Seminarski rad iz Inteligentnog upravljanja- Quadratic classifiers

Autor: Emina Hasanović

***Uvod* — U ovom seminarskom radu obradit će se "Quadratic classifiers" (eng. - Kvadratični klasifikator) metoda koja se koristi u mašinskom učenju i statističkoj klasifikaciji kako bi se razdvojili objekti ili događaji prema nekim mjerama koristeći četverougaonu površinu. Ova metoda zapravo predstavlja širu verziju linearnog klafisikatora.**

Ključne riječi: mašinsko učenje, klasifikacija,LDA metoda, QDA metoda

***Abstract* — In this paper will be analyzed one method, "Quadratic classifiers" which is often used in machine learning and statistical classification to separate measurements of two or more classes of objects or events by a quadric surface. It is more general version of linear classifer.**

Key words: machine learning, classification, LDA method, QDA method

I. UVOD

1.1.Opis problema

Mašinsko učenje je multridisciplinarna oblast, koja koristi rezultate raznih drugih oblasti, kao što su vjerovatnoća i statistika, vještačka inteligencija, teorije kompleksnosti, itd. Pod mašinskim učenjem podrazumijevamo svojstvo programa da poboljša svoju performansu sa iskustvom. Problemi mašinskog učenja su razni, prepoznavanje raznih objekata, lica, slika, klasifikacija velikog broja podataka, itd. Prilikom korištenja bilo koje moetode mašinskog učenja, potrebno je prvenstvno odabrati tip iskustava iz kojeg će algoritam učiti (tj. određeni broj podataka), ciljnu funkciju, odnosno tačan tip znanja koji sistem treba naučiti, reprezentaciju ciljne funkcije i algoritam za aproksimaciju ciljne funkcije (mehanizam učenja).
S obzirom da se u ovom dokumentu obrađuju samo klafikacije podataka i kvadratični klasifikator, u nastavku je data podjela klasifikatora u dvije grupe, te su nabrojani neki predstavnici klasifikatora:

- Klasifikatori koji koriste omjer vjerovatnoće
  - *LDA (eng. **L**inear **D**iscriminant **A**nalysis)*
  - *QDA (eng. **Q**uadratic **D**iscriminant **A**nalysis)*
  - *Naïve-Bayes-ov klasifikator*
- Diskriminativni klasifikatori
  - Logistička regresija
  - Algoritam percepcije
  - Klasifikacijska i regersijska stable
  - Klasifikator na bazi najbližih susjeda
  - Neuronske mreže

Tema ovog seminarskog rada je kvadratični klasifikator koji je baziran na QDA metodi koja će u narednom poglavlju biti detaljno objašnjena. [HYPERLINK "" \l "DMi94" 1]

1.2.Moguće aplikacije u praksi

QDA metoda se koristi kada pri klasifikaciji podataka nije moguće ostvariti linearnu granicu među formiranim grupama podataka nego se umjesto toga koristi nelinearna granica različitih oblika, kao npr. paraboličnog oblika.
Ovaj klasifikator se koristi za grupisanje i prepoznavanje brojeva i slova različitih rukopisa, prepoznavanje i grupisanje drugih 'patterna' itd. [HYPERLINK "" \l "Šuc13" 1]

II. KORIŠTENI ALGORITAM

2.1.Uvod

U predikcijskom učenju, problem je predvidjeti/odrediti vrijednost izlaza (vrijednost odziva u regresiji ili vrijednost labele u klasifikaciji), varijablu Y iz vrijednosti ulaznih, observiranih varijabli **X** (prediktora, uzoraka, atributa), $(X_1, X_2, \dots, X_n) = \mathbf{X}$ Smatramo da $X \in R^n$. Problemi predikcije su klasificirani na osnovu tipa $Y \in y$.
U klasifikaciji, izlaz Y se naziva klasna varijabla ili klasna labela, nekada se označava i sa C.
Postoje tri faze u klasifikaciji podataka:

1) FAZA TRENIRANJA
   Pronađemo početne podatke ('labeled data'), $D = \{(x_1,c_1),(x_2,c_2),\dots,(x_n,c_n)\}$, i odredimo koji ćemo tip klasifikatora koristiti. Zatim estimiramo klasifikator ϕ iz podataka D, tj. cilj je pronaći klasifikator ϕ koji dobro klasificira podatke D.

2) FAZA VALIDACIJE

   Provjerimo koliko je stvarno dobar klasifikator ϕ mjerenjem različitih grešaka pri klasifikaciji itd.
   Pronađemo novi set podataka D_v i provjerimo koliko dobro klasifikator ϕ klasificira nove podatke.

Ukoliko ϕ loše klasificira nove podatke (provjerimo greške pri klasifikaciji) odredimo neki novi klasifikator.

3) FAZA TESTIRANJA
Kada smo konačno odredili dobar klasifiator, koristimo ga. Novim, ulaznim (nelabeliranim) podacima x, pridružimo izlaz kao (formula 2.1.):

$$\hat{c}(x) = \phi(x) \tag{2.1}$$

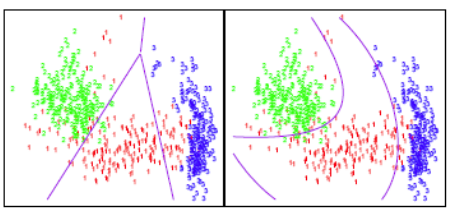
tj. $\hat{c}(x)$ zovemo predicirana labela od x.

U narednom potpoglavlju ćemo opisati linearni I kvadratični klasifikator. [HYPERLINK "" \l "DMi94" 1]

2.2.Opis rada korištenog algoritma

Kako bismo razmatrali kvadratični klasifikator potrebno je opisati i linearni klasifikator i razlike među njima.

Prvi i osnovna razlika između linearnog klasifikatora je ta što linearni klasifikator formira linearnu granicu između klasa dok kvadratični klasifikator formira nelinearnu granicu koja je uglavnom paraboličnog oblika. (slika 2.1.)



Slika 2.1. Razlika između linearnog klasifikatora (lijevo) i kvadratičnog klasifikatora (desno)

A. Linearni klasifikator

Linearni klasifikator je baziran na LDA metodi (metoda linearne diskriminativne analize).

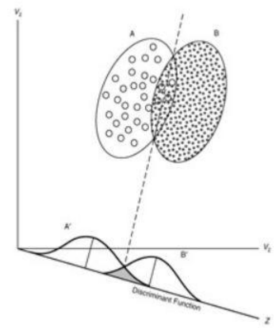
LDA metoda računa diskriminantni "uspjeh" svake obzervacije kako bi je kasificirala u određenu klasu. Ove vrijednosti "uspjeha" obzervacija su dobivene iz linearne kombinacije nezavisnih varijabli, pa je za jednu predikcijsku vrijednost X = x LDA klasifikator estimiran kao:

$$\hat{\delta}(x) = x \cdot \frac{\widehat{\mu_k}}{\widehat{\sigma^2}} - \frac{\widehat{\mu_k^2}}{2\widehat{\sigma^2}} + \log\widehat{\pi_k} \tag{2.2}$$

gdje je

- $\hat{\delta}(x)$ estimirani diskriminativni "uspjeh" da će obzervirana varijabla upasti u k-tu klasu sa određenom izlaznom vrijednošću baziranoj na vrijednosti predikcijske varijable x.
- $\widehat{\mu_k}$ je prosjek svih obzervacija za treniranje iz k-te klase
- $\widehat{\sigma^2}$ je težinski prosjek varijansi uzorka za svaku od K klasa
- $\widehat{\pi_k}$ je vjerovatnoća da obzervacija pripada k-toj klasi (to nije matematička konstanta π).

Klasifikator će dodijeliti obzervaciju k-toj klasi sa izlaznom vrijednošću Y_k za koju je diskriminativni "uspjeh" najveći. Naprimjer, ako imamo dvije klase A i B za izlaznu varijablu Y. Na osnovu nekih ulaznih vrijednosti (predikcijskih varijabli) LDA će izračunati vjerovatnoću raspodjele za klasu A i klasu B. Granica između vjerovatnoća dsitribucije bit će predstavljena pravom linijom. Diskriminativni "uspjesi" lijevo od granice bit će pridruženi klasi A, a desno od granice klasi B (slika 2.2.).



Slika 2.2. Linearna granica među klasama A i B

Ukoliko imamo vise od jedne predikcijskih varijabli , onda LDA klasifikator pretpostavi da su obzervacije u k-toj klasi rapodjeljene multivarijablinom Gausovom raspodjelom N(μ_k, Σ), gdje je μ_k vector srednjih vrijednosti specifičan za svaku klasu, dok je Σ matrica kovarijanski koja je zajednička za sve klase K.Ukoliko se ovo unese u jednačinu za LDA klasifikator dobije se:

$$\widehat{\delta_k}(x) = x^T\Sigma^{-1}\widehat{\mu_k} - \frac{1}{2}\widehat{\mu_k^T}\Sigma^{-1} - \widehat{\mu_k} + \log(\widehat{\pi_k}) \tag{2.3.}$$

gdje će obzervacija biti dodijeljena klasi k gdje diskriminativni "uspjeh" iz jednačine (2.2.) $\widehat{\delta_k}(x)$ najveći.
[HYPERLINK "" \l "Tom97" 1]

B. Nelinearni (kvadratni) klasifikator

Kao što je pomenuto u predhodnom dijelu za LDA metodu, LDA pretpostavlja da su obzervacije u unutar svake od klasa raspoređene multivarijabilnom Gausovom raspodjelom I kovarijansa predikcijskih varijabli su zajedničke za sve k nivoe izlazne varijable Y. Kvadratni klasifikator je baziran na *QDA (eng. **Q**uadratic **D**iscriminant **A**nalysis)* metodi.

QDA daje alterativni pristup ovom problem, tj. predstavlja općenitiju verziju LDA metode. QDA kao i LDA pretpostavlja da su obzervacije unutar svake od klasa izlaza Y raspoređene Gausovom raspodjelom. Međutim, za razliku od LDA, QDA podrazumijeva da svaka od klasa ima posebno svoju matricu kovarijanse. Drugim riječima, predikcijske varijable nemaju zajedničku varijansu duž svih k nivoa u izlazu Y.Matematički, za obzervaciju iz k-te klase vrijedi X~ N(μ_k, Σ_k), gdje je Σ_k matrica kovarijanse za k-tu klasu. Na osnovu predhodnog, klasifikator dodjeljuje obzervaciju klasi za koju je:

$$\widehat{\delta_k}(x) = -\frac{1}{2}x^T\Sigma_k^{-1}x + x^T\Sigma_k^{-1}\widehat{\mu_k} - \frac{1}{2}\widehat{\mu_k^T}\Sigma_k^{-1}\widehat{\mu_k} - \frac{1}{2}\log|\Sigma_k| + \log(\widehat{\pi_k}) \quad (2.4.)$$

najveći. Zašto je ovo bitno? Ako posmatramo sliku 2.1., pokušaj da se obzervacije podlijeve u tri klase (kodirane bojom) .LDA, lijevi dio slike 2.1., iscrtava linearnu granicu što je bazirano na prepostavci da obzervacije/uzorci varijraju konstantno kroz sve tri klase. Međutim ako detaljnije posmatramo podatke vidjet ćemo da je očigledno da se varijabilnost između pojedinih klasa mijenja. Slijedi, QDA (desni dio slike 2.1.) može "uhvatiti" promjenu kovarijansi i formirati precizniju nelinearnu granicu među klasama.
[HYPERLINK "" \l "Tom97" 1]

U nastavku će biti objašnjena implementacija prehodnih metoda.

2.3.Svođenje opisanog problema u formu korištenog algoritma

Linearni i kvadratni klasifikator su implementirani u programskom okruženju MATLAB, koji ima jako puno alata i već implementiranih funkcija za mašinsko učenje, razne metode mašinskog učenja, kao i setove podataka za treniranje, validaciju i testiranje svih poznatih vrsta klasfikatora.

Podaci koji su odabrani za treniranje i testiranje je poznati set podataka koji je zove Fisher-ov iris (iris cvijet). Ovaj set podataka je prikupljen od strane britanskog statističara i biologa Ronalda Fishera-a 1936. godine. Fisher-ov iris se korisiti upravo za probleme linearne i kvadratne diskiminativne analize, tj. predstavlja tipični testni set za mnoge statistične klasifikacijske tehnike u mašinskom učenju.
Fisher-ov iris se sastoji od „species“ , tj. vektora kolone vrsta irisovog cvijeta i postoje tri, a to su:

- Iris setosa (slika 2.3)
- Iris virginica (slika 2.4.)
- Iris versicolor (slika 2.5.)

U matrici „meas“ su smještene mjere čašićnih listova i latica za sve tri vrste cvijeta, tj. širina i dužina čašičnih listova, te širina i dužina latica.

Ovaj set podataka ima 150 uzoraka (3 klase, 50 uzoraka za svaku klasu). [HYPERLINK "" \l "Dua17" 1]

Tri vrste irisovog cvijeta su prikazane na sljedeće tri slike:

Slika 2.3. Iris setosa

Slika 2.4. Iris versicolor

Slika 2.5. Iris virginica

Zadatak je da se klasificiraju Irisovi cvjetovi u tri grupe, koristeći razne kriterije, npr. dužinu i širinu latice ili dužinu i širinu čašičnih listova i slično.

2.4.Zaključak

Bit će korištena oba algoritma, LDA i QDA te ćemo izvršiti njihovu usporedbu. Koristit ćemo različite kriterije kako bismo uporedili performanse oba kalsifikatora, kao što su:

- Greška pri razvrstavanju u klase (greška pri treniranju podataka)
- Testna greška, tj. greška pri testiranju grupe podataka
  Itd.

III. Impelentacija QDA i LDA metode u Matlabu

3.1.Uvod

U ovom poglavlju ćemo prikazati primjenu metoda obrađenih u prehodnom poglavlju, tj. prikazat ćemo kako se to određena grupa podataka klasificira u mašinskom učenju prolazeći kroz sve tri faze (nabrojane u poglavlju 2, potpoglavlje 2.1.): treniranje, validacija i testiranje.

3.2.Treniranje, testiranje i validacija

Set podataka Fisher's Iris postoje već u Matlabu, potrebno ih je samo pozvati kao:

```
load fisheriris
```

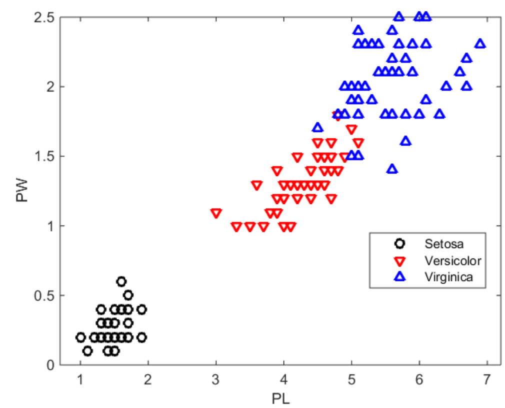
Kako smo već ranije spomenuli u poglavlju 2, potpoglavlje 2.3., set podataka se sastoji od vektora kolone „species“ i matrice mjerenja „meas“ pa je u startu potrebno izdvojiti željene podatke prema kojima ćemo grupirati date klase „species“, to mogu biti ili latice i njhove mjere ili čašični listovi i njhove odgovarajuće mjere, ili neke kombinacije također zavisno koja kalsifikacija je potrebna.

U prvom dijelu zadatka razvrstat ćemo klase cvjetova na osnovu dužine i širine latica (iz matrice „meas“ izdvojimo treću i četvrtu kolonu):

```
%vektor kolona 'species' se sastoji od tri
vrste cvijeta iris : setosa,
%versicolor i virginica
%matrica 'meas' se sastoji od mjerenja dužine
i širine casicnih listova (prve dvije kolone)
i
%dužine i širine latica cvijeta (druge dvije
kolone)
PL = meas(:,3);
PW = meas(:,4);
```

Nakon toga iscrtamo klase cvjetova iris raspoređene prema dužini i širini latica. To je prikazano na slici 3.1. Isctavamo pomoću:

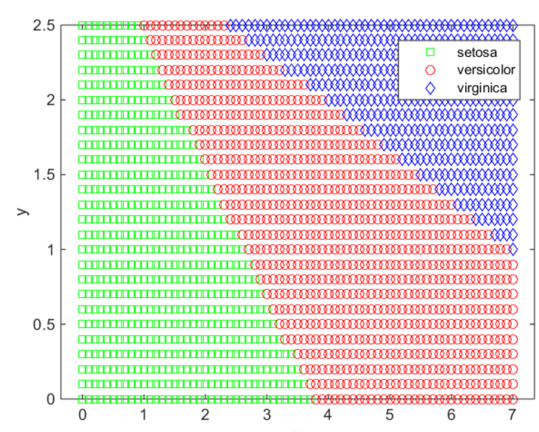
```
%plotanje podataka i klasifikacija po vrstama
tj. 'species'
figure(1);
h1 =
gscatter(PL,PW,species,'krb','ov^',[],'off');
h1(1).LineWidth = 2;
h1(2).LineWidth = 2;
h1(3).LineWidth = 2;
legend('Setosa','Versicolor','Virginica','Loc
ation','best')
```



Slika 3.1. Podjela vrsta irisa na osnovu dužine i širine latica

Moguće je iscrtati područja svake od klasa koristeći određene funkcije za klasifikaciju koje su već implementirane u Matlabu, ovo je prikazano na slici 3.2.:

```
figure(2)
%isrtavanje podrucja koji pripadaju
razlicitim klasama
[x,y] = meshgrid(0:.1:7,0:.1:2.5);
x = x(:);
y = y(:);
j = classify([x y],meas(:,3:4),species);
gscatter(x,y,j,'grb','sod')
```



Slika 3.2. Područja klasa/vrsta cvjetova

Sljedeći korak je primjena linarnog ili kvadratnog klasifikatora na ovaj set podataka.

Čitav set podataka „fisheriris“ predstavlja set podataka za treniranje oba klasifikatora, linearnog i kvadratnog .

Treniranje linearnog klasifikatora je implementirano u Matlabu kao:

```
%kreiranje linearnog klasifikatora
X = [PL,PW];
MdlLinear = fitcdiscr(X,species);
ldaClass = resubPredict(MdlLinear);
```

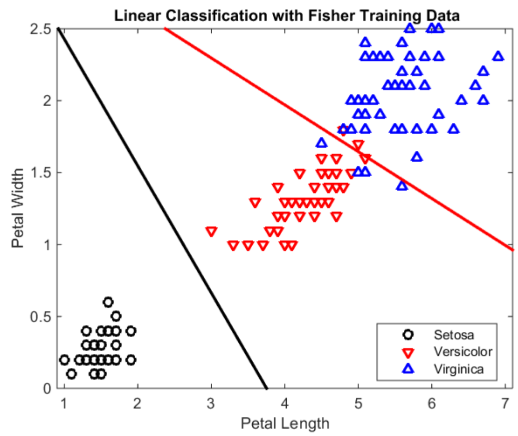
Treniranje kvadratnog klasifikatora je implementirano kao:

```
X = [PL,PW];
%kreiranje kvadraticnog klasifikatora
MdlQuadratic =
fitcdiscr(X,species,'DiscrimType','quadratic')
qdaClass = resubPredict(MdlQuadratic)
```

Nakon treniranja klasifikatora izračunati su odgovarajući koeficjenti koji su potrebni da bi se iscrtane linearne i parabolične granice među uzorcima pojedinih klasa. Računanje linearnih koeficjenata je dato u nastavku:

```
%odredjivanje koeficjenata za linearnu
granicu izmedju druge i trece klase
MdlLinear.ClassNames([2 3])
K = MdlLinear.Coeffs(2,3).Const;
L = MdlLinear.Coeffs(2,3).Linear;

figure(3)
h1 =
gscatter(PL,PW,species,'krb','ov^',[],'off');
h1(1).LineWidth = 2;
h1(2).LineWidth = 2;
h1(3).LineWidth = 2;

legend('Setosa','Versicolor', 'Virginica','Lo
cation','best')
hold on
%plotanje linije izmedju druge i trece klase
f = @(x1,x2) K + L(1)*x1 + L(2)*x2;
h2 = ezplot(f,[.9 7.1 0 2.5]);
h2.Color = 'r';
h2.LineWidth = 2;

%odredjivanje koeficjenata za linearnu
granicu izmedju prve i druge klase
MdlLinear.ClassNames([1 2])
K = MdlLinear.Coeffs(1,2).Const;
L = MdlLinear.Coeffs(1,2).Linear;

%plotanje linije koja razdvaja prvu i drugu
klasu
f = @(x1,x2) K + L(1)*x1 + L(2)*x2;
h3 = ezplot(f,[.9 7.1 0 2.5]);
h3.Color = 'k';
h3.LineWidth = 2;
axis([.9 7.1 0 2.5])
xlabel('Petal Length')
ylabel('Petal Width')
title('{\bf Linear Classification with Fisher
Training Data}')
hold on;
```

Iz predhodnog koda primjećujemo da je prvo iscrtana linearna granica između druge i treće klase pa između prve i druge klase (slika 3.3).



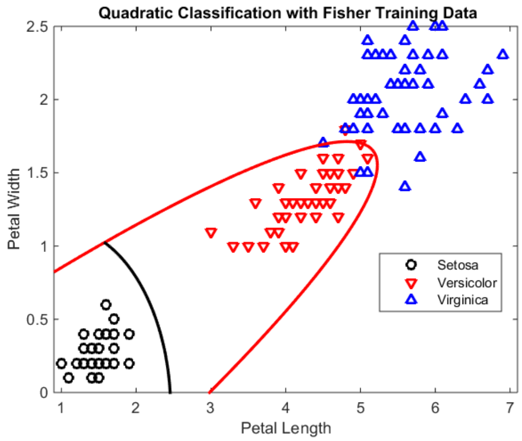
Slika 3.3. Linearna klasifikacija

Računanje paraboličnih koeficjenata je dato u nastavku:

```
%odredjivanje koeficjenata za formiranje
parabolnicne granice medju klasama
%2 i 3
MdlQuadratic.ClassNames([2 3])
K = MdlQuadratic.Coeffs(2,3).Const;
L = MdlQuadratic.Coeffs(2,3).Linear;
Q = MdlQuadratic.Coeffs(2,3).Quadratic;

figure(3);
h1 =
gscatter(PL,PW,species,'krb','ov^',[],'off');
h1(1).LineWidth = 2;
h1(2).LineWidth = 2;
h1(3).LineWidth = 2;
legend('Setosa','Versicolor','Virginica','Loc
ation','best')
hold on;
%plotanje parabole izmedju druge i trece
klase
f = @(x1,x2) K + L(1)*x1 + L(2)*x2 +
Q(1,1)*x1.^2 + ...
    (Q(1,2)+Q(2,1))*x1.*x2 + Q(2,2)*x2.^2;
h2 = ezplot(f,[.9 7.1 0 2.5]);
h2.Color = 'r';
h2.LineWidth = 2;
hold on;

%odredjivanje koeficjenata za formiranje
parabolicne granice izmedju klasa 1
%i 2
MdlQuadratic.ClassNames([1 2])
K = MdlQuadratic.Coeffs(1,2).Const;
L = MdlQuadratic.Coeffs(1,2).Linear;
Q = MdlQuadratic.Coeffs(1,2).Quadratic;

%plotanje parabole izmedju klase 1 i klase 2

f = @(x1,x2) K + L(1)*x1 + L(2)*x2 +
Q(1,1)*x1.^2 + ...
    (Q(1,2)+Q(2,1))*x1.*x2 + Q(2,2)*x2.^2;
h3 = ezplot(f,[.9 7.1 0 1.02]);
h3.Color = 'k';
h3.LineWidth = 2;
axis([.9 7.1 0 2.5])
xlabel('Petal Length')
ylabel('Petal Width')
title('{\bf Quadratic Classification with
Fisher Training Data}')
```

```
hold off
```

Parabolične granice među klasama su prikazane na slici 3.4.:



Slika 3.4. Kvadratna klasifikacija

Kako bismo validirali da li su klasifikatori, LDA i QDA, dobro odredili granicu među klasama, računat ćemo grešku raspodjele obzervacija pa poznatom klasom, tj. grešku raspodjele podataka za treniranje kao:

- Za linearni klasifikator:

```
ldaResubErr = resubLoss(MdlLinear)
```

Dobije se vrijednost greške:

```
ldaResubErr = 0.0400
```

- Za kvadratni klasifikator:

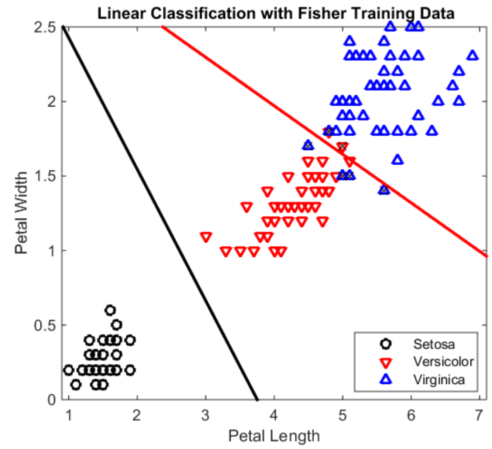
```
qdaResubErr = resubLoss(MdlQuadratic)
```

Dobije se vrijednost greške:

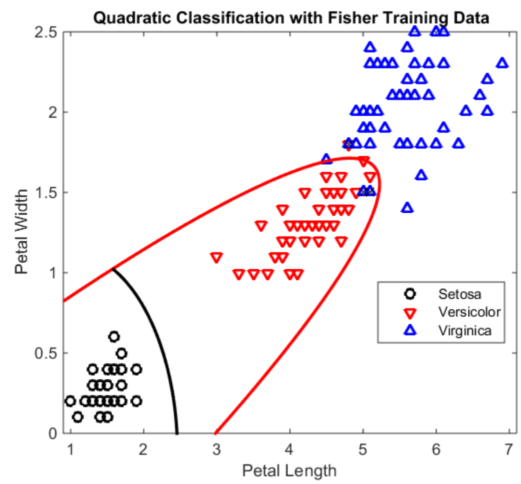
```
qdaResubErr = 0.0200
```

Iz predhodnih podataka u startu vidimo da je greška pri kvadratnom klasifikatoru dosta manja, pa on znatno bolje klasificira podatke za treniranje.
Možemo i grafički prikazati koliko to uzoraka za treniranje nakon klasifikacije nije pripalo svojoj u startu poznatoj klasi. To je prikazano na slici 3.5. i na slici 3.6. za linearni i kvadratni klasifikator respektivno.
Kod:

```
bad = ~strcmp(ldaClass,species);
hold on;
plot(meas(bad,3), meas(bad,4), 'kx');
```



Slika 3.5. Linearni klasifikator, pogrešno grupirani uzorci su precrtani sa „x“ .



Slika 3.6. Kvadratni klasifikator, pogrešno grupirani uzorci su precrtani sa „x“

Drugi kriterij validacije je konfuzijska matrica. Računanje konfuzijske matrice (daje informacije o poznatnoj i predviđenoj klasi uzorka) tj. (i,j) elemenat u matrici predstavljaja broj uzoraka kod kojih je poznata klasa i, a predvidjana klasa j, pa elementi na dijagonali predstavljaju korektne klasifikacije.

- Za linearni klasifikator:

Kod:

```
[ldaResubCM,grpOrder]=
confusionmat(species,ldaClass)
```

Rezultat:

```
ldaResubCM =

    50     0     0
     0    48     2
     0     4    46


grpOrder =

    'setosa'
```

```
    'versicolor'
    'virginica'
```

- Za kvadratni klasifikator:

Kod:

```
[qdaResubCM,grpOrder] =
confusionmat(species,qdaClass)
```

Rezultat:

```
qdaResubCM =

    50     0     0
     0    49     1
     0     2    48


grpOrder =

    'setosa'
    'versicolor'
    'virginica'
```

Vidimo da je kod kvadratnog klasifikatora puno više elemenata korektno klasificirano nego kod linearnog klasifikatora, tj. manje je elemenata izvan glavne dijagonale kod kvadratnog klasifikatora, ukupno samo tri, dok je kod linearnog klasifikatora taj broj veći, ukupno 6.

S obzirom da ovdje nemamo dodatnih podataka za testiranje nakon što smo uradili treniranje i validaciju klasifikatora možemo početni podatkovni set „fisheriris“ da podijelimo u nekoliko grupa pa da jedan dio podataka uzmemo kao dio podataka za testiranje i provjerimo opet naknadno performans klasifikatora.

Razdvojit ćemo početni set podataka „fisheriris“ na 10 podgrupa otprilike iste veličine i iste raspodjele podataka u klase kao kod orginalnog, početnog seta podataka. Nakon toga 9 podgrupa podataka će se koristiti za treniranje dok će 1 podgrupa podataka biti korištena za testiranje.

- Za linearni klasifikator:

```
rng(0,'twister');
cp = cvpartition(species,'KFold',10)
```

Rezultat:

```
cp =
K-fold cross validation partition
NumObservations: 150
NumTestSets: 10
TrainSize: 135  135  135  135  135  135
135  135  135  135
TestSize: 15  15  15  15  15  15  15
15  15  15
```

- Za kvadratni klasifikator:

```
rng(0,'twister');
cp = cvpartition(species,'KFold',10)
```

Rezultat:

```
cp =
K-fold cross validation partition
NumObservations: 150
NumTestSets: 10
TrainSize: 135  135  135  135  135  135
135  135  135  135
TestSize: 15  15  15  15  15  15  15  15
15  15
```

Dalje ćemo koristiti fukcije `crossval` i `kfoldLoss` da estimiramo grešku nekorektne klasifikacije pri testiranju podataka koristeći kreiranu particiju `cp`.

- Za linearni kalsifikator računamo kao:

```
%odredjivanje testne greške
cvlda =
crossval(MdlLinear,'CVPartition',cp);
ldaCVErr = kfoldLoss(cvlda)
```

I dobijemo da je testna greška:

```
ldaCVErr = 0.0467
```

- Za kvadratni klasifikator računamo kao:

```
cvqda =
crossval(MdlQuadratic,'CVPartition',cp);
qdaCVErr = kfoldLoss(cvqda)
```

Dobijemo testnu grešku:

```
qdaCVErr = 0.0267
```

Uočavamo iz predhodnih vrijednosti grešaka da je ponovo kvadratni klasifikator imao bolju performansu i pri testiranju određene grupe podataka.

[HYPERLINK "" \l "Off"   1]

3.3.Zaključak

Iz predhodnog potpoglavlja (3.2.) vidimo da kvadratni klasifikator ima uglavnom bolje performanse nego linerani klasifikator iako je korišteni set podataka relativno mali. Moguće je da pri drugim klasifikacijama, tipa klasifikacija po dužini i širini čašičnih listova, bolju performansu ima linearni klasifikator. Razlog je činjenica da je broj podataka mali i jasno su uočljive razlike među pojedinim klasama.

IV. Zaključak i diskusija

U ovom seminarskom radi obrađen je jako zanimljiv klasifikator koji ima višestruku primjenu, prije svega jer dosta generalniji od običnih linearnih klasifikatora. Koristi se u slučajevima kada se operira nad dosta većim i komplikovanijim setom podataka kada klasne razlike i granice nisu lahko uočljive. Možda jedini nedostatak ovog klasifikatora je činjenica da ipak nije univerzalan i uvijek bolje performanse nego linearni klasifikator, ali u većini slučajeva jeste tako.

APPENDIX

Kod za linearni (LDA) klasifikator:

```
%ucitavanje podataka
load fisheriris
%vektor kolona 'species' se sastoji od tri vrste
cvijeta iris : setosa,
%versicolor i virginica
%matrica 'meas' se sastoji od mjerenja dužine i
širine ?aši?nih listova (prve dvije kolone) i
%dužine i širine latica cvijeta (druge dvije
kolone)
PL = meas(:,3);
PW = meas(:,4);
%plotanje podataka i klasifikacija po vrstama tj.
'species'
figure(1);
h1 = gscatter(PL,PW,species,'krb','ov^',[],'off');
h1(1).LineWidth = 2;
h1(2).LineWidth = 2;
h1(3).LineWidth = 2;
legend('Setosa','Versicolor','Virginica','Location
','best')
%hold on

%kreiranje linearnog klasifikatora
X = [PL,PW];
MdlLinear = fitcdiscr(X,species);
ldaClass = resubPredict(MdlLinear);
%greška pri pogrešnoj klasifikaciji uzoraka u
klase
ldaResubErr = resubLoss(MdlLinear)

%racunanje konfuzijske matrice(daje informacije o
poznatnoj i predvidjenoj
%klasi uzorka) tj. (i,j) elemenat u matrici
predstavljaja broj uzoraka kod
%kojih je poznata klasa i, a predvidjana klasa j,
pa elementi na dijagonali
%predstavljaju korektne klasifikacije
[ldaResubCM,grpOrder] =
confusionmat(species,ldaClass)

figure(2)
%isrtavanje podrucja koji pripadaju razlicitim
klasama
[x,y] = meshgrid(0:.1:7,0:.1:2.5);
x = x(:);
y = y(:);
j = classify([x y],meas(:,3:4),species);
gscatter(x,y,j,'grb','sod')


%odredjivanje koeficjenata za linearnu granicu
izmedju druge i trece klase
MdlLinear.ClassNames([2 3])
K = MdlLinear.Coeffs(2,3).Const;
L = MdlLinear.Coeffs(2,3).Linear;

figure(3)
h1 = gscatter(PL,PW,species,'krb','ov^',[],'off');
h1(1).LineWidth = 2;
h1(2).LineWidth = 2;
h1(3).LineWidth = 2;
legend('Setosa','Versicolor','Virginica','Location
','best')
hold on
%plotanje linije izmedju druge i trece klase
f = @(x1,x2) K + L(1)*x1 + L(2)*x2;
h2 = ezplot(f,[.9 7.1 0 2.5]);
h2.Color = 'r';
h2.LineWidth = 2;
%odredjivanje koeficjenata za linearnu granicu
izmedju prve i druge klase
MdlLinear.ClassNames([1 2])
K = MdlLinear.Coeffs(1,2).Const;
L = MdlLinear.Coeffs(1,2).Linear;

%plotanje linije koja razdvaja prvu i drugu klasu
f = @(x1,x2) K + L(1)*x1 + L(2)*x2;
h3 = ezplot(f,[.9 7.1 0 2.5]);
h3.Color = 'k';
h3.LineWidth = 2;
axis([.9 7.1 0 2.5])
xlabel('Petal Length')
ylabel('Petal Width')
title('{\bf Linear Classification with Fisher
Training Data}')
hold on;

bad = ~strcmp(ldaClass,species);
hold on;
plot(meas(bad,3), meas(bad,4), 'kx');
hold off;


%kreiranje testnog dijela uzoraka od seta za
treniranje jer ne posjedujemo
%jos podataka
%razdvajanje set za treniranje u 10 podgrupa,
otprilike iste velicine i
%iste raspodjele podataka u klase kao i orginalni
set za treniranje
rng(0,'twister');

%nakon poodjele 9 podskupova podataka je koristeno
za treniranje,a jedan za
%testiranje
cp = cvpartition(species,'KFold',10)

%odredjivanje testne greške
cvlda = crossval(MdlLinear,'CVPartition',cp);
ldaCVErr = kfoldLoss(cvlda)
```

Kod za kvadratni (QDA) klasifikator:

```
%ucitavanje podataka
load fisheriris
%vektor kolona 'species' se sastoji od tri vrste
cvijeta iris : setosa,
%versicolor i virginica
%matrica 'meas' se sastoji od mjerenja dužine i
širine cašicnih listova (prve dvije kolone) i
%dužine i širine latica cvijeta (druge dvije
kolone)
PL = meas(:,3);
PW = meas(:,4);
%plotanje podataka i klasifikacija po vrstama tj.
'species'
figure(1);
h1 = gscatter(PL,PW,species,'krb','ov^',[],'off');
h1(1).LineWidth = 2;
h1(2).LineWidth = 2;
h1(3).LineWidth = 2;
legend('Setosa','Versicolor','Virginica','Location
','best')
hold on;


X = [PL,PW];
%kreiranje kvadraticnog klasifikatora
MdlQuadratic =
fitcdiscr(X,species,'DiscrimType','quadratic');
qdaClass = resubPredict(MdlQuadratic)
%greška pri pogrešnoj klasifikaciji uzoraka u
klase
qdaResubErr = resubLoss(MdlQuadratic)
```

```
%racunanje konfuzijske matrice(daje informacije o
poznatnoj i predvidjenoj
%klasi uzorka) tj. (i,j) elemenat u matrici
predstavljaja broj uzoraka kod
%kojih je poznata klasa i, a predvidjana klasa j,
pa elementi na dijagonali
%predstavljaju korektne klasifikacije
[qdaResubCM,grpOrder] =
confusionmat(species,qdaClass)
%crtanje x na pogrešno klasificirane uzorke



%isrtavanje podrucja koji pripadaju razlicitim
klasama
figure(2);
[x,y] = meshgrid(0:.1:7,0:.1:2.5);
x = x(:);
y = y(:);
j = classify([x y],meas(:,3:4),species);
gscatter(x,y,j,'grb','sod')


%uklanjanje linearnih granica medju klasama
% delete(h2);
% delete(h3);

%odredjivanje koeficjenata za formiranje
parabolnicne granice medju klasama
%2 i 3
MdlQuadratic.ClassNames([2 3])
K = MdlQuadratic.Coeffs(2,3).Const;
L = MdlQuadratic.Coeffs(2,3).Linear;
Q = MdlQuadratic.Coeffs(2,3).Quadratic;

figure(3);
h1 = gscatter(PL,PW,species,'krb','ov^',[],'off');
h1(1).LineWidth = 2;
h1(2).LineWidth = 2;
h1(3).LineWidth = 2;
legend('Setosa','Versicolor','Virginica','Location
','best')
hold on;
%plotanje parabole izmedju druge i trece klase
f = @(x1,x2) K + L(1)*x1 + L(2)*x2 + Q(1,1)*x1.^2
+ ...
    (Q(1,2)+Q(2,1))*x1.*x2 + Q(2,2)*x2.^2;
h2 = ezplot(f,[.9 7.1 0 2.5]);
h2.Color = 'r';
h2.LineWidth = 2;
hold on;

%odredjivanje koeficjenata za formiranje
parabolicne granice izmedju klasa 1
%i 2
MdlQuadratic.ClassNames([1 2])
K = MdlQuadratic.Coeffs(1,2).Const;
L = MdlQuadratic.Coeffs(1,2).Linear;
Q = MdlQuadratic.Coeffs(1,2).Quadratic;

%plotanje parabole izme?u klase 1 i klase 2

f = @(x1,x2) K + L(1)*x1 + L(2)*x2 + Q(1,1)*x1.^2
+ ...
    (Q(1,2)+Q(2,1))*x1.*x2 + Q(2,2)*x2.^2;
h3 = ezplot(f,[.9 7.1 0 1.02]); % Plot the
relevant portion of the curve.
h3.Color = 'k';
h3.LineWidth = 2;
axis([.9 7.1 0 2.5])
xlabel('Petal Length')
ylabel('Petal Width')
title('{\bf Quadratic Classification with Fisher
Training Data}')
hold on

bad = ~strcmp(qdaClass,species);
hold on;
plot(meas(bad,3), meas(bad,4), 'kx');
hold off;

%kreiranje testnog dijela uzoraka od seta za
treniranje jer ne posjedujemo
%jos podataka
%razdvajanje set za treniranje u 10 podgrupa,
otprilike iste velicine i
%iste raspodjele podataka u klase kao i orginalni
set za treniranje
rng(0,'twister');

%nakon poodjele 9 podskupova podataka je koristeno
za treniranje,a jedan za
%testiranje
cp = cvpartition(species,'KFold',10)

%odredjivanje testne greške
cvqda = crossval(MdlQuadratic,'CVPartition',cp);
qdaCVErr = kfoldLoss(cvqda)
```

Reference

[1] D. J. Spiegelhalter, C. C. Taylo D. Michie, *Machine Learning, Neural and Statistical Classification*.: Prentice Hall, 1994.

[2] Tomislav Šuc, "Strojno učenje - Linearni model," PMF, Zagreb, Zagreb, 2013.

[3] Tom M. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.

[4] D. and Karra Taniskidou, E Dua, "Iris Data Set," Irvine, CA: University of California, School of Information and Computer Scienc, 2017. Available: http://archive.ics.uci.edu/ml/citation_policy.html

[5] Official MATLAB guide - MathWorks,.Available: https://www.mathworks.com/