

Elektrotehnički fakultet, Univerzitet u Beogradu

# **Prepoznavanje oblika – domaći zadaci**

## **Prvi domaći zadatak – opis problema**

Potrebno je isprojektovati sistem za prepoznavanje rukom pisanih cifara za bazu podataka koja sarži cifre 0, 6 i 8. Sistem treba da se zasniva na testiranju hipoteza, a rezultate treba prikazati u vidu matrice konfuzije.

## **Domaći zadatak – opis rešenja**

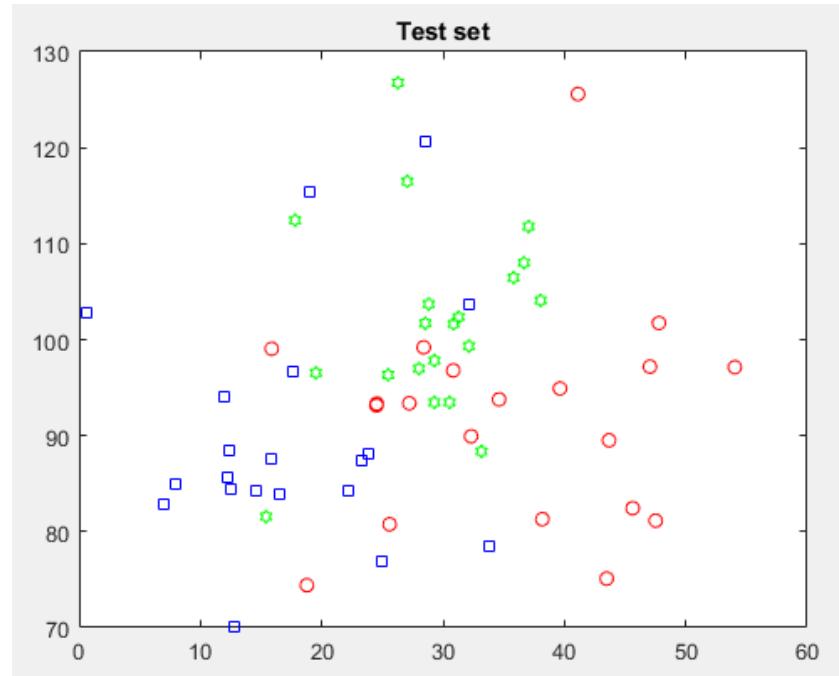
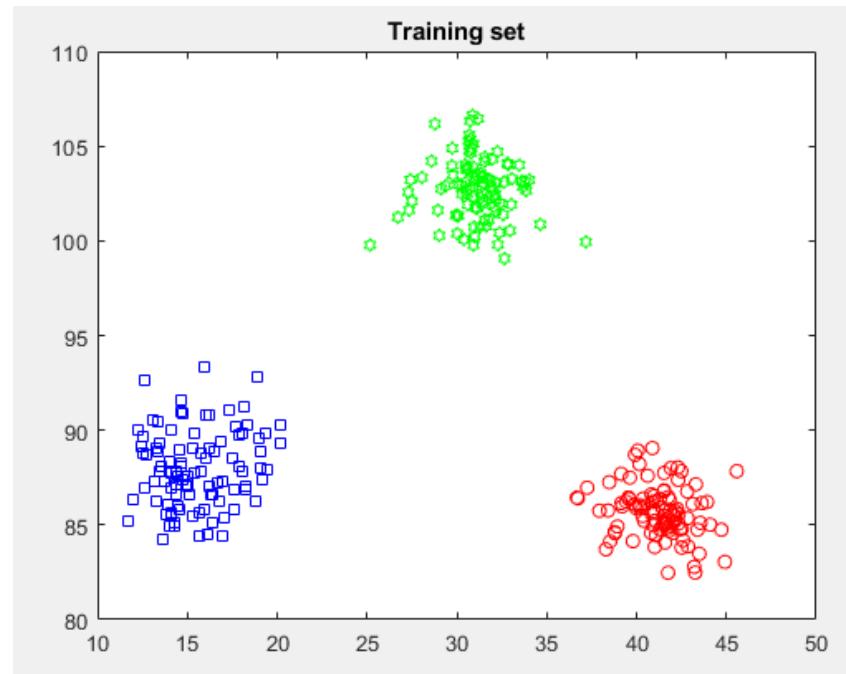
Rešenje prati sledeći algoritam:

1. Učitavaju se slike iz baze podataka i obrađuju. Obrada slike podrazumeva uklanjanje linija koje su posledica skeniranja i belih margini. Data slika se svodi na veličinu 5x5 i svi pikseli se uzimaju kao potencijalna obeležja.
2. Podaci se dele na testirajući i obučavajući skup (20 slika za testirajući i 100 slika za obučavajući skup). Na obučavajućem skupu se određuju vektori matematičkih očekivanja i kovarijacione matrice za svaku klasu.
3. Primjenjuje se redukcija dimenzija na bazi mere rasipanja tako da se merni vektori svedu na dvodimenzione.
5. Nakon redukcije dimenzija cifre iz testirajućeg skupa se klasificuju na bazi Bajesovog pravila odlučivanja minimalne greške pod pretpostavkom da odbirci imaju normalnu raspodelu.

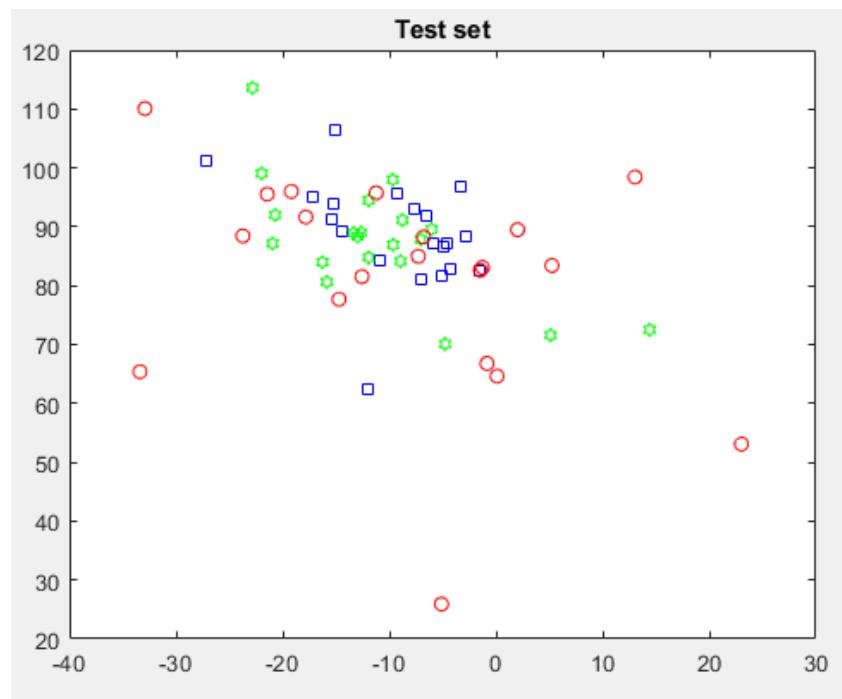
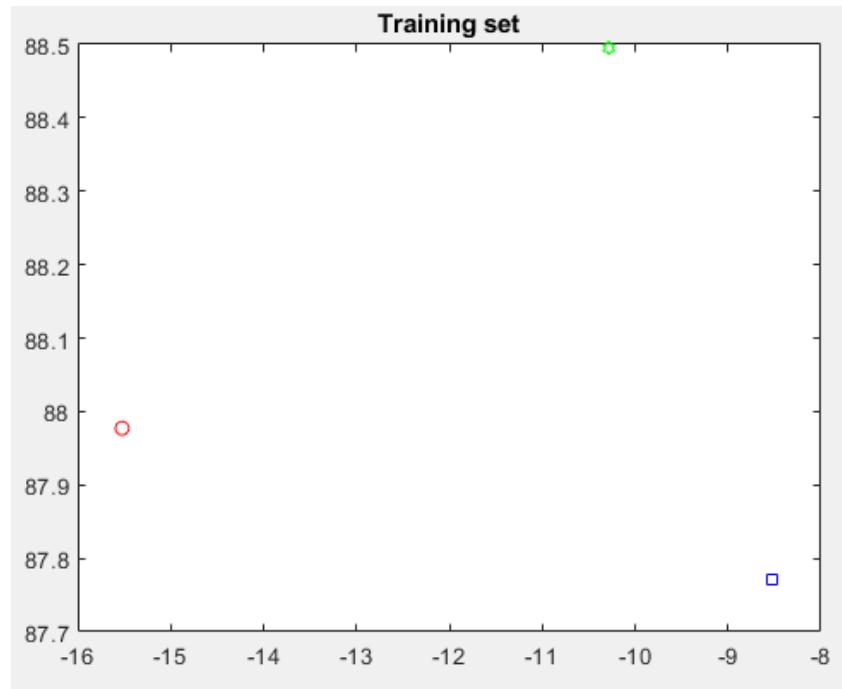
## **Obrada slika i biranje inicijalnih obeležja**

Obrada slike najpre radi uklanjanje tamnih linija koje se javljaju kao posledica skeniranja. Ovo je urađeno odsecanjem 5% piksela sa svake ivice. Nakon toga sledi binarizacija slike. Potom se kreće od krajeva (gornji, donji, levi i desni) i proverava se srednja vrednost boje datog reda, odnosno kolone. Ukoliko je srednja vrednost tog reda, odnosno kolone, i onog ispred pretežno bela, smatra se da je to deo margine i proverava se dalje. Ukoliko to nije slučaj, to je znak da se tu nalazi cifra i da tu treba preseći sliku. Ova obrada je jako dobro izdvojila cifre.

Konačno, slika se smanjuje tako da bude dimenzija 5x5 i formira se vektor obeležja od svih 25 piksela. Kako je obučavajući skup veličine 100 ne sme se uzeti preveliki broj obeležja jer će to dovesti do *overfitting-a*. Kada model ima jako veliki broj obeležja u odnosu na obučavajući skup, tada dolazi do toga da model prosto “nauči napamet” dati skup. Konkretno, u ovom slučaju, sa velikim brojem obeležja moguće je naći transformaciju koja savršeno prevodi obučavajući skup u dve dimenzije, ali je potpuno neprimenjiva na testirajući skup. Dati problem je ilustrovan na slikama 1.1.1. i 1.1.2. gde se vidi kako je moguće izabrati transformaciju tako da se merenja iz različitih klasa gotovo savršeno razdvajaju u obučavajućem skupu, ali i da model ne važi kada se ista transformacija primeni na testirajući skup. Na slici 1.1.2. se vidi problem koji se javlja kada je broj obeležja znatno veći od veličine obučavajućeg skupa (situacija u kojoj je “naučen napamet” obučavajući skup).



*Slika 1.1.1. Problem overfitting-a kada se uzme previše obeležja (slika dimenzija 15x15)*



Slika 1.1.2. Problem overfitting-a kada se uzme previše obeležja (slika dimenzija 20x20)

## Redukcija dimenzija na bazi mere rasipanja

Ova vrsta redukcije dimenzija je od posebnog značaja kada se radi klasifikacija oblika jer vodi računa o tome da se maksimizuje separabilnost različitih klasa. Prepostavimo da je potrebno izvršiti klasifikaciju u okviru  $L$  različitih klasa, pri čemu je svakoj klasi pridružen vektor matematičkog očekivanja  $M_i$  i kovarijaciona matrica  $\Sigma_i$ . Definišimo tada matricu unutar-klasnog rasejanja na sledeći način:

$$S_W = \sum_{i=1}^L P_i \Sigma_i$$

Definišimo takođe združeni vektor matematičkog očekivanja  $M_0$  kao:

$$M_0 = \sum_{i=1}^L P_i M_i$$

Konačno, definišemo i matricu međuklasnog rasejanja kao:

$$S_B = \sum_{i=1}^L P_i (M_i - M_0)(M_i - M_0)^T$$

Još se uvodi i takozvana miksovana matrica rasejanja:

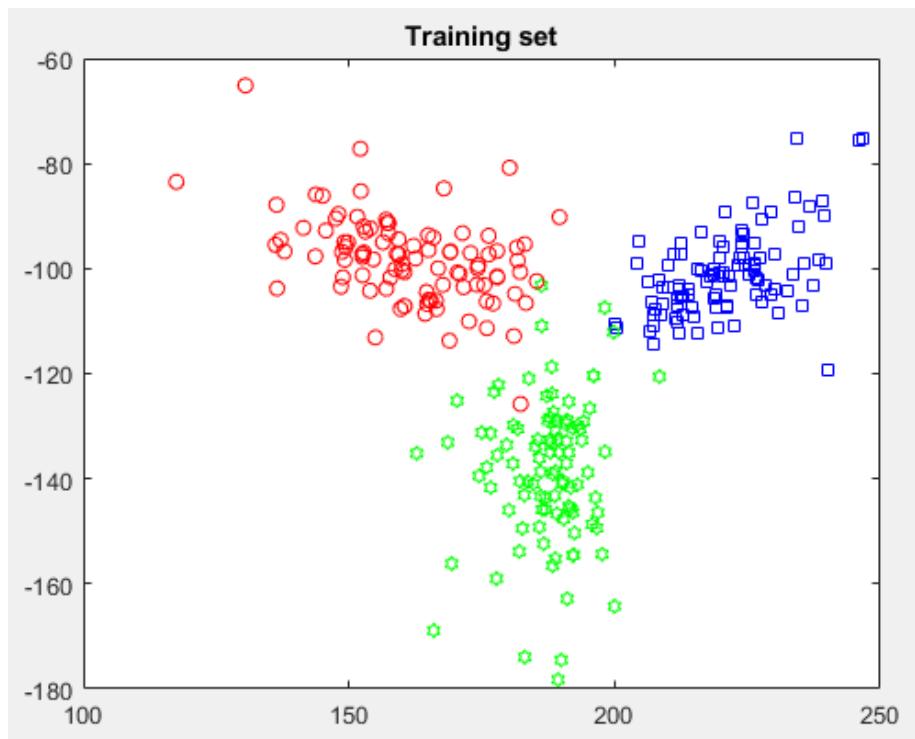
$$S_M = S_W + S_B$$

Tada se postavlja sledeći problem: Kako odabratи transformacionu matricу  $A$ , tako da slučajni vektor  $Y = A^T X$  minimizira određeni kriterijum. U konkretnoj implementaciji je korišćen kriterijum  $J = \text{tr}(S_W^{-1} S_B)$ .

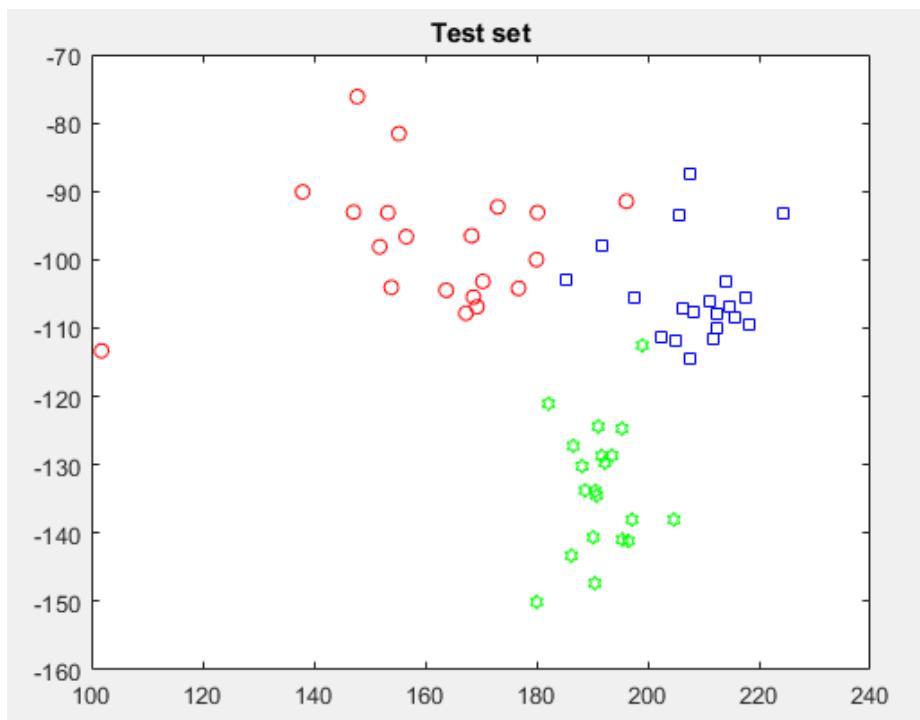
Matrica  $A$  koja minimizuje ovaj kriterijum je matrica sopstvenih vektora matrice  $S_W^{-1} S_B$ . Sortiranjem sopstvenih vrednosti u nerastući niz, biranjem prvih  $m$  vrednosti i zatim formiranjem matrice  $A$  od sopstvenih vektora koji odgovaraju najvećim  $m$  sopstvenih vrednosti može se izvršiti redukcija dimenzija na  $m$  dimenzija tako da se maksimizuje separabilnost između klasa.

U konkretnom slučaju, pokazuje se da su dve dimenzije sasvim dovoljne za klasifikaciju odbiraka. Na slici 1.2.1. je prikazan obučavajući skup koji se dobije kao posledica redukcije dimenzija, a na slici 1.2.2. testirajući skup kada se na njega primeni ista transformacija.

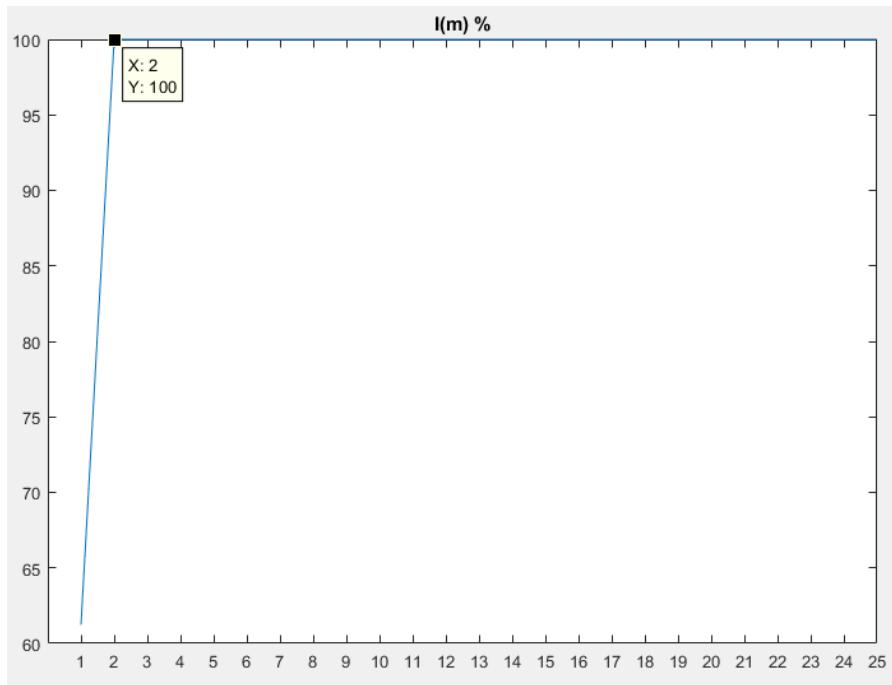
Na slici 1.2.3. je prikazan indeks informativnosti koji pokazuje konzervaciju količine informacija koja je rezultat čuvanja prvih  $m$  koordinata. Sa ovog grafika se jasno vidi da biranjem prve dve koordinate mi čuvamo sve informacije.



Slika 1.2.1. Obučavajući skup nakon redukcije dimenzija



Slika 1.2.2. Test skup nakon redukcije dimenzija



Slika 1.2.3. Indeks informativnosti

Indeks informativnosti se definiše kao:

$$I(m) = \frac{\sum_{i=1}^m \lambda_i}{\sum_{i=1}^n \lambda_i}$$

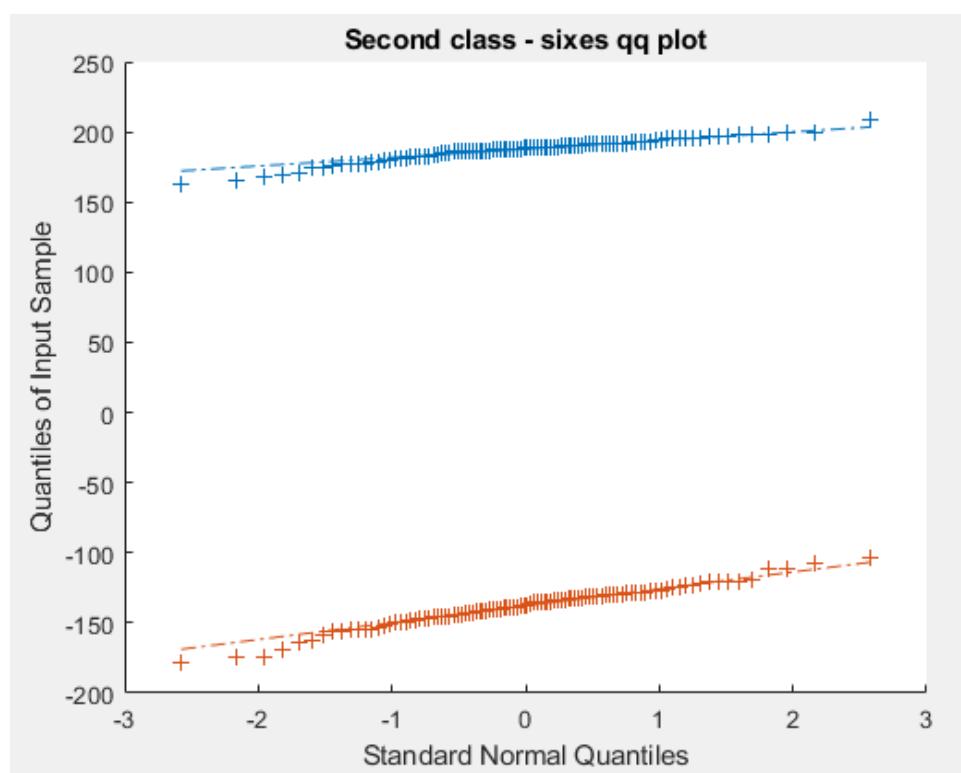
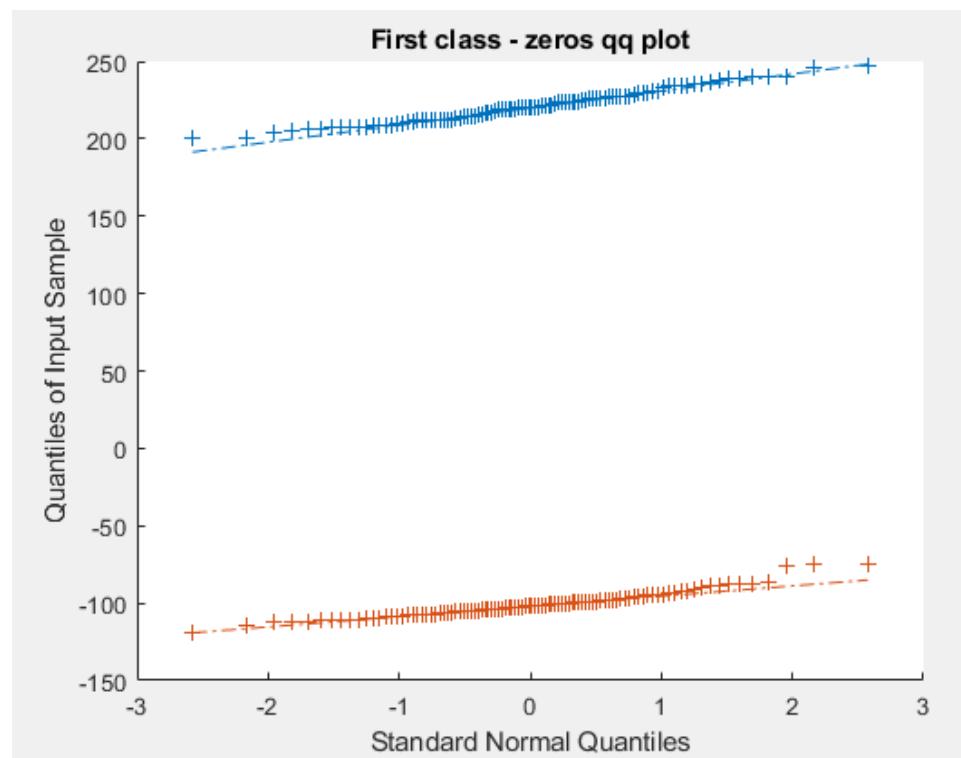
gde je  $n$  početni broj dimenzija, a podrazumeva se sledeći niz sopstvenih vrednosti:

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$$

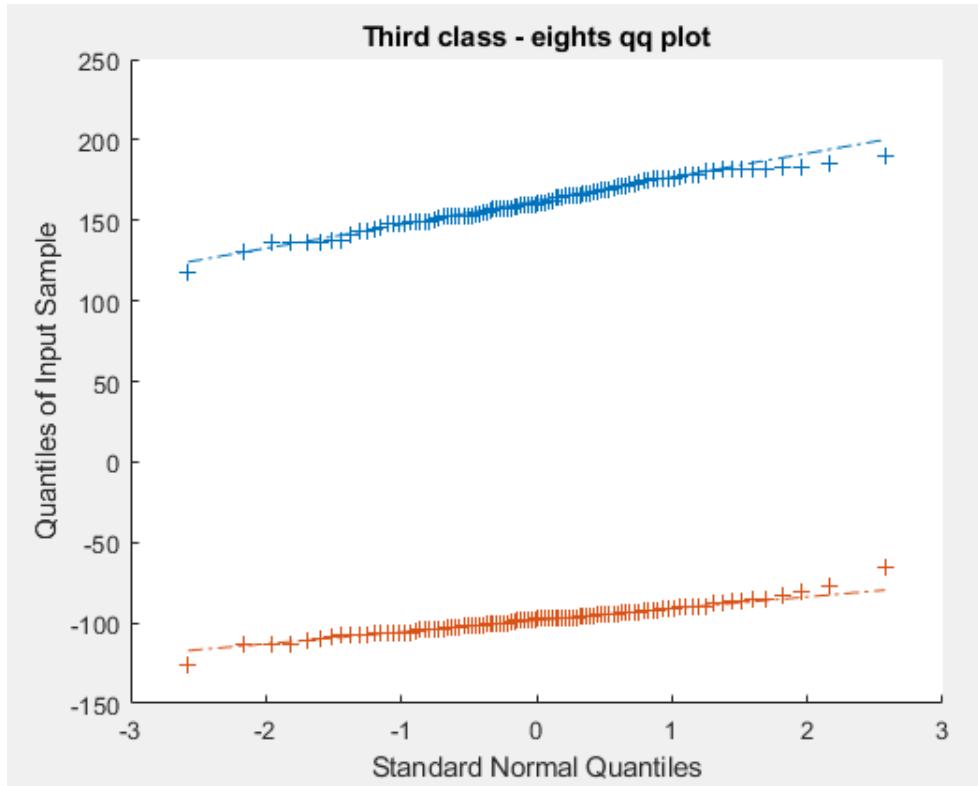
### Klasifikacija odbiraka

Nakon redukcije dimenzija klasifikacija se može uraditi na različite načine. Kako je potrebno klasifikovati odbirke na bazi testiranja hipoteza, izabrano je da se prepostavite normalne raspodele, definisane vektorima matematičkih očekivanja i kovarijacionim matricama dobijenih na osnovu obučavajućeg skupa redukovanih dimenzija.

Na slikama 1.3.1. i 1.3.2. se vidi QQ-kriva za svaku klasu urađena za normalizovanu normalnu raspodelu za obe komponente vektora. Sa grafika se vidi da se podaci jako lepo grupišu oko prave, s tim da su linearno pomereni u odnosu na istu (što je pokriveno vektorima matematičkih očekivanja), što opravdava prepostavljenu raspodelu.



Slika 1.3.1. QQ krive za prvu i drugu klasu



*Slika 1.3.2. QQ krive za treću klasu*

Koristi se Bajesovo pravilo odlučivanja minimalne greške<sup>1</sup>. Za dati test vektor se računa vrednost funkcije gustine verovatnoće za svaku klasu i vektor se pridružuje onoj klasi za koju je data vrednost maksimalna.

Rezultat ovakve klasifikacije prikazan u vidu konfuzione matrice je:

$$M_k = \begin{bmatrix} 18 & 0 & 2 \\ 1 & 19 & 0 \\ 0 & 0 & 20 \end{bmatrix}$$

Uzimajući u obzir to da baza podataka nije velika, dati rezultati su prilično zadovoljavajući.

Na slici 1.3.4. su prikazane pogrešno klasifikovane cifre - nule kao osmice i šestica kao nula.

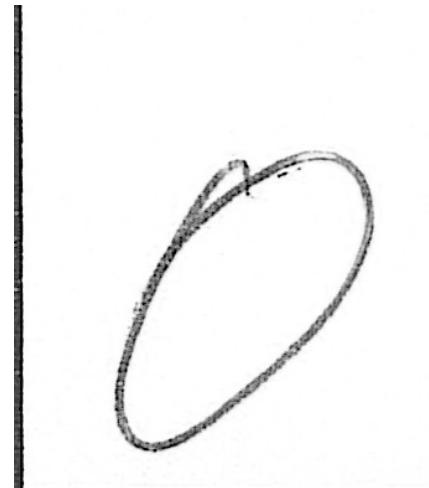
Na slici 1.3.5. se vide primeri cifara koje su dobro klasifikovane.

---

<sup>1</sup> Teorija je objašnjena u sklopu drugog domaćeg zadatka.



*Slika 1.3.5. Pogrešno klasifikovane cifre*



*Slika 1.3.6. Primeri dobro klasifikovanih cifara*

## Drugi domaći zadatak – opis problema

Potrebno je generisati i prikazati 500 odbiraka iz dveju dvodimenzionalnih klasa. Potom treba generisati geometrijsko mesto tačaka sa konstantnom vrednošću funkcija gustina verovatnoće i prikazati ih na dijagramu u prostoru oblika. Generisati Bajesov klasifikator minimalne greške i na dijagramu sa odbircima skicirati klasifikacionu liniju i proceniti verovatnoću greške. Potrebno je ponoviti ovaj postupak i za neki drugi klasifikator po izboru. Za klase odbiraka generisanih u prethodnim tačkama isprojektovati *Wald*-ov sekvencijalni test pa skicirati zavisnost broja potrebnih odbiraka od usvojene verovatnoće grešaka prvog, odnosno drugog tipa.

## Domaći zadatak – opis rešenja

Rešenje prati sledeći algoritam:

1. Generisanje po 500 odbiraka iz dveju dvodimenzionalnih klasa. Odbirci obe klase imaju bimodalnu raspodelu. Prikaz odbiraka na dijagramu.
2. Generisanje geometrijskog mesta tačaka sa konstantnom vrednošću funkcija gustina verovatnoće i prikaz na dijagramu u prostoru oblika.
3. Projektovanje Bajesovog klasifikatora minimalne greške i prikaz klasifikacione linije zajedno sa odbircima na dijagramu. Određivanje greške klasifikacije.
4. Projektovanje klasifikatora koji se bazira na Neyman-Pearson-ovom testu i prikaz klasifikacione linije zajedno sa odbircima na dijagramu. Određivanje greške klasifikacije.
5. Projektovanje *Wald*-ovog sekvencijalnog testa za klase prethodno generisanih odbiraka i skiciranje zavisnosti broja potrebnih odbiraka od usvojene verovatnoće grešaka prvog i drugog tipa.

### Generisanje odbiraka

Zadatak je da se generišu odbirci dveju dvodimenzionalnih klasa čija su matematička očekivanja i kovarijacione matrice poznati. U Matlabu je moguće generisati odbirke jedinične raspodele, dakle potrebno je ove odbirke preslikati u odbirke sa željenom raspodelom. Transformacija kojom se ovo postiže je inverzna transformacija beljenja, takozvana transformacija bojenja.

Neka su  $M$  vektor matematičkog očekivanja, a  $\Sigma$  kovarijaciona matrica željene normalne raspodele.

Kovarijaciona matrica željene raspodele se može napisati u obliku:

$$\Sigma = \Phi \Lambda \Phi^T$$

gde je  $\Phi$  njena matrica sopstvenih vektora, a  $\Lambda$  njena matrica sopstvenih vrednosti.

Neka je  $Y$  slučajni vektor normalne raspodele koji ima nulti vektor matematičkog očekivanja i kovarijacionu matricu koja je jednaka jediničnoj. Prvi korak jeste da se on sleva pomnoži matricom  $\Lambda^{1/2}$  da bi se dobili odbirci koji su u opsegu željene varijanse. Nakon toga rezultat se sleva množi matricom  $\Phi$

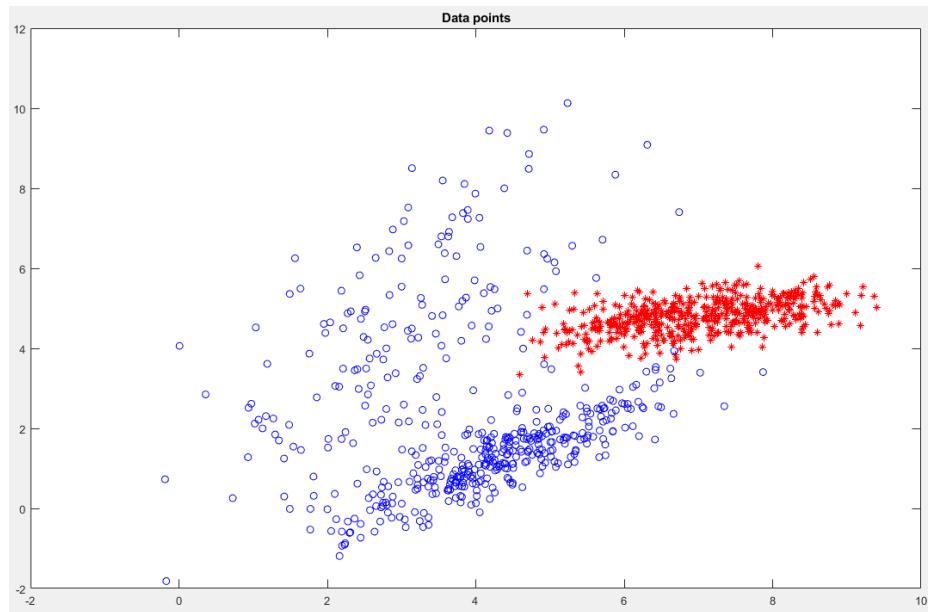
čime se prethodno nekorelisani podaci prevode u korelisane. Konačno, rezultatu se dodaje željeno matematičko očekivanje.

Ovim se dobija slučajni vektor  $X$  sa željenim matematičkim očekivanjem i kovarijacionom matricom.

$$X = \Phi \Lambda^{1/2} Y$$

Matrica sopstvenih vektora i matrica sopstvenih vrednosti su dobijene korišćenjem ugrađene Matlab funkcije *eig*.

Na slici 2.1.1. su prikazani odbirci.



Slika 2.1.1. Generisani odbirci

Funkcije gustina verovatnoća su date kao:

$$\begin{aligned} f_1(X) &\sim \mathcal{N}(M_1, S_1) \\ f_2(X) &\sim P_{21}\mathcal{N}(M_{21}, S_{21}) + P_{22}\mathcal{N}(M_{22}, S_{22}) \end{aligned}$$

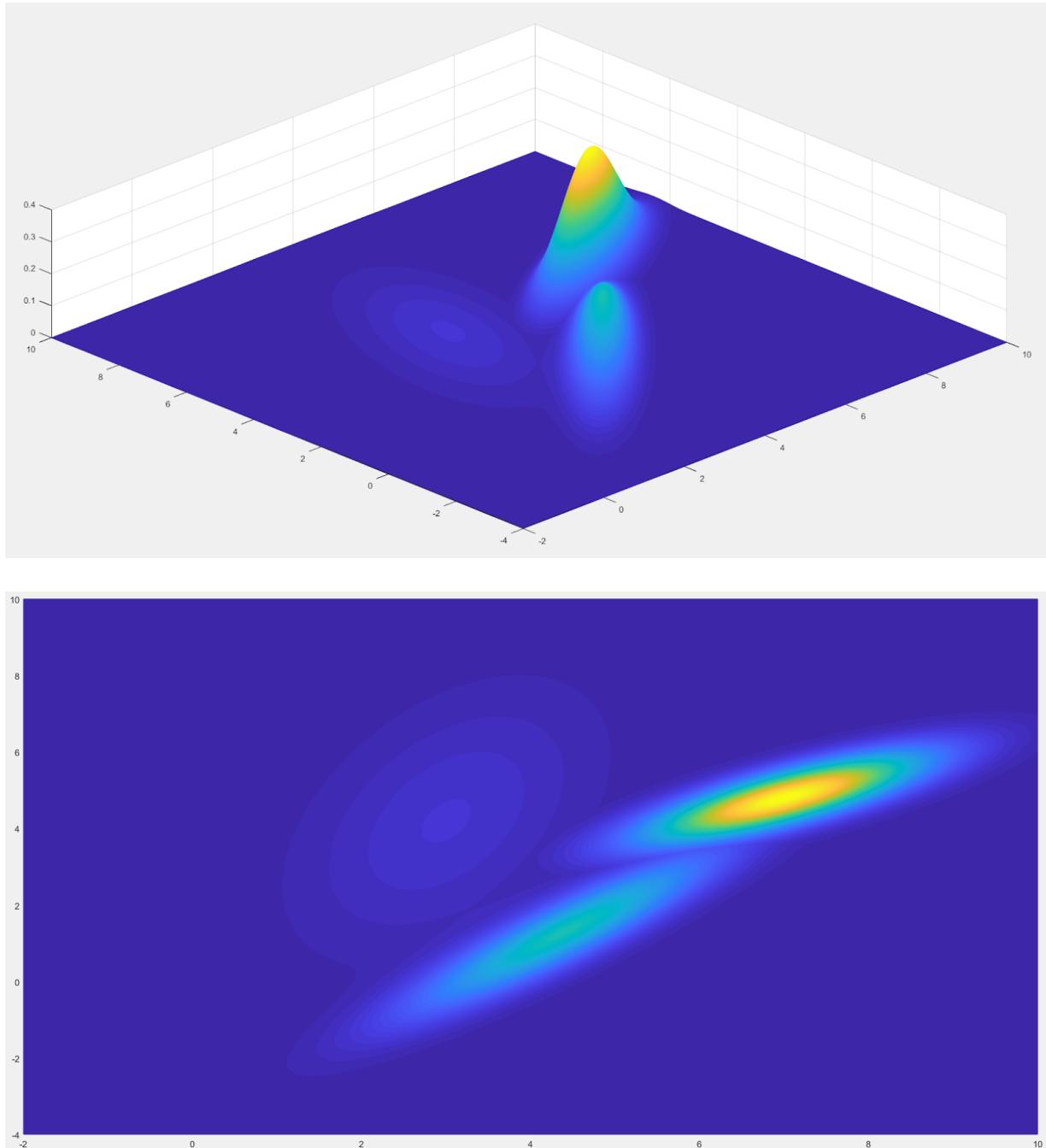
pritom je:

$$M_1 = \begin{bmatrix} 7 \\ 4.8 \end{bmatrix}, M_{21} = \begin{bmatrix} 3 \\ 4.2 \end{bmatrix}, M_{22} = \begin{bmatrix} 4.4 \\ 1.3 \end{bmatrix}$$

$$S_1 = \begin{bmatrix} 0.8 & 0.7 \\ 0.2 & 0.4 \end{bmatrix}, S_{21} = \begin{bmatrix} 1.3 & 1.5 \\ 0.8 & 5 \end{bmatrix}, S_{22} = \begin{bmatrix} 0.9 & 1.3 \\ 1.6 & 1.2 \end{bmatrix}$$

$$P_{21} = 0.35, P_{22} = 1 - P_{21}$$

Na slici 2.I.2. su prikazane izabrane funkcije gustine verovatnoće za date klase u prostoru oblika.



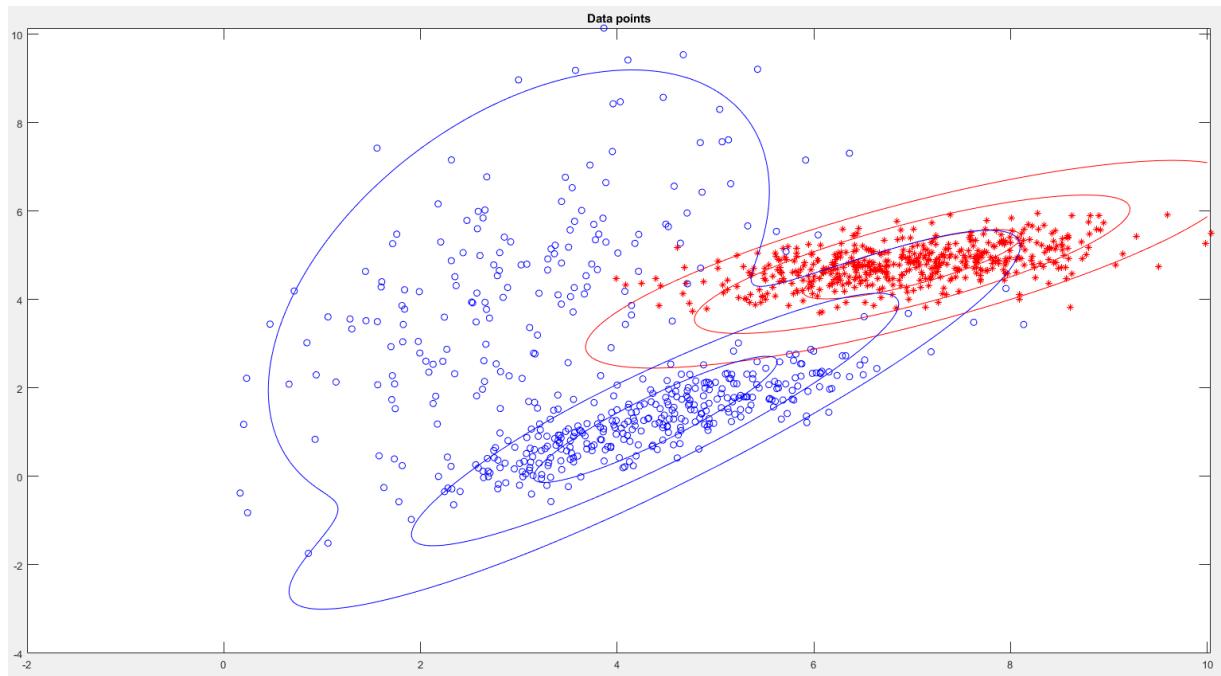
Slika 2.I.2. Funkcije gustine verovatnoće u prostoru oblika

## Generisanje geometrijskog mesta tačaka sa konstantnom vrednošću funkcije gustine verovatnoće

Izabrano je da se na dijagramu iscrtaju geometrijska mesta tačaka čije je statističko rastojanje od centra 1, 2, i 3.

Tačke imaju isto statističko rastojanje ukoliko je vrednost funkcije gustine verovatnoće u njima ista. U slučaju bimodalne raspodele kao centar je izabrana ona tačka u kojoj funkcija dostiže maksimalnu vrednost.

Na slici 2.2.1. su prikazana ova geometrijska mesta tačaka. Korišćena je ugrađena Matlab funkcija *contour* za njihovo izračunavanje.



Slika 2.2.1. Geometrijska mesta tačaka sa konstantnim vrednostima funkcije gustine verovatnoće

## Projektovanje Bajesovog klasifikatora minimalne greške

Pretpostavimo da oblici dolaze iz  $L$  klasa i neka je sa  $f(X/\omega_i)$  označena uslovna funkcija gustine verovatnoće za klasu  $i$ , a sa  $P_i$  apriori verovatnoća pojave  $i$ -te klase. Verovatnoća da će merni vektor uzeti neku vrednost  $X$  i pripadati klasi  $i$  je prema formuli uslovne verovatnoće data kao:

$$\Pr \{X, \omega_i\} = \Pr \{X/\omega_i\} \Pr \{\omega_i\} = f_i(X/\omega_i)P_i$$

Na osnovu Bajesove teoreme važi:

$$Pr \{ \omega_i / X \} = \frac{Pr \{ X, \omega_i \}}{Pr \{ \omega_i \}}$$

Sa druge strane, na osnovu formule totalne verovatnoće je:

$$Pr \{ X \} = \sum_{i=1}^L Pr \{ X / \omega_i \} Pr \{ \omega_i \}$$

Definišimo sa  $q_i(X)$  verovatnoću da merena vrednost  $X$  pripada klasi  $i$ . Zamenom vrednosti iz gornjih jednačina dobijamo izraz za uslovnu verovatnoću, to jest verovatnoću da merena vrednost  $X$  pripada klasi  $\omega_i$ .

$$q_i(X) = \frac{P_i f(X / \omega_i)}{\sum_{i=1}^L P_i f(X / \omega_i)}$$

Bajesovo pravilo odlučivanja minimalne greške se bazira na priduživanju odbirka onoj klasi za koju je uslovna verovatnoća maksimalna. Ovo se može kompaktно zapisati kao:

$$q_k(X) = \max_{i=1,..,L} \{ q_i(X) \} \Rightarrow X \in \omega_k$$

Kako je član u imeniku zajednički za sve uslovne verovatnoće pravilo odlučivanja se može zapisati kao:

$$P_k f(X / \omega_k) = \max_{i=1,..,L} \{ P_i f(X / \omega_i) \} \Rightarrow X \in \omega_k$$

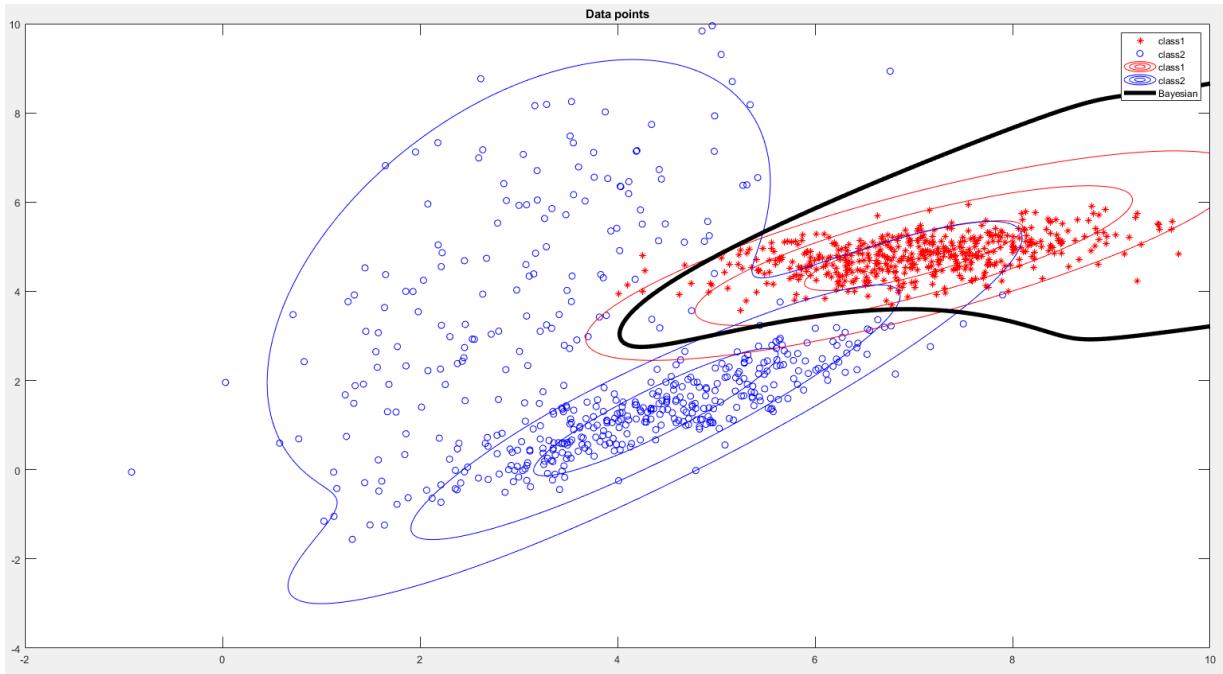
Kada imamo dve klase ovo pravilo dobija sledeći oblik:

$$\frac{f_1(X)}{f_2(X)} \stackrel{\omega_1}{\gtrless} \frac{P_2}{P_1}$$

Često se definiše diskriminaciona funkcija  $h(X)$  kao  $h(X) = -\log(f_1(X)/f_2(X))$ , pa pravilo odlučivanja nakon primene negativnog logaritma postaje:

$$h(X) \stackrel{\omega_2}{\gtrless} \log \frac{P_1}{P_2}$$

U konkretnom primeru su apriori verovatnoće jednake i iznose 0.5, pa je prag odlučivanja jednak nuli. Na slici 2.3.1. je prikazana klasifikaciona linija dobijena primenom Bajesovog pravila odlučivanja minimalne greške.



Slika 2.3.1. Klasifikaciona linija dobijena primenom Bajesovog pravila odlučivanja minimalne greške

### Verovatnoća greške Bajesovog pravila odlučivanja minimalne greške

Bajesovo pravilo zaista minimizira ukupnu grešku odlučivanja i to se lako pokazuje. Neka je  $\varepsilon_1$  verovatnoća greške prvog tipa, to jest verovatnoća da će odbirak iz prve klase biti pogrešno pridružen drugoj klasi, i neka je  $\varepsilon_2$  verovatnoća greške drugog tipa, odnosno verovatnoća da će odbirak iz druge klase biti pogrešno pridružen prvoj. Tada je ukupna greška  $\varepsilon$  data kao:

$$\varepsilon = P_1\varepsilon_1 + P_2\varepsilon_2$$

Na osnovu definicije pravila data greška je jednaka:

$$\varepsilon = \int_{L_2} P_1 f_1(X) dX + \int_{L_1} P_2 f_2(X) dX$$

gde je sa  $L_i$  označena oblast u kojoj se odbirak pridružuje klasi  $i$ , odnosno greška tipa  $i$  je ukupna verovatnoća da će odbirak iz klase  $i$  imati one vrednosti za koje se pridružuje suprotnoj klasi.

Lako se pokazuje da je dato pravilo obezbedilo minimalnu grešku. Pretpostavimo da nismo definisali pravilo odlučivanja, već pokušavamo da minimizujemo grešku. Verovatnoće pojave klasa kao i funkcije gustine verovatnoće su unapred definisane i pozitivne vrednosti. Jedina stvar na koju možemo uticati je oblast integracije.

Ukoliko primetimo da važi:

$$\int_{L_2} f_1(X) dX = 1 - \int_{L_1} f_1(X) dX$$

možemo pisati ukupnu grešku kao:

$$\varepsilon = 1 + \int_{L_1} P_2 f_2(X) - P_1 f_1(X) dX$$

Očigledno, da bi integral, odnosno greška, bio minimalan, oblast  $L_1$  treba izabrati tako da je na njoj podintegralna funkcija negativna, odnosno, oblast koja odgovara prvoj klasi treba izabrati tako da je na njoj  $P_1 f_1(X) > P_2 f_2(X)$ . Ovo je upravo Bajesovo pravilo odlučivanja minimalne greške, pa je ovim tvrđenje dokazano.

Procena greške se može uraditi na nekoliko načina. Najjednostavniji na način je da se greške prvog i drugog tipa procene kao odnos broja pogrešno klasifikovanih i ukupnog broja odbiraka svake klase. Prednost je u tome što se ovakva procena veoma lako računa (ovde je, na primer, to moglo da se vidi sa slike). Mana ovakve procene je u tome što je potreban veliki broj odbiraka da bi ista imala smisla, a sa druge strane, “nije fer” koristiti iste odbirke za obučavanje klasifikatora i njegovo testiranje.

U konkretnom primeru procenjene greške date u vidu konfuzione matrice su:

$$\begin{bmatrix} 494 & 6 \\ 7 & 493 \end{bmatrix}$$

Odnosno, greške prvog i drugog tipa su:

$$\hat{\varepsilon}_1 = 0.012, \hat{\varepsilon}_2 = 0.014$$

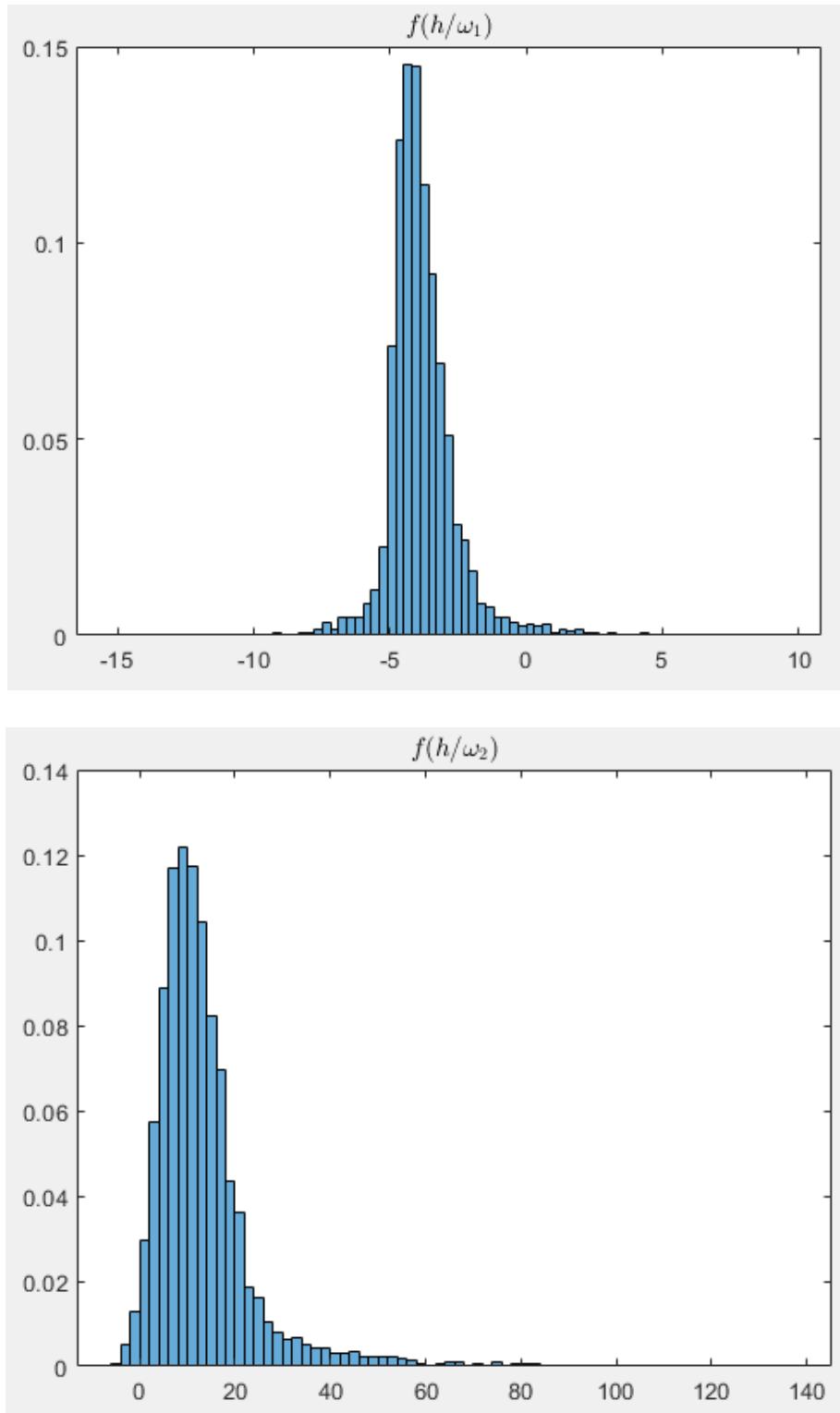
Drugi način je na osnovu definicije, računanjem integrala. Ovde je korišćeno trapezoidno pravilo za računanje integrala. Ovakvom procenom su dobijene sledeće greške prvog i drugog tipa:

$$\hat{\varepsilon}_1 = 0.0322, \hat{\varepsilon}_2 = 0.0705$$

Treći način je na osnovu diskriminacione funkcije. Na osnovu 5000 generisanih odbiraka je određen histogram slučajne promenljive  $h$  i računate greške. Dobijene su sledeće vrednosti:

$$\hat{\varepsilon}_1 = 0.0180, \hat{\varepsilon}_2 = 0.0148$$

Na slici 2.3.2. su prikazane eksperimentalno određene uslovne funkcije gustina verovatnoća promenljive  $h$ .



Slika 3.2.2. Funkcije gustine verovatnoća promenljive  $h$ .

## Projektovanje klasifikatora koji se bazira na Neyman-Pearson-ovom testu

Ovaj test se bazira na minimizaciji jedne greške dok se druga drži konstantnom. Neka se greška  $\varepsilon_2$  drži konstantnom  $\varepsilon_2 = \varepsilon_0 = const.$  a greška prvog tipa minimizuje. Tada je kriterijumska funkcija data kao:

$$r = \varepsilon_1 + \mu(\varepsilon_2 - \varepsilon_0)$$

gde je  $\mu$  Lagranžev multiplikator.

Zamenom definicija grešaka prvog i drugog tipa u gornju jednačinu se dobija:

$$r = \int_{L_2} f_1(X) dX + \mu \left( \int_{L_1} f_2(X) dX - \varepsilon_0 \right)$$

Uvezši u obzir da je

$$\int_{L_2} f_1(X) dX = 1 - \int_{L_1} f_1(X) dX$$

kriterijumska funkcija se konačno može zapisati u obliku:

$$r = (1 - \mu\varepsilon_0) + \int_{L_1} \mu f_2(X) - f_1(X) dX$$

Kao u dokazu Bajesovog pravila odlučivanja minimalne greške, i ovde je potrebno minimizovati funkciju biranjem odgovarajuće oblasti integracije. Istom logikom dolazimo do zaključka da je prvoj klasi potrebno pridružiti slučajeve u kojima je  $\mu f_2(X) < f_1(X)$ , a ostale drugoj klasi.

Ovo pravilo odlučivanja se u domenu diskriminacione funkcije piše kao:

$$h(X) \stackrel{\omega_2}{\underset{\omega_1}{\gtrless}} -\log \mu$$

Parametar  $\mu$  figuriše u oblasti integracije, odnosno važi  $L_1 = L_1(\mu)$ . Takođe, ova oblast je definisana u sklopu greške drugog tipa, pa se dobija veza između ovih parametara kao:

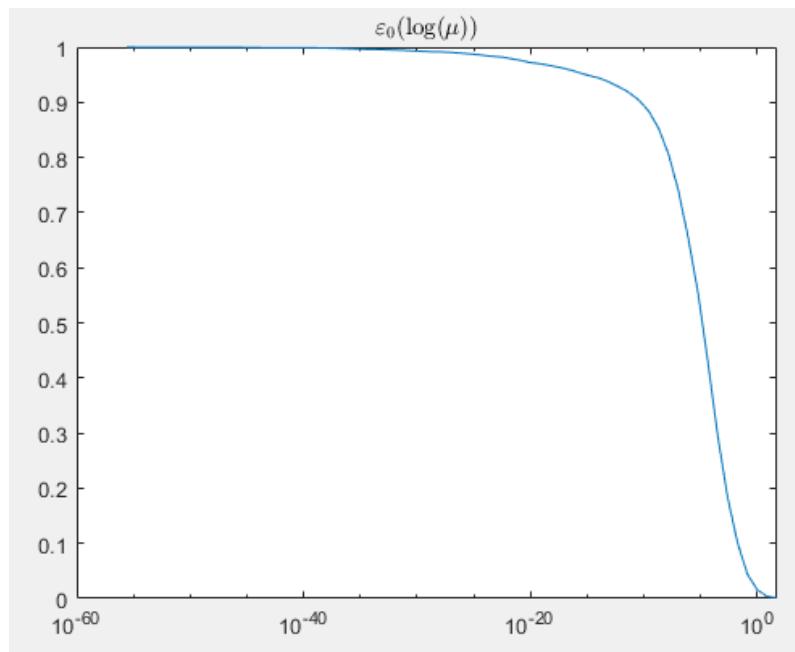
$$\varepsilon_2 = \int_{L_1(\mu)} f_2(X) dX = \varepsilon_0$$

Jasno je da je dati izraz nepogodan za rad. Stoga se koristi treći način za procenu greške koji se bazira na računanju po slučajnoj promenljivoj  $h$  koja predstavlja slučajnu promenljivu koja ima vrednost diskriminacione funkcije. Greška prvog tipa postoji tamo gde odbirci pripadaju prvoj klasi, a diskriminaciona funkcija ima vrednost veću od praga. Analogno, greška drugog tipa je ukupna verovatnoća da će odbirci druge klase imati diskriminacionu funkciju ispod vrednosti praga. Neka je sa  $f(h/\omega_i)$  označena funkcija gustine verovatnoće slučajne promenljive  $h$  onda kada odbirci pripadaju  $i$ -toj klasi. Tada se greške prvog i drugog tipa mogu definisati kao:

$$\varepsilon_1 = \int_{-\log \mu}^{+\infty} f(h/\omega_1) dh, \quad \varepsilon_2 = \int_{-\infty}^{-\log \mu} f(h/\omega_2) dh$$

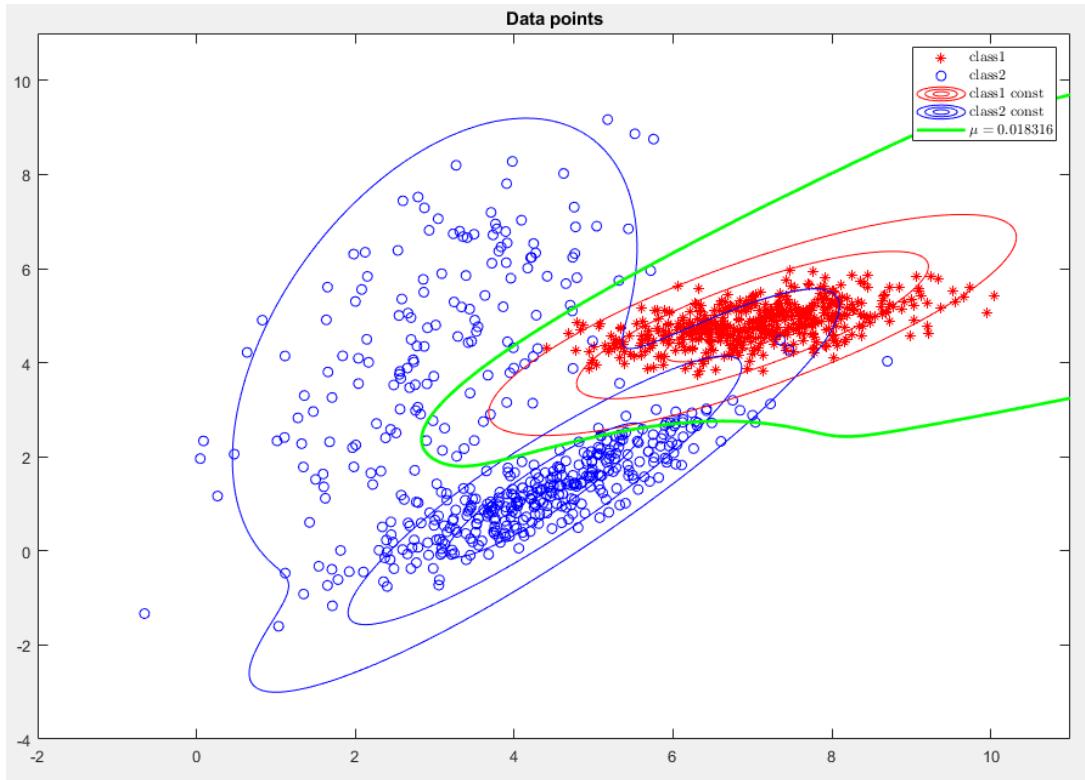
Funkcije gustine verovatnoće slučajne promenljive  $h$  se mogu eksperimentalno odrediti generisanjem odbiraka datih klasa. Broj odbiraka je 5000.

Na osnovu generisanog histograma je određena zavisnost  $\varepsilon_0(\mu)$ . Ova zavisnost je u prikazana na slici 2.4.1. u logaritamskoj razmeri.



Slika 2.4.1. Zavisnost  $\varepsilon_0(\mu)$  u logaritamskoj razmeri

Na slici 2.4.2. je prikazana Neyman-Pearson-ova klasifikaciona linija za  $\mu = 0.0183$ .



Slika 2.4.3.. Neyman-Person-ov klasifikator  $\mu = 0.0183$

Na osnovu procenjene zavisnosti greške drugog tipa od parametra  $\mu$  je očekivana vrednost greške  $\varepsilon_0 = 0.0956$ . Računanjem integrala su dobijene sledeće vrednosti grešaka:

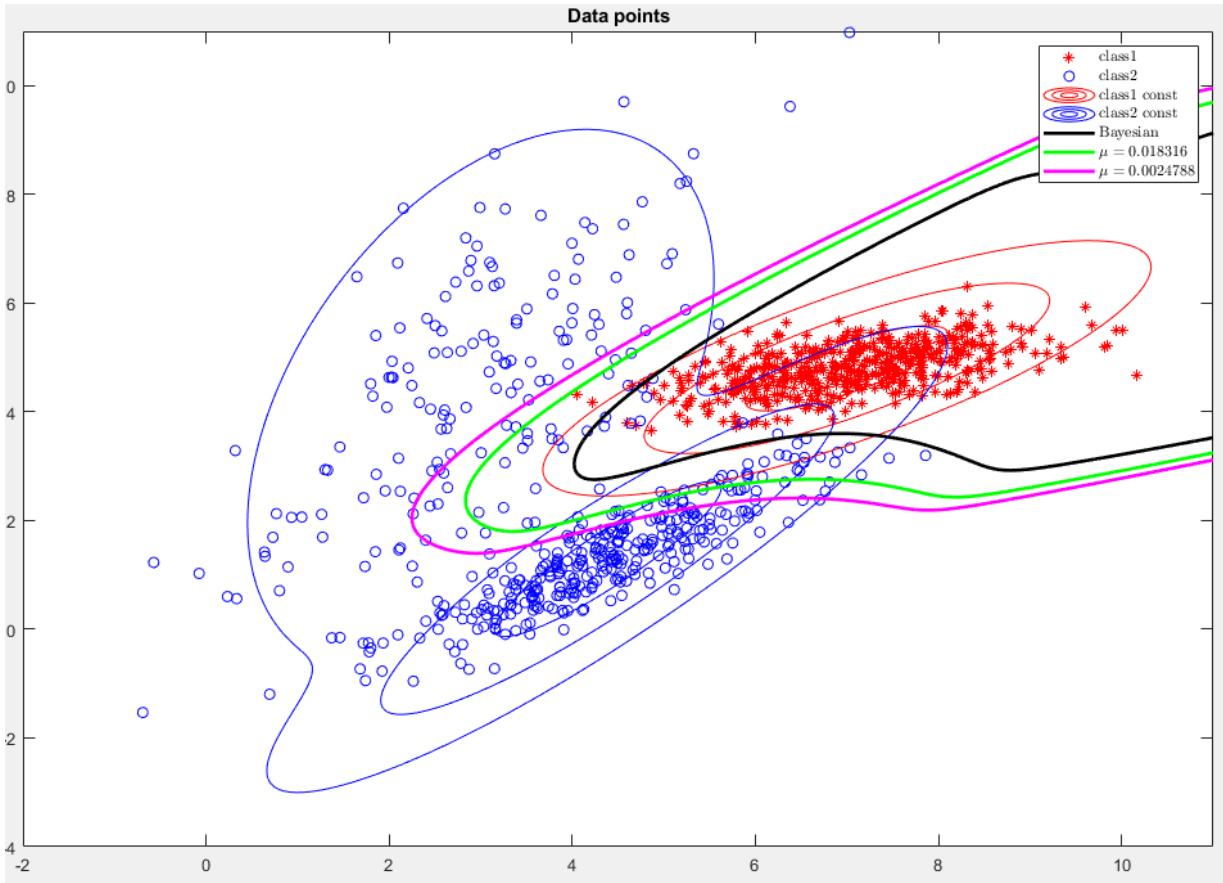
$$\hat{\varepsilon}_1 = 9.1013e - 04, \hat{\varepsilon}_2 = 0.2207$$

Ovakva greška u proceni je posledica toga što rezolucija sa kojom je rađeno nije bila dovoljna i to znatno utiče za manje vrednosti  $\mu$  gde se mala promena  $\mu$  odgovara velikoj promeni greške (čak i u logaritamskoj razmeri ovaj nagib je veoma veliki). Isto važi za procenu greške obzirom da je  $\varepsilon_0$  zavisnost od  $-\log(\mu)$  samo obrnuta funkcija raspodele, a funkcija raspodele je upravo vrednost greške za dati prag.

Kao i u slučaju Bajesovog klasifikatora, određene su i greške na osnovu promenljive  $h$ . Dobijene vrednosti su:

$$\hat{\varepsilon}_1 = 0.0012, \hat{\varepsilon}_2 = 0.1030$$

Konačno, na slici 2.4.4. se vide zajedno oba klasifikatora, pritom su razmatrane dve vrednosti parametra  $\mu$ . Ovde se lepo vidi kako sa smanjenjem ovog parametra greška drugog tipa raste.



Slika 2.4.4. Bayesov i Neyman-Person-ov klasifikator

### Projektovanje Wald-ovog sekvencijalnog testa

Sekvencijalno testiranje ima primenu u slučajevima kada sa prolaskom vremena dobijamo nove informacije. Neka su  $X_1, X_2, \dots, X_m$  izmerene opservacije dobijene sekvencijalnim testiranjem. Pretpostavimo da su ovi slučajni vektori nezavisni i jednakoraspodeljeni sa poznatim aposteriornim funkcijama gustine  $f_1(X)$  i  $f_2(X)$ . Tada možemo da formiramo združenu funkciju gustine verovatnoće za  $m$  ovih slučajnih vektora i da na količnik verodostojnosti napišemo kao:

$$l_m = \frac{f_1(X_1, X_2, \dots, X_m)}{f_2(X_1, X_2, \dots, X_m)} = \frac{\prod_{i=1}^m f_1(X_i)}{\prod_{i=1}^m f_2(X_i)}$$

jer su promenljive nezavisne.

Definišimo  $s_m$  kao negativni logaritam količnika verodostojnosti. Tada je:

$$s_m = -\log l_m = -\sum_{i=1}^m \log \frac{f_1(X_i)}{f_2(X_i)} = \sum_{i=1}^m h(X_i)$$

U najvećem broju merenja za prvu klasu važi  $f_1(X) > f_2(X)$ , pa se može očekivati da  $s_m$  opada kako dolazi sve više odbiraka iz prve klase. I obrnuto,  $s_m$  raste kako se pojavljuju novi odbirci druge klase. Dokažimo ovo tvrđenje tako što ćemo posmatrati srednju vrednost i varijansu slučajne promenljive  $s_m$ .

$$E \{s_m/\omega_i\} = \sum_{j=1}^m E \{h(X_j)/\omega_i\} = m\eta_i$$

gde je  $\eta_i = E \{h(X_j)/\omega_i\}$ .

Po definiciji, matematička očekivanja  $\eta_1$  i  $\eta_2$  su:

$$\eta_i = E \left\{ -\log \frac{f_1(X)}{f_2(X)} / \omega_i \right\} = \int_{-\infty}^{\infty} -\log \frac{f_1(X)}{f_2(X)} f_i(X) dX$$

Važi nejednakost  $\log x \leq x - 1$ , pa se može pisati:

$$\eta_1 \leq \int_{-\infty}^{\infty} \left( \frac{f_2(X)}{f_1(X)} - 1 \right) f_1(X) dX = \int_{-\infty}^{\infty} f_2(X) dX - \int_{-\infty}^{\infty} f_1(X) dX = 0$$

$$-\eta_2 \leq \int_{-\infty}^{\infty} \left( \frac{f_1(X)}{f_2(X)} - 1 \right) f_2(X) dX = \int_{-\infty}^{\infty} f_1(X) dX - \int_{-\infty}^{\infty} f_2(X) dX = 0$$

Ovim je prethodno tvrđenje dokazano.

Na osnovu ovoga možemo formirati test kao:

$$\begin{aligned} s_m \leq a &\Rightarrow X \in \omega_1 \\ s_m \in (a, b) &\Rightarrow \text{take another sample} \\ s_m \geq b &\Rightarrow X \in \omega_2 \end{aligned}$$

gde su  $a$  i  $b$  granice odlučivanja i pomoću njih se kontroliše greška.

Neka je  $a = -\log A$  i  $b = -\log B$ . Tada se gore navedeno pravilo prevodi u pridruživanje prvoj klasi kada je  $l_m \geq A$ , pridruživanje drugoj klasi kada je  $l_m \leq B$  i uzimanje novog odbirka ako je količnik verodostojnosti između  $A$  i  $B$ .

Posmatrajmo slučaj pridruživanja prvoj klasi.

$$l_m = \frac{f_1(X_1, X_2, \dots, X_n)}{f_2(X_1, X_2, \dots, X_n)} \geq A \Rightarrow X \in \omega_1$$

Ako pomnožimo obe strane funkcijom  $f_2(X_1, X_2, \dots, X_n)$ , zatim integralimo po oblasti gde je  $l_m \geq A$  i konačno sumiramo svaku stranu za svako  $m$  dobijamo:

$$\sum_{m=1}^{\infty} \int_{l_m \geq A} f_1(X_1, X_2, \dots, X_n) dX_1 dX_2 \dots dX_n \geq A \sum_{m=1}^{\infty} \int_{l_m \geq A} f_2(X_1, X_2, \dots, X_n) dX_1 dX_2 \dots dX_n$$

Izraz sa leve strane predstavlja verovatnoću da će odbirci prve klase biti dobro klasifikovani, dok izraz sa desne strane predstavlja verovatnoću greške drugog tipa.

$$1 - \varepsilon_1 \geq A\varepsilon_2$$

Na isti način, polazeći od odluke za drugu klasu, dobijamo:

$$\varepsilon_1 \leq B(1 - \varepsilon_2)$$

Iz ove veze dobijamo vrednosti granica u zavisnosti od željene greške.

$$a = \log \frac{\varepsilon_2}{1 - \varepsilon_1}, \quad b = \log \frac{1 - \varepsilon_2}{\varepsilon_1}$$

Wald-ov sekvenčijalni test minimizira srednju vrednost potrebnog broja merenja za zadate greške.

Podsetimo se da je

$$E \{ s_m / \omega_i \} = \sum_{j=1}^m E \{ h(X_j) / \omega_i \} = m\eta_i$$

U ovom izrazu promenljiva  $m$  takođe zavisi od klase, pa je ovo njena srednja vrednost. Sa druge strane, promenljiva  $s_m$  može uzeti jednu od dve vrednosti -  $a, b$ . Stoga je

$$\begin{aligned} E \{ s_m / \omega_1 \} &= a(1 - \varepsilon_1) + b\varepsilon_2 \\ E \{ s_m / \omega_2 \} &= a\varepsilon_2 + b(1 - \varepsilon_2) \end{aligned}$$

Na osnovu ovoga nalazimo izraz za očekivani broj merenja:

$$E \{m/\omega_1\} = \frac{a(1 - \varepsilon_1) + b\varepsilon_2}{\eta_1}$$

$$E \{m/\omega_2\} = \frac{a\varepsilon_2 + b(1 - \varepsilon_2)}{\eta_2}$$

Test je projektovan na sledeći način:

1. Izabrane su određene vrednosti grešaka.
2. Za date greške su određeni parametri  $a$  i  $b$ .
3. Za date vrednosti greške generišu se obirci prve klase dokle god je  $s_m$  takvo da je potrebno još odbiraka da se doneše odluka. Onda kada se doneše 20 odluka prelazi se na generisanje odbiraka druge klase po istom principu. Isto tako, kada se doneše 20 odluka, test se završava.
4. Nakon tačke 3 možemo odrediti broj merenja koji je u proseku bio potreban da se doneše odluka za date greške, kao i broj napravljenih grešaka.
5. Proces se ponavlja za druge vrednosti grešaka.

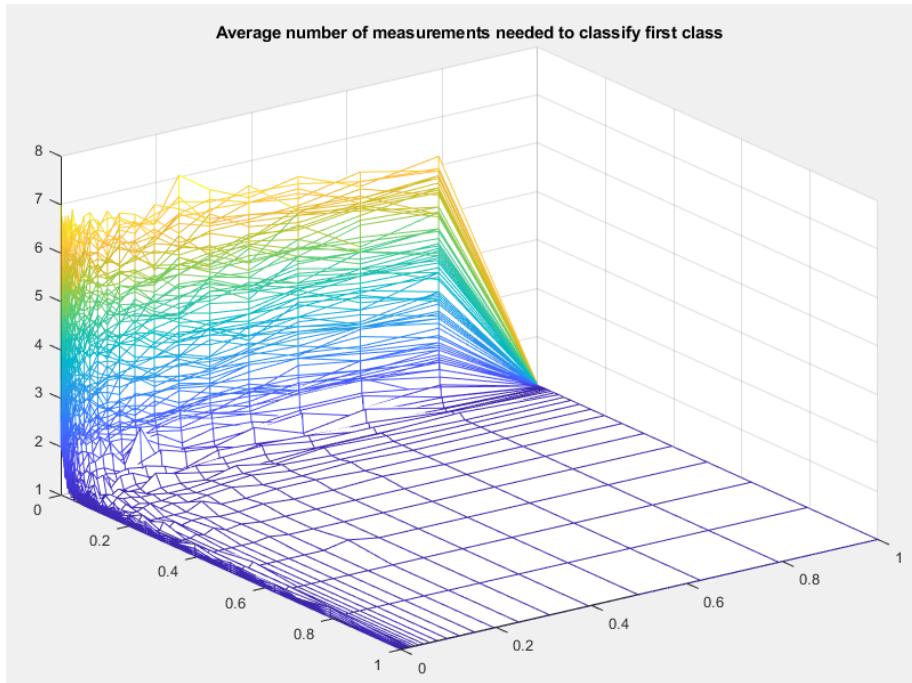
Na slici 2.5.1. se može videti zavisnost prosečnog broja potrebnih odbiraka za klasifikovanje odbiraka prve klase u zavisnosti od grešaka. Dok se na slici 2.5.2. vidi isto kada je u pitanju druga klasa.

Primetimo da izraz za broj merenja za male vrednosti grešaka ima oblik:

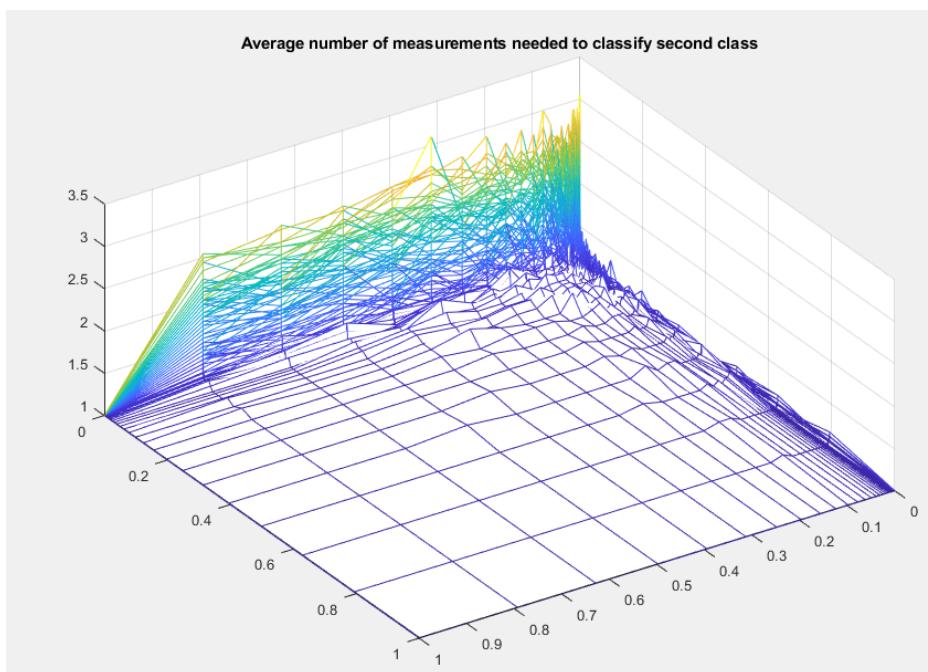
$$E \{m/\omega_1\} \approx \frac{\log \varepsilon_2}{\eta_1}$$

Ovo se i vidi sa grafika, promena jedne greške gotovo da ne utiče na broj merenja, dok smanjenje druge naglo povećava broj potrebnih odbiraka.

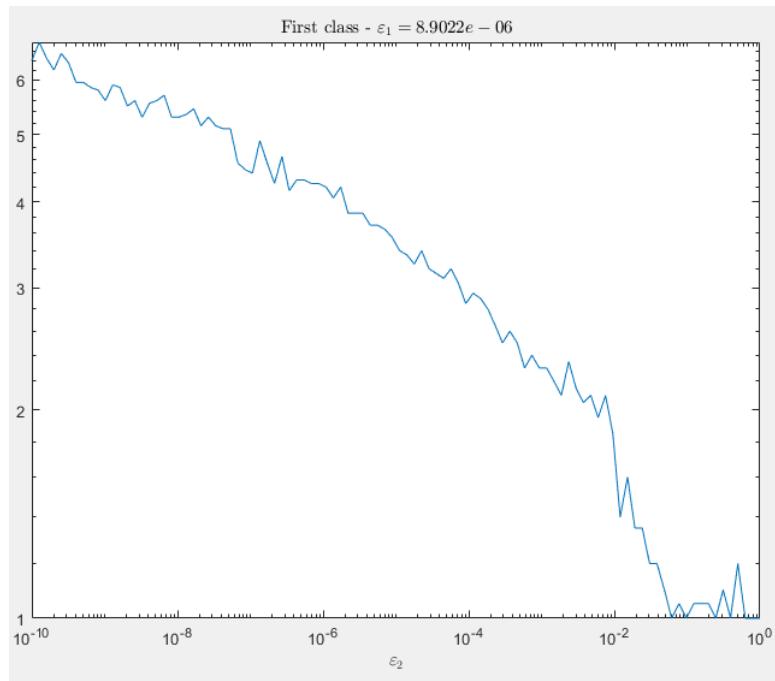
Na slici 2.5.3. se vidi kako se broj potrebnih merenja menja kada je  $\varepsilon_1 = \text{const}$  i odbirci su iz prve klase, dok se na slici 2.5.4. vidi broj merenja kada je to slučaj sa odbircima iz druge klase.



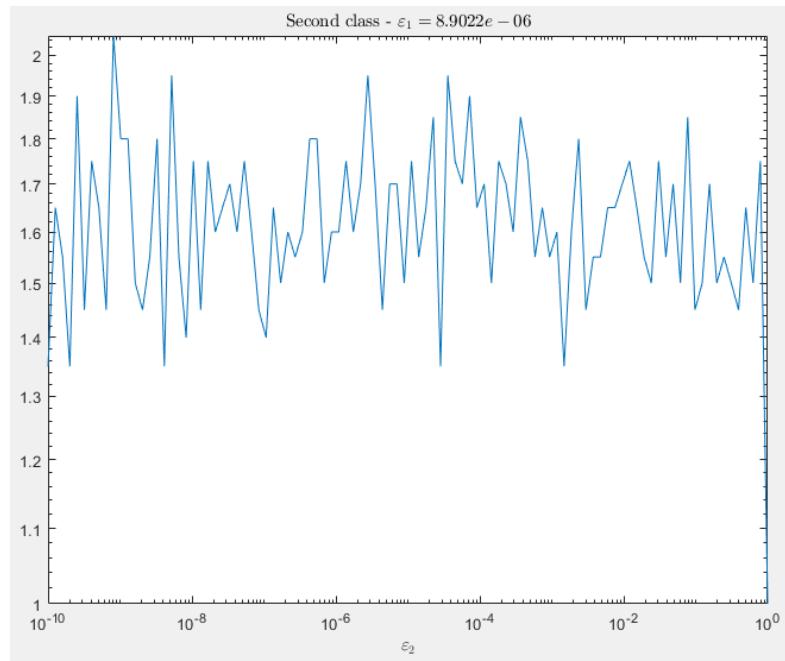
Slika 2.5.1. Zavisnost prosečnog broja odbiraka potrebnih da se doneše odluka u zavisnosti od grešaka prvog i drugog tipa kada odbirci pripadaju prvoj klasi



Slika 2.5.2. Zavisnost prosečnog broja odbiraka potrebnih da se doneše odluka u zavisnosti od grešaka prvog i drugog tipa kada odbirci pripadaju drugoj klasi



*Slika 2.5.3. Zavisnost broja merenja od  $\varepsilon_2$  kada  $\varepsilon_1 = \text{const} = 8.9022e - 06$  i odbirci dolaze iz prve klase*



*Slika 2.5.4. Zavisnost broja merenja od  $\varepsilon_2$  kada  $\varepsilon_1 = \text{const} = 8.9022e - 06$  i odbirci dolaze iz druge klase*

Zanimljivo je uporediti dobijene rezultate sa teorijskim brojem prosečno potrebnih merenja.

Srednje vrednosti diskriminacione funkcije su određene eksperimentalno. Generisano je po 500 odbiraka iz svake klase, zatim je određena vrednost diskriminacione funkcije uz pomoć histograma za svaku klasu i konačno njena srednja vrednost.

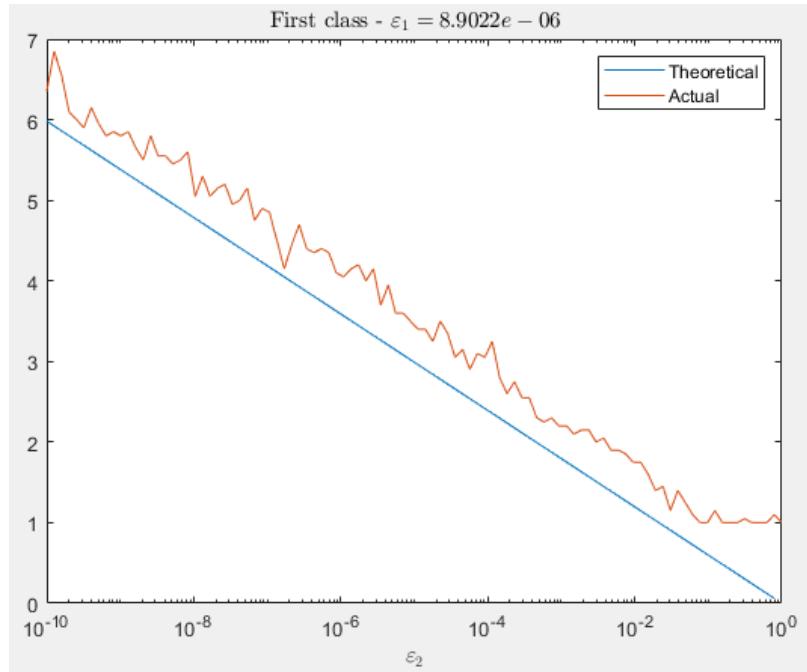
Nakon određivanja srednjih vrednosti broj potrebnih odbiraka je računat za istu konstantnu vrednost greške prvog i drugog tipa. Na slikama 2.5.6. i 2.5.7. se mogu videti zavisnost broja potrebnih merenja za donošenje odluke u logaritamskog zavisnosti od greške drugog tipa za prvu klasu, odnosno prvog tipa. Na slikama 2.5.8. i 2.5.9. se vidi zavisnost od grešaka prvog i drugog tipa za drugu klasu. Na istim graficima se mogu i videti vrednosti koje su prethodno dobijene izvođenjem *Wald*-ovog testa.

Kao što se vidi sa slike, dobijeni broj potrebnih merenja je u okviru onih koje predviđa teorija (broj odbiraka svakako ne može da bude manji od 1 jer nam je potreban bar jedan oblik da bismo mogli da radimo klasifikaciju).

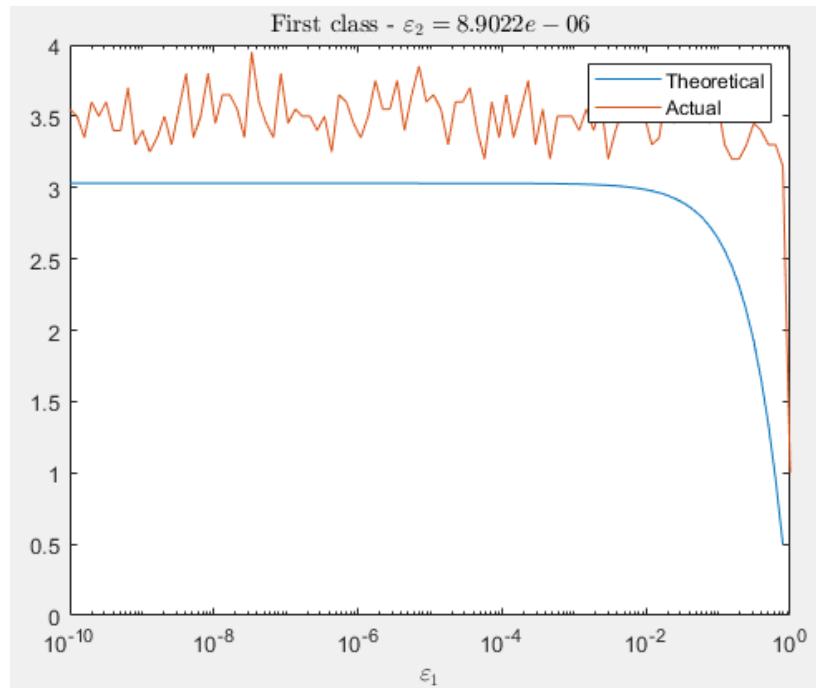
Sa grafika se takođe vidi da greška suprotne klase znatno više utiče na srednji broj potrebnih odbiraka.

Konačno, napomenimo tri veoma važne osobine *Wald*-vog sekvencijalnog testa.

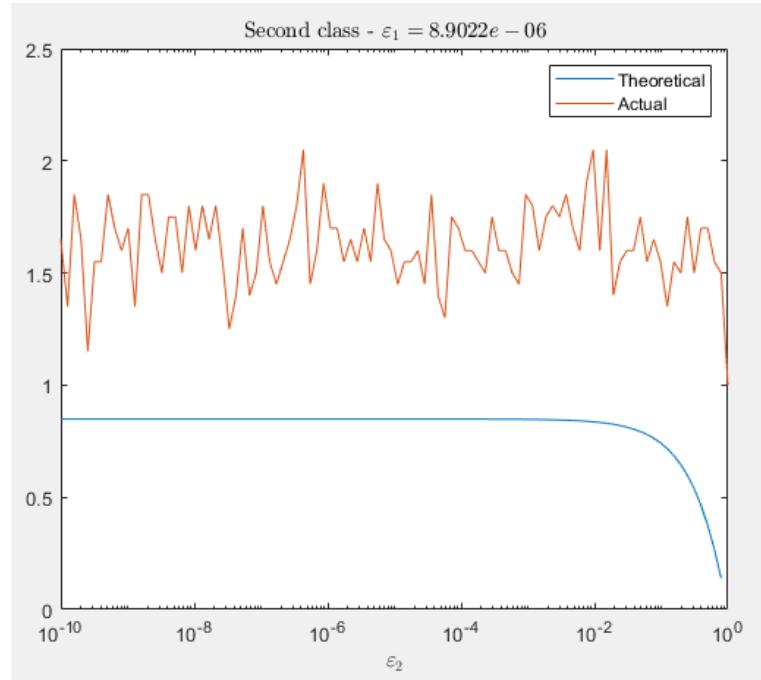
1. Može se pokazati da izrazi za vezu između verovatnoće greške prvog i drugog tipa sa parametrima A i B važe i ukoliko merenja nisu nezavisni jednak raspodeljeni.
2. *Wald*-ov sekvencijalni test se završava sa verovatnoćom 1, što znači da je verovatnoća događaja u kome broj odbiraka m neograničeno raste, a izraz ne dostiže graničnu vrednost jednaku nuli.
3. *Wald*-ov sekvencijalni test minimizira srednju vrednost potrebnog broja merenja za zadato  $\varepsilon_1$  i  $\varepsilon_2$ , što znači da među svim drugim testovima koji rade sa istim greškama, *Wald*-ov zahteva najmanji broj odbiraka u srednjem smislu.



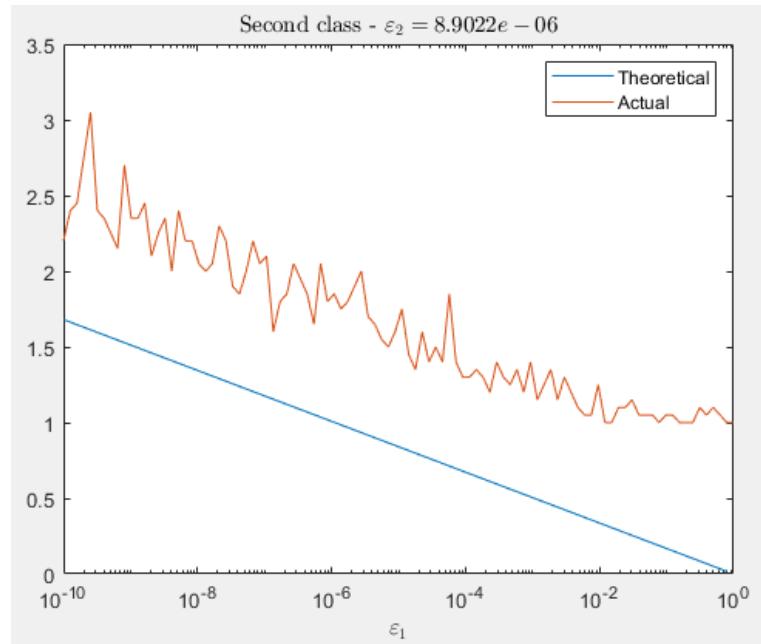
Slika 2.5.6. Teorijski i realni srednji broj potrebnih merenja za donošenje odluke u zavisnosti od greške drugog tipa za prvu klasu ( $\varepsilon_1 = \text{const} = 8.9022e - 06$ )



Slika 2.5.7. Teorijski i realni srednji broj potrebnih merenja za donošenje odluke u zavisnosti od greške prvog tipa za prvu klasu ( $\varepsilon_2 = \text{const} = 8.9022e - 06$ )



Slika 2.5.8. Teorijski i realni srednji broj potrebnih merenja za donošenje odluke u zavisnosti od greške drugog tipa za drugu klasu ( $\varepsilon_1 = \text{const} = 8.9022e - 06$ )



Slika 2.5.9. Teorijski i realni srednji broj potrebnih merenja za donošenje odluke u zavisnosti od greške prvog tipa za drugu klasu ( $\varepsilon_2 = \text{const} = 8.9022e - 06$ )

## Treći domaći zadatak – opis problema

Potrebno je generisati dve klase dvodimenzionalnih oblika. Izabratи funkciju gustine verovatnoće oblika tako da klase budu linearно separabilne. Za tako generisane oblike izvršiti projektovanje linearног klasifikatorа jednom od tri iterativne procedure. Ponoviti prethodni postupak korišćenjem metode željenog izlaza. Analizati uticaj elemenata u matrici željenih izlaza na konačnu formu linearног klasifikatorа. Generisati dve klase dvodimenzionalih oblika koje jesu separabilne ali ne linearно pa projektovati kvadratni klasifikator metodom po želji.

## Treći domaći zadatak – opis rešenja

Rešenje prati sledeći algoritam:

1. Generisano je po 1000 odbiraka svake klase.
2. Obe klase imaju normalnu raspodelu, ali su linearно separabilne. Prikazani su generisani odbirci. Projektovani su linearni klasifikatori prvom, drugom i trećom iterativnom procedurom i poređeni su rezultati.
3. Projektovan je klasifikator na bazi redukcija dimenzija.
4. Projektovan je klasifikator metodom željenog izlaza.
5. Generisano je po 1000 odbiraka iz dve klase koje su separabilne ali ne linearно. Generisani odbirci su prikazani. Za ove odbirke je projektovan kvadratni klasifikator.

### Linearni klasifikator

Veoma je čest slučaj u kome uslovne gustine funkcija verovatnoće klase nisu poznate. Jedno od rešenja je da se na osnovu obučavajućeg skupa one procene (npr. prvi domaći zadatak), ali to zahteva veliki broj podataka i može biti numerički veoma zahtevno. Otuda se često koriste klasifikatori koji nisu optimalni ali su numerički jednostavniji i dovoljno je poznavati kovarijacionu matricu i vektor matematičkog očekivanja za njihovo projektovanje.

Linearni klasifikator osim svoje jednostavnosti ima prednost u tome što je veoma robustan, to jest neosetljiv na promene u statističkim karakteristikama oblika koji se klasifikuju.

Ideja je da se odbirci klase projektuju na neki vektor tako da se što više odbiraka prve klase nađe ispod neke granične vrednosti, a što više odbiraka druge klase nađe iznad te vrednosti. Prema tome diskriminaciona funkcija se može zapisati kao:

$$\begin{aligned} h(X) &= V^T X + v_0 < 0 \Rightarrow X \in \omega_1 \\ h(X) &= V^T X + v_0 > 0 \Rightarrow X \in \omega_2 \end{aligned}$$

Vektor  $V$  je vektor na koji projektujemo mereni vektor  $X$ . Skalar  $v_0$  je prag sa kojim poredimo projekciju.

Ovo se može i shvatiti kao analogno nalaženju prave koja deli dve funkcije na dve oblasti tako da je svaka funkcija maksimalna u njoj odgovarajućoj oblasti kao što se radi Bajesovim pravilom u slučaju minimalne

greške za vektor koji ima dimenziju jedan. U ovom slučaju uslovne gustine verovatnoće nisu poznate, pa se uz pomoć projekcije na vektor one estimiraju na nivou jedne dimenzije. Ovo je ujedno i linearna transformacija gde transformaciona matrica praktično radi redukciju dimenzija na jedan. Odnosno, linearni klasifikator se može shvatiti kao redukcija dimenzija i primena Bayesovog pravila odlučivanja minimalne greške na dobijene slučajne vektore.

Promenljiva  $h(X)$  je takođe slučajna promenljiva, a njeno matematičko očekivanje i varijansa su date kao:

$$\begin{aligned}\eta_i &= E\{h(X)/\omega_i\} = E\{V^T X + v_0/\omega_i\} = V^T M_i + v_0 \\ \sigma_i^2 &= \text{var}\{h(X)/\omega_i\} = \text{var}\{V^T X + v_0/\omega_i\} = V^T \Sigma_i V\end{aligned}$$

### **Optimalni linearni klasifikator u slučaju normalno raspodeljenih oblika**

Kada su vektori oblika  $X$  normalno raspodeljeni ili ako je njihova dimenzija dovoljno velika da važi centralna granična teorema diskriminaciona funkcija će takođe imati normalnu raspodelu. Tada je moguće odrediti optimalni linearni klasifikator.

Verovatnoća greške za dati linearni klasifikator, gde je sa  $f(h/\omega_i)$  označena funkcija uslovne gustine verovatnoće promenljive  $h$ , je data sledećim izrazom:

$$\varepsilon = P_1 \varepsilon_1 + P_2 \varepsilon_2 = P_1 \int_0^\infty f(h/\omega_1) dh + P_2 \int_{-\infty}^0 f(h/\omega_2) dh$$

U slučaju normalne raspodele promenljive  $h$ :

$$\varepsilon = P_1 \int_0^\infty \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{1}{2}\frac{(h-\eta_1)^2}{\sigma_1^2}\right) dh + P_2 \int_{-\infty}^0 \frac{1}{\sqrt{2\pi}\sigma_2} \exp\left(-\frac{1}{2}\frac{(h-\eta_2)^2}{\sigma_2^2}\right) dh$$

Uvođenjem smena u integralima dobija se:

$$\varepsilon = P_1 \int_{-\frac{\eta_1}{\sigma_1}}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\zeta^2}{2}\right) d\zeta + P_2 \int_{-\infty}^{-\frac{\eta_2}{\sigma_2}} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\zeta^2}{2}\right) d\zeta$$

Kako je ovde moguće naći eksplisitni oblik greške, najprirodnije je iskoristiti taj izraz kao funkciju koju treba minimizovati.

Greška je eksplisitno funkcija promenljivih  $\eta_1, \eta_2, \sigma_1, \sigma_2$ , a implicitno funkcija promenljivih  $V$  i  $v_0$ . Potrebno je minimizirati ovu funkciju u zavisnosti od promenljivih  $\eta_1, \eta_2, \sigma_1, \sigma_2$ .

Pri tome je:

$$\frac{d}{d\alpha} \int_{f(\alpha)}^{g(\alpha)} h(x) dx = h(g(\alpha)) \frac{dg(\alpha)}{d\alpha} - h(f(\alpha)) \frac{df(\alpha)}{d\alpha}$$

Dobijaju se sledeći parcijalni izvodi:

$$\frac{\partial \varepsilon}{\partial \sigma_1^2} = \frac{\partial \varepsilon}{\partial (-\eta_1/\sigma_1)} \frac{\partial (-\eta_1/\sigma_1)}{\partial \sigma_1^2} = \frac{P_1}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{1}{2}\left(\frac{\eta_1}{\sigma_1}\right)^2\right) \left(-\frac{\eta_1}{\sigma_1^2}\right)$$

$$\frac{\partial \varepsilon}{\partial \sigma_2^2} = \frac{\partial \varepsilon}{\partial (-\eta_2/\sigma_2)} \frac{\partial (-\eta_2/\sigma_2)}{\partial \sigma_2^2} = \frac{P_2}{\sqrt{2\pi}\sigma_2} \exp\left(-\frac{1}{2}\left(\frac{\eta_2}{\sigma_2}\right)^2\right) \left(\frac{\eta_2}{\sigma_2^2}\right)$$

$$\frac{\partial \varepsilon}{\partial \eta_1} = \frac{\partial \varepsilon}{\partial (-\eta_1/\sigma_1)} \frac{\partial (-\eta_1/\sigma_1)}{\partial \eta_1} = \frac{P_1}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{1}{2}\left(\frac{\eta_1}{\sigma_1}\right)^2\right)$$

$$\frac{\partial \varepsilon}{\partial \eta_2} = \frac{\partial \varepsilon}{\partial (-\eta_2/\sigma_2)} \frac{\partial (-\eta_2/\sigma_2)}{\partial \eta_2} = -\frac{P_2}{\sqrt{2\pi}\sigma_2} \exp\left(-\frac{1}{2}\left(\frac{\eta_2}{\sigma_2}\right)^2\right)$$

Greška implicitno zavisi i od promenljivih  $V$  i  $v_0$ . To se može zapisati kao:

$$\frac{\partial \varepsilon}{\partial V} = \frac{\partial \varepsilon}{\partial \eta_1} \frac{\partial \eta_1}{\partial V} + \frac{\partial \varepsilon}{\partial \eta_2} \frac{\partial \eta_2}{\partial V} + \frac{\partial \varepsilon}{\partial \sigma_1^2} \frac{\partial \sigma_1^2}{\partial V} + \frac{\partial \varepsilon}{\partial \sigma_2^2} \frac{\partial \sigma_2^2}{\partial V}$$

$$\frac{\partial \varepsilon}{\partial v_0} = \frac{\partial \varepsilon}{\partial \eta_1} \frac{\partial \eta_1}{\partial v_0} + \frac{\partial \varepsilon}{\partial \eta_2} \frac{\partial \eta_2}{\partial v_0} + \frac{\partial \varepsilon}{\partial \sigma_1^2} \frac{\partial \sigma_1^2}{\partial v_0} + \frac{\partial \varepsilon}{\partial \sigma_2^2} \frac{\partial \sigma_2^2}{\partial v_0}$$

Na osnovu formula za matematičko očekivanje i varijansu slučajne promenljive  $h$  važi:

$$\frac{\partial \sigma_i^2}{\partial V} = \frac{\partial (V^T \Sigma_i V)}{\partial V} = 2\Sigma_i V$$

$$\frac{\partial \eta_i}{\partial V} = \frac{\partial (V^T M_i + v_0)}{\partial V} = M_i$$

$$\frac{\partial \sigma_i^2}{\partial v_0} = \frac{\partial (V^T \Sigma_i V)}{\partial v_0} = 0$$

$$\frac{\partial \eta_i}{\partial v_0} = \frac{\partial (V^T M_i + v_0)}{\partial v_0} = 1$$

Ove parcijalne izvode je potrebno izjednačiti sa nulom.

$$\frac{\partial \varepsilon}{\partial V} = 0 \Leftrightarrow \frac{\partial \varepsilon}{\partial \eta_1} M_1 + \frac{\partial \varepsilon}{\partial \eta_2} M_2 + \frac{\partial \varepsilon}{\partial \sigma_1^2} 2\Sigma_1 V + \frac{\partial \varepsilon}{\partial \sigma_2^2} 2\Sigma_2 V = 0$$

$$\frac{\partial \varepsilon}{\partial v_0} = 0 \Leftrightarrow \frac{\partial \varepsilon}{\partial \eta_1} + \frac{\partial \varepsilon}{\partial \eta_2} = 0$$

Zamenom vrednosti parcijalnih izvoda i daljim sračunavanjem se konačno dobija:

$$s = \frac{\partial \varepsilon / \partial \sigma_1^2}{\partial \varepsilon / \partial \sigma_1^2 + \partial \varepsilon / \partial \sigma_2^2} = \frac{-\eta_1 / \sigma_1^2}{-\eta_1 / \sigma_1^2 + \eta_2 / \sigma_2^2}$$

$$V = [s\Sigma + (1-s)\Sigma_2]^{-1}(M_2 - M_1)$$

Ukoliko odredimo  $V$  i  $v_0$  tako da su poslednje relacije zadovoljene funkcija greške će biti minimizovana. Nažalost, ovo je nije moguće eksplicitno rešiti obzirom da su  $\eta_1$  i  $\sigma_1^2$  funkcije  $V$  i  $v_0$ . Zbog toga se koristi neka od iterativnih procedura. Ipak, pogodno je pre toga odrediti zavisnost  $v_0$  od  $s$ . Koristi se sledeći izraz:

$$v_0 = -\frac{s\sigma_1^2 V^T M_2 + (1-s)\sigma_2^2 V^T M_1}{s\sigma_1^2 + (1-s)\sigma_2^2}$$

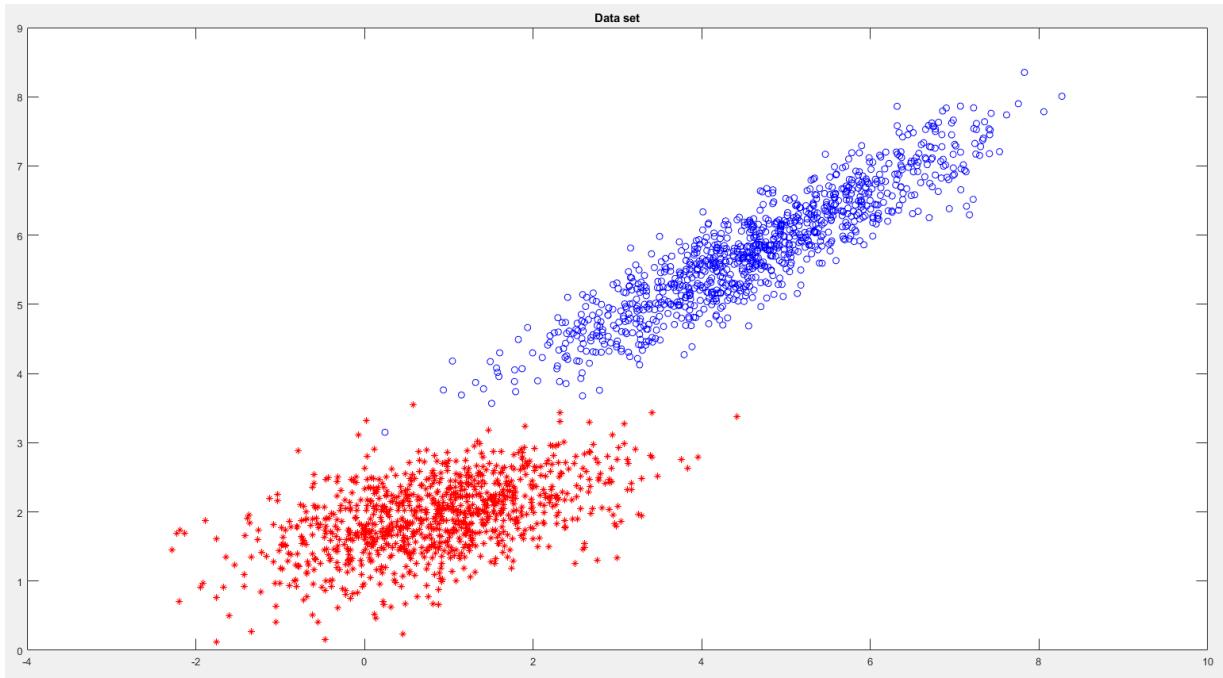
## Generisanje linearно separabilnih klasa

Projektovanje linearog klasifikatora nekom od tri iterativne procedure daje optimalni linearни klasifikator kada je diskriminaciona funkcija normalno raspodeljena (to je u slučaju kada su vektori oblika normalno raspodeljeni ili ako je njihova dimenzija dovoljno velika da važi centralna granična teorema). Tada je moguće naći optimalnu vrednost vektora i parametra takve da minimiziraju verovatnoću greške. Ova metoda se može primeniti i kada to nije slučaj, ali onda ovi parametri neće nužno biti optimalni. Zbog toga je izabранo da se generišu klase čije su raspodele normalne.

Na slici 3.1.1. su prikazani generisani odbirci. Vektori matematičkih očekivanja i kovarijacione matrice su redom:

$$M_1 = \begin{bmatrix} 0.8 \\ 1.9 \end{bmatrix}, M_2 = \begin{bmatrix} 4.7 \\ 5.8 \end{bmatrix}$$

$$S_1 = \begin{bmatrix} 1 & 0.7 \\ 0.3 & 0.5 \end{bmatrix}, S_2 = \begin{bmatrix} 1.3 & 1.5 \\ 0.8 & 1.1 \end{bmatrix}$$



*Slika 3.1.1. Generisani odbrići*

### Prva iterativna procedura

1. Sračuna se vektor  $V$  za zadatu vrednost parametra  $s$  koristeći relaciju:

$$V = [s\Sigma_1 + (1-s)\Sigma_2]^{-1}(M_2 - M_1)$$

2. Na osnovu dobijenog vektora  $V$  sračunavaju se redom sledeće statistike:

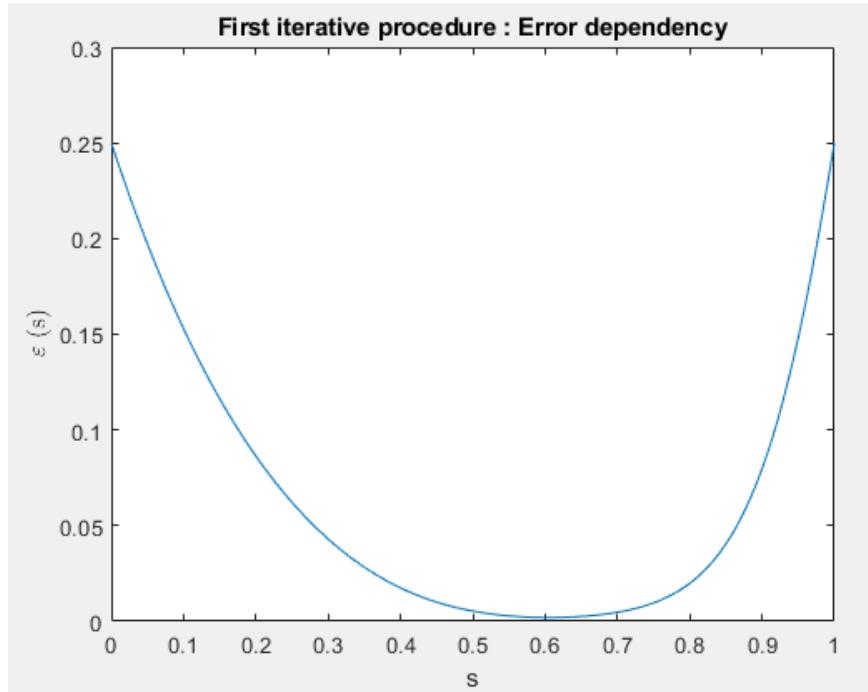
$$\begin{aligned}\sigma_1^2 &= V^T \Sigma_i V, i = 1, 2 \\ v_0 &= -\frac{s\sigma_1^2 V^T M_2 + (1-s)\sigma_2^2 V^T M_1}{s\sigma_1^2 + (1-s)\sigma_2^2} \\ \eta_i &= V^T M_i + v_0, i = 1, 2\end{aligned}$$

3. Sračuna se verovatnoća greške definisana integralima, a koja implicitno zavisi od parametra  $s$ .

$$\varepsilon(s) = P_1 \int_{-\frac{\eta_1}{\sigma_1}}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\zeta^2}{2}\right) d\zeta + P_2 \int_{-\infty}^{-\frac{\eta_2}{\sigma_2}} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\zeta^2}{2}\right) d\zeta$$

4. Parametar  $s$  se menja u intervalu  $[0, 1]$  sa korakom  $\Delta s$ , skicira se zavisnost  $\varepsilon(s)$  i odredi optimalna vrednost parametra  $s$ .

Na slici 3.2.1. se može videti zavisnost greške od parametra  $s$ .

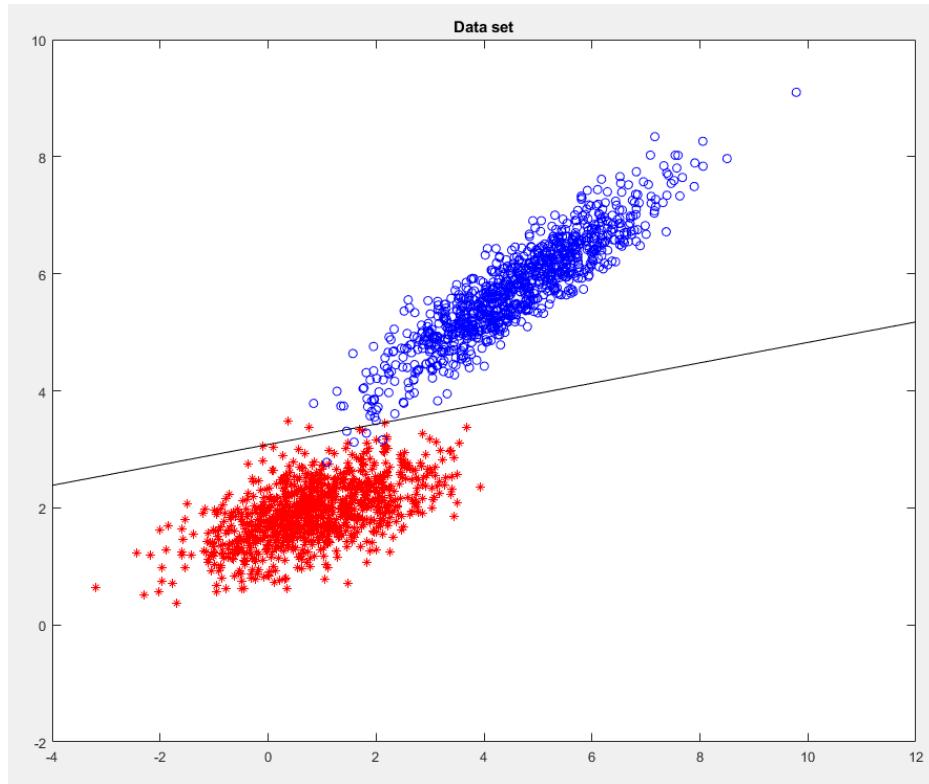


Slika 3.2.1. Zavisnost ukupne greške od parametra  $s$

Na osnovu određene zavisnosti je pronađena optimalna vrednost parametra  $s$  i na osnovu nje definisan linearni klasifikator kao:

$$h(X) = V^T X + v_0 = [-1.8801 \quad 10.7754] X - 33.2094$$

Na slici 3.2.2. je prikazan ovaj klasifikator u prostoru oblika.



### 3.2.2. Linearni klasifikator dobijen primenom prve iterativne procedure

Nedostatak ove metode je računanje integrala koje je izuzetno skupa operacija. Zbog toga se uvode druga i treća iterativna procedura koje prevazilaze ovaj problem. Dalje je izložena treća, obzirom da je gotovo ista kao druga, s tim da je razlika u tome što se podaci dele na obučavajući i testirajući skup, dok se u drugoj iterativnoj proceduri svi podaci koriste i kao „obučavajući“ i „testirajući“.

#### Druga i treća iterativna procedura

1. Podelimo raspoložive odbirke na dve grupe – obučavajući i testirajući skup (najčešće u odnosu 70:30 ili 80:20 u korist obučavajućeg skupa).
2. Na osnovu obučavajućeg skupa se izvrši procena vektora srednjih vrednosti i kovarijacionih matrica  $\hat{M}_i$  i  $\hat{\Sigma}_i$ .
3. Usvoji se vrednost za  $s$  i na osnovu toga i procenjenih vektora srednje vrednosti i kovarijacionih matrica se sračuna vrednost vektora  $V$ .

$$V = [s\hat{\Sigma}_1 + (1-s)\hat{\Sigma}_2]^{-1}(\hat{M}_2 - \hat{M}_1)$$

4. Na osnovu dobijenog vektora  $V$  sračunaju se

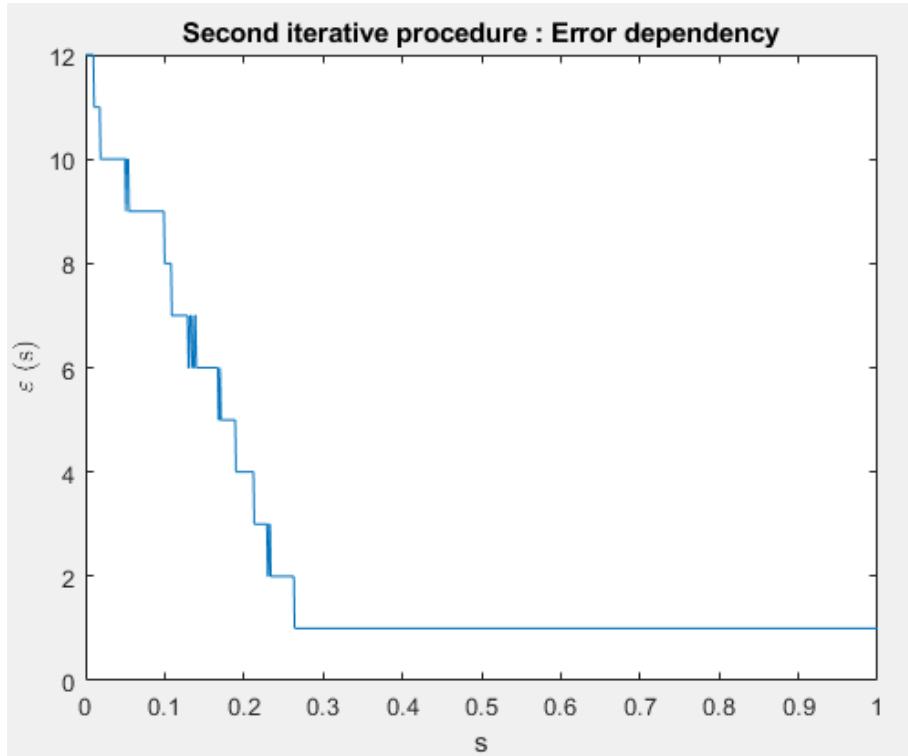
$$y_j^{(i)} = V^T X_j^{(i)} \quad i = 1, 2 \quad j = 1, \dots, N_i$$

gde je  $N_i$  broj raspoloživih oblika iz svake klase ponaosob, a  $X_j^{(i)}$  je  $j$ -ti obučavajući vektor iz  $i$ -te klase.

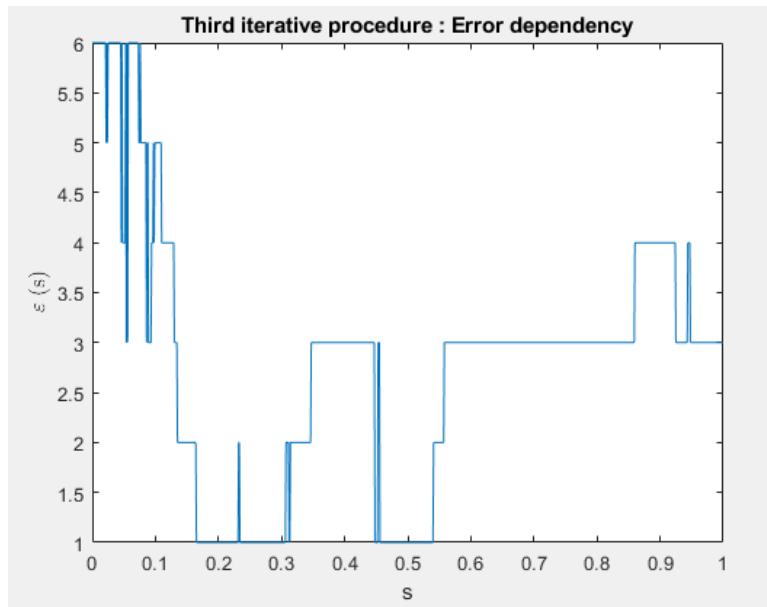
5. Oni  $y_i^{(1)}$  i  $y_i^{(2)}$  koji ne zadovoljavaju  $y_i^{(1)} < -v_0$  i  $y_i^{(2)} > v_0$  se broje kao greške, pri čemu se  $v_0$  menja u opsegu od  $-\max(\max(y_j^{(1)}), \max(y_j^{(2)}))$  do  $-\min(\min(y_j^{(1)}), \min(y_j^{(2)}))$  i pri tome se pamti ona vrednost  $v_0$  koja rezultuje najmanjim brojem grešaka. Broj grešaka se određuje na osnovu testirajućeg skupa.

5. Menja se vrednost parametra  $s$  u opsegu  $[0, 1]$  sa korakom  $\Delta s$ . Skicira se zavisnost pogrešno klasifikovanih odbiraka u zavisnosti od  $s$  i određuje optimalna vrednost parametra  $s$ .

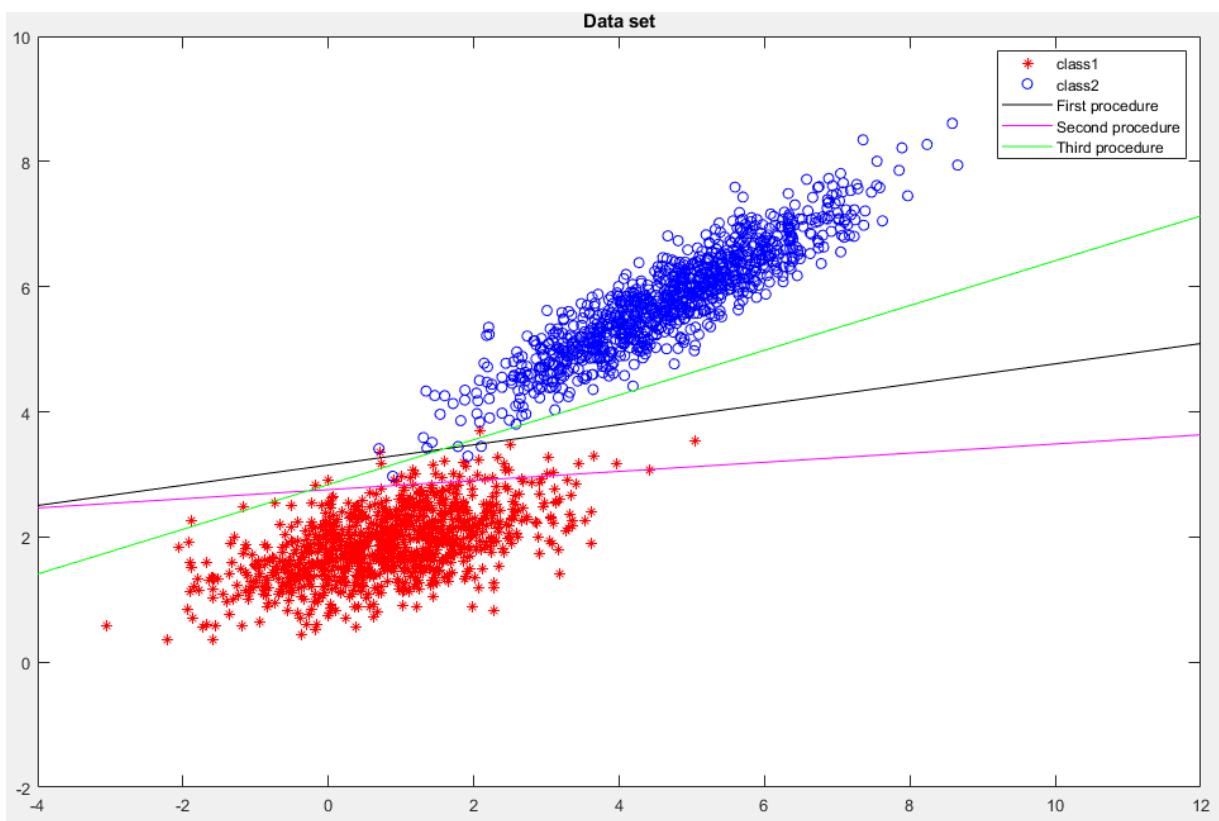
Na slici 3.2.3. je prikazana zavisnost greške od parametra  $s$  kada se primeni druga, a na slici 3.2.4. kada se primeni treća iterativna procedura. Na slici 3.2.5. se vide klasifikacione linije dobijene prvom, drugom i trećom iterativnom procedurom.



Slika 3.2.3. Zavisnost greške od parametra  $s$  dobijena primenom druge iterativne procedure

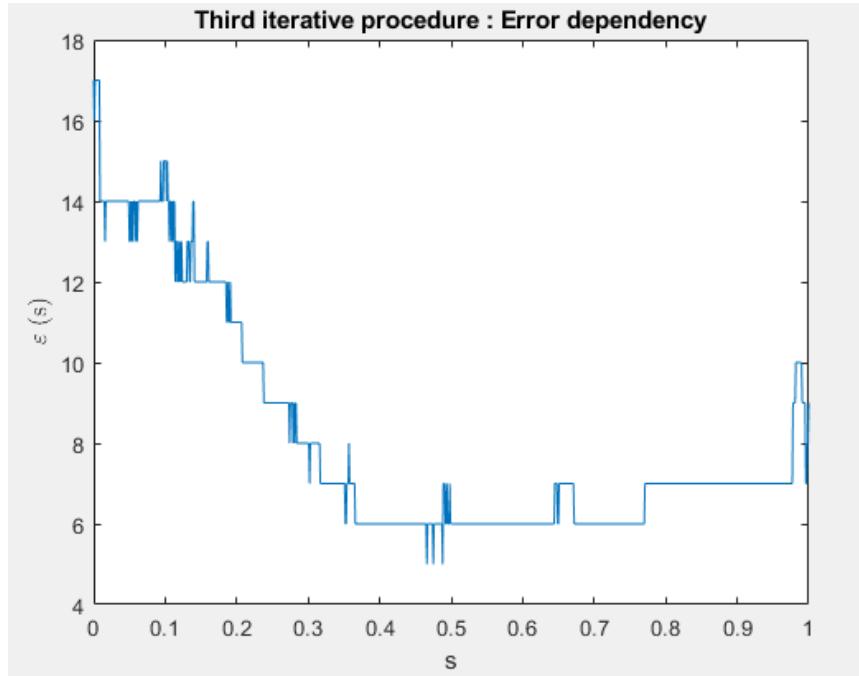


*Slika 3.2.4. Zavisnost greške od parametra  $s$  dobijena primenom treće iterativne procedure*



*Slika 3.2.5. Klasifikacione linije dobijene primenom prve, druge i treće iterativne procedure (1000 odbiraka)*

Zavisnost greške od parametra  $s$  u slučaju treće iterativne procedure nema dovoljno dobru rezoluciju kada je broj odbiraka 1000 (vidi se da velike promene slabo utiču na grešku i nema dovoljno informacija da se informisano doneše odluka o optimalnoj vrednosti parametra). Zbog toga je bilo zanimljivo videti šta se dešava kada se broj odbiraka poveća na 5000. Na slici 3.2.6. je prikazana zavisnost greške od parametra  $s$ , a na slici 3.2.7. dobijena klasifikaciona linija.



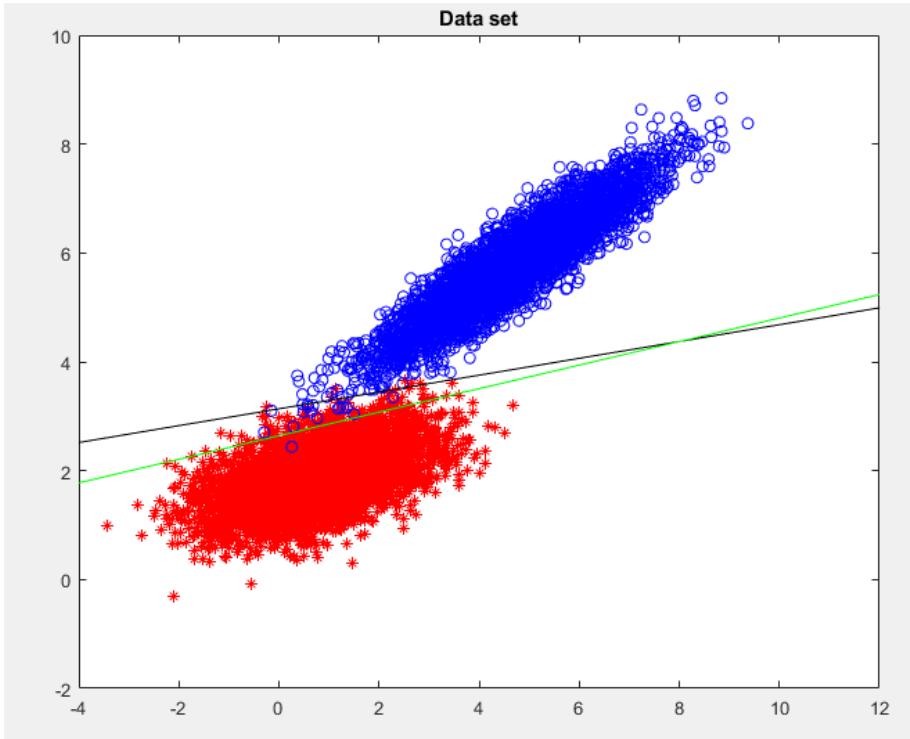
Slika 3.2.4. Zavisnost greške od parametra  $s$  primenom treće iterativne procedure (5000 odbiraka)

Nakon primene oba algoritma zaključak je da je u konkretnom slučaju prva iterativna procedura dala daleko bolje rezultate. Glavna prednost je u tome što nije neophodan veliki broj odbiraka da bi se primenila. Problem sa trećom iterativnom procedurom je u podeli skupa odbiraka na obučavajući i testirajući, pa je zbog toga potrebno mnogo više odbiraka da bi uopšte moglo da se govori od biranju optimalnog parametra.

Treba primetiti da je relativna tačnost klasifikatora dobijenih drugom i trećom procedurom posledica uzimanja prve vrednosti parametra za koju je greška minimalna. Rezultati ovih algoritama govore o tome da vrednosti koje su bliske jedinici mogu da daju malu grešku iako to nije slučaj ako pogledamo grafik dobijen primenom prve procedure.

Osim problema sa odbircima koji se može prevazići korišćenjem druge iterativne procedure (ali tada se isti odbirci koriste i za testiranje i za obučavanje, pa to nije baš opravdano), u praksi ove procedure zahtevaju više vremena nego prva. Ispostavlja se da na ovom nivou programiranja, Matlab-ova ugrađena funkcija za računanje integrala radi dovoljno brzo da prevaziđe implementaciju računanja greške na osnovu odbiraka.

Moguće da na nižem nivou programiranja računanje integrala predstavlja vremenski skupljaju operaciju, ali na ovom nivou to nije slučaj.



*Slika 3.2.7.. Klasifikacione linije dobijene prvom (crna) i trećom (zelenom) iterativnom procedurom kada se koristi 5000 odbiraka*

Konačno kada se uporede dobijene klasifikacione linije, prva procedura daje rezultate koji deluju kao bolja klasifikacija. To nije veliko iznenadenje obzirom da oblici zaista imaju normalnu raspodelu.

Na osnovu ovoga, zaključak je da je u konkretnom primeru prva iterativna procedura daleko bolja opcija nego treća ili druga.

### **Projektovanje linearog klasifikatora na bazi redukcija dimenzija**

Kao što je već pomenuto, linearni klasifikator se može shvatiti kao preslikavanje slučajne promenljive  $X$  linearном transformacijom definisanom vektorom  $V$  u jednu dimenziju, gde je  $v_0$  granica koja deli funkcije gustine verovatnoće novodobijenog slučajnog vektora. Zato je bilo zanimljivo proveriti šta se dešava kada projektujemo linearni klasifikator na bazi redukcija dimenzija i to na bazi kriterijuma rasipanja.

Definišimo matricu unutar-klasnog rasejanja kao:

$$S_w = P_1 \Sigma_1 + P_2 \Sigma_2$$

i matricu međuklasnog rasejanja kao:

$$S_B = P_1(M_1 - M_0)(M_1 - M_0)^T + P_2(M_2 - M_0)(M_2 - M_0)^T$$

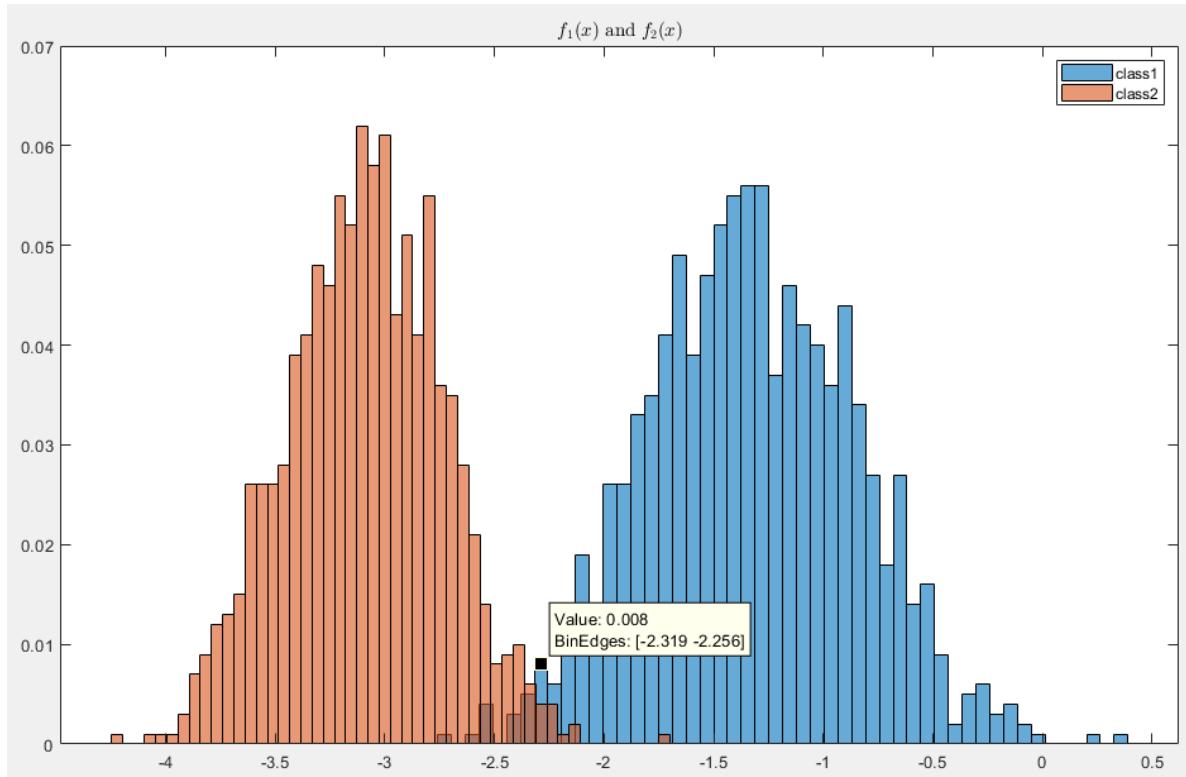
gde je  $M_0$  združeni vektor matematičkog očekivanja za obe klase zajedno:

$$M_0 = P_1M_1 + P_2M_2$$

Tada se  $V$  može uzeti kao sopstveni vektor matrice  $S_W^{-1}S_B$ , takav da mu odgovara maksimalna sopstvena vrednost iste matrice.

Na slici 3.3.1. su prikazani histogrami, odnosno funkcije gustina verovatnoća vektora  $Y$  dobijenog primenom linearne transformacije:

$$Y = V^T X$$



*Slika 3.3.1. Funkcije gustine verovatnoća nakon redukcije na jednu dimenziju*

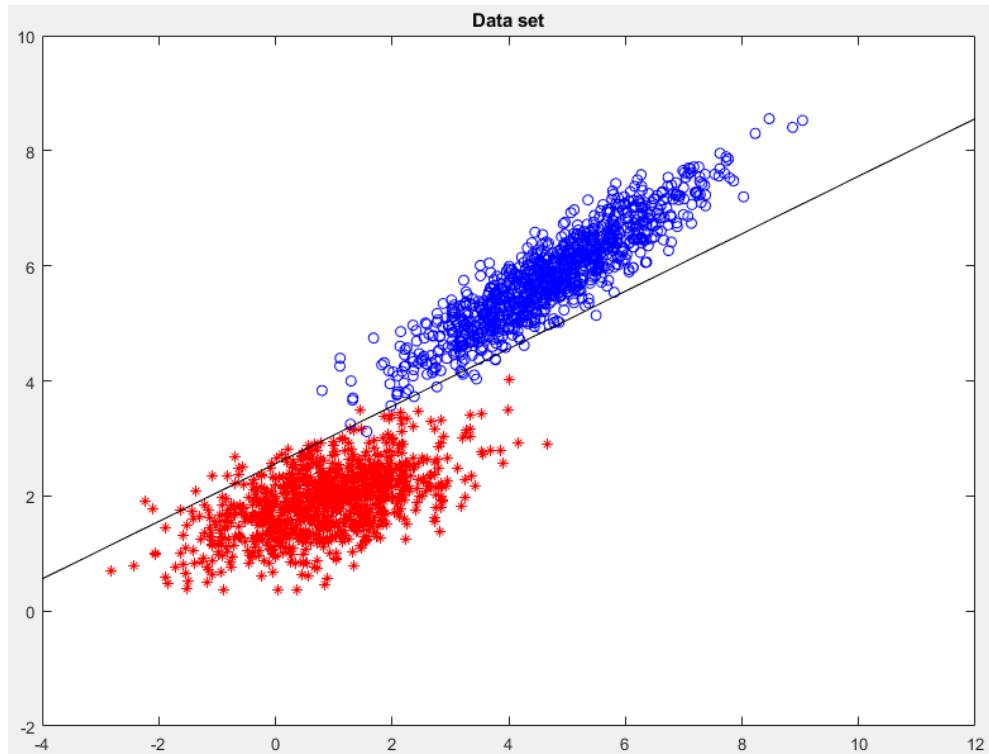
Na osnovu histograma možemo naći vrednost parametra  $v_0$  takvu da sa minimalnom greškom klasifikujemo slučajan vektor  $Y$  kao granicu gde postaje  $f_1(Y) > f_2(Y)$ . Tada se linearni klasifikator može definisati kao:

$$h(X) = V^T X - v_0 < 0 \Rightarrow X \in \omega_1$$

$$h(X) = V^T X - v_0 > 0 \Rightarrow X \in \omega_2$$

Primetimo da stoji  $-v_0$  jer za odbirke prve klase važi  $f_1(Y) > v_0$ .

Na slici 3.3.2. je prikazan dobijen klasifikator. Primećuje se da je rezultat najsličniji onom koji se dobije primenom treće iterativne procedure (s tim da je ovo brže rešenje i zahteva manje odbiraka).



*Slika 3.3.2. Linearni klasifikator dobijen primenom redukcije dimenzija na 1*

### Projektovanje linearnog klasifikatora na bazi željenog izlaza

Usvojimo linearni klasifikator oblika:

$$h(X) = -V^T X - v_0 > 0 \Rightarrow X \in \omega_1$$

$$h(X) = V^T X + v_0 > 0 \Rightarrow X \in \omega_2$$

i uvedimo novi vektor  $Z$  kojim se predstavljaju oblici:

$$Z = \begin{bmatrix} -1 & -X_1 & -X_2 & \dots & -X_n \end{bmatrix}^T; X \in \omega_1$$

$$Z = \begin{bmatrix} 1 & X_1 & X_2 & \dots & X_n \end{bmatrix}^T; X \in \omega_2$$

Tada se diskriminaciona funkcija može napisati u obliku:

$$h(Z) = W^T Z = \sum_{i=0}^n w_i z_i > 0$$

a projektovanje klasifikatora se svodi na sledeća dva koraka:

1. Generisati vektore  $Z$  na osnovu oblika  $X$  iz klase  $\omega_1$  i  $\omega_2$  na osnovu gore definisane relacije.
2. Odrediti vektor  $W$  tako da je  $W^T Z > 0$  za što je moguće više vektora  $Z$ .

Uvedimo i takozvani željeni izlaz  $\gamma(Z)$  koji pridružujemo svakom semplu  $Z$ , pri čemu u ovom slučaju ovaj željeni izlaz poseduje ograničenje  $\gamma(Z) > 0$ . Izbor vektora  $W$  se vrši minimizacijom unapred postavljene funkcije koja insistira na tome da diskriminaciona funkcija  $h(Z)$  generiše vrednosti koje treba da budu što bliže želenom izlazu  $\gamma(Z)$ . Uobičajeni oblici kriterijumske funkcije su:

$$\bar{\varepsilon^2} = \frac{1}{N} \sum_{j=1}^N [W^T Z_j - |W^T Z_j|]^2$$

$$\bar{\varepsilon^2} = \frac{1}{N} \sum_{j=1}^N [\text{sgn}(W^T Z_j) - 1]^2$$

$$\bar{\varepsilon^2} = \frac{1}{N} \sum_{j=1}^N [W^T Z_j - \gamma(Z_j)]^2$$

gde je  $N$  broj semplova. Ukoliko koristimo treću kriterijumsku funkciju, optimalni vektor  $W$  se može eksplicitno odrediti.

Definišimo  $U = [Z_1 Z_2 Z_N]$  i  $\Gamma = [\gamma(Z_1) \gamma(Z_2) \gamma(Z_N)]^T$ . Tada se kriterijumska funkcija može pisati kao:

$$\bar{\varepsilon^2} = \frac{1}{N} \sum_{j=1}^N [W^T Z_j - \gamma(Z_j)]^2 = \frac{1}{N} (U^T W - \Gamma)(U^T W - \Gamma)$$

Pa parcijalni izvod kriterijumske funkcije po vektoru  $W$  dobija oblik:

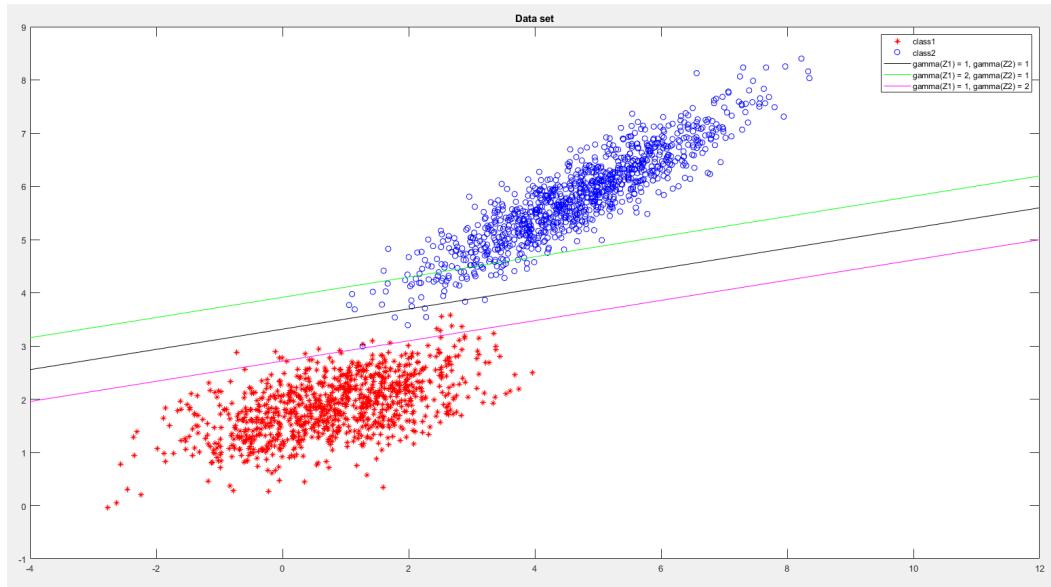
$$\frac{\partial \varepsilon^2}{\partial W} = \frac{2}{N} U(U^T W - \Gamma)$$

Izjednačavanjem parcijalnog izvoda sa nulom dobijamo eksplicitno rešenje za optimalni vektor  $W$ :

$$W = (UU^T)^{-1}\Gamma$$

Konačno, napomenimo da se izborom  $\gamma(Z_i)$  može birati značaj odbirka  $Z_i$ . U ovoj implementaciji je izabrano da svi odbirci imaju istu težinu, odnosno  $\gamma(Z_i) = 1$  za svako  $i$ .

Na slici 3.4.1. je prikazana klasifikaciona linija dobijena metodom željenih izlaza. Crnom bojom je prikazana klasifikaciona linija koja se dobije kada svi odbirci imaju istu težinu, zelenom kada se prednost dà odbircima prve klase, a roze linija koja se dobije kada se odbricima druge klase dà veća težina.



Slika 3.4.1. Klasifikaciona linija dobijena metodom željenih izlaza

Sa slike se lepo vidi kako postavljanje veće težine jednoj klasi smanjuje taj tip greške, a povećava drugi tip, odnosno kako promenom vrednosti u matrici  $\Gamma$  mogu da se kontrolišu greške prvog i drugog tipa. Dakle, kao i uvek, projektovanje zahteva da se napravi neki kompromis.

### Projektovanje kvadratnog klasifikatora

U uslovima kada klase nisu linearno separabilne, potrebno je koristiti složenije forme klasifikatora. Jedan od njih je kvadratni klasifikator. Njegova opšta forma je:

$$h(X) = X^T Q X + V^T X + v_0 < 0 \Rightarrow X \in \omega_1$$

$$h(X) = X^T Q X + V^T X + v_0 > 0 \Rightarrow X \in \omega_2$$

Ako raspakujemo gornju jednačinu vidimo da se ovaj klasifikator zapravo može linearizovati:

$$h(X) = \sum_{i=1}^n \sum_{j=1}^n q_{ij} x_i x_j + \sum_{i=1}^n v_i x_i + v_0 = \sum_{i=1}^{\frac{n(n+1)}{2}} \alpha_i y_i + \sum_{i=1}^n v_i x_i + v_0$$

gde su  $q_{ij}$  i  $v_i$  elementi matrice  $Q$  i vektora  $V$ , dok su sa  $y_i$  označeni proizvodi oblika  $x_j x_k$ .

Možemo definisati vektor  $Z$  kao  $Z = [Y^T \quad X^T]^T$  i gornji problem svesti na problem projektovanja linearog klasifikatora za slučajni vektor  $Z$ .

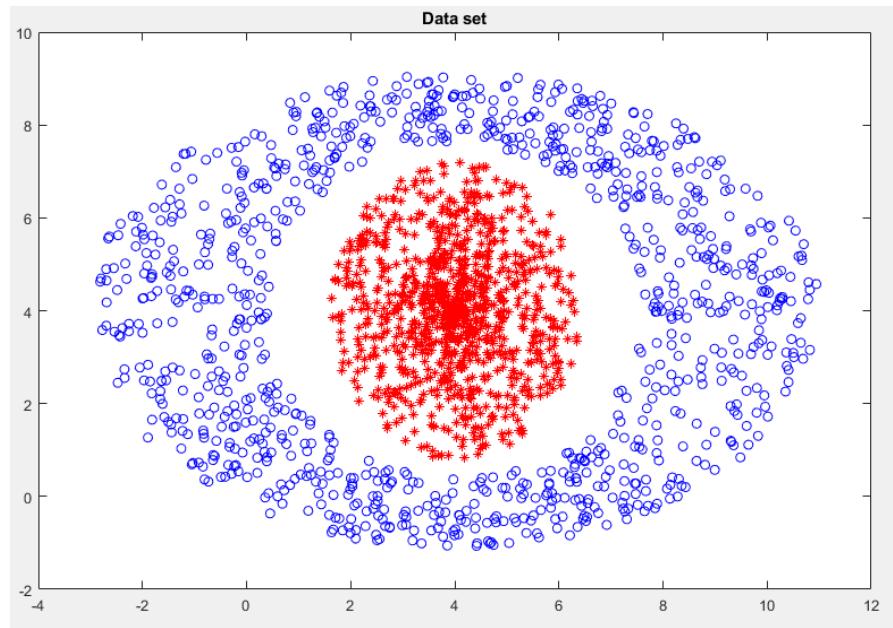
Tada će vektor  $V$  imati oblik:

$$V = [\alpha_1 \quad \alpha_2 \quad \dots \quad \alpha_{n(n+1)/2} \quad v_1 \quad \dots \quad v_n]^T$$

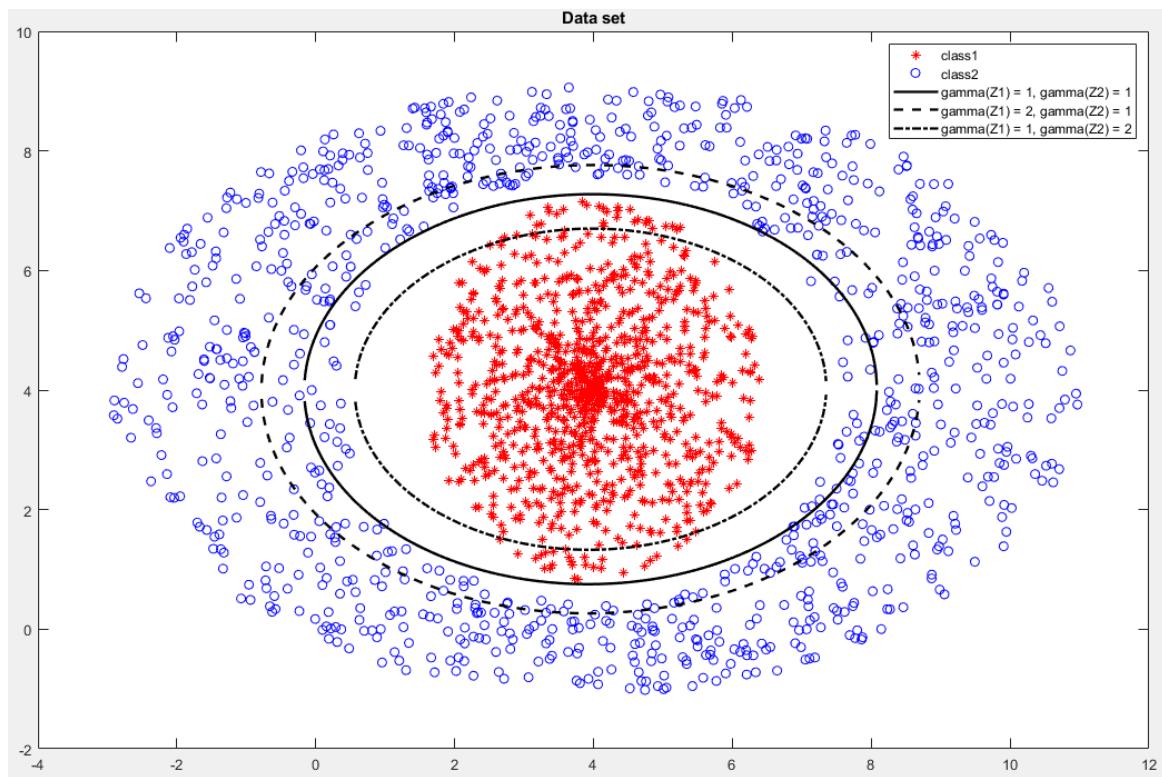
Sada se može se primeniti neka od gotovih procedura za projektovanje linearog klasifikatora. Ovde je konkretno izabran metod željenog izlaza obzirom da je najjednostavniji.

Izabrano je da oblici jedne klase imaju oblik elipse sa centrom u tački  $(4, 4)$  i  $a = 2.4$ ,  $b = 3.2$ . Druga klasa ima oblik elipse sa centrom u istoj tački, parametrima  $a = 7$ ,  $b = 5$  i rupom u vidu kruga sa centrom u istoj tački, poluprečnika  $r = 3.6$ . Na slici 3.5.1. su prikazani generisani odbirci. Generisano je 1000 odbiraka iz svake klase.

Klasifikator je kvadratni, određen metodom željenih izlaza, prikazano je kako se rezultati menjaju u zavisnosti od težina u matrici  $\Gamma$ . Na slici 3.5.2. je prikazan ovaj klasifikator.



Slika 3.5.1. Generisani odbirci



Slika 3.5.2. Generisani odbirci i kvadratni klasifikator dobijen metodom željenih izlaza

## Četvrti domaći zadatak – opis problema

Potrebno je generisati po 500 odbiraka iz četiri klase koje će biti linearно separabilne. Izabratи jednu od metoda za klasterizaciju (c mean metod, metod kvadratne dekompozicije, metod maksimalne verodostojnosti ili metod grana i granica) i primeniti je na formirane uzorke klasa. Izvršiti analizu osetljivosti izabranog algoritma na početnu klasterizaciju kao i srednji broj potrebnih iteracija. Takođe izvršiti analize slučaja kada se prije ne poznaje broj klasa.

Generisati po 500 dvodimenzionih odbiraka iz dve klase koje su nelinearno separabilne. Izabratи jednu od metoda za klasterizaciju koje su primenjive za nelinearno separabilne klase (metod kvadratne dekompozicije ili metod maksimalne verodostojnosti) i ponoviti analizu iz prethodne tačke.

## Četvrti domaći zadatak – opis rešenja

Rešenje prati sledeći algoritam:

1. Generisano je po 500 odbiraka svake klase. Sve klase imaju normalnu raspodelu, ali su linearne separabilne. Prikazani su generisani odbirci.
2. Projektovan je c mean algoritam klasterizacije i primenjen na generisane odbirke.
3. Na istim odbircima je analiziran prethodni algoritam za slučajeve kada nije unapred poznat broj klasa.
4. Generisano je po 500 odbiraka iz dve klase koje su separabilne ali ne linearne. Generisani odbirci su prikazani.
5. Projektovan je klasifikator na bazi kvadratne dekompozicije.
6. Prethodni klasifikator je analiziran u slučajevima kada broj klasa nije unapred poznat.
7. Poređena su oba klasifikatora na osnovu broja potrebnih iteracija.

### C-mean algoritam klasterizacije

Klasterizacija se može posmatrati kao tehnika za grupisanje semplova koji će maksimizirati separabilnost klasa. Prepostavimo da želimo da klasifikujemo  $N$  semplova  $X_1, X_2, \dots, X_N$ . Za ove vektore se ne prepostavlja da su slučajni, već da su fiksni i poznati. Svaki od semplova treba klasifikovati u jednu od  $L$  klase  $(\omega_1, \dots, \omega_L)$ , gde se smatra da je  $L$  dato. Označimo klasu kojoj je dodeljen sempl  $X_i$  sa  $\omega_{k_i}$  gde  $i = 1, \dots, N$ . Klasifikacija  $\Omega$  je vektor  $\omega_{k_i}$ -ova koji su pridruženi konfiguraciji  $X^* : \Omega = [\omega_{k_1} \ \omega_{k_2} \ \dots \ \omega_{k_N}]^T$ ;  $X^* : \Omega = [X_1^T \ X_2^T \ \dots \ X_N^T]^T$ .

Klasterizacioni kriterijum  $J$  je funkcija  $\Omega$  i  $X^*$  i može se pisati:

$$J = J(\omega_{k_1}, \dots, \omega_{k_N}; X_1, \dots, X_N) = J(\Omega, X^*)$$

Kao kriterijumi se mogu koristiti oni bazirani na matricama unutarskog i međuklasnog rasejanja. Neka je dato  $L$  klasa pri čemu je svakoj od klase pridružen vektor matematičkog očekivanja  $M_i$  i kovarijaciona matrica  $\Sigma_i$ .

Tada se matrica unutarklasnog rasejanja definiše kao:

$$S_W = \sum_{i=1}^L P_i E \left\{ (X - M_i)(X - M_i)^T / \omega_i \right\} = \sum_{i=1}^L P_i \Sigma_i$$

Matrica međuklasnog rasejanja se definiše kao:

$$S_B = \sum_{i=1}^L P_i (M_i - M_0)(M_i - M_0)^T$$

gde je  $M_0$  združeni vektor matematičkog očekivanja za sve klase zajedno:

$$M_0 = E \{ X \} = \sum_{i=1}^L P_i M_i$$

Još se uvodi i miksovana matrica rasejanja:

$$S_M = E \{ (X - M_0)(X - M_0)^T \} = S_W + S_B$$

Algoritam klasterovanja u opštem slučaju glasi:

Neka je u  $l$ -toj iteraciji  $\Omega(l) = [\omega_{k_1(l)} \dots \omega_{k_N(l)}]^T$ . Ako se  $i$ -ti sempl reklassificuje iz  $k_i(l)$ -te klase u  $j$ -tu klasu, klasterizacioni kriterijum će doživeti priraštaj:

$$\Delta J(i, j, l) = J(\omega_{k_1(l)}, \dots, \omega_j, \dots, \omega_{k_N(l)}, X^*) - J(\Omega(l), X^*)$$

Ako je ovaj priraštaj negativan, jasno da reklassifikacija  $i$ -tog sempla smanjuje kriterijum i poboljšava klasterizaciju. Algoritam se može sistematizovati kroz sledeće korake:

1. Izabere se inicijalna klasifikacija  $\Omega(0)$ .
2. Za  $l$ -tu klasifikaciju  $\Omega(l)$  sračuna se  $\Delta J(i, j, l)$  za  $j = 1, \dots, L$  i  $i = 1, \dots, N$ .
3. Za  $i = 1, \dots, N$  izvrši se reklassifikacija  $i$ -tog sempla u klasu  $t$ , gde je

$$\Delta J(i, t, l) = \min_j (i, j, l)$$

Na taj način se formira nova klasifikacija  $\Omega(l+1)$ .

4. Ako je  $\Omega(l+1) \neq \Omega(l)$  ponavlja se korak 2, u protivnom se klasterizacija završava.

Dobar kriterijum za klasterizaciju je onaj koji omogućava isti rezultat nezavisno od toga u kom se koordinatnom sistemu nalazimo. Kriterijum  $\text{tr}(S_M^{-1} S_W)$  zadovoljava ovaj uslov.

Bez gubitka opštosti usvojimo da je  $M_0 = 0$  i  $S_M = I$ , tada se kriterijum može zapisati u sledećem obliku:

$$\begin{aligned} J = \text{tr} \{S_W\} &= \sum_{r=1}^L \frac{N_r}{N} \frac{1}{N} \sum_{j=1}^{N_r} (X_j^{(r)} - M_r)^T (X_j^{(r)} - M_r) \\ &= \frac{1}{N} \sum_{r=1}^L \sum_{j=1}^{N_r} \|X_j^{(r)} - M_r\|^2 \end{aligned}$$

Ako izvršimo reklassifikaciju sempla  $X_i$  iz klase  $K_i$  u  $j$ -tu klasu tokom  $l$ -te iteracije, priraštaj kriterijuma postaje:

$$\Delta J(i, j, k) = \frac{1}{N} (\|X_i - M_j(l)\|^2 - \|X_i - M_{k_i}(l)\|^2)$$

Kako je drugi član nezavistan od  $j$ , reklassifikacija sempla  $X_j$  u  $l$ -toj iteraciji je određena na sledeći način:

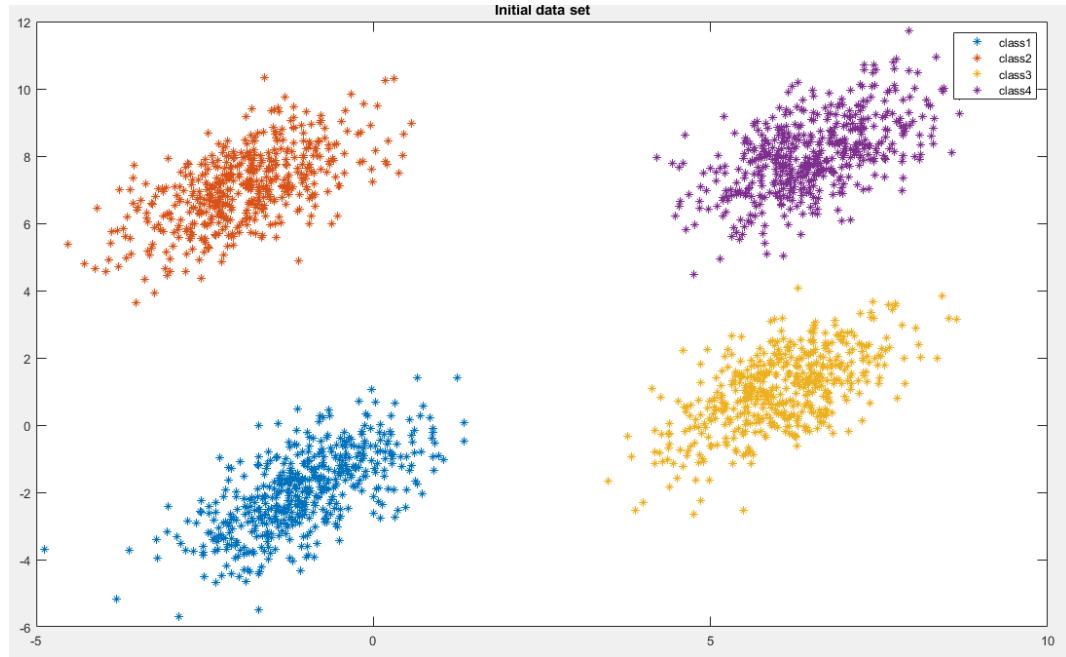
$$\|X_i - M_t(l)\| = \min_j \|X_i - M_j(l)\| \Rightarrow X \in \omega_t$$

Algoritam se može formalizovati na sledeći način:

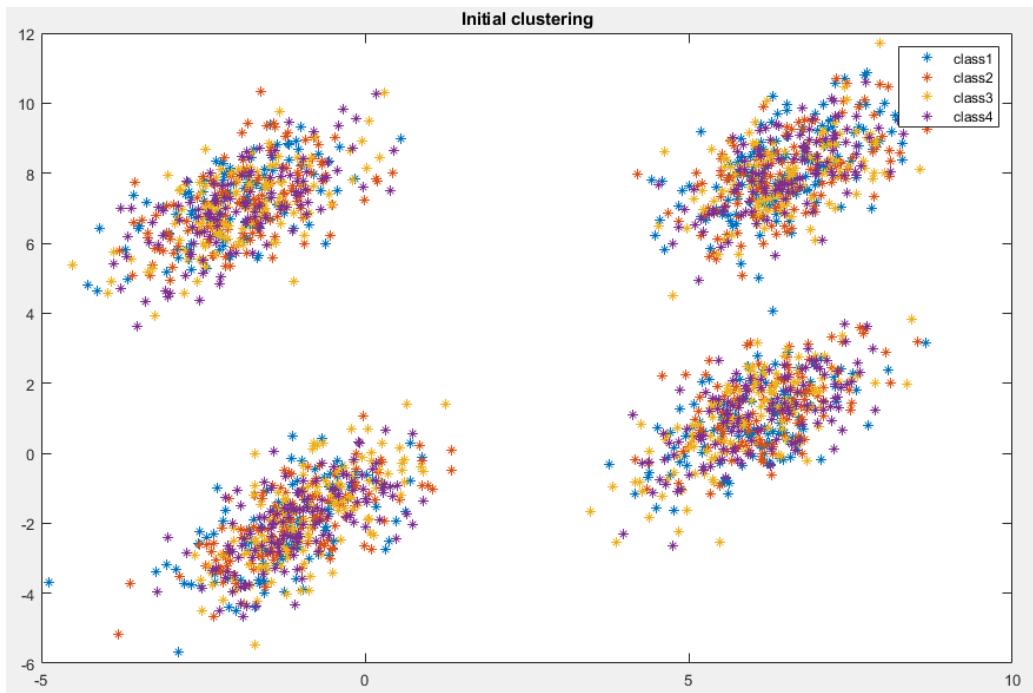
1. Izabere se inicijalna klasifikacija  $\Omega(0)$  i sračuna  $M_1(0), M_2(0), \dots, M_L(0)$ .
2. Na osnovu sračunatih  $M_1(l), \dots, M_L(l)$  u  $l$ -toj iteraciji reklassifikacija za svaki sempl  $X_i$  se vrši prema najbližem vektoru srednjih vrednosti  $M_j(l)$ .
3. Ako je bar neki sempl  $X_i$  reklassifikovan u  $l$ -toj iteraciji ulazi se u novu  $l + 1$ -u iteraciju, i vraća se na korak 2. Ukoliko u tekućoj iteraciji nijedan sempl nije reklassifikovan, algoritam se završava.

Na slici 4.1.1. su prikazani generisani odbirci. Na slici 4.1.2. se može videti početna inicijalizacija, dok se na slikama 4.1.3. i 4.1.4. mogu pratiti rezultati svake iteracije i konačan rezultat. Broj iteracija je 3. Kao što se vidi iz izvođenja, neophodno je poznavati broj klastera. Isto tako, samo srednje vrednosti, odnosno vektori matematičkih očekivanja utiču na formiranje granica, dok to nije slučaj sa kovarijacionim matricama. Početna klasterizacija se može uzeti stohastički (i to je urađeno u konkretnom primeru). Problem nalaženja optimalnih klastera je NP složen, stoga se primenjuje varijanta opisana gore (gde se u jednoj iteraciji reklassificuje više semplova). Ovaj algoritam garantuje konvergenciju, ali ne i optimalno rešenje. Algoritam se završava u jednom od lokalnih minimuma. Zanimljivo da je u jednom trenutku

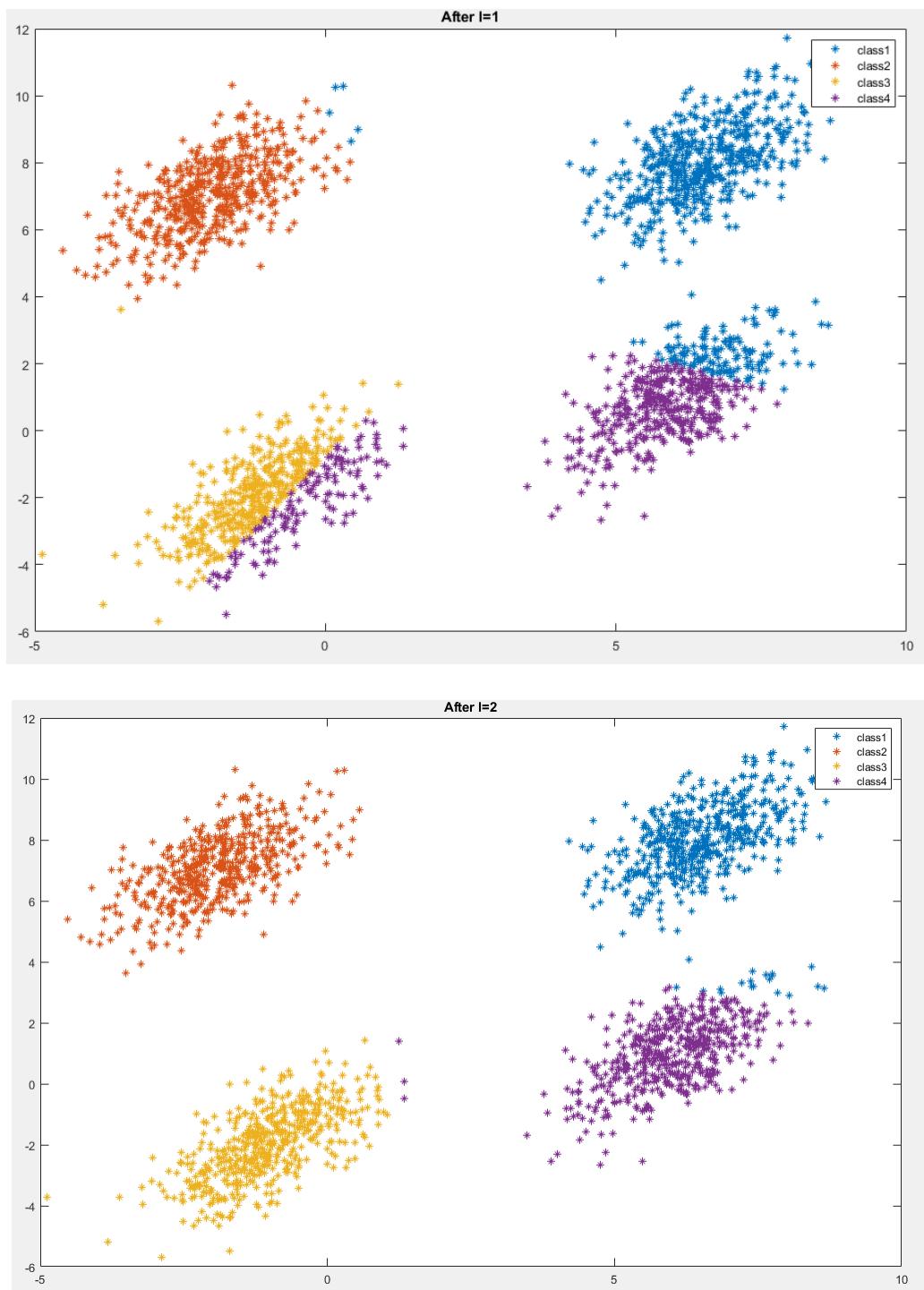
program dobio klasterizaciju koja je odbirke podelila na samo tri klase. Dakle, kao i sa većinom algoritama koji ne garantuju optimalno rešenje, već samo lokalni ekstremum, poželjno je pustiti program nekoliko puta.



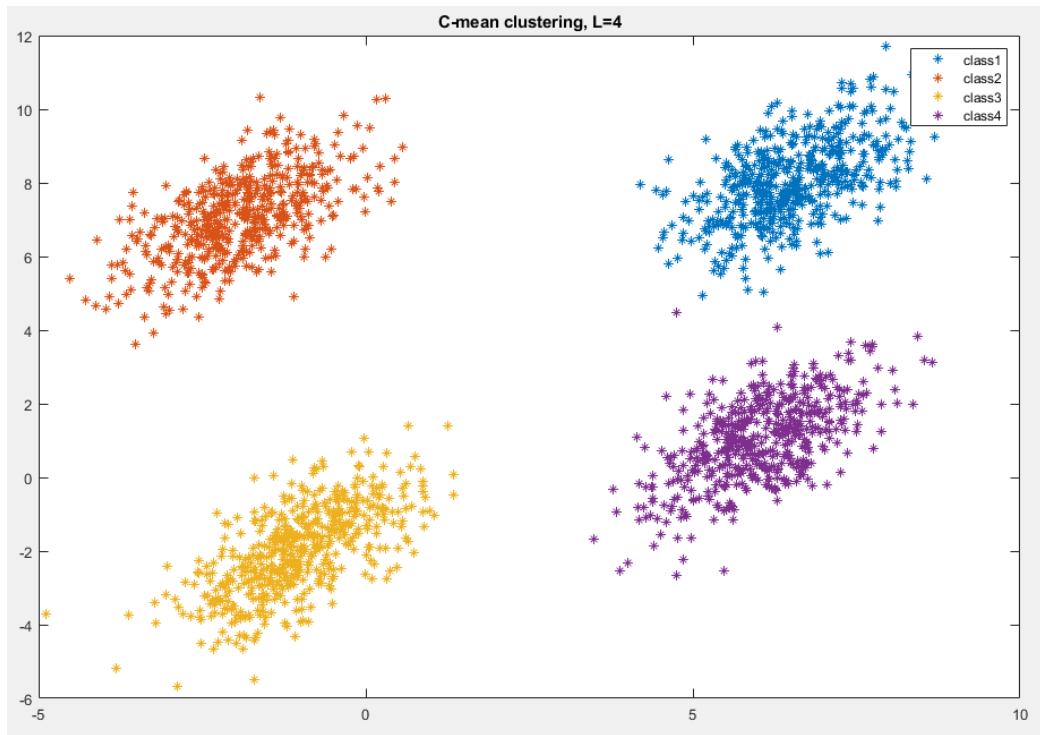
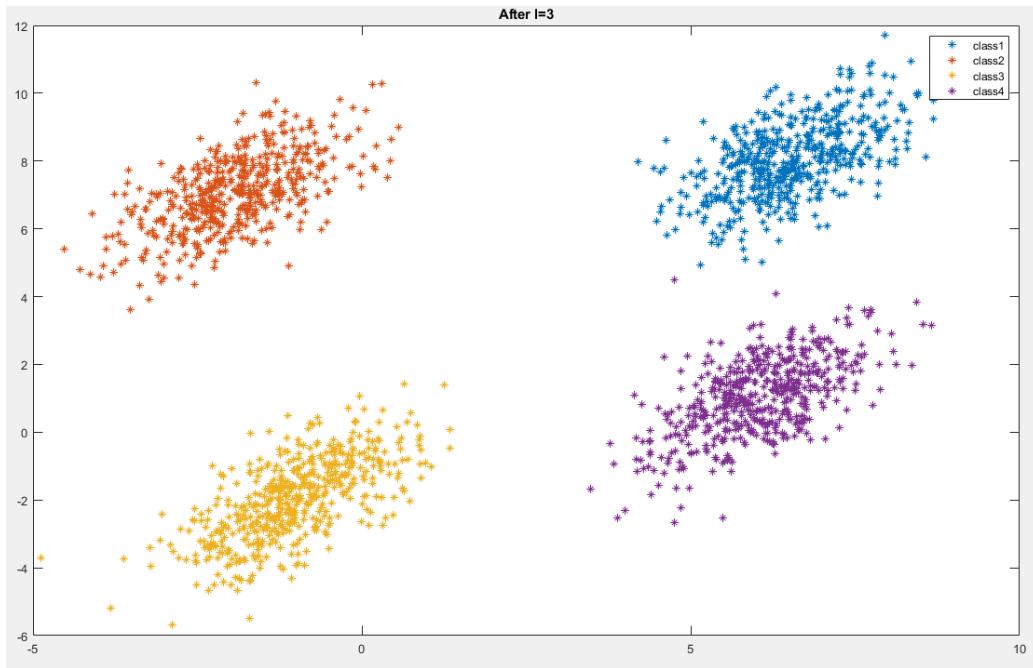
*Slika 4.1.1. Generisani odbirci*



*Slika 4.1.2. Inicijalna klasterizacija*



Slika 4.1.3. Klasterizacija nakon prve i druge iteracije



*Slika 4.1.4. Klasterizacija nakon treće iteracije i finalni rezultat*

Nakon izvršavanja programa 20 puta, zaključak je da je u proseku potrebno 6.5 iteracija da bi se došlo do rešenja. Klasterizacija je uglavnom veoma dobra, greške su reda jednog odbirka, ali u pojedinim

slučajevima se dešava da se odbirci klasifikuju u tri umesto četiri klase. Obzirom da je program jako brz i da unapred znamo broj klasa koji postoji, ovo nije tako veliki problem, jer prosto možemo da ponovimo postupak.

Problem koji se javlja je svojstvo algoritama koji rade po principu rasta/opadanja gradijenta, ovakvi algoritmi ne mogu da vide dalje od svoje okoline (na primer, nemoguće je doći do maksimuma ukoliko je potrebno da se preskoči deo gde vrednost gradijenta opada). Ovakvi problemi se rešavaju višestrukim puštanjem programa ili pametnje izabranim inicijalizacijama. Konkretno, za *c-mean* metodu se koriste različite inicijalizacije. Neke su bazirane na tome da se početna matematička očekivanja nalaze što bliže centru svih podataka (*Random Partition Initialization*), a neke kao *Froggy* metoda na tome da su centri klastera u početnoj iteraciji što rasejaniji. Konačan odgovor na to koja vrsta inicijalizacije je bolja ne postoji. Ovde je korišćena *random* funkcija za inicijalizaciju.

Osim gore pomenutih rešenja uvek se može izabrati opcija u kojoj se reklasificuje samo jedan sempl, i tada je garantovano da će algoritam konvergirati ka optimalnoj vrednosti, ali se to plaća ogromnim vremenom izvršavanja programa. Kao kod sličnih NP problema, poput problema trgovackog putnika, moguće je ograničiti broj opcija koje se pretražuju uz pomoć pametno osmišljenih stabla pretraživanja. Konkretno, za datu kriterijumsku funkciju se može koristiti *branch and bound* algoritam koji može značajno da redukuje graf pretraživanja ako je kriterijum koji se minimizuje monoton (što ovde jeste slučaj).

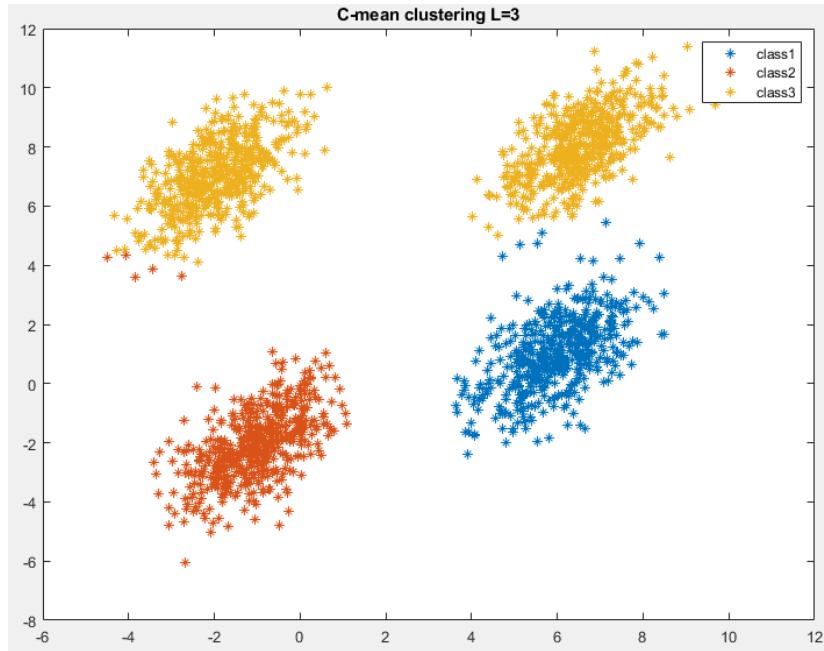
### **Primena *c-mean* algoritma u slučajevima kada broj klasa nije unapred poznat**

Kriterijumska funkcija direktno zavisi od broja klasa  $L$ . Nažalost, kriterijumska funkcija korišćena za *c-mean* klasterizaciju je takva da monotno opada sa povećanjem broja klasa (sve dok ne postane nula jer je svaki odbirak posebna klasa i samim tim sva odstojanja od srednjih vrednosti nula). Dakle, potrebno je na neki način ograničiti algoritam.

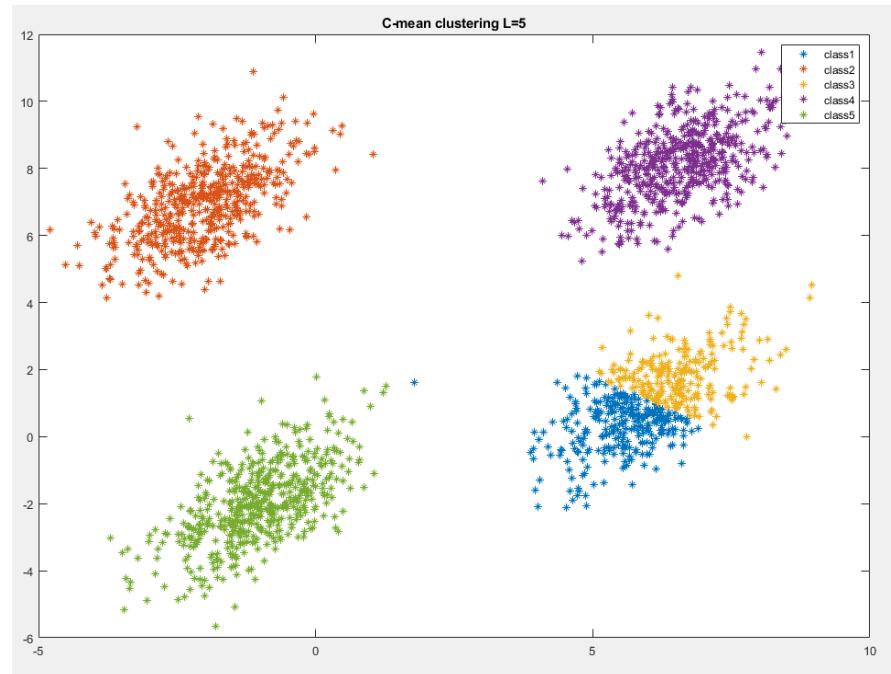
U ovom primeru je to urađeno tako što je postavljen početan broj klasa na dva. Algoritam se izvršava dva puta i ako oba puta završi u lokalnom minimumu takvom da je klasifikovao manje klasa nego što je zadato smatra se da je to jer je stvarno smisleno podeliti podatke u taj broj klasa. Rezultati su zadovoljavajući, algoritam se uglavnom zaustavlja na tome da postoje 4 klase podataka, ređe 3 ili 5. Program nije prepostavio da postoji više od 5 klasa, što ne isključuje da to može da se desi.

Ovo rešenje je prilično jednostavno i brzo, a ne daje loše rezultate. Svakako pruža dobar uvid u to koliko klasa može da postoji, pa može da bude dobar način za restrikciju potencijalnog broja klasa.

Na slici 4.1.5. je prikazana klasterizacija kada se pretpostavi da postoje 3, a na slici 4.1.6. se vide rezultati kada je pretpostavka da postoji 5 klasa.



*Slika 4.1.5. Rezultati primene c-mean algoritma kada je prepostavljeno da se oblici mogu podeliti na tri klase*



*Slika 4.1.6. Rezultati primene c-mean algoritma kada je prepostavka da se oblici mogu podeliti na pet klasa*

## Projektovanje klasifikatora na bazi metode kvadratne dekompozicije

Prednost ovog algoritma je u tome što su granice između klastera deo po deo kvadratne krive, stoga se može koristiti u slučajevima kada oblici nisu linearno separabilini.

Algoritam se može formalizovati na sledeći način:

1. Izabere se inicijalna klasterizacija  $\Omega(0)$  i na osnovu nje procene apriorne verovatnoće pojave klasa  $P_i(0)$ , vektori srednjih vrednosti klasa  $M_i(0)$  i kovarijacione matrice za pojedine klase  $\Sigma_i(0)$ ,  $i = 1, 2, \dots, L$ .
2. U  $t$ -toj iteraciji svaki sempl  $X_j$  se reklasificuje prema sledećem izrazu:

$$\frac{1}{2}(X_j - M_t(l))^T \Sigma_t^{-1}(l)(X_j - M_t(l)) - \frac{1}{2} \log |\Sigma_t(l)| + \frac{1}{2} \log P_t(l)$$

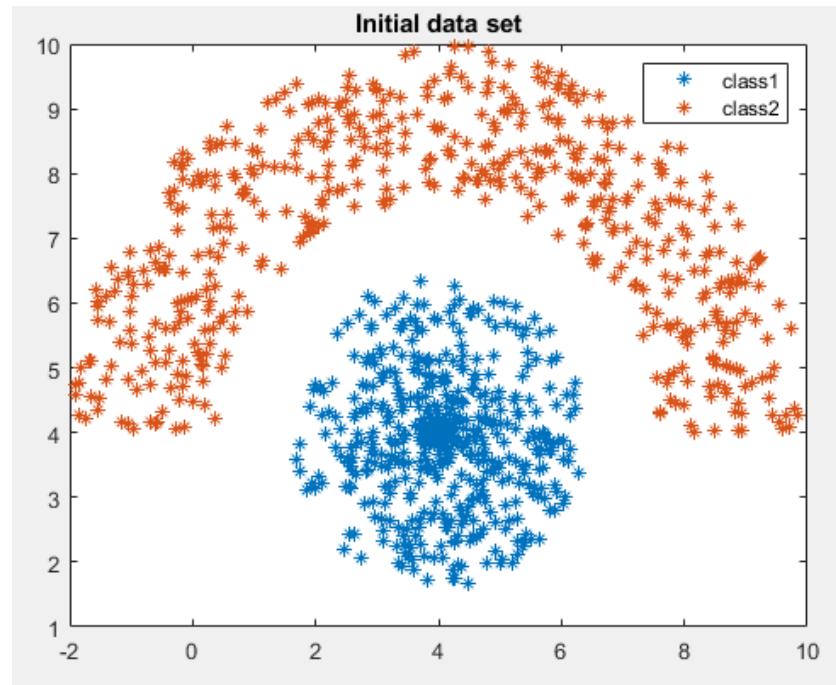
gde  $t = 1, \dots, L$ . Sempl se pridružuje onoj klasi za koju je ovaj izraz najmanji.

3. Ako je  $X_j$  reklasifikovan ide se u sledeću  $t + 1$ -u iteraciju sa novim procenjenim parametrima apriornih verovatnoća, matematičkih očekivanja i kovarijacionih matrica i ponavlja korak 2.

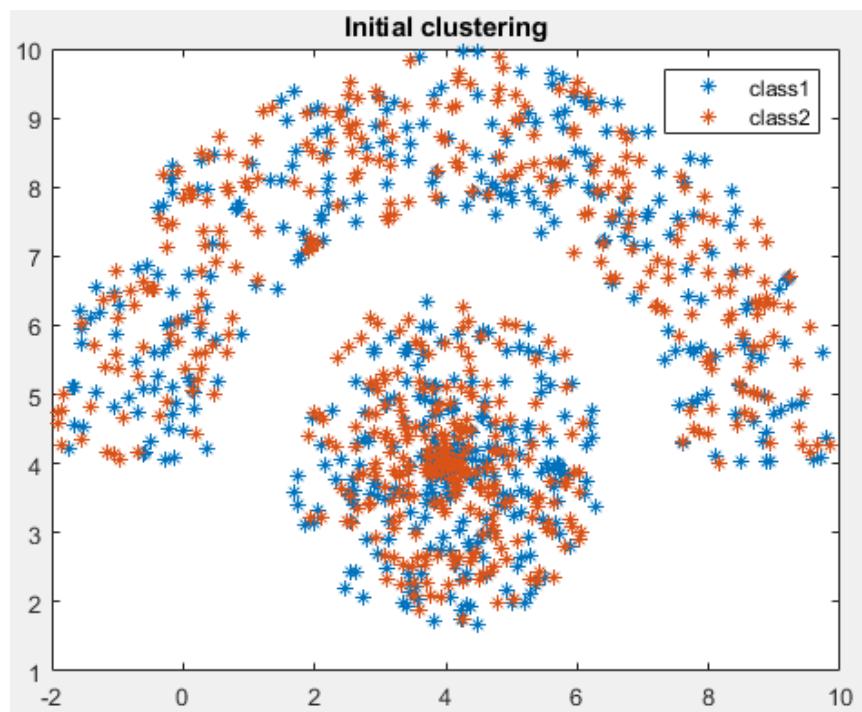
Na slici 4.2.1. je prikazan inicijalni set podataka. Na slici 4.2.2. se može videti inicijalna klasterizacija, na slici 4.2.3. su prikazani rezultati svake iteracije, i konačan rezultat, do kog se dolazi nakon 4 iteracije slići 4.3.4. (izabran je slučaj u kome se kroz 4 iteracije dolazi do rešenja jer je najlakše prikazati međukorake, broj iteracija je uglavnom dosta veći).

Pokretanjem algoritma 20 puta pronađen je prosečan broj iteracija koje su potrebne da se dođe do rešenja. U proseku je potrebno 9.5 iteracija. Za razliku od c-mean algoritma koji daje konzistentne rezultate, izuzev nekoliko odbiraka, metod kvadratne dekompozicije pri svakom pokretanju programa daje vidno drugačiju klasterizaciju. Finalna klasterizacija dosta zavisi od početne inicijalizacije. Osim same klasifikacije odbiraka, i broj iteracija je takođe veoma promenljiv i osetljiv na početne uslove.

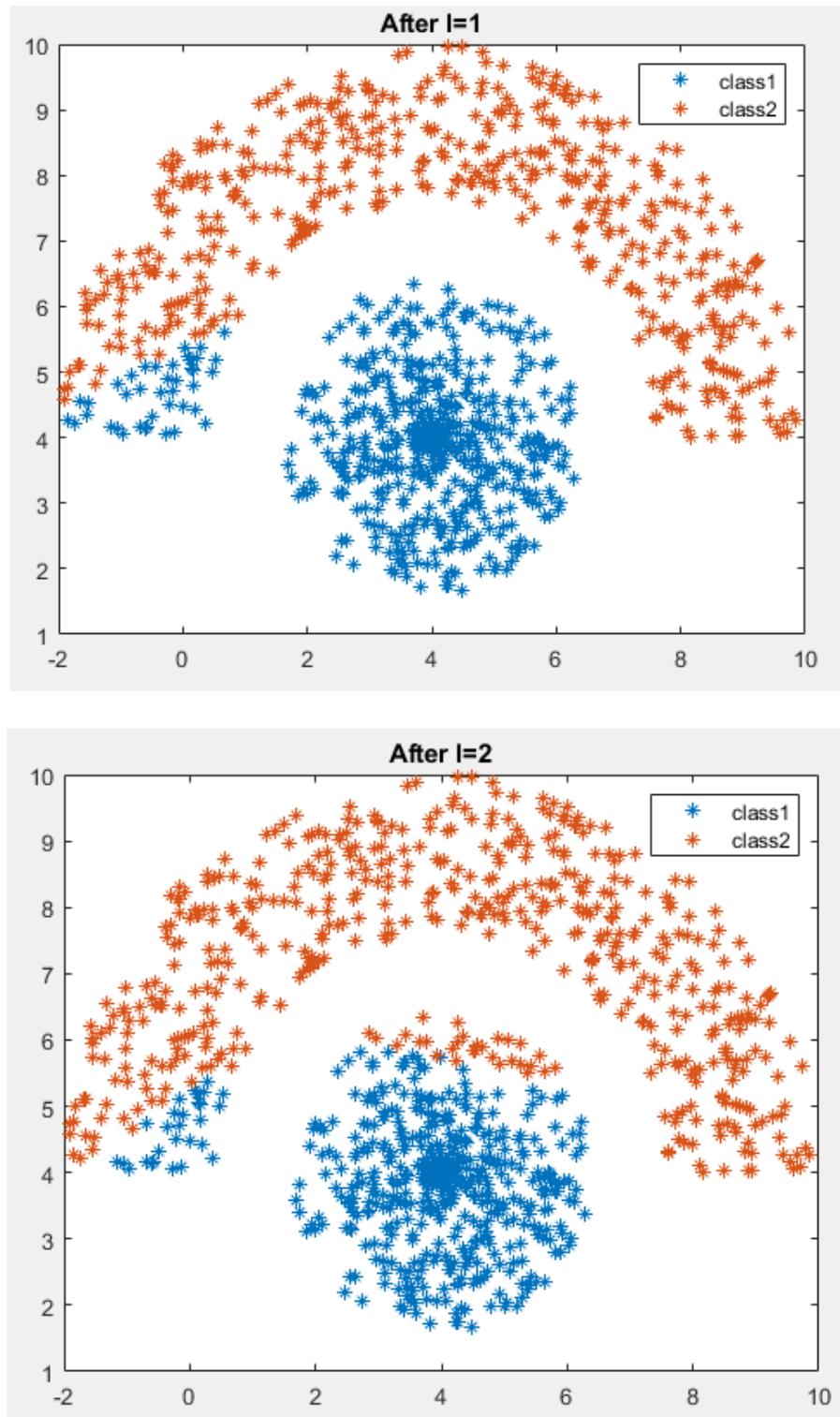
Kada uporedimo kako rezultati klasifikatora baziranog na c-mean metodi i onog baziranog na metodi kvadratne dekompozicije zavise od inicijalizacije možemo da zaključimo da kriterijumska funkcija u drugom slučaju ima mnogo više lokalnih ekstremuma nego u prvom.



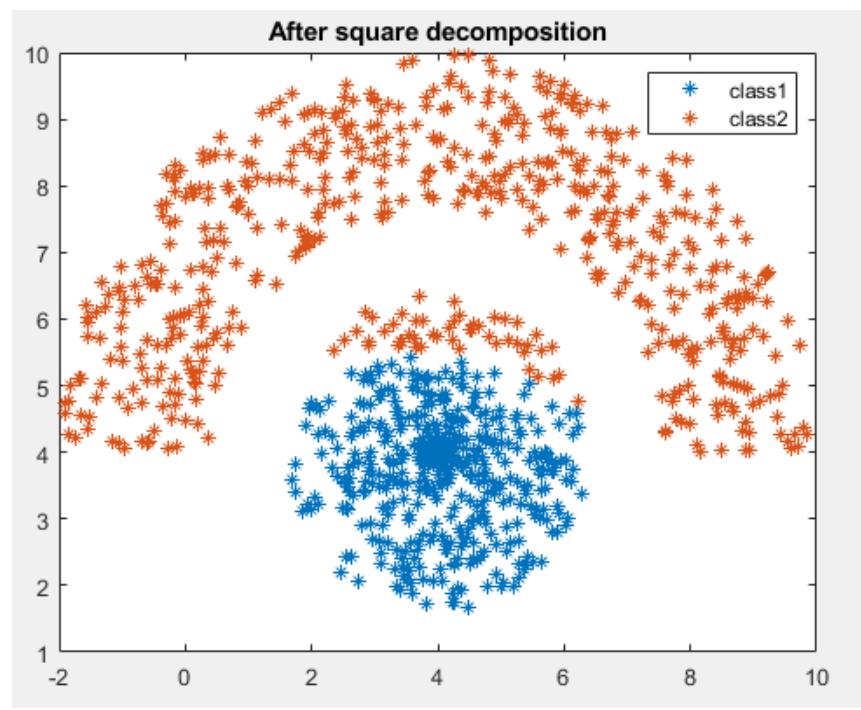
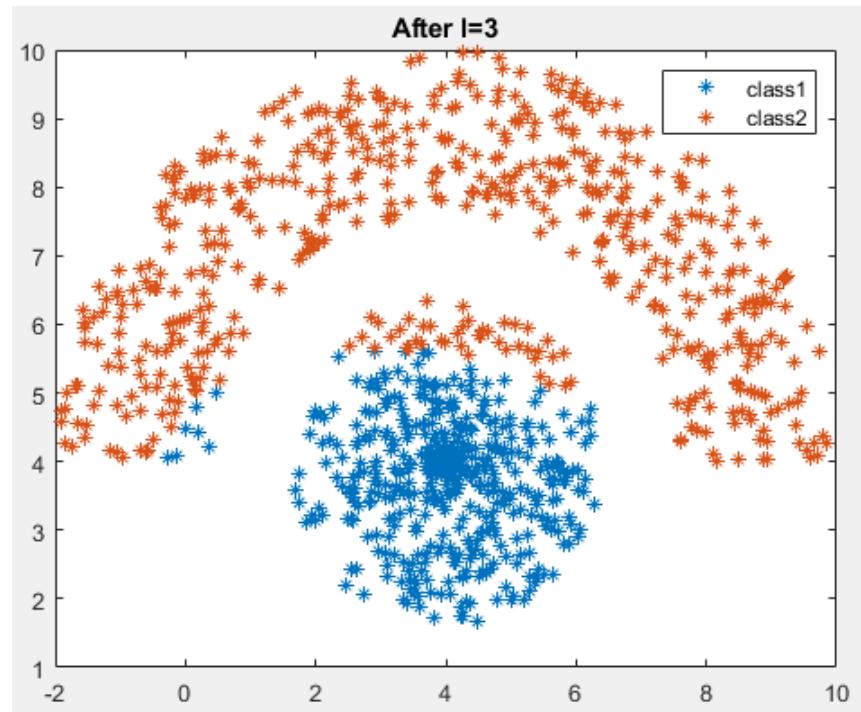
*Slika 4.2.1. Generisani odbirci*



*Slika 4.2.2. Inicijalna klasterizacija*



Slika 4.2.3. Rezultati klasterizacije nakon prve i druge iteracije



*Slika 4.2.4. Rezultati nakon treće iteracije i konačni rezultati klasterizacije dobijeni metodom kvadratne dekompozicije*

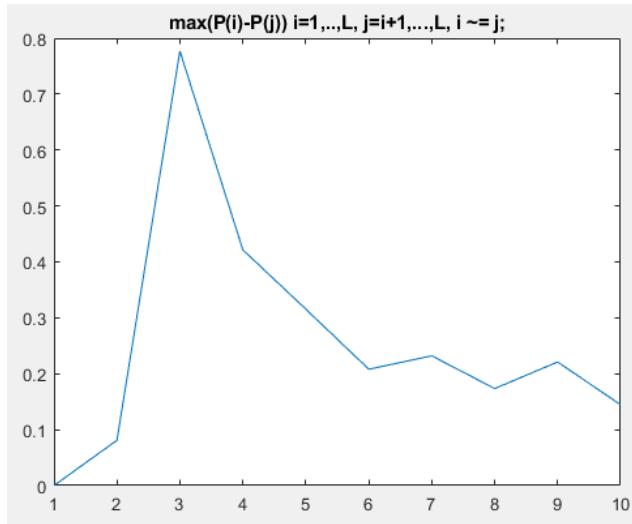
## **Primena metoda kvadratne dekompozicije kada broj klasa nije unapred poznat**

Kao što je već pomenuto, kriterijumska funkcija u slučaju metoda kvadratne dekompozicije ima mnogo više lokalnih ekstremuma. Zato algoritam koji je primenjen na c-mean klasterizaciju u ovom slučaju ne daje dobre rezultate. Računanje kriterijumske funkcije pokazuje da ona opada sa povećanjem broja klasa, pa nije moguće pronaći takvo  $L_0$  posle kog ona postaje relativno konstantna. Za kvadratnu dekompoziciju je potrebno više informacija da bi se donela odluka o broju klasa ako on nije unapred poznat.

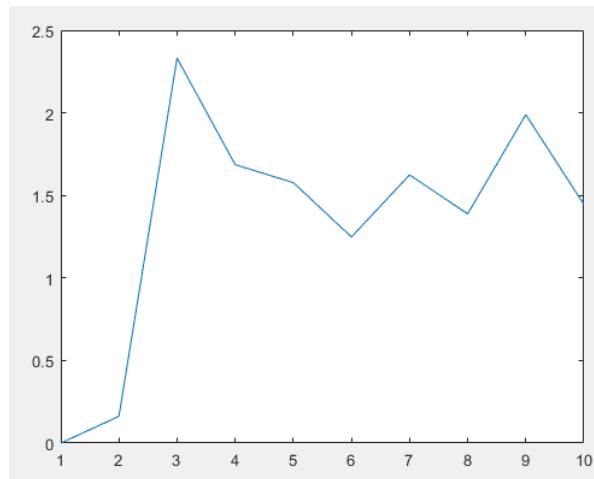
Kako razmatranje kriterijumske funkcije nije opcija u ovom slučaju, potrebne su dodatne informacije. Ovde je izabранo da se koristi činjenica da je broj odbiraka u klasama isti, odnosno prepostavlja se da je verovatnoća pojave svake klase ista. Ispostavlja se da ovo nije loš kriterijum i zapravo daje dobre rezultate. Za svaku klasterizaciju sa određenim brojem klasa je izračunata maksimalna vrednost razlike između bilo koje dve verovatnoće pojave svake klase. Za svaki broj klasa je ova vrednost izračunata dva puta i uzeta srednja vrednost. Na slici 4.3.5. je prikazana zavisnost ove vrednosti od broja klasa.

U ovom slučaju se može izvesti tačan zaključak o stvarnom broju klasa. Vidi se da je ova razlika verovatnoća pojave svake klase jedino u slučaju dve klase dovoljno mala da bude bliska onoj gde je verovatnoća pojave svake klase ista. Skok koji se dešava pri inicijalnom povećanju broja klasa pre nego što vrednost postane skoro konstantna je takođe dobar pokazatelj (razlike verovatnoća 0.2 za veliki broj klasa nisu toliko značajne kao kada je broj klasa manji obzirom da su sve verovatnoće tada dosta manje). Kada se dobijena maksimalna razlika pomnoži sa brojem klasa, da bi se razlici dodelila veća težina kada je i broj klasa veći, dobija se se zanimljiv grafik sa kog je moguće tačno očitati kako je rasipanje odbiraka po klasama zaista najmanje za dve klase. Rezultat je prikazan na slici 4.3.6.

Konačno, ova metoda bi mogla da se i iskoristi u slučaju c-mean algoritma ako bi se dozvolila dodatna informacija o tome kako su raspodeljeni brojevi odbiraka po klasama.



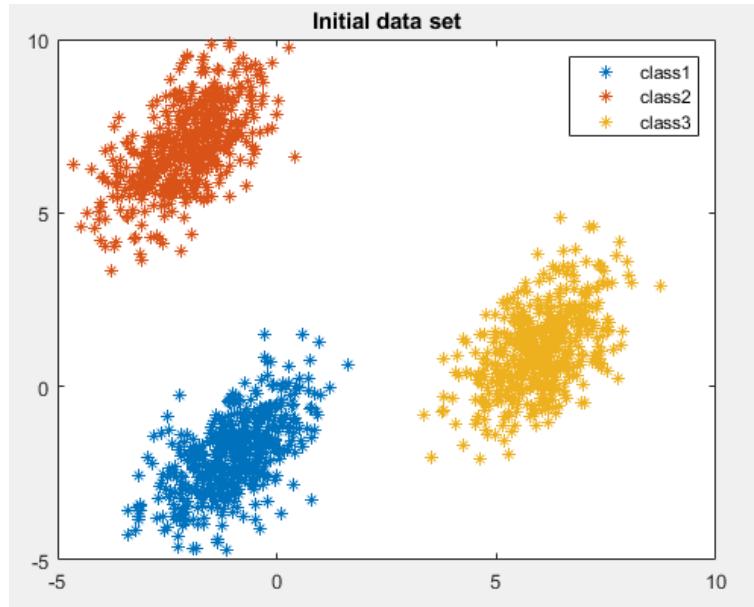
Slika 4.3.5. Zavisnost maksimalne vrednosti razlike ma koje dve verovatnoće pojave klase od broja klasa



Slika 4.3.6. Zavisnost ravnomernosti rasipanja odbiraka po klasama u zavisnosti od broja klasa

### Poređenje c-mean algoritma i metoda kvadratne dekompozicije

Zanimljivo je uporediti kako svaki od ova dva algoritma zavisi od početne inicijalizacije i prosečan broj iteracija koji je potreban da se uradi klasterizacija podataka. U te svrhe su generisane tri linearno separabilne klase (pošto c-mean algoritam ne radi dobro ako to nije slučaj). Na slici 4.4.1. je mogu videti generisani odbirci.



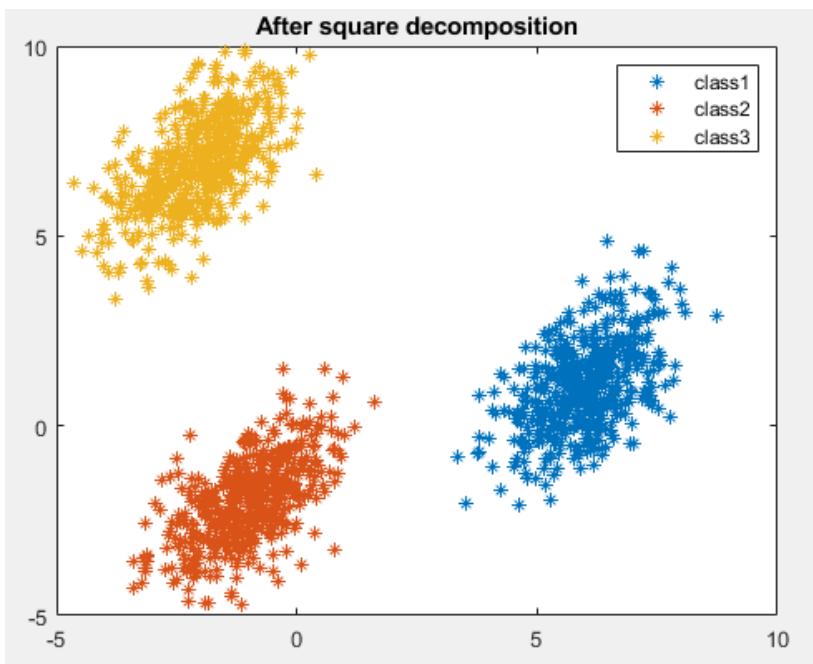
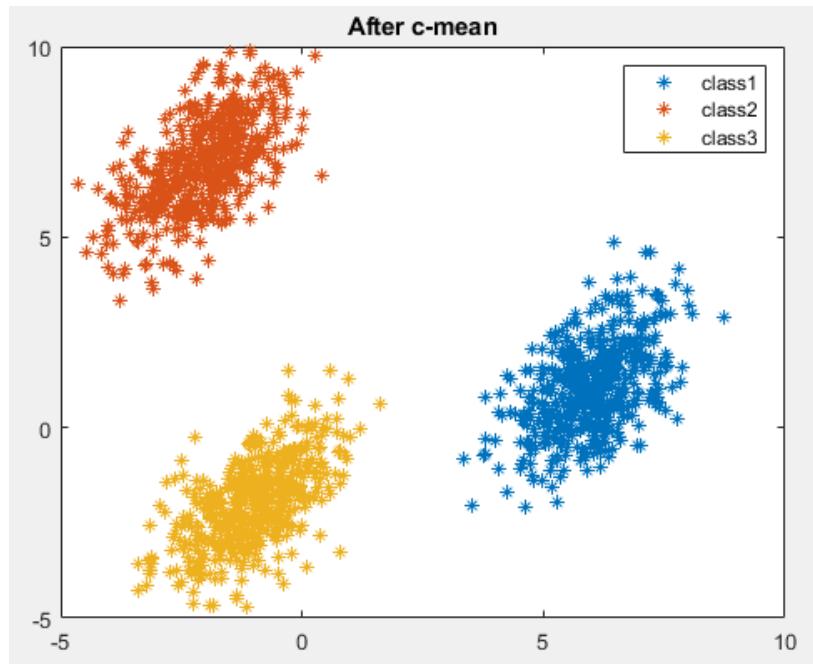
Slika 4.4.1. Generisani odbirci

Na slici 4.4.2. su prikazani rezultati klasterizacije. Obzirom da su klase jasno linearne separabilne oba algoritma ovde daju odlične rezultate i ne prave greške. Sama reklasifikacija odbiraka je daleko brža kada se koristi c-mean algoritam obzirom da se računa samo Euklidsko rastojanje. Osim toga, c-mean algoritam na početku svake iteracije računa samo vektore matematičkih očekivanja, dok metod kvadratne dekompozicije zahteva i da se pronađu kovarijacione matrice i verovatnoće pojavljivanja svake klase. Dakle, što se tiče same brzine izvršavanja, c-mean metod je daleko bolji.

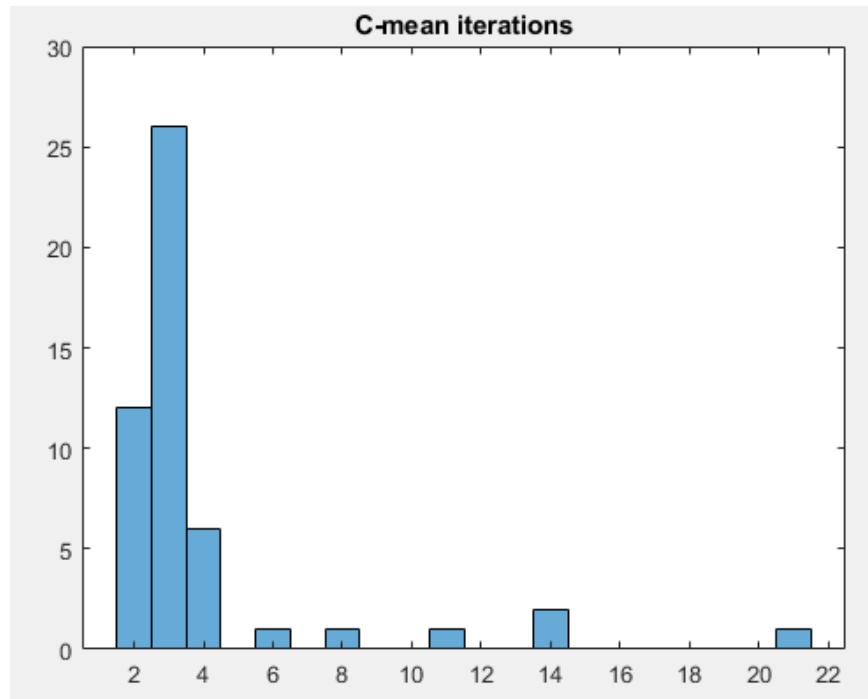
Smatrajući da je broj klasa poznat, izvršena je klasterizacija primenom oba algoritma po 50 puta i formirane su slučajne promenljive koje predstavljaju broj iteracija potreban svakom algoritmu da dođe do rešenja. Na slici 4.4.3. je prikazan histogram broja iteracija kada se koristi c-mean algoritam, a na slici 4.4.4. kada se koristi metod kvadratne dekompozicije.

Broj iteracija u slučaju metoda normalne dekompozicije ne zavisi mnogo od inicijalne klasifikacije i to se jako lepo vidi na histogramu. Broj iteracija je, osim nekih izuzetaka, praktično konstantan - median broj iteracija je 3. Sa druge strane, kada se pogleda histogram iteracija za metodu normalne dekompozicije vidi se da je dosta teže predvideti koliko će iteracija biti. Razlog je to što je ovaj algoritam veoma osteljiv na početnu klasterizaciju.

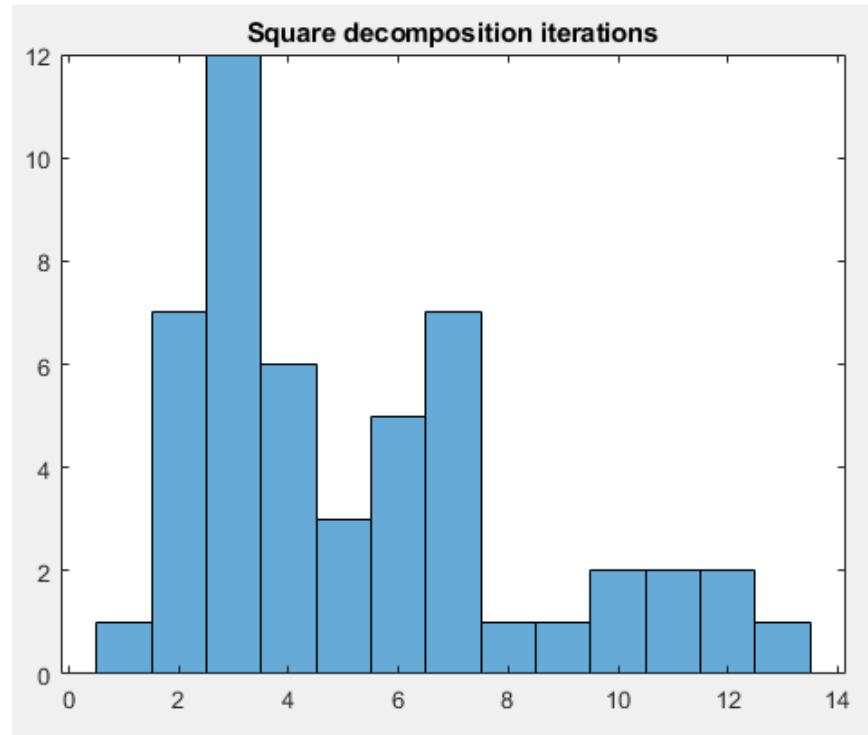
Zaključak je da u slučajevima u kojima je moguće primeniti c-mean algoritam ovaj algoritam bolja opcija. Ako, ipak, klase nisu linearne separabilne, metod kvadratne dekompozicije ostaje kao moguće rešenje obzirom da se tada dobijaju granice između klastera koje su deo po deo kvadratne krive.



Slika 4.4.2. Rezultati klasterizacije nakon primene c-mean algoritma i metoda kvadratne dekompozicije



*Slika 4.4.3. Histogram broja iteracija kada se koristi c-mean algoritam*



*Slika 4.4.4. Histogram broja iteracija kada se koristi metod kvadratne dekompozicije*

