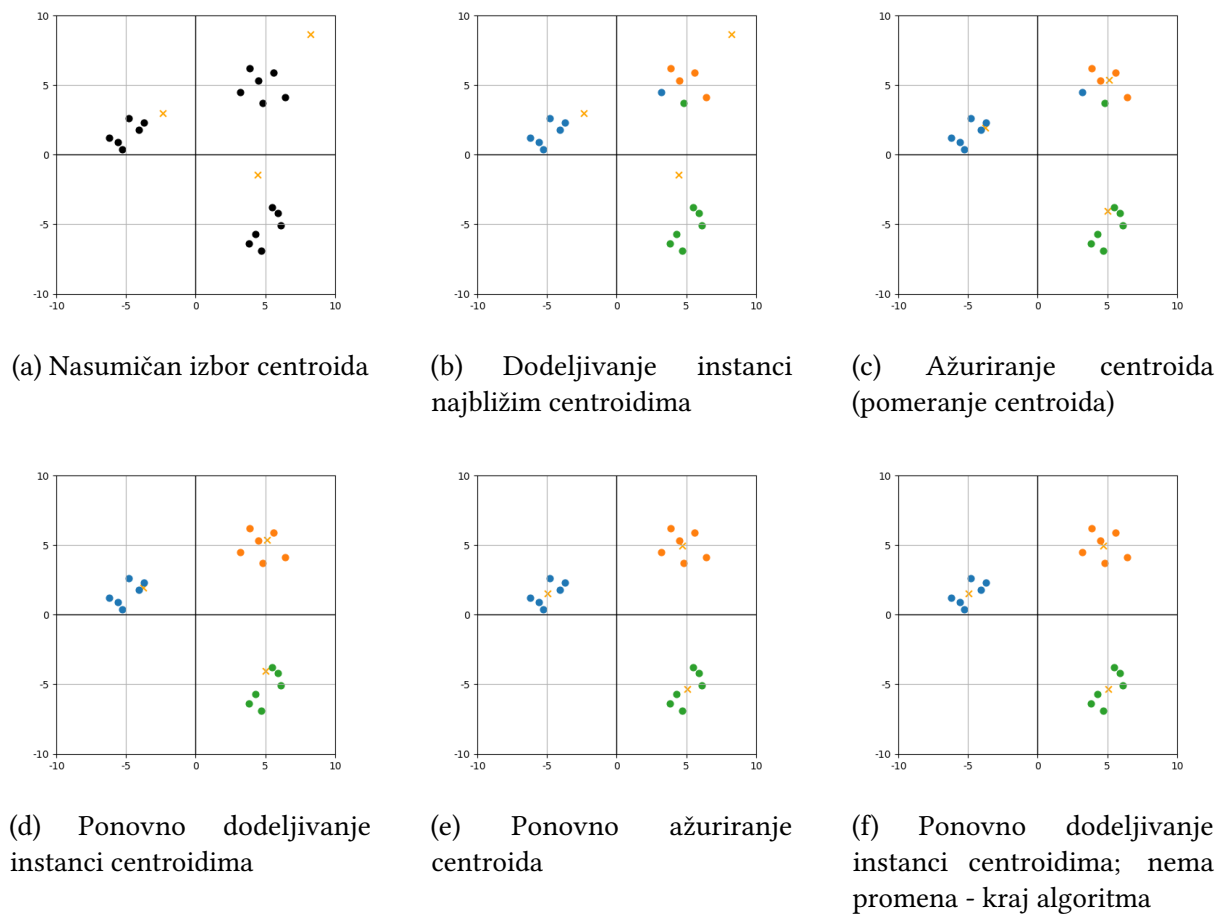
Centroid svakog klastera izračunava se kao srednja vrednost svih instanci koje pripadaju tom klasteru:

$$\mu_k = \frac{1}{N_k} \sum_{q=1}^{N_k} x_q$$

gde je μ_k srednja vrednost (centroid) klastera k , N_k broj instanci koje pripadaju klasteru k , a x_q instanca koja pripada klasteru k .

Mogući su različiti uslovi za prekid algoritma. Na primer, algoritam se zaustavlja kada daljim ponavljanjem postupka ne dolazi do promene u raspodeli instanci po klasterima, odnosno kada se centri klastera više ne menjaju. To znači da je algoritam dostigao stabilno rešenje. Takođe, algoritam se može zaustaviti ako se prekorači unapred definisani broj iteracija.

Na Slici 2.1 je prikazan jedan primer rada k -means algoritma.



Slika 2.1: Primer k -means algoritma

Pseudokod k -means algoritma:

Ulaz: S (skup instanci), k (broj klastera)

Izlaz: klasteri

- 1: Inicijalizuj k centroida.
- 2: **while** uslov zaustavljanja nije zadovoljen **do**
- 3: Dodeli instance najbližem centroidu.
- 4: Ažuriraj centroide na osnovu dodela.
- 5: **end while**

Pre nego što se definiše problem koji ovaj rad istražuje, potrebno je uvesti formalnu definiciju Voronojevih dijagrama.

2.2 Formalna definicija Voronojevih dijagrama

Neka je S konačan skup tačaka (generatora) u ravni. Za svaki par tačaka $p, q \in S$, $p \neq q$, definiše se dominantna oblast tačke p nad tačkom q kao skup svih tačaka x u ravni koje su bar jednako blizu p nego q :

$$\text{dom}(p, q) = \{x \in \mathbb{R}^2 \mid d(x, p) \leq d(x, q)\},$$

gde d označava neku metriku (videti [1]). Za Euklidsku metriku ta oblast je polu-ravan koja je ograničena simetralnom pravom između p i q .

Voronojeva ćelija za tačku $p \in S$ se dobija kao presek svih takvih dominantnih oblasti u odnosu na ostale tačke iz S :

$$V(p) = \bigcap_{q \in S \setminus \{p\}} \text{dom}(p, q),$$

tj. to je skup svih tačaka u ravni koje su bliže p nego bilo kojoj drugoj tački iz skupa S .

Voronojev dijagram $V(S)$ skupa S je skup svih Voronojevih ćelija (jedna Voronojeva ćelija po svakoj tački skupa S), koji zajedno formiraju particiju ravni u poligonalne ćelije (videti Sliku 1.2):

$$V(S) = \{V(p) \mid p \in S\}.$$

2.3 Postavka problema

Poznato je da k -means algoritam u svojoj standardnoj formi koristi Euklidsku distancu za dodelu tačaka najbližem centroidu. Sa druge strane, Voronojev dijagram predstavlja podelu ravni na regione (Voronojeve ćelije) u zavisnosti od izabrane distance i skupa generatora. Kada se kao generatori Voronojevog dijagrama uzmu centriodi dobijeni k -means algoritmom, prirodno se nameće pitanje koliko se podudara klasteraska pripadnost tačaka sa pripadnošću Voronojevim ćelijama, posebno pri korišćenju različitih metrika.

Problem koji se razmatra u ovom radu jeste empirijska analiza podudarnosti klastera dobijenih primenom k -means algoritma i Voronojevih ćelija konstruisanih na osnovu centroida klastera. Konkretno, u drugoj iteraciji k -means algoritma, dobijeni centriodi se posmatraju kao generatori Voronojevog dijagrama pri čemu se rastojanje tačaka od generatora računa

korišćenjem distance Minkovskog za različite vrednosti parametra p , $p \in \{1, 2, 2.5, 3, \infty\}$.

Cilj istraživanja je da se empirijski utvrdi za koju vrednost parametra p najveći broj tačaka istovremeno pripada istom klasteru i odgovarajućoj Voronojevoj ćeliji.

U nastavku se uvodi formalna definicija distance Minkovskog i razmatra se njen uticaj na Voronojeve dijagrame.

2.4 Distanca Minkovskog i Voronojevi dijagrami

Distanca Minkovskog predstavlja generalizaciju više poznatih metrika i definisana je kao ([15], [16]):

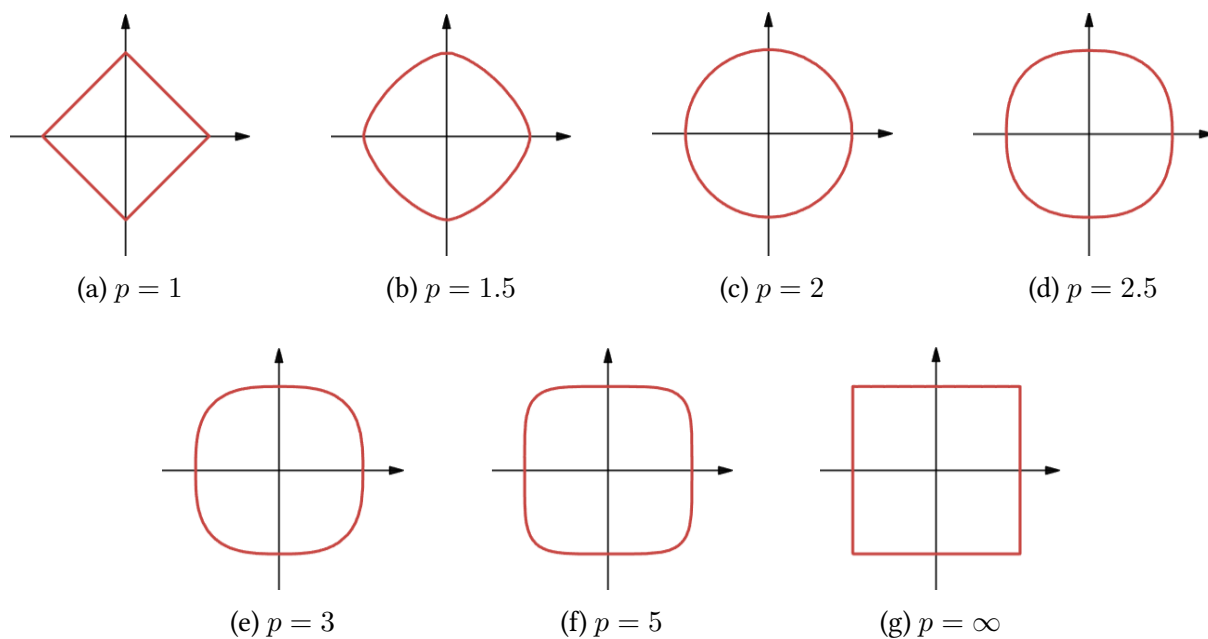
$$d_p(x, y) = \begin{cases} \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}}, & 1 \leq p < \infty, \\ \max_{1 \leq i \leq n} |x_i - y_i|, & p = \infty, \end{cases} \quad x, y \in \mathbb{R}^n, p \in \mathbb{R},$$

gde je $n \in \mathbb{N}$ dimenzija prostora, odnosno broj komponenti vektora x i y .

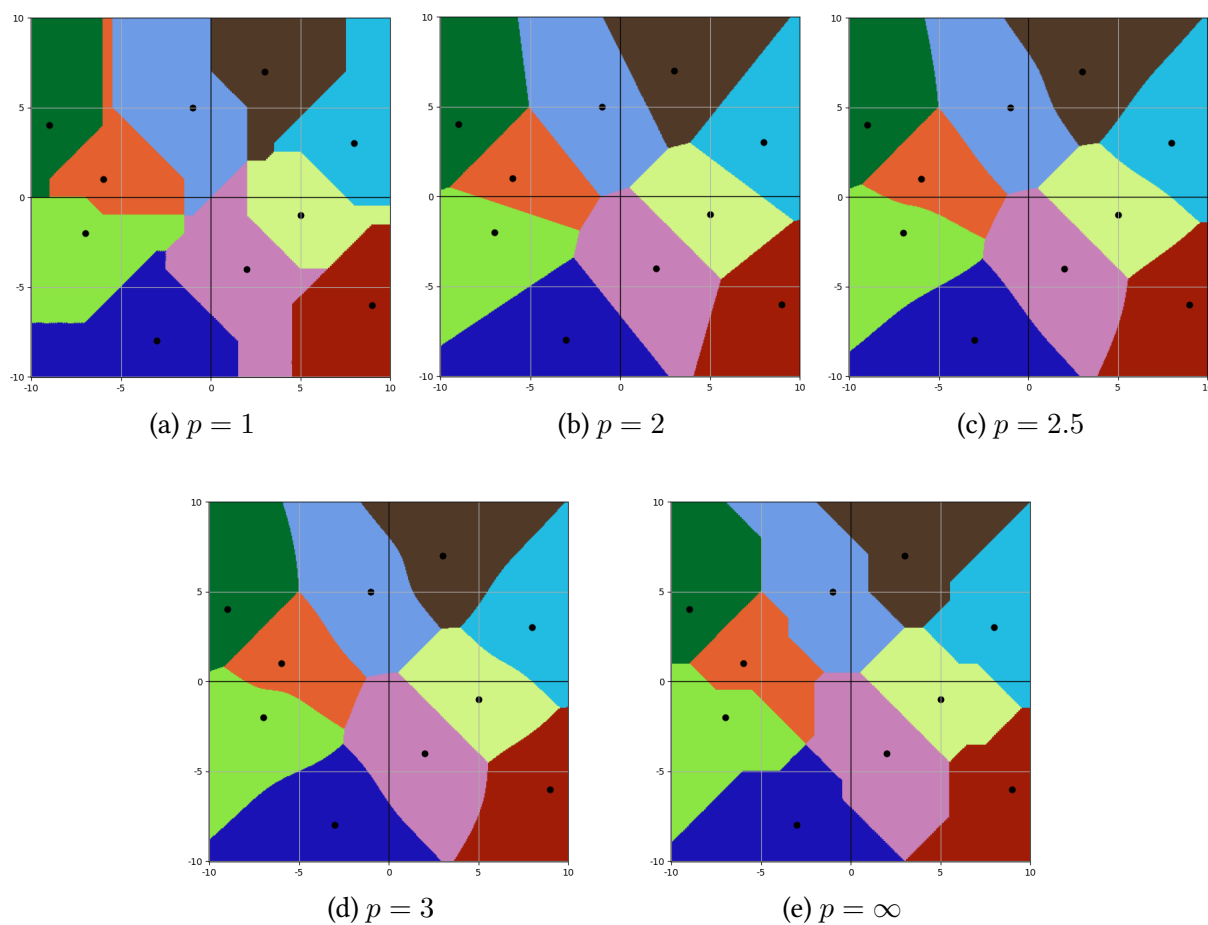
Za posebne vrednosti parametra p dobijaju se poznate distance:

- Ako je $p = 1$, onda je to Menhetn (Manhattan) distanca
- Ako je $p = 2$, onda je to Euklidska distanca
- Ako je $p = \infty$, onda je to Čebiševljeva (Chebyshev) distanca.

Promene vrednosti parametra p pri računanju distance u konstruisanju Voronojevog dijagrama utiču na oblik Voronojevih ćelija (videti Slike 2.2 i 2.3). Zbog toga određene tačke mogu promeniti pripadnost Voronojevoj ćeliji, iako njihova pripadnost klasteru prema k -means algoritmu ostaje nepromenjena.



Slika 2.2: Skup tačaka u dvodimenzionalnom prostoru ($n = 2$) koje su jednako udaljene od centra koordinatnog sistema za različite vrednosti parametra p



Slika 2.3: Voronojev dijagram u dvodimenzionalnom prostoru konstruisan korišćenjem distance Minkovskog za različite vrednosti parametra p

U sledećoj glavi rada će se analizirati uticaj primene različitih distanci Minkovskog na pripadnost tačaka Voronojevih ćelijama u dvodimenzionalnom prostoru.

2.5 Postojeća rešenja

U postojećoj literaturi često se razmatra veza između Voronojevih dijagrama i k -means algoritma. Poznato je da k -means algoritam, pri korišćenju Euklidske distance, implicitno generiše Voronojevu particiju ravni, gde centri klastera predstavljaju generatore Voronojevih ćelija. Ova veza je teorijski dobro utemeljena i detaljno analizirana u brojnim radovima (npr. [1]).

Značajan broj istraživanja fokusira se na unapređenje k -means algoritma, posebno u pogledu njegovih poznatih nedostataka, kao što su nasumičan izbor početnih centara klastera i potreba da se unapred zada broj klastera k . U jednom od predloženih pristupa ([17]), autori koriste Voronojeve dijagrame za određivanje početnih centara. Najpre se pronalaze Voronojeva temena, a zatim se oni iterativno spajaju dok se ne dobije odgovarajući broj tačaka koje se koriste kao početni centri za k -means. Problem unapred zadatog k rešavaju uvođenjem ograničenja poluprečnika Voronojevih krugova¹. Eksperimentalni rezultati ukazuju da ovakav pristup dovodi do stabilnijeg i efikasnijeg klasterovanja u poređenju sa standardnim k -means algoritmom.

Međutim, većina postojećih radova fokusira se isključivo na Euklidsku distancu. Iako postoje istraživanja koja se bave konstrukcijom Voronojevih dijagrama u različitim metrikama (npr. [15]), kao i radovi koji razmatraju k -means algoritam sa alternativnim metričkim funkcijama ([18], [19]), njihov međusobni odnos izvan okvira Euklidske metrike nedovoljno je istražen. Upravo odsustvo ovakvih istraživanja u literaturi predstavlja osnovnu motivaciju za ovaj rad.

2.6 Primena k -means algoritma i Voronojevih dijagrama

K -means algoritam ima široku primenu u raznim oblastima. Uspešno se koristi za segmentaciju kupaca ([20]), segmentaciju slika u računarskom vidu ([21]), koristi se u astronomiji ([22]), agrokulturi ([23]), detekciji obrazaca u podacima (data mining) [24],

¹Voronojevi krugovi su najveći mogući krugovi čiji su centri Voronojeva temena i koji ne sadrže nijedan generator u svojoj unutrašnjosti.

biomedicini ([25]), prepoznavanju govora ([26]), pa čak i za analizu zagađenja vazduha ([27]).

Voronoevi dijagrami takođe imaju praktičnu i teorijsku primenu u velikom broju oblasti, pre svega u nauci, tehnologiji i likovnoj umetnosti. Primenuju se u računarstvu i računarskoj grafici, geometriji, epidemiologiji, biologiji, geografskim informacionim sistemima (GIS), geofizici i meteorologiji ([2], [28]). Iako se pojavljuju pod različitim imenima, Voronoevi dijagrami se javljaju u fizici kondenzovane materije i u proučavanju Lijeviskih grupa ([29]). Glavna vrednost Voronoevih dijagrama je u tome što omogućavaju analizu prostora i raspodelu resursa na osnovu blizine generatora.

Jedan od istorijskih primera primene Voronoevih dijagrama je analiza epidemije kolere u Londonu 1854. godine. Lekar Džon Snou (John Snow) je koristio koncept sličan Voronoevim ćelijama da bi identifikovao izvor infekcije – pumpu na Broad Street-u. On je mapirao pumpe i okolne oblasti i pokazao da su smrtni slučajevi kolere bili najčešći u delu grada najbližem toj pumpi (videti [30]).

3. IMPLEMENTACIJA I ANALIZA PROBLEMA

U ovoj glavi rada prikazaće se autorova implementacija k -means algoritma i konstrukcija Voronojevih dijagrama na osnovu centroida dobijenih u drugoj iteraciji algoritma. Takođe, biće analizirani rezultati dobijeni eksperimentalnim putem.

3.1 Implementacija algoritma

Implementacija je realizovana u Python programskom jeziku. Radi lakše vizualizacije rađeno je sa podacima u dvodimenzionalnom prostoru. Nakon unetog broja klastera i tačaka za klasterovanje, početni centroidi u k -means algoritmu inicijalizuju se nasumično. Sve faze algoritma su grafički prikazane kako bi se omogućila jasna analiza ponašanja klastera i odgovarajućih Voronojevih ćelija. Takođe, važno je napomenuti da je prikaz ograničen na koordinatni sistem dimenzija 20×20 , sa vrednostima od -10 do 10 na obe ose (x i y), pri čemu se podrazumeva da se i ulazne tačke nalaze unutar navedenog opsega, radi jednostavnije i preglednije analize rezultata.

Delovi implementacije koji se odnose na unos podataka i vizuelni prikaz faza algoritma neće biti prikazani u ovom radu. Razlog za to je fokusiranje na suštinske delove implementacije, kao i izbegavanje nepotrebnog opterećenja rada prikazom kompletnog koda. Najvažniji koraci koji utiču na formiranje klastera i odgovarajućih Voronojevih ćelija detaljnije će biti objašnjeni uz prikaz delova implementacije.

Dodeljivanje tačaka najbližem centroidu

U ovoj fazi k -means algoritma svaka tačka se dodeljuje klasteru čiji je centroid najbliži. Funkcija `dodeli_tacke_najblizem_centroidu` prolazi kroz sve tačke i za svaku računa Euklidsku distancu do svih centroida, nakon čega se tačka pridružuje klasteru sa minimalnom distancom. Funkcija vraća promenljivu `klasteri` koja predstavlja rečnik (dictionary) u kome svaki ključ odgovara jednom centroidu, dok vrednost predstavlja skup tačaka koje su mu dodeljene u tekućoj iteraciji algoritma.

```
def dodeli_tacke_najblizem_centroidu():  
    klasteri = {k: [] for k in centroidi}
```

```
for tacka in tacke:
    x, y = tacka
    min_dist = float('inf')
    min_c = (float('inf'), float('inf'))

    for c in klasteri.keys():
        dist = math.sqrt((c[0] - tacka[0])**2 + (c[1] - tacka[1])**2)

        if dist < min_dist:
            min_dist = dist
            min_c = c

    klasteri[min_c].append(tacka)

return klasteri
```

Pored funkcije za dodeljivanje tačaka najbližem centroidu, implementirana je i funkcija za vizualizaciju klastera oboji_klastere. Ona iscrtava tačke svakog klastera različitom bojom koristeći kolornu mapu tab10 koja obezbeđuje do deset različitih boja.

Generisanje Voronojevog dijagrama

Funkcija za generisanje Voronojevog dijagrama poziva se u drugoj iteraciji k -means algoritma. Generatori postaju centriodi trenutne iteracije k -means algoritma, nakon čega se na osnovu njih prostor deli na Voronojeve ćelije. Za računanje rastojanja se koristi distanca Minkovskog kao što je i definisana u prethodnoj glavi rada, pri čemu je dimenzija prostora $n = 2$.

Funkcija iterira kroz niz vrednosti $p = \{1, 2, 2.5, 3, \infty\}$, tako da se u svakoj iteraciji crta Voronojev dijagram nad trenutnom vrednošću parametra p .

Za vizualizaciju je korišćena dvodimenzionalna mreža (grid), gde svaka tačka mreže predstavlja jednu koordinatu u prostoru. Promenljive width i height su dimenzije mreže, dok je (i, j) trenutna pozicija na mreži. Prolaskom kroz mrežu, za svaku tačku se računa njeno rastojanje do svakog generatora koristeći trenutnu vrednost p . Svaka tačka mreže ima svoju listu rastojanja koja sadrži rastojanje te tačke do svakog generatora. Najbliži generator se utvrđuje tako što se uzme najmanje rastojanje iz liste. Zatim se toj trenutnoj tački dodeljuje boja svog najbližeg generatora. Time su sve tačke koje pripadaju istom generatoru obojene u

istu boju.

U nastavku funkcije za generisanje Voronojevog dijagrama, na istoj mreži, prikazuju se i tačke klastera iz trenutne iteracije k -means algoritma, koje su obojene u odgovarajuće boje, što omogućava lakšu analizu.

```
def izgenerisi_voronojev_dijagram():
    p_vrednosti = [1, 2, 2.5, 3, np.inf]
    generatori = np.array(centroidi)

    #distanca Minkovskog za trenutno p
    def rastojanje(A, B, p):
        if p == np.inf:
            return max(abs(A[0]-B[0]), abs(A[1]-B[1]))
        return ((abs(A[0]-B[0])**p) + (abs(A[1]-B[1])**p) )**(1/p)

    colors = np.random.rand(len(generatori), 3)

    for p in p_vrednosti:
        print(f"\n\nVoronojev dijagram nad centroidima za p = {p}\n")

        #normalizovanje tacaka nam je potrebno zbog grafickog prikaza;
        #nije kljucno za razumevanje glavne logike
        normalizovani_generatori = np.array(
            [normalizuj_tacku(g) for g in generatori])

        #prolazimo kroz mrezu, da bismo videli sta cemo da bojimo
        for i in range(height):
            for j in range(width):
                #normalizovana pozicija (i,j) na gridu
                pos = np.array([j/width, i/height])

                rastojanja = []
                #racunamo rastojanje trenutne tacke (i,j) (pos)
                #od generatora (za svaki generator)
                for g in normalizovani_generatori:
                    d = rastojanje(g, pos, p)
                    rastojanja.append(d)
```

```
najblizi_generator = np.argmin(rastojanja)
grid[i, j] = colors[najblizi_generator]

#Ovde se nalazi nastavak funkcije koji na mrezi prikazuje i tacke
#klastera iz trenutne iteracije k-means algoritma
```

Ažuriranje centroida

Funkcija `azuriraj_centroide` prolazi kroz sve klasterne i za svaki klaster računa srednju vrednost po aritmetičkoj sredini svih tačaka u klasteru. Ta srednja vrednost postaje nova pozicija trenutnog centroida. Ažurirani centroidi su smešteni kao ključevi u rečnik `novi_klasteri` koji sadrži iste vrednosti kao `klasteri`, ali sa novim pozicijama centroida. Fleg `postoji_pomeren_centroid` proverava da li je centroid promenio poziciju i koristi se za uslov zaustavljanja k -means algoritma.

```
def azuriraj_centroide():
    postoji_pomeren_centroid = False
    novi_klasteri = {}

    for i, (k, v) in enumerate(klasteri.items()):
        if not v:
            novi_klasteri[(k[0], k[1])] = v
            continue

        x_suma = 0
        y_suma = 0
        for x, y in v: #za svaku tacku (x,y) iz v
            x_suma += x
            y_suma += y

        srednja_x = x_suma / len(v)
        srednja_y = y_suma / len(v)

        novi_klasteri[(srednja_x, srednja_y)] = v

        if srednja_x != k[0] or srednja_y != k[1]:
            postoji_pomeren_centroid = True
```

```
return novi_klasteri, postoji_pomeran_centroid
```

Za vizualizaciju ažuriranih centroida implementirana je funkcija `nacrtaj_pomerene_centroide`.

Glavna petlja k -means algoritma

Glavna petlja k -means algoritma izvršava se iterativno sve dok postoji pomeranje barem jednog centroida. U svakoj iteraciji vrši se dodeljivanje tačaka najbližem centroidu i ažuriranje centroida, sve to uz vizualizaciju. U drugoj iteraciji konstruiše se Voronojev dijagram na osnovu trenutnih centroida. Algoritam se zaustavlja kada više ne dolazi do promene položaja centroida.

```
iteracija = 0
postoji_pomeran_centroid = True
while postoji_pomeran_centroid:

    iteracija += 1
    print(f"\n\nITERACIJA {iteracija}")

    print("\n\n4. Dodeljivanje tacaka najblizem centroidu \
(bojenje klastera)\n")
    klasteri = dodeli_tacke_najblizem_centroidu()
    oboji_klastere()

    if iteracija == 2:
        izgenerisi_voronojev_dijagram()

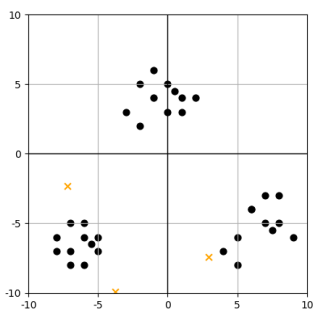
    print("\n\n5. Azuriranje centroida\n")
    novi_klasteri, postoji_pomeran_centroid = azuriraj_centroide()

    if postoji_pomeran_centroid == False:
        print("\nNijedan centroid se nije pomerio. Kraj algoritma.")
        continue

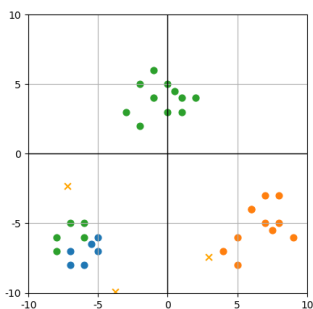
    centroidi = list(novi_klasteri.keys())
    normalizovani_centroidi = nacrtaj_pomerene_centroide()
```

Primer vizuelnog prikaza izvršavanja programa

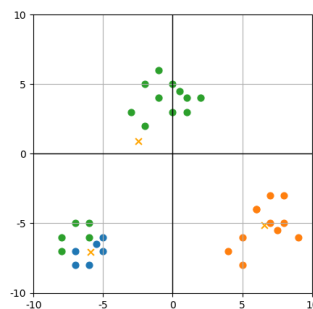
Prikazan je jedan primer vizuelnog prikaza koji se dobija pokretanjem programa, pri čemu je korisnik uneo broj klastera $k = 3$ i skup od 30 tačaka. Centroidi su označeni znakom \times i istaknuti su narandžastom bojom.



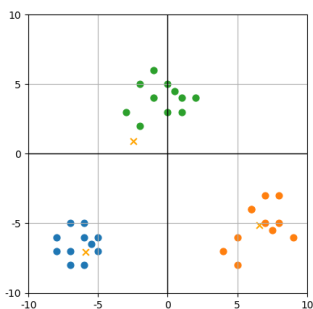
(a) Nasumičan izbor centroida



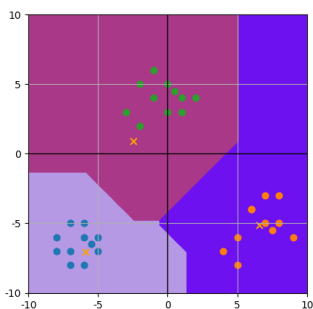
(b) Dodeljivanje tačaka centroidima (prva iteracija)



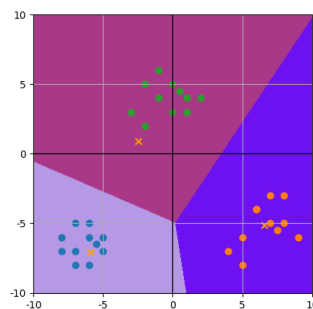
(c) Ažuriranje centroida (prva iteracija)



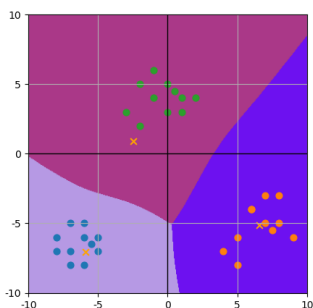
(d) Dodeljivanje tačaka centroidima (druga iteracija)



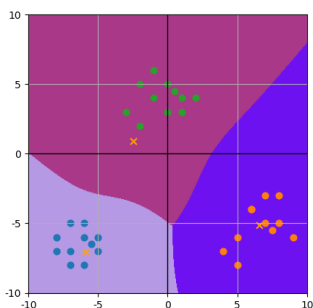
(e) Generisanje Voronojevog dijagrama za $p = 1$



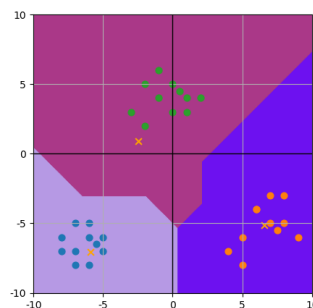
(f) Generisanje Voronojevog dijagrama za $p = 2$



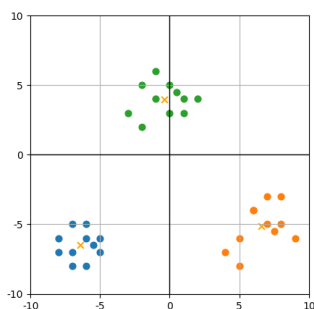
(g) Generisanje Voronojevog dijagrama za $p = 2.5$



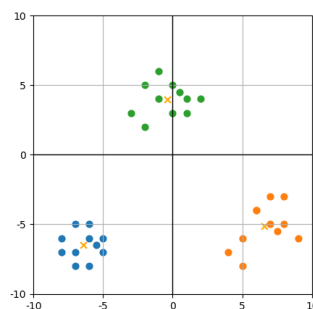
(h) Generisanje Voronojevog dijagrama za $p = 3$



(i) Generisanje Voronojevog dijagrama za $p = \infty$



(j) Ažuriranje centroida (druga iteracija)



(k) Dodeljivanje tačaka centroidima (treća iteracija); nakon ove iteracije ne dolazi do promene; kraj algoritma

Slika 3.2: Primer vizuelnog prikaza izvršavanja programa

U nastavku rada neće biti prikazani svi grafički rezultati, već samo oni najvažniji za dalju analizu.

3.2 Eksperimentalna analiza

Uvodne napomene i termini

Pre početka analize uvode se osnovni termini i napomene neophodne za dalji tok rada.

Voronoevi dijagrami se u ovom radu konstruišu u drugoj iteraciji k -means algoritma, nakon dodele tačaka centroidima, a pre ažuriranja (pomeranja) centroida.

Druga iteracija je izabrana jer centriodi u tom koraku više nisu isključivo rezultat nasumične inicijalizacije, već su već delimično prilagođeni tačkama.

Važno je uočiti da k -means algoritam radi sa diskretnim skupom tačaka, dok Voronoevi dijagrami dele celu ravan, odnosno kontinualni prostor. Zbog toga se podudarnost klastera i Voronoevih ćelija ne može posmatrati nad celom ravni, već se posmatra isključivo u odnosu na skup tačaka koje se klasteruju.

Neka je c_1 centroid, C klaster čiji je centroid c_1 , a $V(c_1)$ Voronoeva ćelija čiji je generator c_1 . Za fiksiranu vrednost parametra p , klaster C i Voronoeva ćelija $V(c_1)$ su podudarni ako i samo ako sve tačke klastera C pripadaju Voronoevoj ćeliji $V(c_1)$. Dakle, klaster i Voronoeva ćelija nisu podudarni ako postoji bar jedna tačka klastera koja ne pripada toj Voronoevoj ćeliji.

Tokom analize će se koristiti termin *prelaz*, odnosno *promena pripadnosti tačke*, koji označava situaciju u kojoj tačka klastera ne pripada Voronojevoj ćeliji koja je generisana nad njegovim centroidom. Drugim rečima, kaže se da je tačka prešla u drugu Voronojevu ćeliju.

Takođe, koristiće se termin *potpuna podudarnost*, koji označava situaciju u kojoj ne postoji nijedan prelaz, odnosno kada su svaki klaster i njemu odgovarajuća Voronojeva ćelija, generisana nad njegovim centroidom, međusobno podudarni.

Važno je primetiti da na izgled Voronojevog dijagrama utiču nasumično inicijalizovani centriodi na početku algoritma. To znači da za isti ulaz, odnosno isti broj klastera i isti skup tačaka, rezultat može biti drugačiji. Razlog za to je što izgled Voronojevog dijagrama zavisi od broja i položaja centroida u drugoj iteraciji, odnosno generatora nad kojima je konstruisan. Sa druge strane, same tačke klastera nemaju uticaj na izgled Voronojevog dijagrama. Njegov izgled zavisi isključivo od centroida u drugoj iteraciji i vrednosti parametra p .

Cilj analize i slučaj potpune podudarnosti

Cilj analize je utvrditi za koju vrednost parametra p će najveći broj tačaka istovremeno pripadati istom klasteru i Voronojevoj ćeliji koja je generisana nad njegovim centroidom, odnosno da se odredi vrednost parametra p za koju će najmanji broj tačaka promeniti svoju pripadnost.

Kao što je ranije navedeno, k -means algoritam u svakoj iteraciji implicitno konstruiše Voronojev dijagram nad svojim centroidima, jer za dodelu tačaka centroidima koristi Euklidsku distancu. Distanca Minkovskog za $p = 2$ je upravo Euklidska distanca. To znači da su, u slučaju kada se za konstrukciju Voronojevih ćelija koristi distanca Minkovskog sa parametrom $p = 2$, klaster i odgovarajuća Voronojeva ćelija uvek podudarni. Drugim rečima, za $p = 2$ uvek će sve tačke pripadati istom klasteru i Voronojevoj ćeliji koja je generisana nad njegovim centroidom, odnosno u ovom slučaju neće postojati nijedan prelaz. Ovo je primer potpune podudarnosti.

Za ostale vrednosti parametra p to se ne može očigledno utvrditi, pa će se dalje analizirati kroz primere.

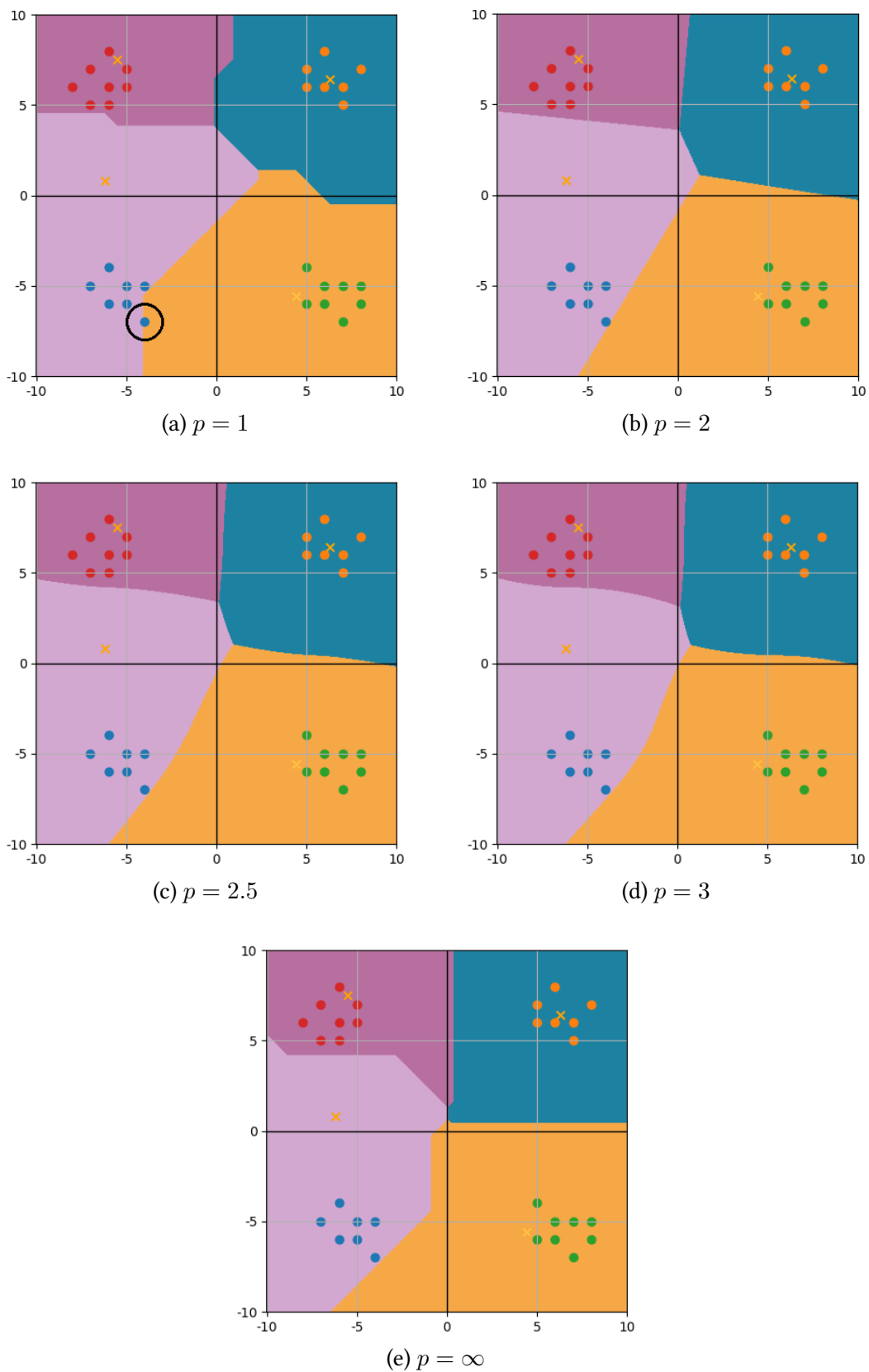
Takođe, važno je napomenuti da promena vrednosti parametra p ne utiče na pripadnost tačaka klasterima.

Uticaj parametra p na oblik Voronojevih ćelija

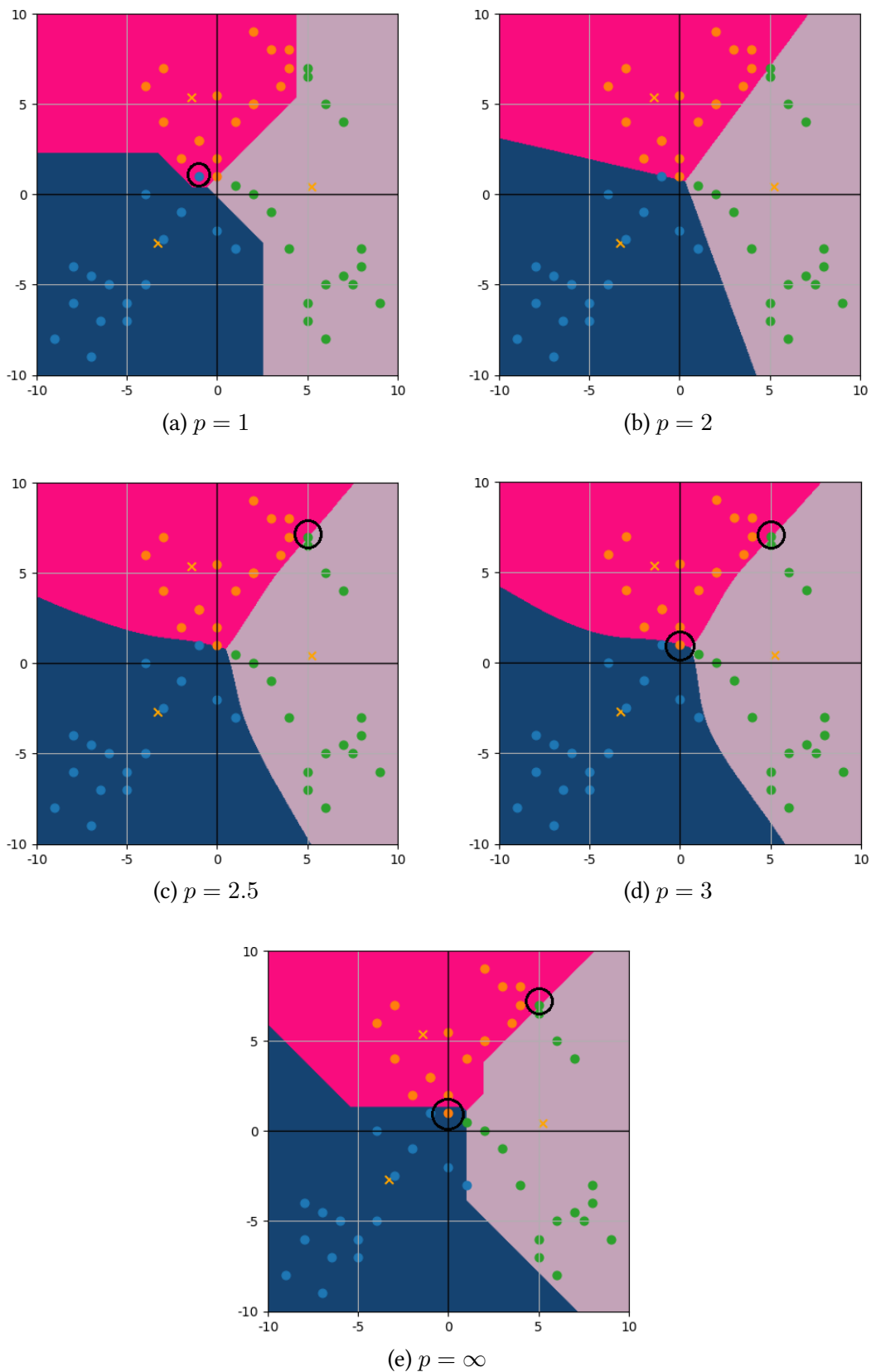
Postavlja se pitanje u kojim situacijama tačka može da promeni pripadnost. Da bi se dobio odgovor na to pitanje, potrebno je posmatrati kako se ponašaju Voronojeve ćelije prilikom promene vrednosti parametra p .

Promena vrednosti parametra p utiče na oblik Voronojevih ivica. Na Slici 2.2 iz prethodne glave rada primećuje se da je Voronojeva ivica za $p = 2.5$ i $p = 3$ kriva linija, a za $p = 1$ i $p = \infty$ izlomljena linija. Sa porastom vrednosti parametra p ($p > 2$), Voronojeve ivice postaju sve izraženije zakrivljene i u graničnom slučaju $p = \infty$ prelaze u izlomljene linije.

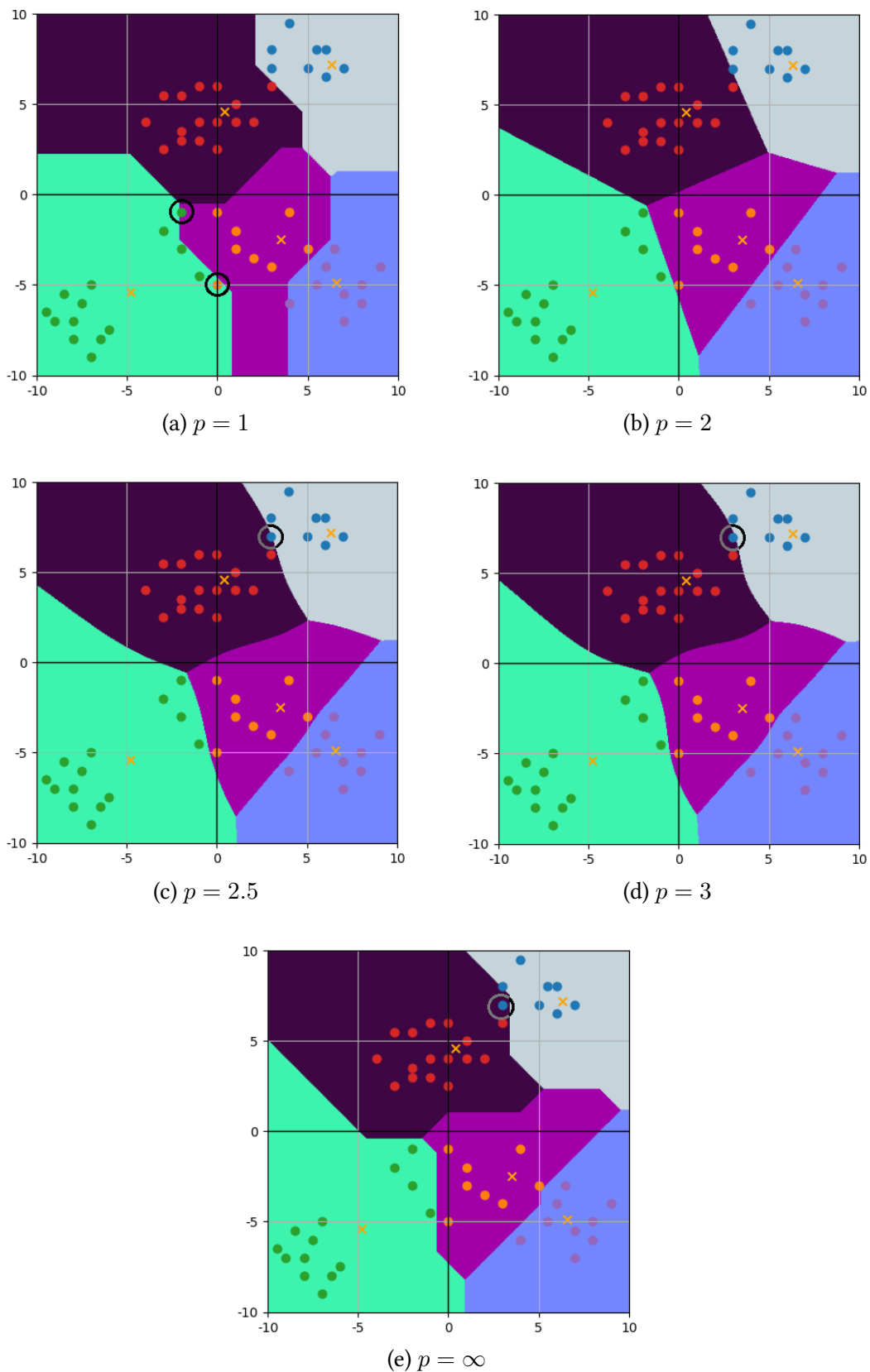
Na Slikama 3.3, 3.4 i 3.5 prikazani su primeri za različite ulaze kod kojih dolazi do prelaza tačaka u susednu Voronojevu ćeliju. Tačke koje prelaze u susednu Voronojevu ćeliju su obeležene kružnicama. Posmatranjem oblika Voronojevih ivica na tim primerima, može se uočiti da se promena pripadnosti tačaka najčešće dešava kod tačaka koje se nalaze u blizini Voronojevih ivica.



Slika 3.3: Generisan Voronojev dijagram u drugoj iteraciji za $k = 4$ i 30 tačaka



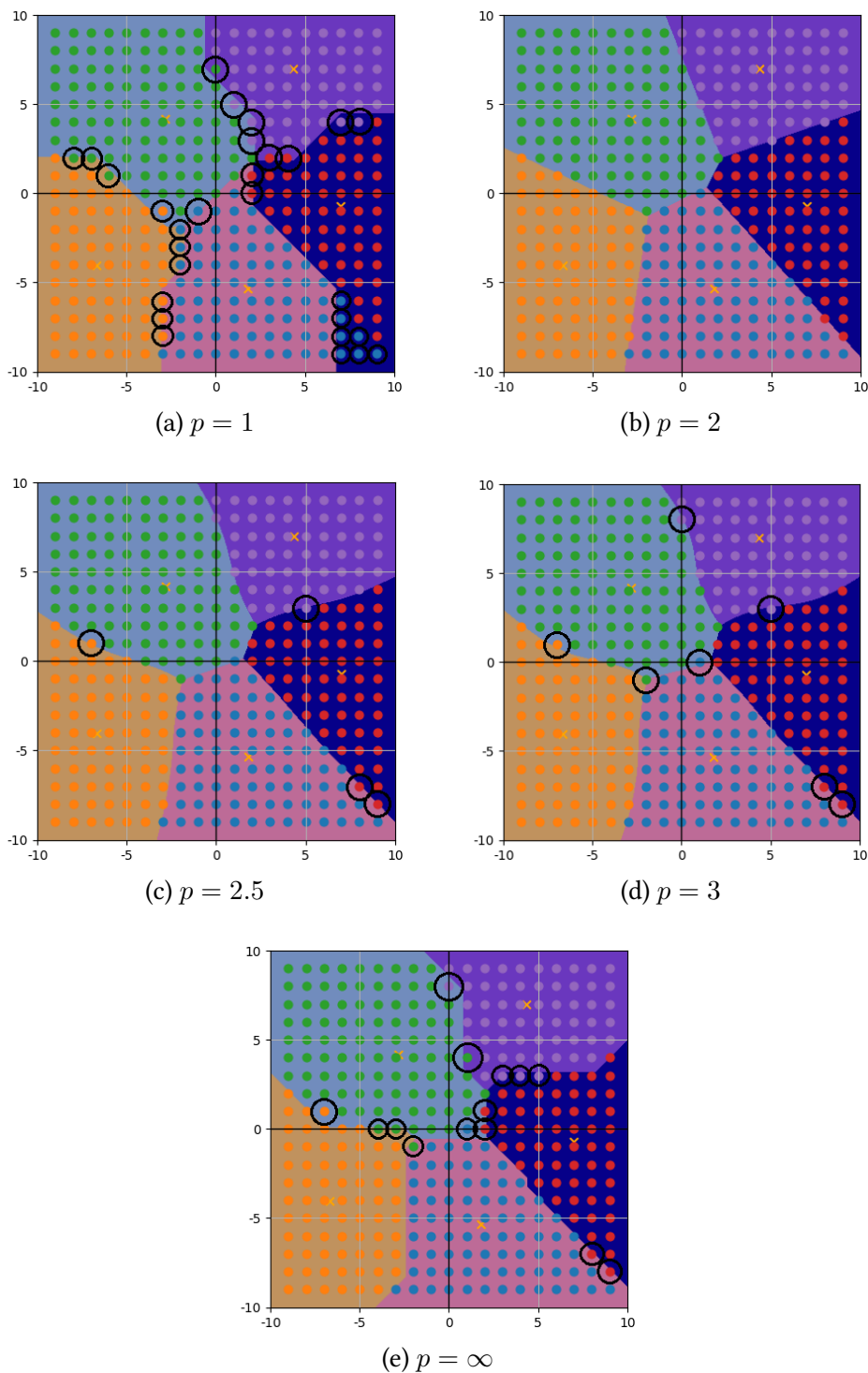
Slika 3.4: Generisan Voronojev dijagram u drugoj iteraciji za $k = 3$ i 48 tačaka



Slika 3.5: Generisan Voronojev dijagram u drugoj iteraciji za $k = 5$ i 55 tačaka

Primer skupa tačaka sa celobrojnim koordinatama

Kao primer posmatra se skup svih tačaka sa celobrojnim koordinatama na mreži 20×20 . Iako ovakav primer nema direktnu primenu u realnim problemima klasterovanja, on je koristan za ilustraciju jer jasno pokazuje ponašanje tačaka u blizini Voronojevih ivica.



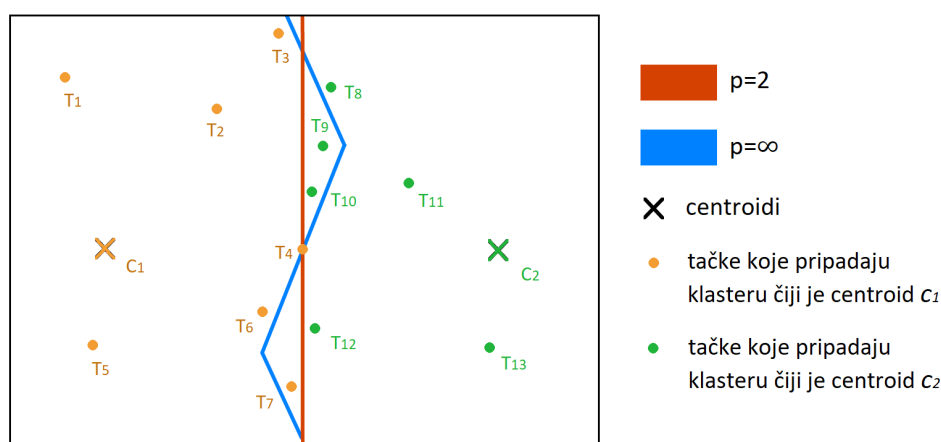
Slika 3.6: Voronojev dijagram za ulaz $k = 5$ i skup tačaka koji čine sve tačke sa celobrojnim koordinatama na mreži 20×20

Može se uočiti da veći broj tačaka u ovom primeru menja pripadnost za $p = 1$ i $p = \infty$ nego za vrednosti $p = 2.5$ i $p = 3$.

Kako vrednost parametra p raste i udaljava se od vrednosti 2, a linija sve više krivi, ona postaje sve različitija u odnosu na liniju dobijenu za $p = 2$. Kako je za $p = 2$ uvek prisutna potpuna podudarnost klastera i odgovarajućih Voronojevih ćelija, može se steći utisak da sa porastom vrednosti parametra p raste i verovatnoća da će tačka koja se nalazi u blizini Voronojeve ivice promeniti pripadnost. Međutim, ova pretpostavka ne mora biti tačna.

Analiza promena pripadnosti tačaka

U nastavku se razmatra ponašanje tačaka kada je Voronojeva ivica najrazličitija od Voronojeve ivice dobijene za $p = 2$, odnosno slučaj kada je $p = \infty$. Na Slici 3.7 prikazane su Voronojeve ivice dobijene za $p = 2$ i $p = \infty$. Prikazan je i skup tačaka na različitim pozicijama koje su obojene bojom najbližeg centroida. Za slučaj $p = 2$ već je utvrđeno da nijedna tačka ne menja pripadnost, pa će se u nastavku ispitivati ponašanje tačaka za slučaj $p = \infty$.

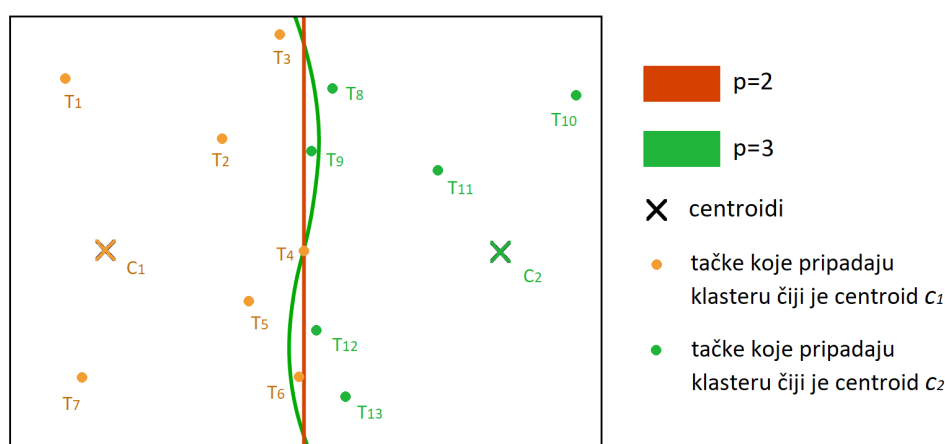


Slika 3.7: Primer Voronojevih ivica između dva centroida (generatora) dobijenih za $p = 2$ i $p = \infty$

Neka je posmatrana tačka T_1 sa slike koja je dovoljno udaljena od Voronojevih ivica. Ona pripada klasteru čiji je centroid C_1 i Voronojevoj ćeliji čiji je generator takođe C_1 . Dakle, nije promenila pripadnost. Da ne dolazi do promene pripadnosti primećuje se i kod tačaka T_2 , T_5 , T_{11} i T_{13} , koje su takođe dovoljno udaljene od Voronojevih ivica. Neka je sada posmatrana tačka u blizini Voronojevih ivica, npr. T_6 . Ona pripada klasteru čiji je centroid C_1 i Voronojevoj ćeliji čiji je generator takođe C_1 . Ni ona ne menja pripadnost. Isto važi i za tačke T_3 , T_4 , T_8 i

T_{12} , koje se takođe nalaze u blizini Voronojevih ivica. Dalje se analizira tačka T_7 . Ona takođe pripada klasteru čiji je centroid C_1 , ali ona sada, za $p = \infty$, pripada Voronojevoj ćeliji čiji je generator C_2 . Dakle, ova tačka menja svoju pripadnost. Isto se može primetiti i za tačke T_9 i T_{10} .

Dalje se razmatra slučaj kada je $p = 3$, odnosno kada je Voronojeva ivica sličnija Voronojevoj ivici dobijenoj za $p = 2$ nego što je Voronojeva ivica dobijena za $p = \infty$. Na Slici 3.8 su prikazane Voronojeve ivice dobijene za $p = 2$ i $p = 3$. Ispituje se ponašanja tačaka za slučaj $p = 3$.



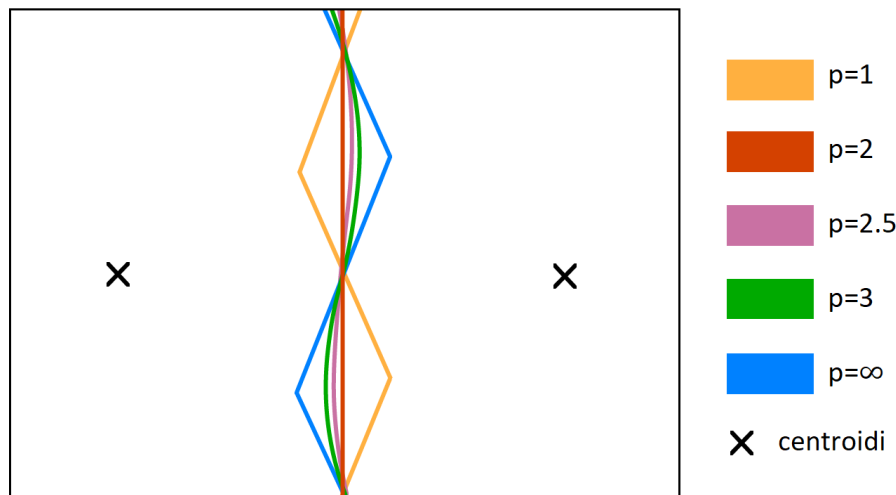
Slika 3.8: Primer Voronojevih ivica između dva centroida (generatora) dobijenih za $p = 2$ i $p = 3$

Tačke koje su dovoljno udaljene od Voronojevih ivica, T_1 , T_2 , T_7 , T_{10} i T_{11} , pripadaju klasterima i Voronojevim ćelijama čiji su centroid i generator isti. Kod tih tačaka ne dolazi do promene pripadnosti (slično kao i u slučaju kada je $p = \infty$). Tačke T_3 , T_8 , T_4 , T_5 , T_{12} i T_{13} takođe pripadaju klasterima i Voronojevim ćelijama čiji su centroid i generator isti. One takođe ne menjaju svoju pripadnost. Promena pripadnosti se može videti kod tačaka T_9 (koja pripada centroidu C_2 i Voronojevoj ćeliji koja je generisana nad centroidom C_1) i T_6 (koja pripada centroidu C_1 i Voronojevoj ćeliji koja je generisana nad centroidom C_2).

Može se primetiti da su tačke koje menjaju pripadnost kada je $p = \infty$ one koje se nalaze u oblasti između Voronojevih ivica dobijenih za $p = 2$ i $p = \infty$. Slično, kada je $p = 3$, primećuje se da su tačke koje menjaju pripadnost one koje se nalaze u oblasti između Voronojevih ivica dobijenih za $p = 2$ i $p = 3$. Analogno bi važno i za $p = 2.5$ i $p = 1$.

Posmatranjem Voronojevih ivica na Slici 3.9 i prethodnih primera, dolazi se do zaključka

da za svako p , $p \neq 2$, sve tačke koje se nalaze između Voronojeve ivice dobijene za to p i Voronojeve ivice dobijene za $p = 2$ menjaju svoju pripadnost. Takođe, sve tačke koje se ne nalaze u tim oblastima neće promeniti pripadnost.



Slika 3.9: Primer Voronojevih ivica između dva centroida (generatora) za različite vrednosti parametra p

Važno je primetiti da se javlja potpuna podudarnost ukoliko su sve tačke van navedenih oblasti. To znači da $p = 2$ ne mora biti jedina vrednost parametra p za koju se javlja potpuna podudarnost (videti Sliku 3.2).

Analiza tačaka u oblastima između Voronojevih ivica

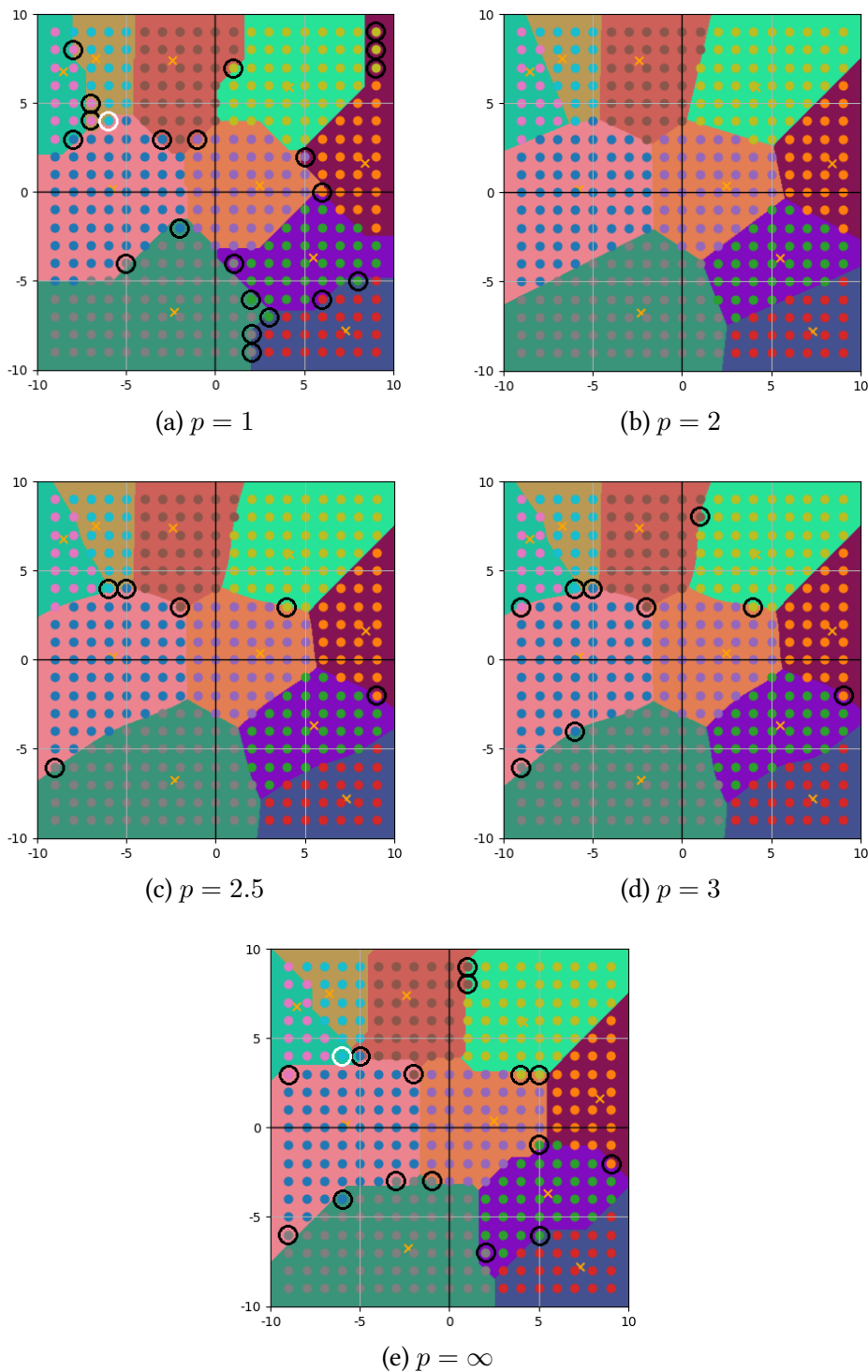
Pošto je utvrđeno da tačke van navedenih oblasti nikada ne menjaju pripadnost, u nastavku će se analizirati samo tačke koje se nalaze u tim oblastima.

Radi lakše analize uvodi se pojam $oblast_p$, koji označava oblast između Voronojeve ivice dobijene za $p = 2$ i Voronojeve ivice dobijene za posmatranu vrednost p .

Posmatranjem Slike 3.9 može se primetiti da važi: $oblast_{2.5} \subset oblast_3$ i $oblast_3 \subset oblast_\infty$, pa samim tim sledi da je $oblast_{2.5} \subset oblast_\infty$.

Na osnovu toga zaključuje se da tačke koje se nalaze u $oblast_{2.5}$ menjaju pripadnost za $p = 2.5$, $p = 3$ i $p = \infty$. Takođe, tačke koje pripadaju $oblast_3$, ali ne i $oblast_{2.5}$, menjaju pripadnost za $p = 3$ i $p = \infty$, ali ne i za $p = 2.5$. Analogno, tačke koje se nalaze u $oblast_\infty \setminus (oblast_{2.5} \cup oblast_3)$ menjaju pripadnost samo za $p = \infty$.

Na primeru 3.10 (kao i na ostalim primerima u ovom radu) može se videti da sve tačke koje su promenile pripadnost za $p = 2.5$ menjaju pripadnost i za $p = 3$, a sve koje su promenile pripadnost za $p = 3$ menjaju i za $p = \infty$.



Slika 3.10: Voronojev dijagram za ulaz $k = 10$ i skup tačaka koji čine sve tačke sa celobrojnim koordinatama na mreži 20×20

Na osnovu navedenog zaključuje se da je, od razmatranih vrednosti parametra

$p \in \{2.5, 3, \infty\}$, vrednost $p = 2.5$ ona za koju najmanji broj tačaka menja pripadnost. Takođe, za $p = 3$ manji broj tačaka menja pripadnost nego za $p = \infty$.

Za vrednost $p = 1$ ponašanje tačaka se ne može jasno utvrditi. Iako na Slici 3.9 deluje da $oblast_1$ ne sadrži nijednu drugu oblast, kao i da tačke koje menjaju pripadnost za ostale vrednosti parametra p nikad neće promeniti pripadnost za $p = 1$, i obrnuto, to ne mora uvek biti slučaj. Na primer, postoje tačke koje u isto vreme pripadaju i $oblast_1$ i $oblast_\infty$ (na Slici 3.10 takve tačke su obeležene belim kružnicama). Još jedan razlog zbog koga se ponašanje tačaka za $p = 1$ ne može jasno utvrditi je što postoje situacije gde postoji više prelaza za $p = 1$ nego za $p = \infty$ (npr. u primeru 3.6), a takođe i situacije u kojima je broj prelaza za $p = 1$ jednak broju prelaza za $p = 2.5$ (npr. u primeru 3.4).

4. ZAKLJUČAK

U ovom radu analizirana je veza između klastera k -means algoritma i Voronojevih ćelija konstruisanih nad njegovim centroidima. Ispitivan je uticaj različitih vrednosti parametra p distance Minkovskog na njihovu međusobnu podudarnost. Poseban fokus je bio na posmatranju promena pripadnosti tačaka, odnosno prelaza tačaka u susednu ćeliju, prilikom promene vrednosti parametra p . Cilj istraživanja je bio empirijski ispitati za koju vrednost parametra p će najviše tačaka pripadati istom klasteru i Voronojevoj ćeliji koja je generisana nad njegovim centroidom.

Analizom je potvrđeno da u slučaju kada je $p = 2$, odnosno kada se koristi Euklidska distanca, dolazi do potpune podudarnosti, odnosno sve tačke će pripadati istom klasteru i odgovarajućoj Voronojevoj ćeliji. Ovo je očekivan rezultat jer je to direktna posledica načina rada k -means algoritma.

Za ostale vrednosti parametra p broj promena pripadnosti tačaka zavisi samo od njihovih konkretnih položaja. Primećeno je da će sve tačke pripadati istom klasteru i odgovarajućoj Voronojevoj ćeliji, odnosno, doći će do potpune podudarnosti, ako se sve tačke nalaze van oblasti između Voronojeve ivice dobijene za posmatranu vrednost parametra p i Voronojeve ivice dobijene za $p = 2$.

U ostalim slučajevima, ako postoje tačke u navedenim oblastima, pokazano je da je $p = 2.5$ vrednost parametra p za koju će najviše tačaka pripadati istom klasteru i odgovarajućoj Voronojevoj ćeliji, u odnosu na $p = 3$ i $p = \infty$. Takođe, za $p = 3$ veći broj tačaka će pripadati istom klasteru i odgovarajućoj Voronojevoj ćeliji nego za $p = \infty$. Za slučaj $p = 1$ se nije moglo jasno utvrditi.

Dobijeni rezultati u ovom radu doprinose boljem razumevanju odnosa između klasterovanja i geometrijskih podela prostora. Oni mogu predstavljati osnovu za dalja istraživanja, posebno za ispitivanje veze između drugih algoritama klasterovanja i Voronojevih dijagrama pri upotrebi različitih metrika. Takođe, otvaraju mogućnost za razmišljanje o proširenju na višedimenzionalne prostore i realne podatke.

BIBLIOGRAFIJA

1. Aurenhammer, F. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM computing surveys (CSUR)* **23**, 345–405 (1991).
2. Wikipedia contributors. *Voronoi diagram* — *Wikipedia, The Free Encyclopedia* https://en.wikipedia.org/w/index.php?title=Voronoi_diagram&oldid=1321826822. [Online; accessed 21-November-2025]. 2025.
3. De Berg, M., Cheong, O., Van Kreveld, M. & Overmars, M. *Computational geometry: algorithms and applications* (Springer, 2008).
4. Rokach, L. & Maimon, O. in *Data mining and knowledge discovery handbook* 321–352 (Springer, 2005).
5. Sud, K., Erdogmus, P. & Kadry, S. *Introduction to data science and machine learning* (BoD–Books on Demand, 2020).
6. Jain, A. K. Data clustering: 50 years beyond K-means. *Pattern recognition letters* **31**, 651–666 (2010).
7. MacQueen, J. *Multivariate observations in Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability* **1** (1967), 281–297.
8. Aurenhammer, F., Klein, R. & Lee, D.-T. *Voronoi diagrams and Delaunay triangulations* (World Scientific Publishing Company, 2013).
9. Syta, H. & van de Weygaert, R. Life and Times of Georgy Voronoi. *arXiv preprint arXiv:0912.3269* (2009).
10. Ito, J. *Dragonfly wings [Fotografija]* <https://www.flickr.com/photos/35034362831@N01/698898343/>. [Online; accessed 23-January-2026]. 2007.
11. World Wildlife Fund. *Sea Turtle* <https://www.worldwildlife.org/species/sea-turtle>. [Online; accessed 23-January-2026]. 2026.
12. Sousa, V. M. *INTERACTIVE TRANSMEDIA vs. VORONOI DIAGRAM EXPRESSIONS [Conference paper]* https://www.researchgate.net/figure/Dry-earth-Voronoi-diagram_fig4_316167811. [Online; accessed 23-January-2026]. 2017.

13. Bellelli, F. S. *The fascinating world of Voronoi diagrams* <https://fbellelli.com/posts/2021-07-08-the-fascinating-world-of-voronoi-diagrams/>. [Online; accessed 23-January-2026]. 2021.
14. CamillaC. *Camilla: Voronoi Diagram Knowledge Analysis: In nature* <https://teamhoudini.wordpress.com/2018/12/06/camilla-voronoi-diagram-knowledge-analysis-in-nature/>. [Online; accessed 23-January-2026]. 2018.
15. Nair, V. G. Metric-Driven Voronoi Diagrams: A Comprehensive Mathematical Framework. *Computation* **13**. ISSN: 2079-3197. <https://www.mdpi.com/2079-3197/13/9/212> (2025).
16. Zezula, P., Amato, G., Dohnal, V. & Batko, M. *Similarity search the metric space approach* (Springer, 2006).
17. REDDY, D. & PAWAR, P. Improved K-means Clustering Algorithm for Biological Data using Voronoi Diagram. *International Journal of Computational and Experimental Science and Engineering* **2**, 9–18. <https://ijcesen.com/index.php/ijcesen/article/view/223> (Mar. 2016).
18. Oti, E. U., Olusola, M. O., Godwin, O.-O. & Nwankwo, C. H. New k-means clustering method using minkowski's distance as its metric. *Information Technology* **4**, 28–41 (2020).
19. Singh, A., Yadav, A. & Rana, A. K-means with Three different Distance Metrics. *International Journal of Computer Applications* **67** (2013).
20. Omol, E., Onyangor, D., Mburu, L. & Abuonji, P. Application of K-means clustering for customer segmentation in grocery stores in Kenya. *International Journal of Science, Technology & Management* **5**, 192–200 (2024).
21. Burney, S. A. & Tariq, H. K-means cluster analysis for image segmentation. *International Journal of Computer Applications* **96** (2014).
22. Ordovás-Pascual, I. & Almeida, J. S. A fast version of the k-means classification algorithm for astronomical applications. *Astronomy & Astrophysics* **565**, A53 (2014).
23. Shedthi, B. S., Shetty, S. & Siddappa, M. *Implementation and comparison of K-means and fuzzy C-means algorithms for agricultural data in 2017 international conference on inventive communication and computational technologies (ICICCT)* (2017), 105–108.

24. Annas, M. & Wahab, S. N. Data mining methods: K-means clustering algorithms. *International Journal of Cyber and IT Service Management* **3**, 40–47 (2023).
25. Kalyani, C., Ramudu, K. & Reddy, G. R. A review on optimized K-means and FCM clustering techniques for biomedical image segmentation using level set formulation. *Biomedical Research* **29**, 3660–3668 (2018).
26. Li, X.-g., Yao, M.-f. & Huang, W.-t. *Speech recognition based on k-means clustering and neural network ensembles* in *2011 Seventh International Conference on Natural Computation* **2** (2011), 614–617.
27. Govender, P. & Sivakumar, V. Application of k-means and hierarchical clustering techniques for analysis of air pollution: A review (1980–2019). *Atmospheric Pollution Research* **11**, 40–56. ISSN: 1309-1042. <https://www.sciencedirect.com/science/article/pii/S1309104219304556> (2020).
28. Pokojski, W. & Pokojaska, P. Voronoi diagrams–inventor, method, applications. *Polish Cartographical Review* **50** (2018).
29. Wikipedia contributors. *Georgy Voronoy* — *Wikipedia, The Free Encyclopedia* https://en.wikipedia.org/w/index.php?title=Georgy_Voronoy&oldid=1310201647. [Online; accessed 21-November-2025]. 2025.
30. Wikipedia contributors. *1854 Broad Street cholera outbreak* — *Wikipedia, The Free Encyclopedia* https://en.wikipedia.org/w/index.php?title=1854_Broad_Street_cholera_outbreak&oldid=1318330470. [Online; accessed 21-November-2025]. 2025.