Katedra za Signale i sisteme
Elektrotehnički fakultet
Univerzitet u Beogradu

Prepoznavanje oblika

Domaći zadatak

Student:
Aleksandar Đorđević 2019/0086

Mentori:
prof. dr Željko Đurović
dipl. ing Natalija Đorđević

Sadržaj

1 Zadatak 1

1.1 Postavka problema

Na sajtu predmeta dostupna je baza zabelezenih brzina po x i po y osi, kao i pritiska prilikom ispisivanja slova na grafičkoj tabli pod nazivom PO_slova_brzine (u njoj se nalazi fajl "PO_slova.mat'). Za jedno slovo prva vrsta su brzine po x osi u ekvidistantnim trenucima, druga vrsta brzine po y osi, a treća vrsta pritisak. Za datu bazu projektovati inovativni sistem za prepoznavanje 10 slova po izboru zasnovan na testiranju hipoteza. Zbog malog broja dostupnih odabiraka za svako slovo, nije potrebna podela na trening i test skup. Uzeti približno jednak broj odabiraka za svako od slova. Nije dozvoljeno koristiti obeležja korišćena na času vežbi($max(v_{x/y} - min(v_{x/y}))$, kao i $max(v_{x/y})$ i $min(v_{x/y})$).

a) Za jedan primerak svakog od odabranih slova prikazati njegov oblik (pozicije na y osi u zavisnosti od pozicije na x) i njegove brizne po x i y osi. Prokomentarisati povezanost brzine po x i y osi sa načinom ispisivanja slova.

b) Rezultate klasifikacije prikazati u obliku konfuzione matrice.

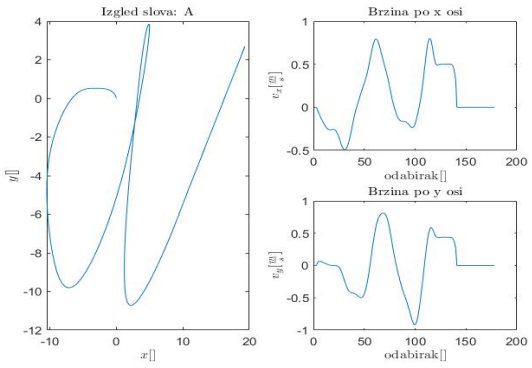
c) Opisati projektovani siste, obrazložiti izbor obeležja i prikazati i prokomentarisati karakteristične primere pravilno i nepravilno klasifikovanih slova (ukoliko ih ima).

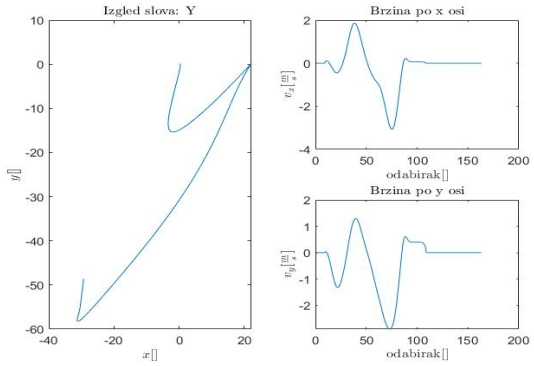
d) Odabrati dva slova i dva obeležja takva da su odabrana dva slova separabilna u tom prostoru.

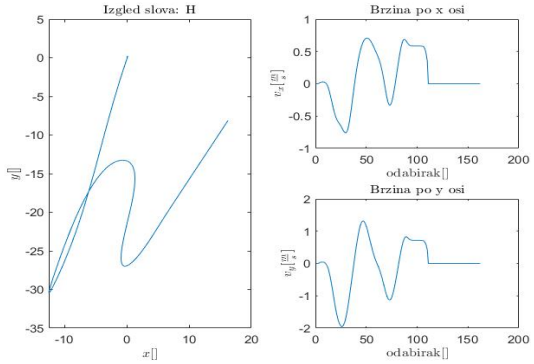
e) Za slova i obeležja pod c) projektovati parametarski klasifikator po izboru i iscrtati klasifikacionu liniju.

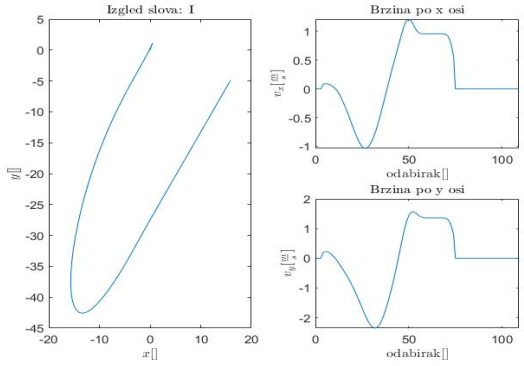
1.2 Rešenje

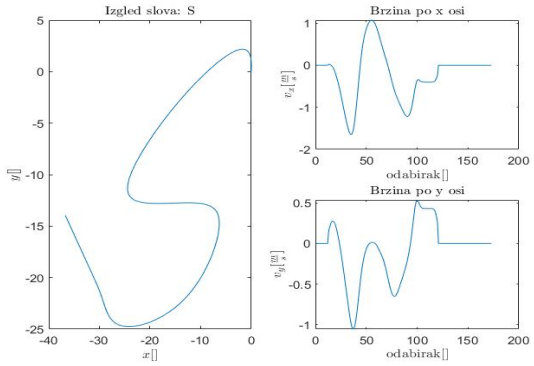
Izabrana slova su : a, y, h, i, s, w, b, z, v, c. Na sledećim graficima su prikazani oblici isipisanih slova kao i brzine pisanja po x odnosno y osi:

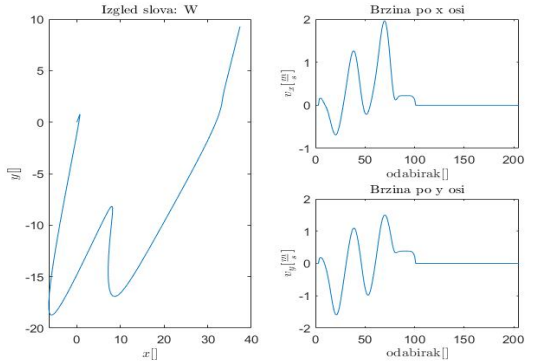


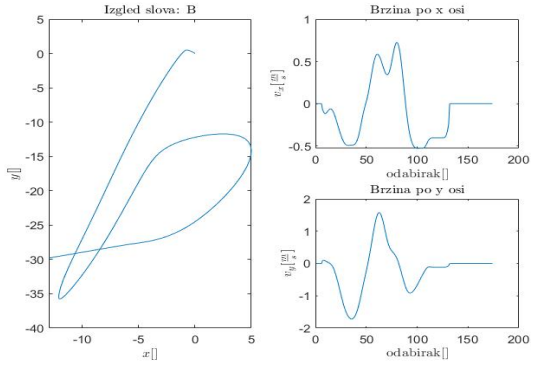

Izgled slova: Y
y[]
x[]
Brzina po x osi
odabirak[]
Brzina po y osi
odabirak[]

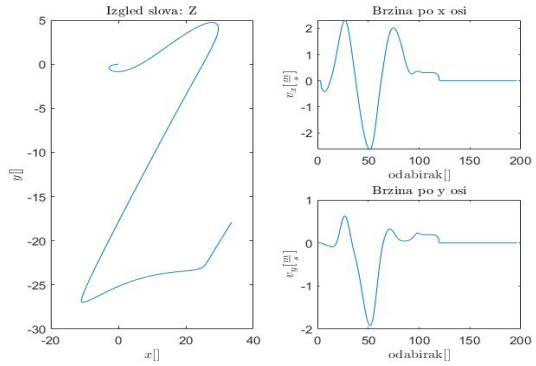

Izgled slova: H
y[]
x[]
Brzina po x osi
odabirak[]
Brzina po y osi
odabirak[]

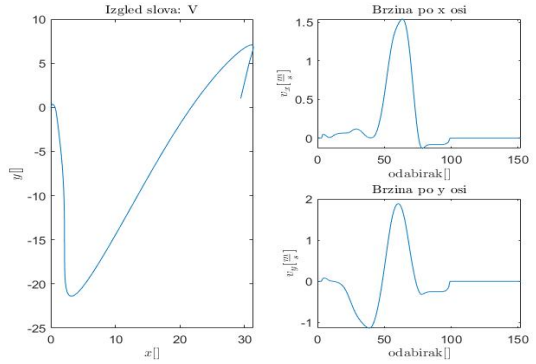

Izgled slova: I
y[]
x[]
Brzina po x osi
odabirak[]
Brzina po y osi
odabirak[]

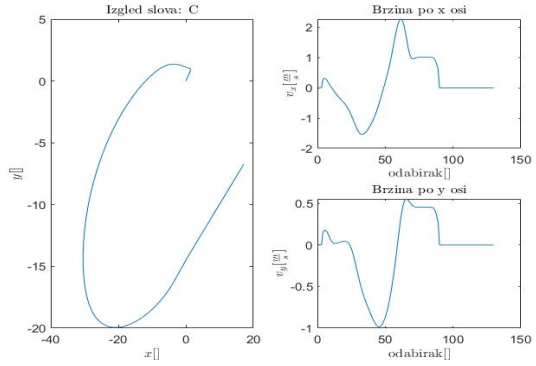

Izgled slova: S
y[]
x[]
Brzina po x osi
odabirak[]
Brzina po y osi
odabirak[]


Izgled slova: W
y[]
x[]
Brzina po x osi
odabirak[]
Brzina po y osi
odabirak[]


Izgled slova: B
y[]
x[]
Brzina po x osi
odabirak[]
Brzina po y osi
odabirak[]







Možemo primetiti da što je slovo jedostavnijeg oblika to su brzine pisanje veće. Na osnovu ovoga se mogu izvući razna obeleźja koja će veoma uspešno vršiti klasifikaciju slova. Jedna od najboljih obeležja za ovu vrstu problema jesu maksimalne i minimalne brzine po x odnosno y osi, ali kako u ovom zadatku njihovo korišćenje nije dozvoljeno primenjena su druga obeležja takođe izvučena iz brzina po x i y osi.

1.2.1 Izbor obeležja

Sva korišćena obeležja su izvučena iz brzina, jer nije nađena neka značajnija korelacija između pritiska i vrste slova koja su ispisana.
Korišćena obeležja su:

- Suma brzina po x osi
- Suma brzina po y osi
- Broj promena znaka brzine po x osi
- Broj promena znaka brzine po y osi
- Razlika između maksimalnog i minimalnog ubrzanja po x osi
- Razlika između maksimalnog i minimlanog ubrzanja po y osi

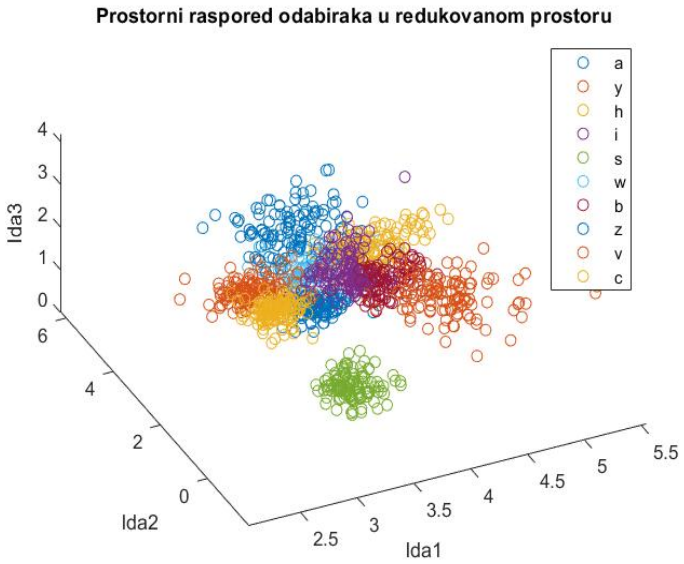
Suma brzina ima smisao srednje brzine pisanja slova što smo mogli da vidimo da je dobro korelisano sa vrstom ispisanog slova.
Broj promena znaka brzine jeste informativno jer pokazuje na to koliko promena smera smo imali tokom pisanja slova što se zaista razlikuje od slova do slova. Međutim, ono je diskretno i relativno malo te nema mnogo mogućnosti za separaciju odabiraka. Pored ovoga, neki ljudi prilikom pisanja dodaju kružiće, kvačice na krajevima slova što značajno utiče na ovo obeležje.
Razlika maksimalnog i minimalnog ubrzanja ima sličan smisao kao i maksimalna i minimalna brzina to jest što je slovo jednostavnije ljudi imaju tendenciju da ga brže pišu. Ipak ispostavlja se da je ono ipak manje informativno od maksimalne i minimalne brzine.

1.2.2 Redukcija dimenzija

Zbog numeričke složenosti nije zgodno da problem posmatramo u $6D$ prostoru te smo pre projektovanja klasifikatora izvršili redukciju dimenzija na 3. Pored toga što nam numerički uprošćava problem redukcija na 3 dimenzije nam da je mogućnost da problem vizuelizujemo što svakako nije bio slučaj sa 6 dimenzija.

Za redukciju dimenzija korišćena je LDA metoda. LDA predstavlja redukckiju dimenzija na bazi matrica rasejanja i ona je vid ekstrakcije obeležja. Naime, ona uzima u obzir sva obeležja i kombinuje ih u manji broj novih obeležja koja ne moraju nužno imati fizički smisao. Postupak je taj da se na osnovu matrica unutarklasnog i međuklasnog rasejanja formira nova matrica čiji sopstveni vektori koji odgovaraju najvećim sopstvenim vredonstima čine matricu transformacije.

Nakon LDA redukcije odabirci izgledaju ovako: Kao što vidimo odabirci nisu u potpunosit



separabilni, ali su dovoljno razdvojeni da možemo izvršiti klasifikaciju sa pristojnom tačnošću.

1.2.3 Procena funkcije gustine verovatnoće

Uslov za primenu bilo kakvog testiranja hipoteza jeste poznavanje funkcije gustine verovatnoće. Kako je mi ne znamo eksplicitno procenićemo je nekom neparametarskom metodom, a zatim primeniti testiranje hipoteza.
Primenićemo KNN(K Nearest Neighbours) metod tako što ćemo napraviti trodimenzionalni grid i svakoj njegovoj tački pomoću KNN metoda proceniti fgv.
KNN proširuje zapreminu koju posmatra sve dok u njega ne upaden određeni broj odabiraka. Zatim, procenjuje fgv na osnovu sledeće relacije:

$$\hat{f} = \frac{k-1}{Nv}$$

, gde je k broj suseda koje posmatramo, N ukupan broj odabiraka, a v zapremina oblasti koja obuhvata sve susede (u našem slučaju sfera poluprečnika jednakom udaljenosti do k-tog najbližeg suseda).
Još se postavlja pitanje izbora k. Pokazuje se da je dobar izbor $k = [N^{\alpha}]$. Uzeto je $\alpha = 0.6$ i dobijeno $k = 70$.

1.2.4 Bayes-ov test minimalne greške

Sada kada smo procenili fgv možemo primeniti neki od testova hipoteza. Primenićemo Bayes-ov test minimalne greške jer on daje teorijski najmanju ukupnu grešku. Njegova mana jeste ta što ne vodi računa o tome kolike su pojedinačne greške određenog tipa, ali nama to ni ne smeta jer su na greške svih tipova jednako loše.

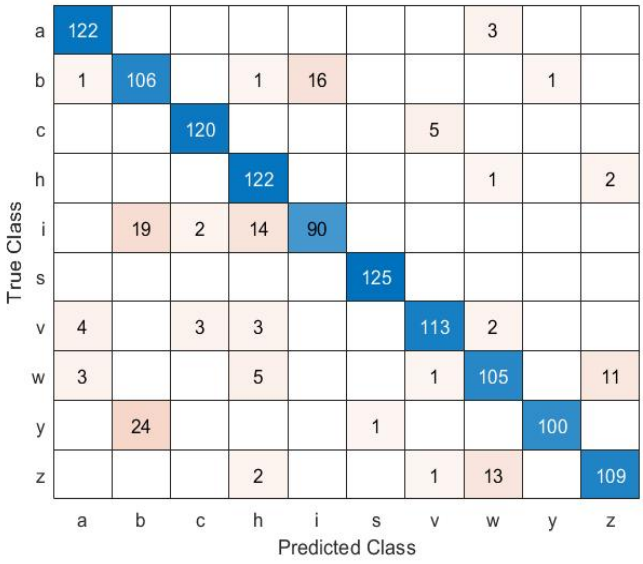
Bayes-ov test funckioniše tako što upoređuje aposteriorne verovatnoće pripadnosti pojedinačnim klasama i odabirak smešta u onu sa maksimalnom aposteriornom verovatnoćom. Aposteriorna verovatnoća se računa na osnovu relacije:

$$q_i(x) = \frac{p_i f_i(x)}{f(x)}$$

, gde je p_i verovatnoća pojavljivanja odabirka iz i-te klase, f_i fgv odabiraka iz i-te klase, a f kumulativna verovatnoća odabiraka.
Kako je f isto za sve klase njega možemo zanemariti, i u našem slučaju su i p-ovi svi isti jer smo uzimali isti broj odabiraka iz svake klase. Dakle, naš test se svodi na upoređivanje marginalnih fgv koje smo već procenili.

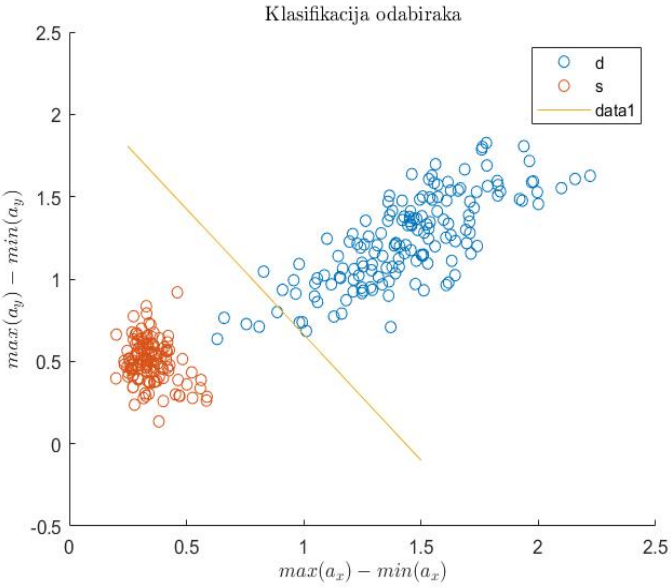
1.2.5 Rezultati klasifikacije testiranjem hipoteza

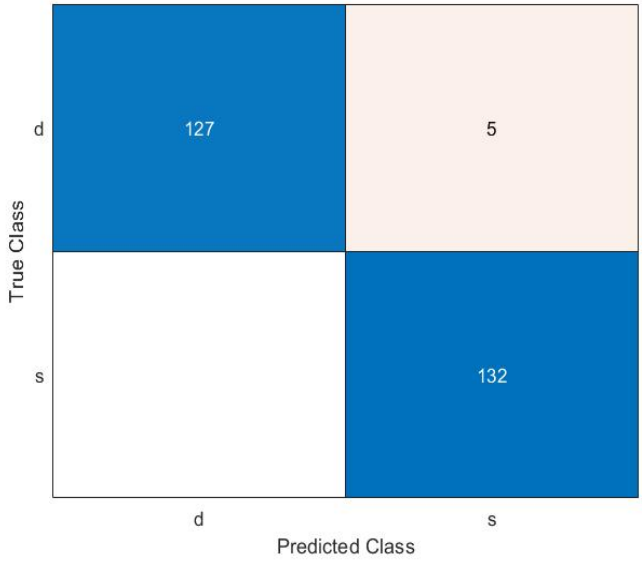
Rezultate ćemo predstaviti u vidu matrice konfuzije: Ukupna tačnvst iznosi 88.96%.



1.2.6 Parametarski klasifikator

U ovom delu su izabrana dva slova (d i s) i dva obelezja (razlika maksimalnog i minimalnog ubrzanja po x i y osi). Izbor je izvršen tako da data slova budu linearno separabilna u prostoru obeležja.
Kako su obeležja linearno separabilna možemo projektovati linearni klasifikator na primer klasifikator distance. Klasifikator distance funkcioniše tako što odabirak smešta u onu klasu čijem centru je bliži te će klasifikaciona linija biti simetrala duži koja spaja centre (matematička očekivanja) klasa. Rezultate možemo predstaviti grafički ili preko konfuzione matrice:


Klasifikacija odabiraka
d
s
data1
$max(a_y) - min(a_y)$
$max(a_x) - min(a_x)$
2.5
2
1.5
1
0.5
0
-0.5
0
0.5
1
1.5
2
2.5


127
5
132
d
s
True Class
d
s
Predicted Class

2 Zadatak 2

2.1 Postavka problema

Generisati po $N = 500$ odabiraka iz dveju dvodimenzionalnih bimodalnih klasa:

$$\Omega_1 : P_{11}N(M_{11}, \Sigma_{11}) + P_{12}N(M_{12}, \Sigma_{12})$$
$$\Omega_2 : P_{21}N(M_{21}, \Sigma_{21}) + P_{22}N(M_{22}, \Sigma_{22})$$

Parametre klasa samostalno izabrati.

a) Na dijagramu prikazati odabirke.

b) Iscrtati kako teorijski izgledaju funkcije gustine verovatnoće za raspodele klasa i uporediti ih sa histogramom generisanih odabiraka.

c) Projektovati Bayes-ov klasifikator minimalne greške i na dijagramu, zajedno sa odabircima, skicirati klasifikacionu liniju. Uporediti greške klasifikacije konkretnih odabiraka sa teorijskom greškom klasifikacije prve i druge vrste za datu postavku.

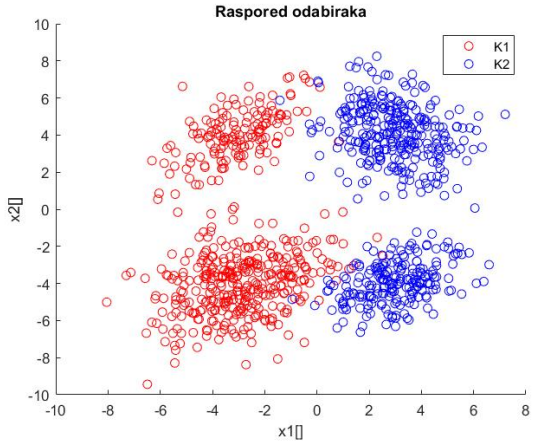
d) Projektovati klasfikator minimalne cene tako da se više penalizuje pogrešna klasifikacija odabiraka iz prve klase.

e) Ponoviti prethodnu tačku za Neyman-Pearson-ov klasifikator. Obrazložiti izbor $\epsilon_2 = \epsilon_0$.

f) Za klase oblika generisanih u prethodnim tačkama, projektovati Wald-ov sekvencijalni test pa skicirati zavisnost broja potrebnih odabiraka od usvojene verovatnoće greške prvog odnosno drugog tipa.

2.2 Rešenje

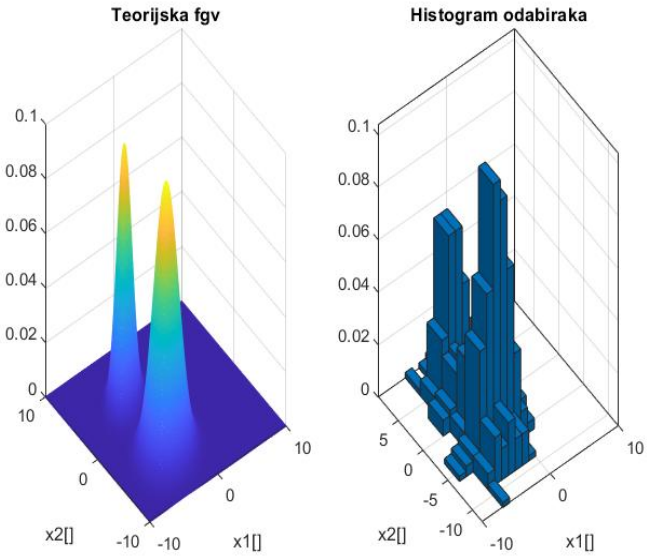
Na slici su prikazani odabirci: Teorijski fgv za dvodimenzionalnu Gausovu raspodelu se

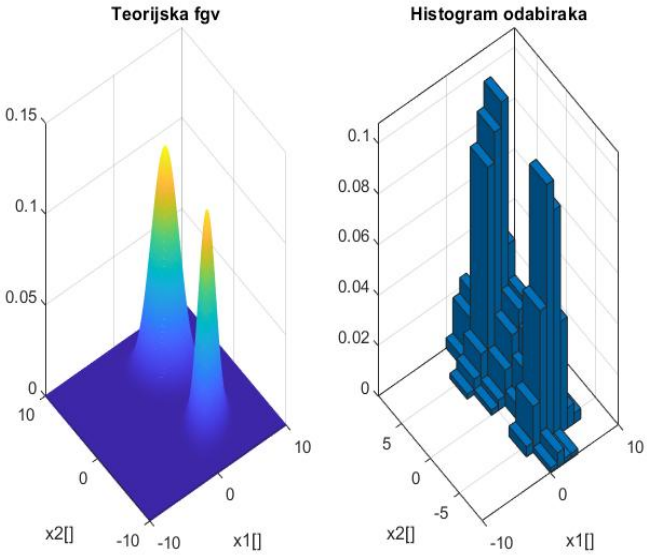


računa kao:

$$f(\mathbf{X}) = \frac{1}{2\pi|\Sigma|^{\frac{1}{2}}} e^{(-\frac{1}{2}(\mathbf{X}-M)^T\Sigma^{-1}(\mathbf{X}-M))}$$

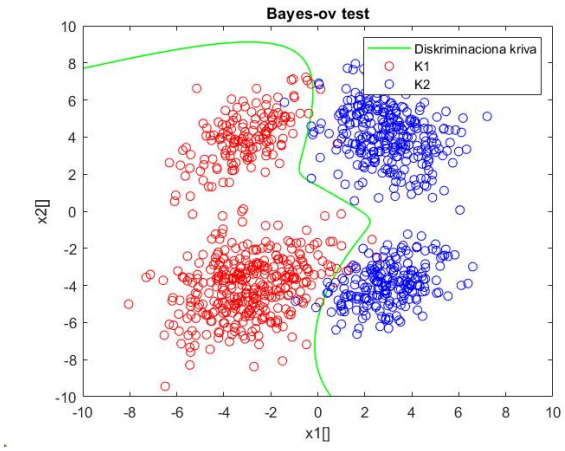
Procenu fgv za odabirke možemo izvršiti pomoću histograma normalizovanog deljenjem sa ukupnim brojem odabiraka. Uporedimo teorijsku fgv sa fgv generisanih odabiraka:

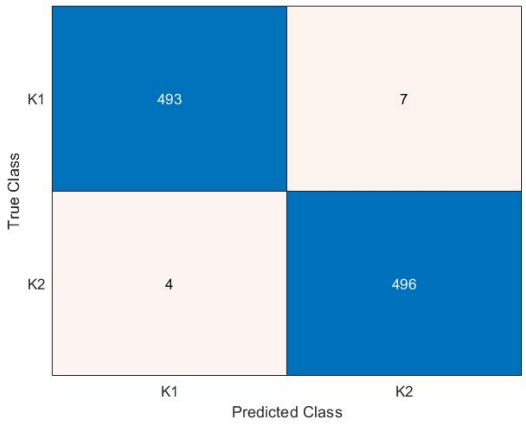

Teorijska fgv
Histogram odabiraka
0.1
0.08
0.06
0.04
0.02
0
10
0
-10
x2[]
-10
0
10
x1[]
5
-5


Teorijska fgv
Histogram odabiraka
0.15
0.1
0.05
0
10
0
-10
x2[]
-10
0
10
x1[]
0.08
0.06
0.04
0.02
5
-5

2.2.1 Bayes-ov test minimalne greške

Kako smo već u prethodnom zadatku definisali Bayes-ov test minimalne greške to ćemo ovde preskočiti.
Rezultati Bayes-ovog test izgledaju ovako: Ukupna verovatnoća greške iznosi 1.1%, dok je





greška prve vrste 1.4%, a druge 0.8%.

2.2.2 Teorijska minimalna greška

Minimalna greška koju smo dobili nije teorijski minimalne jer je rezultat klasifikacije realnih odabiraka. Da bi se sračunala teorijski minimalna greška potrebno je sračunati površine ispod krivih aposteriornih verovatnoća (q_2 u oblasti odluke prve klase, q_1 u oblasti odluke druge klase) odnosno treba sračnati izraz:

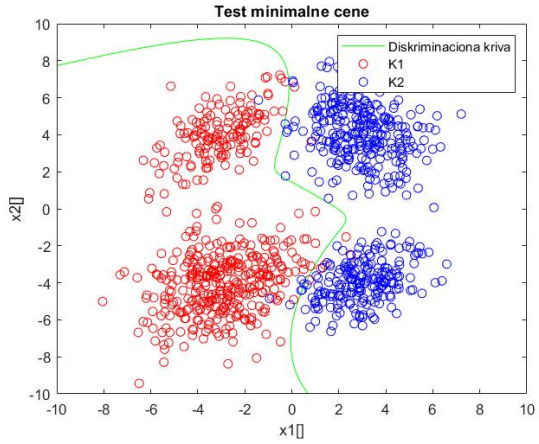
$$\epsilon = p_1 \int_{L2} f_1(x)dx + p_2 \int_{L1} f_2(x)dx$$

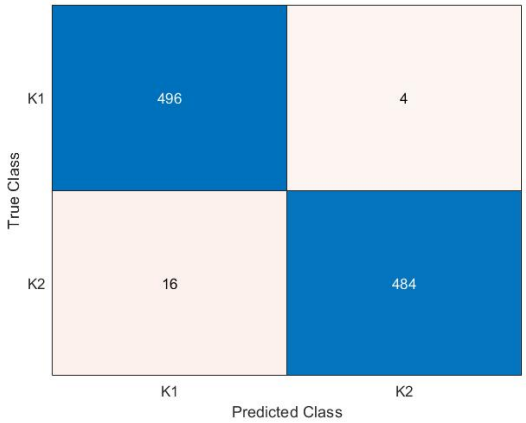
Integrale smo sračunali numerički tako što smo zapreminu ispod krive aproksimirali zbirom zapremina kvadrova sa osnovom kvadrata 0.1×0.1 i visinom koja je jednaka srednjoj vrednosti

vrednostima fgv u tačkama osnove.
Za teorijsku vrednost minimlane greške dobijamo 0.5% što je očekivano manje od greške koju smo dobili prilikom izvršavanja Bayes-ovog testa.

2.2.3 Test minimalne cene

Test minimalne cene nam omogućava da određene vrste greške penalizujemo više nego druge. To je suštinska razlika u odnosu na Bayes-ov test, a tehnička je samo u promeni praga koji nije više $\frac{p_2}{p_1}$, već $\frac{p_2}{p_1}\frac{c_{12}-c_{22}}{c_{21}-c_{11}}$, gde je c_{ij} cena odluke da odabirak pripada i-toj klasi, a on je zaista iz j-te klase.
c_{11} i c_{22} su cene dobrih odluka te ćemo njih staviti na nulu. Nama je traženo da više penalizujemo loše odluke odabiraka iz prve klase te nam c_{21} treba biti veće od c_{12}. Uzeto je da je $c_{12} = 1$, a $c_{21} = 4$.
Rezultati ovog testa su: Ukupna verovatnoća greške iznosi 2%, dok je greška prve vrste 0.8%,





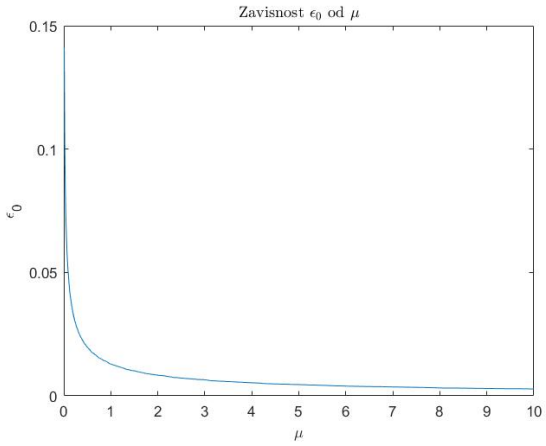
a druge 3.2%.
Možemo primetiti da se ukupna greška povećala, ali se greška prve vrste smanjila što je i bio cilj.

2.2.4 Neyman-Pearson-ov test

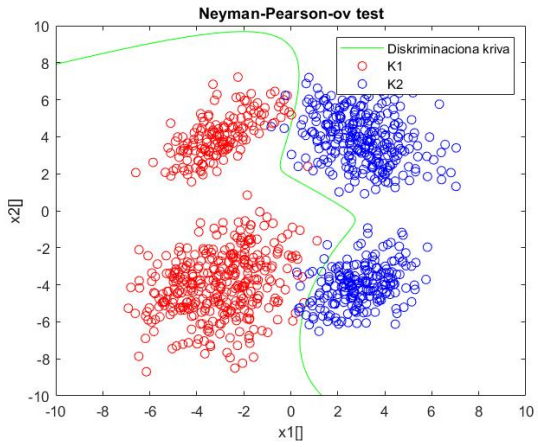
Neyman-Pearson-ov test nam daje mogućnost da jednu vrstu greške držimo konstantnom dok drugo minimizujemo. U raznim problemima je zgodno lažne alarme držati konstantnim dok propuštene detekcije minimizujemo te je poznat pod nazivom CFAR(Constant False Alarm Rate) test.
U odnosu na prethodne testove opet se razlikuje samo u pragu. Prag μ se nalazi iz izbora vrednosti vrste greške koju želimo da držimo konstantnom iz sledeće relacije:

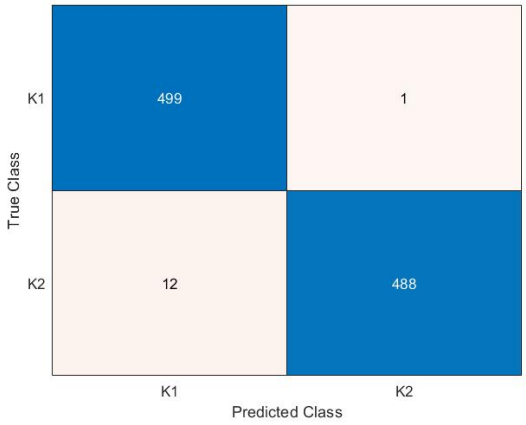
$$\epsilon_0 = \int_{-\infty}^{-ln\mu} f(h|w_2)dh$$

Još ostaje pitanje izbora ϵ_0. Jedan od mogućih izbora jeste da se ona postavi na vrednost ukupne minimalne greške jer tada će drugi tip grške biti jako mali. Tehnički ovo se izvodi tako što se gornji integral izračuna numerički (primenjena je isto metoda kao i kod računanja teorijske minimalne greške) za različite vrednosti μ, a zatim se sa grafika nađe ona koja odgovara odabranom ϵ_0.
Kako za ϵ_0 jednakom minimalnoj greški rezultati nisu bili zadovoljavajući za njega je uzeta vrednost od 3%.
Na osnovu sledećeg grafika dobijamo $\mu = 0.23$.



Rezultati ovog testa izgledaju ovako:





Ukupna verovatnoća greške iznosi 1.3%, dok je greška prve vrste 0.2%, a druge 2.4%. Tako da smo uspešno smanjili grešku prve vrste, dok grešku druge vrste ne držimo na zadatoj vrednosti već na još manjoj što je svakako povoljno.

2.2.5 Wald-ov sekvencijalni test

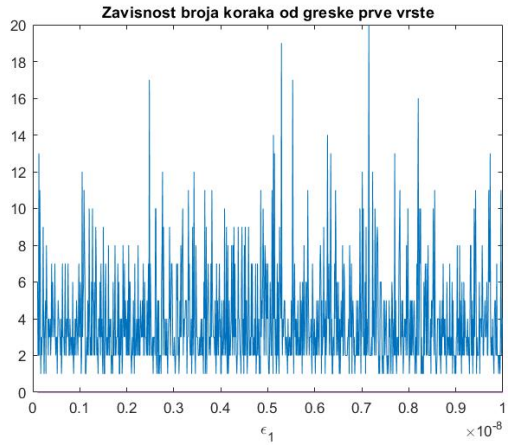
Wald-ov test kao sekvencijalni test podrazumeva da nema na raspolaganju sve odabirke odjednom već ih dobija jedan po jedan tokom vremena. Njegova najveća prednost jeste što za zadate verovatnoće prve i druge vrste nalazi pragove tako da minimizuje broj potrebnih odabiraka za donošsenje odluke. Pragovi se određuju pomoću sledćih izraza:

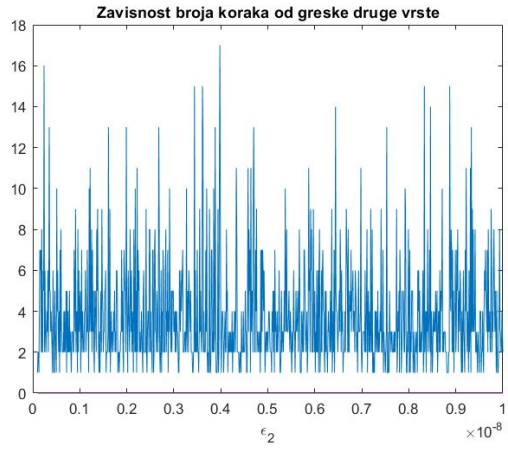
$$A = \frac{1-\epsilon_1}{\epsilon_2}$$

$$B = \frac{\epsilon_1}{1-\epsilon_2}$$

Sada se nakon svakog odabirka računa funkcija verodostojnosti kao: $lm(k) = lm(k-1)\frac{f_1(x_k)}{f_2(x_k)}$
Ako je ona veća od A onda se donosi odluka da su odabirci iz prve klase, ako je manja od B

donosi se odluka da je iz druge klase, a ako nije nijedno čeka se naredni odabirak.
Kako bi oslikali realnu situaciju u sekvencu koja se testira ubačeno je 70% odabiraka prve klase i 30% odabiraka druge klase.
Na graficima su prikazane zavisnosti broja potrebnih odabiraka od greške prve odnosno druge vrste.





3 Zadatak 3

3.1 Postavka problema

Generisati tri klase dvodimenzionalnih oblika. Izabrati funkciju gustine verovatnoće oblika tako da klase budu linearno separabilne.

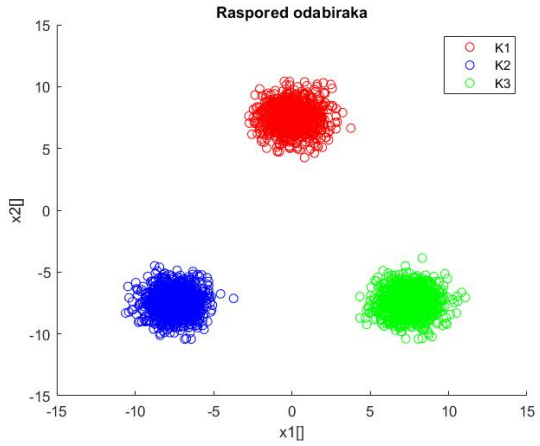
a) Za tako generisane oblike izvršiti projektovanje linearnog klasifikatora jedom od tri iterativen procedure. Rezultate prikazati u obliku matrice konfuzije. Detaljno opisati postupak klasifikacije.

b) Ponoviti prethodni postupak korišćenjem metode željenog izlaza. Analizirati uticaj elemenata u matrici željenih izlaza na konačnu formu linearnog klasifikatora.

c) Generisati dve klase dvodimenzionalnih oblika koje jesu separabilne, ali ne linearno pa isprojektovati kvadratni klasifikator metodom po želji.

3.2 Rešenje

Generisani su odabirci iz tri različite klase tako da oni budu linearno separabilni. Centri klase se nalaze redom u (0,7.5),(-7.5,-7.5) i (7.5,7.5) i imaju jedinične kovariacione matrice.



Ako želimo da projektujemo linearni klasifikator, očigledno moramo projektovati 3 njih (po jedana za svaki par klasa) odnosno projektovaćemo deo po deo linearan klasifikator. I na kraju ćemo klasifikaciju vršiti tako što za određeni odabirak vršimo klasifikaciju za svaki par klasa i na kraju ga svrstamo u onu klasu koja je "pobedila"najviše puta.

3.2.1 Metod resupstitucije

Metod resupstitucije predstavlja algoritam za projektovanje linearnog klasifikatora gde se računanje parametara klasifikator i računanje greške vrši na istom skupu odnosno nema podele na trening i test skup. Mi ćemo projektovati optimalni linearni klasifikator između svakog para klasa.
Linearni klasifikator je oblika:

$$V^T X + v_0 = 0$$

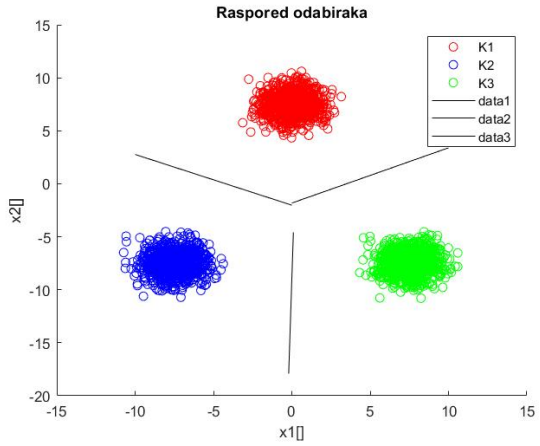
V računamo kao:

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

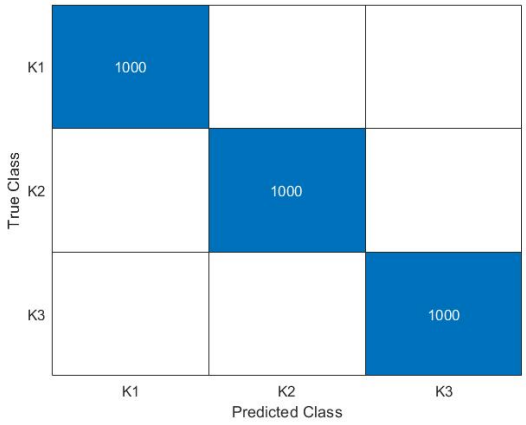
Algoritam izgleda ovako:

1. Na osnovu odabiraka procenimo M_1,M_2,Σ_1 i Σ_2
2. Zavrtimo petlju po s od 0 do 1
3. Izračunamo V na osnovu relacije gore
4. Izračunamo $y = V^T X$ za sve odabirke
5. Zavrtimo petlju po v_0 od $-max(y)$ do $-min(y)$
6. Izračunamo broj pogrešno klasifikovanih odabiraka
7. Zapamtimo ono v_0 koje daje minimalnu grešku
8. Zapamtimo ono s koje daje minimalnu grešku
9. Na osnovu tog s izračunamo V, a v_0 već imamo

Kada smo projektovali deo po deo linearan klasifikator on izgleda ovako:



A matrica konfuzije ovako:



3.2.2 Metod željenog izlaza

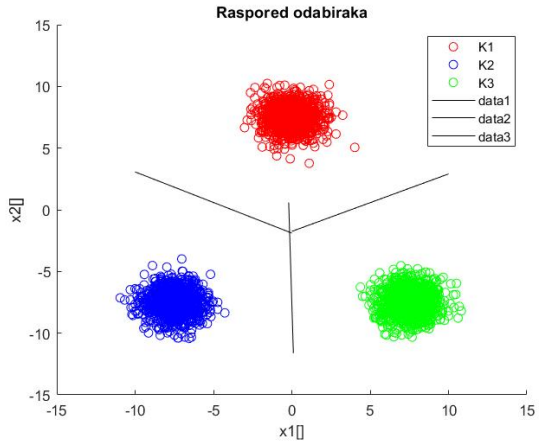
Metod željenog izlaza je metod projektovanja linearnog klasifikatora koji radi tako što se odabirci iz prve klase množe sa minus 1 kako bi se obe klasifikacije svele na isti znak nejednakosti. Formira se matrica $W = [v_0 V^T]$, matrica $U = [Z_1 Z_2 \dots Z_N]$ gde je $Z_i = [-1 - X_i]^T$ za odabirke iz prve klase, a $Z_i = [1 X_i]^T$ za odabirke iz druge klase. Na kraju se formira matrica na bazi željenog izlaza Γ obično sa svim jedinicama, ali je moguće određenim odabircima dati veću težinu tako što odgovarajuće elemente ove matrice postavimo i na više vrednosti. Minimizacijom srednjeg kvadratnog kriterijuma dobijamo relaciju:

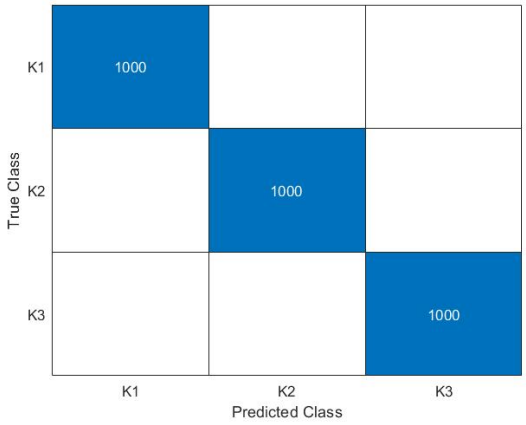
$$U^T W = \Gamma$$

Kako matrica U nije nužno kvadratna umesto inverzije radimo pseudo-inverziju pa matricu W dobijamo kao:

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

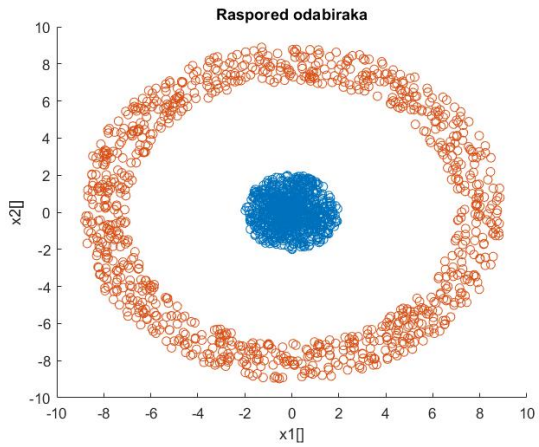
Rezultati izgledaju ovako:





3.2.3 Kvadratni klasifikator

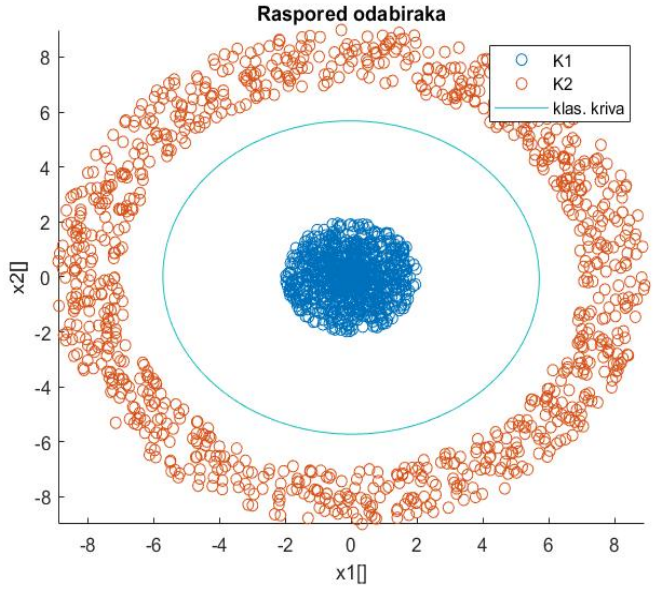
Ukoliko klase nisu linearno separabilne nije ih moguće klasifikovati linearnim klasifikatorom. Umesto njega se projektuju klasifikatori višeg reda na primer kvadratni klasifikator.
Sada smo generisali odabirke iz dve klase tako da oni nisu linearno separabilni, ali jesu kvadratno separabilni odnosno mogu se odvojiti krivom drugog reda.

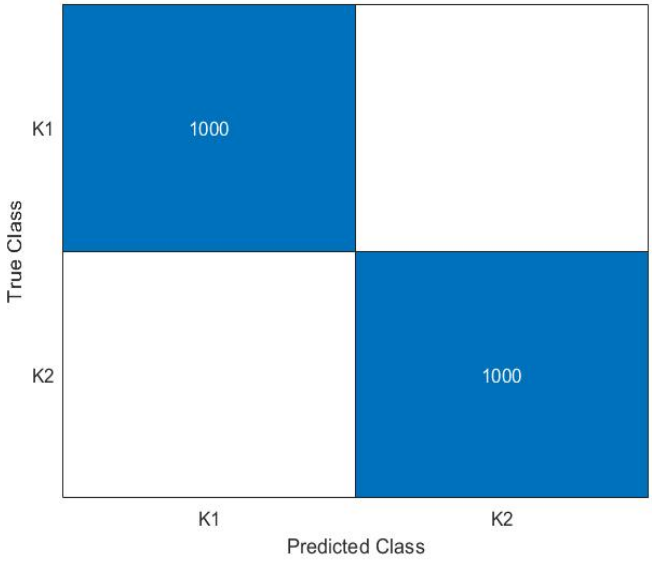


Projektovanje kvadratnog klasifikatora se može izvršiti tako što se on svede na linearni, a zatim se primene jedne od postojećih metoda.
Kvadratni klasifikator ima oblik:

$$X^T Q X + V^T X + v_0 = 0$$

Možemo ga svesti na linearni tako što u matricu V pored linearnih članova dodamo i članove drugog reda odnosno $V = [x_1^2 x_2^2 \dots x_n^2 x_1 x_2 \dots x_{n-1} x_n x_1 \dots x_2]^T$. Sada ćemo projektovati klasifikator metodom željenog izlaza.
Rezultati izgledaju ovako:


Raspored odabiraka
K1
K2
klas. kriva
x2[]
x1[]
8
6
4
2
0
-2
-4
-6
-8


1000
1000
K1
K2
True Class
K1
K2
Predicted Class

4 Zadatak 4

4.1 Postavka problema

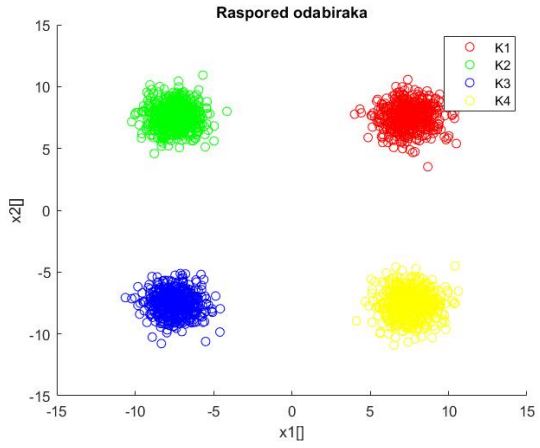
1. Generisati po $N = 500$ dvodimenzionih odbiraka iz četiri klase koje će biti linearno separabilne. Preporuka je da to budu Gausovski raspodeljeni dvodimenzioni oblici. Izabrati jednu od metoda za klasterizaciju (c mean metod, metod kvadratne dekompozicije) i primeniti je na formirane uzorke klasa. Izvršiti analizu osetljivosti izabranog algoritma na početnu klasterizaciju kao i srednji broj potrebnih iteracija. Takođe izvršiti analize slučaja kada se apriorno ne poznaje broj klasa.

2. Na odbircima iz prethodne tačke izabrati jednu od metoda klasterizacije (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 apriorno ne poznaje broj klasa.

3. Generisati po $N = 500$ dvodimenzionih odbiraka iz dve klase koje su nelinearno separabilne. Izabrati jednu od metoda za klasterizaciju koje su primenjive za nelinearno separabilne klase (metod kvadratne dekompozicije ili metod maksimalne verodostojnosti) i ponoviti analizu iz prethodnih tačaka.

4.2 Rešenje

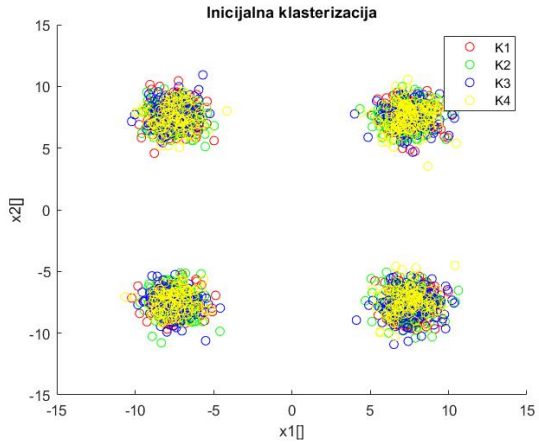
Klasterizacija je oblik učenja bez nadgledanja odnosno vid učenja kada nam odabirci nisu labelirani već ih pridružujemo klasama na osnovu njihovih sličnosti. Generisali smo odabirke koji pripadaju 4 različite klase, ali to koji odabirak kojoj klasi pripada nije bilo poznato algoritmu za klasterizaciju.

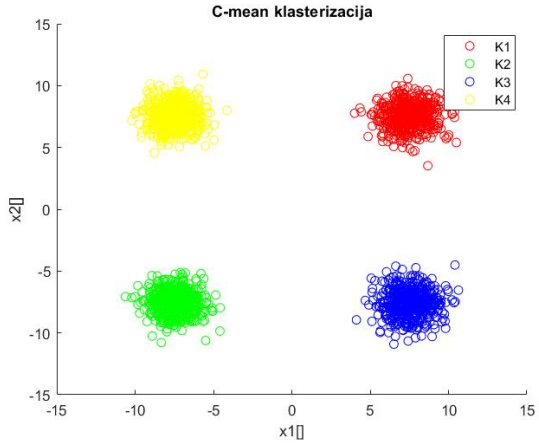


4.2.1 C-mean klasterizacija

C-mean klasterizacija je metod koji se primenjuje kada su klasteri linearno separabilni. Svaki odabirak pridružuje onom klasteru koji mu je centar najbliži(slično kao i klasifikator distance). To povlači da su granice između klastera krive prvog reda.

Algoritam se vrti sve dok se odabirci reklasifikuju, prvi put kada se to ne desi algoritam se prekida. Treba napomenuti da postoji opasnost od oscilacija ali to se u našem slučaju nije događalo.
Velika prednost C-mean klasterizacije jeste njena numerička jednostavnost, ali i to što je manje osetljiva na inicijalnu klasterizaciju (može biti i nasumična) u odnosu na druge metode. Sada su prikazane inicijalna klasterizacija i klasterizacija nakon C-mean algoritma. Još treba napo-





menuti da je prosečcan broj iteracija bio oko 3, ali i da ni ovaj metod nije potpuno neosetljiv na inicijalnu klasterizaciju jer se u nekim slučajevima dešavalo da spoji 2 klastera i tako završ sa 3 umesto 4 klastera.
Takođe za izvršavanje ovog algoritam je neophodno definisati broj klaster unapred. Ako je on nepoznat moguće ga je proceniti na osnovu "lakat"metoda. Prvo skiciramo zavisnost WCSS(Within Cluster Sum of Square) u zavisnosti od broja klastera, a zatim uzmemo onaj broj klastera tamo gde se nalazi lakat krive odnosno tamo gde dinamičnost krive značajno opada.

4.2.2 ML klasterizacija

Maximum Likelihood je metod gde se pretpostavlja da odabirci potiču iz polimodalne Gausove raspodele što je veoma dobra pretpostavka jer i u slučaju kada oni zaista ne potiču iz Gausove raspodele za veliki broj odabiraka centralna granična teorema nam govori da će raspodela ličiti na Gasovu. Kriterijumska funkcija izgleda ovako:

$$J = maxf(x_1, x_2, \ldots, x_N)$$

Aposteriornu verovatnoću računamo kao:

$$q_i = \frac{p_i f_i}{f}$$

Sada je potrebno naći izvode po p_i,M_i i Σ_i i izjednačiti ih sa nulom.
Nakon svega ovoga dobijamo jednačine:

$$P_i = \frac{1}{N} \sum_{j=1}^{N} q_i(X_j)$$

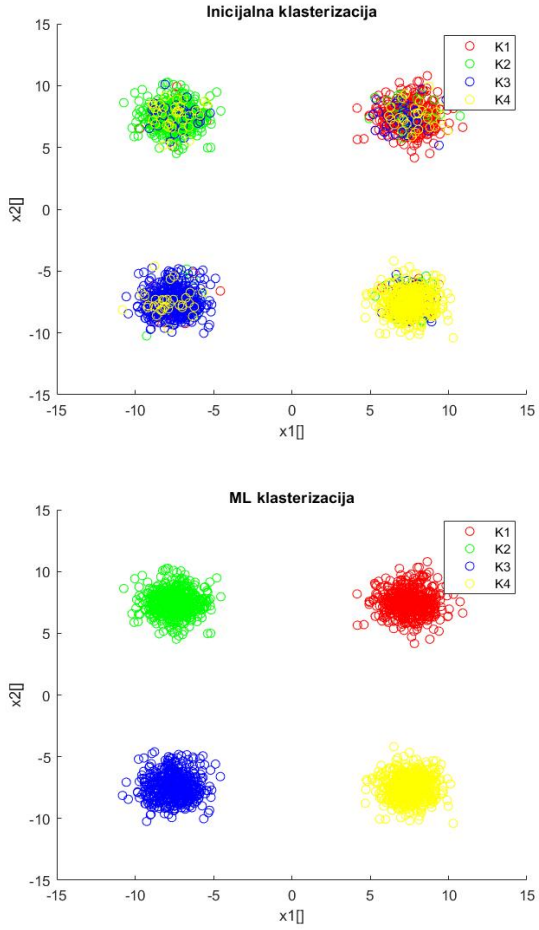
$$M_i = \frac{1}{N_i} \sum_{j=1}^{N} q_i(X_j)$$

$$\Sigma_i = \frac{1}{N_i} \sum_{j=1}^{N} q_i(X_j)(X_j - M_i)(X_j - M_i)^T$$

Gornje jednačine zajedno sa jednačinom za aposteriornu verovatnoću čine sistem koji je potrebno rešiti, a zatim odabirak smestiti u klasu sa maksimalnom aposteriornom verovatnoćom. Kako su jednačine isprepletane njih nije moguće rešiti analitički već pribegavamo sledećom iterativnom metodom:

1. Formiramo inicijalnu klasterizaciju
2. Na osnovu uzorka procenimo P_i,M_i i Σ_i
3. Izračunamo q_i
4. Izračunamo P_i,M_i i Σ_i na osnovu q_i
5. Izračunamo q_i sa novim parametrima
6. Reklasifikujemo odabirke tako da oni odgovaraju onoj klasi gde je q_i maksimalno
7. Ako je maksimalna promena q u odnosu na prethodnu iteraciju manja od nekog praga završavamo, a ako nije vraćamo se na korak 4

Algoritam nije davao zadovoljavajuće rezultate sa nasumičnom inicijalnom klasterizacijom. U realnom slučaju bi pre njega bilo dobro primeniti neku jednostavniju metodu kako bi generisali inicijalnu klasterizaciju (na primer C-mean), ali kako u našem slučaju C-mean već perfektno klasterizuje odabirke ovo ne bi imalo smisla te smo potpomogli ovaj algoritam tako što smo jedan deo odabiraka već smestili u odgovarajuće klastere.
Izgled inicijalne i finalne klasterizacije je dat na slikama: Prosečan broj potrebnih iteracija je bio oko 4.

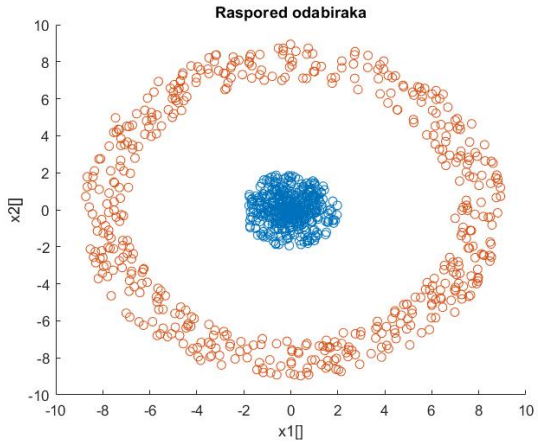


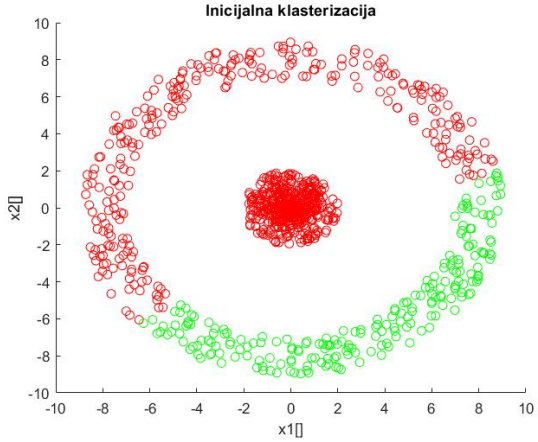
4.2.3 Kvadratna dekompozicija

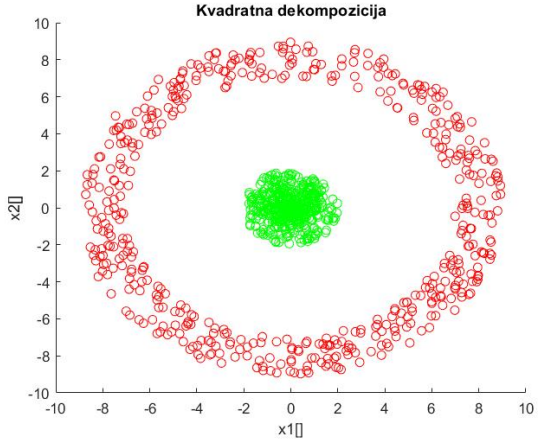
Kada odabirci nisu linearno separabilni primena C-mean klasterizacije nije u opticaju. U ovom slučaju podaci su kvadratno separabilni te je moguće primeniti kvadratnu dekompoziciju. Mana ove metode je njena numerička složenost kao i to što je veoma osetljiva na početnu klasterizaciju. Zbog ovog drugog pre same kvadratne dekompozicije primenjena je C-mean klasterizacija pa je njen izlaz upotrebljen kao inicijalna klasterizacija za kvadratnu dekompoziciju.
Algoritam za kvadratnu dekompoziciju je isti kao i za C-mean samo sa drugom kriterijumskom funkcijom. Naime, sada kriterijumska funkcija pored matematičkih očekivanja u obzir uzima i kovariacione matrice i izgleda ovako:

$$J = -\frac{1}{2} ln p_i + \frac{1}{2} ln|\Sigma_i| + \frac{1}{2}(X - M_i)^T \Sigma_1^{-1}(X - M_i)$$

Generisani odabirci izgledaju ovako: Inicijalna klasterizacija: Krajnja klasterizacija: Prosečan broj iteracija je oko 7 što je više nego duplo više nego za C-mean.


Raspored odabiraka
x2[]
x1[]


Inicijalna klasterizacija
x2[]
x1[]


Kvadratna dekompozicija
x2[]
x1[]