Bayesov klasifikator u OpenCV biblioteci

Brestovac, Jelena

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

Bayesov klasifikator u OpenCV biblioteci

Završni rad

Jelena Brestovac

Osijek, 2021.

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 15.09.2021.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

Ime i prezime studenta:	Jelena Brestovac
Studij, smjer:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R4180, 23.07.2018.
OIB studenta:	37345996637
Mentor:	Izv. prof. dr. sc. . Damir Filko
Sumentor:	
Sumentor iz tvrtke:	
Naslov završnog rada:	Bayesov klasifikator u OpenCV biblioteci
Znanstvena grana rada:	**Umjetna inteligencija (zn. polje računarstvo)**
Predložena ocjena završnog rada:	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	15.09.2021.
Datum potvrde ocjene Odbora:	22.09.2021.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis: Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

IZJAVA O ORIGINALNOSTI RADA

Osijek, 24.09.2021.

Ime i prezime studenta:	Jelena Brestovac
Studij:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R4180, 23.07.2018.
Turnitin podudaranje [%]:	28

Ovom izjavom izjavljujem da je rad pod nazivom**: Bayesov klasifikator u OpenCV biblioteci**

izrađen pod vodstvom mentora Izv. prof. dr. sc. . Damir Filko

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

Sadržaj

1. UVOD

Matematika, kao jedna od najstarijih znanosti u svijetu, može na razne načine olakšati život. Mnoge se pojave mogu iskazati brojkama što čovjeku, ukoliko se dovoljno poznaju osnovne zakonitosti, omogućava da znanje o tim pojavama i proširi. Matematika daje algoritme koji uveliko pomažu i pridonose čovjeku u svakodnevici, a da pritom čovjek nije ni svjestan toga. Vjerojatnost, kao posebna grana matematike čovjeku omogućuje donošenje najpogodnijih odluka u određenom trenutku na temelju prethodnog iskustva. Tome posebno pridonosi Bayesov teorem koji je opisan u ovom radu. Prema spomenutom teoremu se može implementirati klasifikator koji je jedna od najkorisnijih primjena teorema. Računanjem pojedinih vjerojatnosti hipoteza, daje pouzdan uvid u vjerojatnost događaja uvjetovan hipotezama. Time olakšava čovjeku donošenje najpogodnije odluke u određenom trenutku kako bi došao do cilja.

1.1. Zadatak rada

Zadatak rada je objasniti rad Bayesovog klasifikatora implementiranog u OpenCV biblioteci.

2. TEORIJA VJEROJATNOSTI

Često se mogu čuti rečenice: „Vjerojatnost da će danas padati kiša je 60%.“, „Vjerojatnost da ću položiti ispit je 70% “ ili „Vjerojatnost da će Hrvatska nogometna reprezentacija pobijediti je 80%.“. U svim trima rečenicama je riječ o slučajnim pokusima, odnosno slučajevi nisu u potpunosti određeni uvjetima u kojima se pokus održava. Vjerojatnost takvih pokusa može biti problematična za izračunati. Takvim problemima se bavi teorija vjerojatnosti te je njezin osnovni zadatak formiranje i proučavanje matematičkog modela kroz povijest.

2.1. Povijesni aspekti teorije vjerojatnosti

Teorijom vjerojatnosti su se bavili mnogi matematičari, a 1560. godine talijanski liječnik, profesor geometrije i strastveni kockar Girolamo Cardano napisao knjigu „Liber de luda aleae“ („Knjiga o bacanju kocke“). Cardano se u toj knjizi bavi problemom kako u igrama na sreću izračunati vjerojatnost dobitka, odnosno bavi se kombinatornom vjerojatnosti. U knjizi je objavljena i klasična definicija vjerojatnosti definirana kao omjer broja povoljnih i broja mogućih ishoda.

1620. godine talijanski matematičar, fizičar i astronom Galileo Galilei je objavio knjigu „Razmišljanja o igrama kockom“ gdje je objašnjavao vjerojatnost različitih ishoda ako se igra dvjema kockama. Galilei u knjizi navodi da ako zajedno s jednom kockom, koja može pasti na bilo koju od šest strana, a za što je vjerojatnost podjednaka, bacamo još jednu kocku koja također ima šest strana, može se dobiti 36 različitih ishoda jer se svaka strana jedne kocke može javiti u kombinaciji sa svakom stranom druge kocke.

U 17. stoljeću su uspostavljeni temelji kombinatorne teorije vjerojatnosti u dopisivanju Pascala i de Fermata. Christian Huygens je Pascalove i de Fermatove rezultate nastavio, a Jacob Bernoulli 1713. godine prezentira kombinatornu vjerojatnost u potpuno sređenom obliku.

1933. godine ruski matematičar Andrej Nikolajevič Kolmogorov uvodi aksiomatsku definiciju vjerojatnosti te se od tada teorija vjerojatnosti smatra granom matematike.

2.2. Pojam i osnovna svojstva vjerojatnosti

Osnovni pojmovi koji definiraju teoriju vjerojatnosti su pokus (eksperiment) i ishod (elementarni događaj). Najbitnije je pri analizi provedenog pokusa razumjeti odnos između uzroka i posljedice. Poznavanje veze između uzroka i posljedice omogućava predviđanje ishoda ili rezultata pokusa na osnovu definiranog skupa uvjeta u kojima se pokus izvodi.

2.2.1. Prostor događaja

Teorija vjerojatnosti promatra slučajne pokuse, odnosno pokuse čiji ishodi, odnosno rezultati nisu jednoznačno određeni uvjetima u kojima se izvodi . Rezultati takvih slučajnih pokusa se nazivaju slučajnim događajima (A,B,…).

PRIMJER 2.1. *Slučajni pokus: bacanje novčića*

P: palo je „pismo“ – slučajni događaj

G: pala je „glava“ -slučajni događaj

Elementarnim događajem nazivamo svaki mogući ishod slučajnog pokusa. Dva elementarna događaja se ne mogu dogoditi istodobno. Ukoliko se dva elementarna događaja sastoje od istih ishoda, kaže se da su ta dva događaja jednaka. Skup svih elementarnih događaja nekog pokusa se naziva prostor elementarnih događaja (oznaka S, Ω). [1]

$$\Omega=\{\omega_1, \omega_{2,\dots,}\ \omega_N\} \tag{2-1}$$

Prema prethodnom primjeru, skup svih elementarnih događaja za bacanje novčića je:

$$\Omega=\{P, G\} \tag{2-2}$$

Pokusi ili eksperimenti mogu biti deterministički ili slučajni. Deterministički pokusi jesu pokusi koji imaju samo jedan mogući ishod. Slučajni pokusi jesu oni pokusi čiji ishodi, odnosno rezultati nisu jednoznačno određeni, ali ukazuju na skup mogućih ishoda (rezultata). Kod svakog pokusa je nužno imati definirane uvjete pokusa te moguće rezultate pokusa.

Ishod, odnosno elementarni događaj može biti determiniran, nedeterminiran ili stohastičan. Determiniran ishod je jednoznačno određen ishod, nedeterminiran ishod jest onaj ishod koji nije jednoznačno određen, a stohastičan ishod označava slučajan ishod. Pri tom, ishodi čine skup koji može biti konačan, beskonačan ali prebrojiv ili neprebrojiv. Sigurnim događajem se naziva skup svih mogućih ishoda. Također, postoji i nemoguć događaj kojeg označava prazan skup (∅).

Definicija 2.1. *Za svaki podskup skupa svih elementarnih događaja kažemo da je slučajni događaj. Događaj A je realiziran ako je u pokusu realiziran bilo koji od ishoda* $\omega_{ai} \in A$, $i = 1, \ldots r$.[2]

$$A \subset \Omega;\ A = \{\omega_{a1}, \omega_{a2}, \ldots, \omega_{ar}\} \qquad (2\text{-}3)$$

Definicija 2.2. *Skup događaja u nekom pokusu skupa s operacijama unije, presjeka i komplementa se naziva algebrom događaja- (S, ∪, ∩, c), gdje je S skup svih događaja .*

2.2.2. Pojam i aksiomi vjerojatnosti

Vjerojatnost služi određivanju postupka mogućnosti ishoda nekog događaja. Mogućnost ishoda se kvantificira dodjeljivanjem ishodu broja iz intervala [0, 1], odnosno postotka od 0% do 100%. Što je broj bliži 1 ili 100%, vjerojatnost da će se taj izraz pojaviti kao ishod pokusa je veća.

Definicija 2.3. *Kada prostor događaja sadržava N mogućih ishoda koji su jednako vjerojatni, vjerojatnost svakog od ishoda je* $\frac{1}{N}$.

Definicija 2.4. *Za diskretni prostor događaja A vjerojatnost događaja A, s oznakom P(A) jest jednak sumi vjerojatnosti ishoda u A.*

Definicija 2.5. *Funkcija* $P : A' \to R$ *definirana na algebri događaja A' s vrijednostima u skupu realnih brojeva R, naziva se vjerojatnost na algebri A' ako vrijede sljedeći aksiomi:*

1. $\forall A \in A'$ *slijedi* $P(A) \geq 0$ *(nenegativnost)*
2. $P(\Omega)=1$ *(normiranost)*
3. $A \in A', B \in A', A \cap B = \emptyset \Rightarrow P(A \cup B) = P(A) + P(B)$. *(aditivnost)*

Broj P(A) ∈ R je vjerojatnost događaja A. Prostor vjerojatnosti na algebri A označava uređena trojka (Ω, A', P).

Definicija 2.6. *Neka je* $A \in A'$, $n \in N$ *broj ponavljanja pokusa u kojem se registriraju realizacije događaja A. Neka n(A) broj realizacija događaja A u n ponavljanja. Tada broj n(A) označava apsolutnu frekvenciju nastupa događaja A, a kvocijent* $\frac{n(A)}{n}$ *označava relativnu frekvenciju. Tada se vjerojatnost događaja A označava kao:*

$$P(A) = \lim_{n \to \infty} \frac{n(A)}{n} \qquad (2\text{-}4)$$

Definicija 2.7. *KLASIČNA DEFINICIJA VJEROJATNOSTI. Neka je Ω konačan skup svih mogućih jednako vjerojatnih ishoda nekog eksperimenta i neka je A događaj određen podskupom ishoda od Ω. Tada je vjerojatnost događaja A broj:*

$$P(A) = \frac{|A|}{|\Omega|} = \frac{broj\ ishoda\ povoljnih\ za\ događaj\ A}{broj\ svih\ mogućih\ ishoda} \qquad (2\text{-}5)$$

3. BAYESOV TEOREM

Bayesova formula služi za aposteriorno izračunavanje vjerojatnosti pojedinih hipoteza A_i, ako je poznato da se dogodio događaj B. Drugim riječima, ukoliko se želi pronaći vjerojatnost ranijeg događaja, pod uvjetom da je nastupio kasniji događaj, koristi se Bayesov teorem. Da bi se razumjela Bayesova formula, potrebno je znati osnove uvjetne vjerojatnosti kojom se bavio i sam Bayes prije nego što je dokazao svoj poznati teorem.

3.1. Zavisnost događaja

Ako ostvarivanje jednog događaja nekog eksperimenta ne utječe na vjerojatnost ostvarivanja drugog događaja, tada se kaže da su ti događaji nezavisni. Analogno tome, ukoliko ostvarivanje jednog događaja nekog eksperimenta utječe na vjerojatnost ostvarivanja drugog događaja, tada se kaže da su ti događaji zavisni. Matematički gledano, događaj A je zavisan o događaju B ukoliko vrijedi $P(A|B) \neq P(B)$. Analogno tome, ukoliko vrijedi relacija $P(A|B)=P(B)$, tada je događaj A nezavisan o događaju B.

Za zavisne događaje vrijedi sljedeća relacija:

$$P(AB)=P(B)*P(A|B)=P(A)*P(B|A) \tag{3-1}$$

Za nezavisne događaje vrijedi da je vjerojatnost istovremenog pojavljivanja događaja A_1 i A_2 jednaka umnošku vjerojatnosti pojedinih događaja, matematički prikazano:

$$P(A_1 \cap A_2)=P(A_1)*P(A_2) \tag{3-2}$$

Implicitan uvjet je da se događaji A_1 i A_2 ne isključuju, odnosno događaji koji se isključuju ne mogu biti nezavisni.

3.2. Uvjetna vjerojatnost

Pri definiciji zavisnosti dvaju događaja A i B je korišten izraz $P(A|B)$ koji označava vjerojatnost ishoda događaja A ako se dogodio događaj B. Time je iskazana uvjetna vjerojatnost. Pojam uvjetne vjerojatnosti je uveo Thomas Bayes (1702.-1762.).

Definicija 3.1. *Neka je (Ω, A', P) vjerojatnosni prostor i neka je $A \in A'$ takav da je $P(A)>0$. Tada se funkcija $P_A: A' \rightarrow [0,1]$ naziva uvjetnom vjerojatnosti, a svaki $B \in A'$ se definira kao vjerojatnost od B uz uvjet da se dogodio A:*

$$P_A(B)=P(B|A)=\frac{P(B\cap A)}{P(A)} \tag{3-3}$$

PRIMJER 3.1. Kocka se baca jednom. Potrebno je ispitati zavisnost događaja A: {1,2,3,4} o događajima B: {3,4,5,6} i C: {4,5,6}.

Broj svih mogućih ishoda (|Ω|) u pokusu bacanja kocke je 6. Mogući ishodi bacanja kocke su {1,2,3,4,5,6}.

$P(A)=\frac{4}{6}$ – vjerojatnost događaja A

$P(A|B)=\frac{2}{3}$ - od triju mogućnosti (3,4,5) dvije su povoljne (3,4)

$P(A|C)=\frac{1}{3}$ – od triju mogućnosti (4,5,6) jedna je povoljna (4)

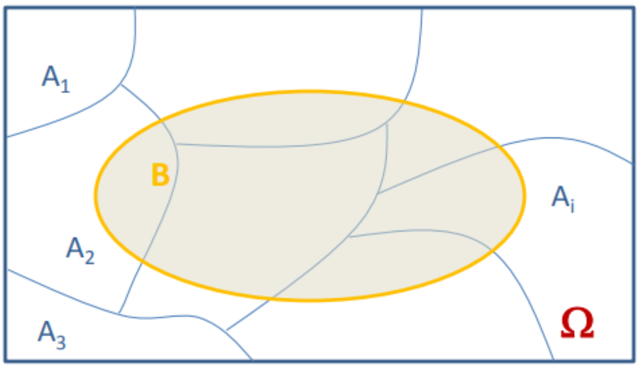
Dakle, P(A|B)=P(A) pa je A nezavisan o B, a P(A|C)≠P(A) pa je A zavisan o C.

3.3. Formula potpune vjerojatnosti

Formula potpune vjerojatnosti se primjenjuje samo kada se događaj A može realizirati samo zajedno s jednim od događaja $B_1, B_2, \ldots, B_n$, koji su međusobno disjunktni i u uniji čine čitav prostor elementarnih događaja. Tada se elementi potpunog skupa događaja $\{B_1, B_2, \ldots, B_n\}$ nazivaju hipotezama. U svakom izvođenju slučajnog pokusa se točno jedna od hipoteza mora dogoditi.

Definicija 3.2. *Neka je P: A'→R vjerojatnost na algebri i neka skup A={$A_1, A_2, \ldots, A_n$ } čini potpunu familiju događaja u A'. Tada za proizvoljni događaj B∈A' vrijedi formula potpune vjerojatnosti koja ima oblik:*

$$P(B) = \sum_{i=1}^{n} P(A_i) \cdot P(B|A_i) \qquad (3\text{-}4)$$



Slika 1 Grafički prikaz vjerojatnosti B na skupu događaja A

Da bi neki skup činio potpunu familiju događaja u vjerojatnosnom prostoru, moraju vrijediti sljedeći uvjeti:

1. Niti jedan podskup skupa B ne smije biti prazan, odnosno:

$$B_i \neq \emptyset, \forall i \in \{1,2,\ldots,n\} \qquad (3\text{-}5)$$

2. Dva podskupa skupa B ne smiju imati zajedničke elemente, odnosno:

$$B_i \cap B_j=\emptyset, \forall i \neq j \tag{3-6}$$

3. Unija svih podskupova skupa B čini čitav prostor elementarnih događaja, odnosno:

$$\bigcup_{i=1}^{n} B_i = \Omega \tag{3-7}$$

Primjer 3.2. Tri tvornice proizvode proizvod iste marke. Prva proizvodi 30% proizvoda, druga 50%, a treća 20%. U prodaji je iz prve tvornice 2% neispravnih proizvoda, iz druge 3%, a iz treće 4%. Kolika je vjerojatnost da je slučajno kupljeni proizvod neispravan?

A: slučajno kupljeni proizvod je neispravan,
H1: slučajno kupljeni proizvod je proizveden u prvoj tvornici,
H2: slučajno kupljeni proizvod je proizveden u drugoj tvornici,
H3: slučajno kupljeni proizvod je proizveden u trećoj tvornici

Tada je:

P(H1)=0.3
P(H2)=0.5
P(H3)=0.2

P(A|H1)=0.02
P(A|H2)=0.03
P(A|H3)=0.04

Prema formuli potpune vjerojatnosti se lako dobije rješenje:

P(A)=P(A|H1)*P(H1)+P(A|H2)*P(H2)+P(A|H3)*P(H3)=0.02*0.3+0.03*0.5+0.04*0.2=0.029

Dakle, u prodaji je 2.9% neispravnih proizvoda.

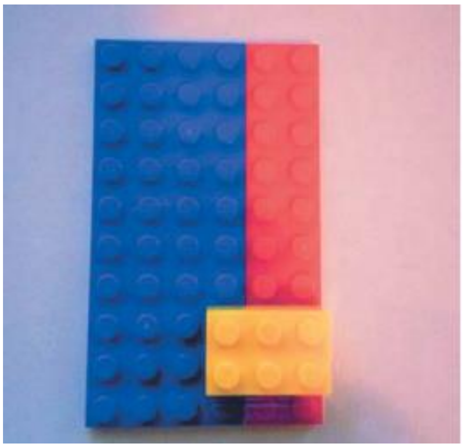
3.4. Bayesova formula

Engleski statističar i filozof Thomas Bayes (1702.-1762.) je dokazao teorem teorije vjerojatnosti koji je po njemu nazvan Bayesov teorem ili češće Bayesova formula. Bayesova formula daje vjerojatnost hipoteze ako se zna da se dogodio neki događaj i može se korisno primijeniti, npr. u medicinskoj dijagnostici, ekonomiji…

Definicija 3.2. *Neka je (Ω, A', P) vjerojatnosni prostor i neka skupovi $H_1, H_2, \ldots, H_n \in A'$ čine potpun sustav događaja. Neka događaj $A \in A'$ ima pozitivnu vrijednost P(A)>0. Tada za proizvoljni netrivijalni događaj $A \in A'$ vrijedi Bayesova formula :*

$$P(H_i|A) = \frac{P(H_i)\cdot P(A|H_i)}{P(A)} = \frac{P(H_i)\cdot P(A|H_i)}{\sum_{j=1}^{n} P(H_j)\cdot P(A|H_j)} \tag{3-8}$$

Primjer 3.3. Potrebno je izračunati vjerojatnost da se ispod žutog elementa nalazi crveni element, ako je dano područje 6 × 10 LEGO elemenata od kojih je 40 plavih i 20 crvenih, te 6 žutih elemenata koji prikrivaju crvene i plave kao na slici 2.



Slika 2 Spojene lego kocke

Zadano područje od 6 × 10 LEGO elemenata predstavlja vjerojatnosni prostor koji se sastoji od crvenih, plavih i žutih elemenata. Žuti elementi se nalaze na crvenim ili plavim elementima. Budući da je pokrivenost plavim elementima 40, a cijeli prostor je veličine 60, lako je izračunati vjerojatnost za plavi element:

$$\text{P(plavi)}=\frac{40}{60}=\frac{2}{3}$$

Analogno tome, prekrivenost crvenih elemenata je 20 te se lako izračuna i vjerojatnost za crveni element:

$$\text{P(crveni)}=\frac{20}{60}=\frac{1}{3}$$

Crveni i plavi mogu opisati skup mogućih događaja budući da je zbroj vjerojatnosti jednak 1. Pokrivenost žutim elementima je 6, a cijeli prostor je veličine 60, što znači da je vjerojatnost žutih elemenata:

$$\text{P(žuti)}=\frac{6}{60}=\frac{1}{10}$$

Budući da žuti elementi prekrivaju jedan dio plavih i crvenih, žuti elementi se moraju promatrati s njima. Stoga vjerojatnost pojavljivanja žutih elemenata ovisi o tome razmatra li se one elemente koji su na plavom ili crvenom dijelu promatranog vjerojatnosnog prostora. Vjerojatnost da se žuti element nalazi na plavom dijelu se označava s P(žuti|plavi), a vjerojatnost da se žuti element nalazi na crvenom dijelu vjerojatnosnog prostora se označava s P(žuti|crveni). Kako bi se odredila vjerojatnost P(žuti|crveni), potrebno je razdvojiti crvene elemente od plavih. Pokrivenost crvenim je 20, a pokrivenost žutih elemenata je 4. Dijeljenjem žutih s prekrivenosti crvenih elemenata se dobije:

$$\text{P(žuti|crveni)}=\frac{4}{20}=\frac{1}{5} \text{ -uvjetna vjerojatnost žutog i crvenog elementa}$$

Uvrštavanjem izračunatih vjerojatnosti u Bayesovu formulu, dobiva se tražena vjerojatnost:

$$\text{P(crveni|žuti)}=\frac{P(\check{z}uti|crveni)*P(crveni)}{P(\check{z}uti)}=\frac{\frac{1}{5}*\frac{1}{3}}{\frac{1}{10}}=\frac{2}{3}$$

Sada se lako može izračunati i vjerojatnost da se ispod žutog nalazi plavi element:

$$\text{P(plavi|žuti)=1-P(crveni|žuti)=1-}\frac{2}{3}=\frac{1}{3}$$

4. OPIS BAYESOVOG KLASIFIKATORA

Bayesov klasifikator je jedna od metoda strojnog učenja koje se zasniva na Bayesovom teoremu opisanom u prošlom poglavlju. Bayesov klasifikator se koristi kada je potrebna klasifikacija pojedinog primjera na temelju više hipoteza. Bayesovi algoritmi zahtijevaju inicijalno znanje mnogih vjerojatnosti. Postoje tri vrste Bayesovog klasifikatora- naivni Bayesov klasifikator, normalni Bayesov klasifikator i Bayesova mreža. U ovom radu bit će korišten i objašnjen normalni Bayesov klasifikator.

4.1. Naivni Bayesov klasifikator

Naivni Bayesov klasifikator jest jedna od najkorisnijih primjena Bayesovog pravila. Ovakva vrsta klasifikatora se može upotrijebiti za utvrđivanje vjerojatnosti klasa na temelju niza opažaja. Pretpostavka je da su varijable značajaka uvjetno nezavisne s obzirom na klasu: [3]

$$P(d|h)=P(a_1,\ldots,a_T\,|h)=\prod_t P(a_t|h). \tag{4-1}$$

Ovaj model Bayesovog klasifikatora se naziva naivnim upravo zbog toga što pretpostavka o uvjetnoj nezavisnosti u praksi uglavnom ne vrijedi. Primjerice, u kontekstu klasifikacije dokumenata, ova pretpostavka kaže da , ako je poznato da dokument pripada klasi „Sport“, vjerojatnost pojavljivanja riječi „nogomet“ jednaka je neovisno o tome pojavljuje li se u istom tekstu riječ „lopta“. Pretpostavka je sasvim sigurno pogrešna, no ipak se pokazuje da naivan Bayesov klasifikator u praksi vrlo dobro funkcionira. Model naivnog Bayesovog klasifikatora glasi:

$$h(x_1, \ldots, x_n)=\text{argmax } P(x_1, \ldots, x_n|C_j)P(y=C_j) \tag{4-2}$$

Potrebno je procijeniti parametre diskretnih razdioba $P(x_1, \ldots, x_n|C_j)$ i $P(C_j)$. Za procjenu apriornih vjerojatnosti $P(C_j)$ se koristi sljedeći izraz:

$$P(C_j) = \frac{1}{N}\sum_{i=1}^{N} 1\,\{y^{(i)} = C_j\}=\frac{N_j}{N} \tag{4-3}$$

Ukupan broj parametara se izračunava kao razlika broja klasa (K) i jedinice. Kod naivnog Bayesovog klasifikatora je problem što broj mogućih stanja varijable x raste eksponencijalno s dimenzijom n što znači da i broj parametara modela raste eksponencijalno. Binarni vektor x može primiti 2^n različitih vrijednosti. Za svaki takav vektor je potrebno procijeniti vjerojatnost pripadanja jednoj od K klase, što za sobom povlači činjenicu da ukupno treba procijeniti $(2^n - 1)K$ parametara. Kako bi se model mogao generalizirati, nužno je uvesti neke pretpostavke. Potrebno je uvesti uvjetnu nezavisnost varijable za zadanu klasu, odnosno mora vrijediti izraz:

$$P(x_1,\ldots,x_n|C_j)=P(x_i|C). \tag{4-4}$$

Višestrukom primjenom prethodno napisane jednakosti na izraz:

$$P(x_1, \dots, x_n | C_j) = \prod_{k=1}^{n\cdot} P(x_k | x_1, \dots, x_{k-1}, C_j) \tag{4-5}$$

Dobije se izraz:

$$P(x_1, \dots, x_n | C_j) = \prod_{k=1}^{n} P(x_k | C_j) \tag{4-6}$$

Te se dolazi do modela naivnog Bayesovog klasifikatora:

$$h(x_1, \dots, x_n) = \text{argmax } P(C_j) \prod_{k=1}^{n} P(x_k | C_j) \tag{4-7}$$

Vjerojatnosti $P(x_k|C_j)$ se mogu jednostavno procijeniti metodom najveće izglednosti:

$$P(x_k | C_j) = \frac{\sum_{i=1}^{N} 1\{x_k^{(i)} = x_k \wedge y^{(i)} = C_j\}}{\sum_{i=1}^{N} 1\{y^{(i)} = C_j\}} = \frac{N_{kj}}{N_j} \tag{4-8}$$

Ukupan broj ovakvih vjerojatnosti koji se moraju procijeniti je $\sum_{k=1}^{n} (K_k - 1)K$. Ukoliko su značajke binarne, taj broj iznosi nK. Sada broj parametara linearno ovisi o dimenziji n, a ne više eksponencijalno.

4.2. Normalni Bayesov klasifikator

Normalni Bayesov klasifikator se koristi u klasifikacijskom problemu gdje su vlastite vrijednosti kontinuirane. Spomenuti klasifikator uzima u obzir da svi atributi (ili vektori) svake kategorije (ili klasifikacije) poštuju višemjernu normalnu Gaussovu distribuciju iskazanu jednadžbom [4]:

$$p(x_1, \dots, x_n | C_k) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_k|}} exp\left(-\frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)\right) \tag{4-9}$$

U formuli μ_k predstavlja n-dimenzionalni srednji vektor koji odgovara k-toj kategoriji i $|\Sigma_k|$ predstavlja vrijednost determinante $n \times n$ matrice kovarijance Σ_k koja odgovara k-toj kategoriji.

Stoga, atributi ne moraju biti neovisni za razliku od naivnog Bayesovog klasifikatora. Cijela funkcija raspodjele, uključujući sve kategorije, mješavina je Gaussove distribucije, a svaka kategorija je komponenta.

Prema Bayesovom pravilu (3-8), aposteriorna vjerojatnost je proporcionalna umnošku apriorne vrijednosti i vjerojatnosti. Međutim, u nekim slučajevima se apriorna vjerojatnost ne može uzeti u obzir, poput klasifikacije vrlo male ili velike dimenzije, odnosno aposteriorna vjerojatnost je proporcionalna samo vjerojatnosti. Sada je potrebno dobiti funkciju vjerojatnosti svake kategorije (4-9) i unijeti nove uzorke predviđanja. Ovisno o tome koja je vrijednost veća, uzorak pripada klasifikaciji koja odgovara funkciji vjerojatnosti, odnosno:

$$\hat{y} = \arg\max p(x_1, \ldots, x_n | C_k) \tag{4-10}$$

Za potrebe izračuna, formula (4-9) se može pretvoriti u sljedeću funkciju vjerojatnosti:

$$ln(L_k) = -\frac{1}{2} ln(|\Sigma_k|) - \frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k) - \frac{n}{2} ln(2\Pi)$$

$$ln(L_k) = -\frac{1}{2}[ln(|\Sigma_k|) + (x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k) + n\, ln(2\Pi)] \tag{4-11}$$

Kako bi se jednadžba (4-11) mogla izračunati, potrebno je procijeniti dva skupa parametara: srednji vektor μ_k i matricu kovarijance Σ_k. Procjena najveće vjerojatnosti i-tog elementa srednjeg vektora μ_k (tj. i-tog obilježja) μ_{ki} je:

$$\hat{\mu}_{ki} = \frac{\sum_{j=1}^{Nk} x_{ki}^{(j)}}{N_k} \tag{4-12}$$

U formuli, $x_{ki}^{(j)}$ predstavlja vrijednost i-tog atributa j-tog uzorka koji pripada klasifikaciji k u uzorku za obuku. Konačni n-dimenzionalni (ukupno n značajki atributa) srednji vektor μ_k se procjenjuje:

$$\hat{\mu}_k = (\hat{\mu}_{k1}, \hat{\mu}_{k2}, \ldots, \hat{\mu}_{kn})^T \tag{4-13}$$

Nepristrana procjena n × n matrice kovarijance Σ_k je:

$$\hat{\Sigma}_k = \frac{1}{N_k - 1} \begin{bmatrix} cov_k^{(1,1)} & \cdots & cov_k^{(1,n)} \\ \vdots & \ddots & \vdots \\ cov_k^{(n,1)} & \cdots & cov_k^{(n,n)} \end{bmatrix} \tag{4-14}$$

U prethodnoj jednadžbi p-ti red i q-ti stupac elementa $cov_k^{(p,q)}$ predstavljaju kovarijancu p-tog atributa i q-tog atributa u skupu podataka koji se sastoji od k-te klasifikacije u obuci uzorka. Ukoliko su p i q jednaki, tada je varijanca $cov_k^{(p,q)}$ jednaka:

$$cov_k^{(p,q)} = \sum_{j=1}^{N_k} [\left(x_{kp}^{(j)} - \hat{\mu}_{kp}\right)\left(x_{kq}^{(j)} - \hat{\mu}_{kq}\right)]$$

$$cov_k^{(p,q)} = \sum_{j=1}^{N_k} \left(x_{kp}^{(j)} x_{kq}^{(j)} - x_{kp}^{(j)} \hat{\mu}_{kq} - x_{kq}^{(j)} \hat{\mu}_{kp} + \hat{\mu}_{kp}\hat{\mu}_{kq}\right)$$

$$cov_k^{(p,q)} = \sum_{j=1}^{N_k} \left(x_{kp}^{(j)} x_{kq}^{(j)}\right) - \hat{\mu}_{kq} \sum_{j=1}^{N_k} x_{kp}^{(j)} - \hat{\mu}_{kp} \sum_{j=1}^{N_k} x_{kq}^{(j)} + N_K \hat{\mu}_{kp}\hat{\mu}_{kq}\Big) \tag{4-15}$$

4.2.1. Primjer

Sljedeća tablica [5] predstavlja skup statističkih podataka o pokazateljima tjelesnih karakteristika stanovnika zemlje:

Redni broj	Spol (S)	Visina (V) [ft]	Težina (T) (lbs)	Dužina stopala (D) (inči)
1	M	6	180	12
2	M	5.92	190	11
3	M	5.58	170	12
4	M	5.92	165	10
5	F	5	100	6
6	F	5.5	150	8
7	F	5.42	130	7
8	F	5.75	150	9

Tablica 1 Tjelesne karakteristike stanovnika zemlje

Tablicu čine osam muškaraca i žena podijeljenih u dva dijela. Broj svakog uzorka jest četiri, odnosno N_M=4 i N_F = 4.

Uvrštavanjem u formulu (4-12) dobiva se:

$$\hat{\mu}_{M,V} = \frac{6 + 5.92 + 5.58 + 5.92}{4} = \frac{23.42}{4} = 5.855$$

$$\hat{\mu}_{M,T} = \frac{180 + 190 + 170 + 165}{4} = \frac{705}{4} = 176.25$$

$$\hat{\mu}_{M,D} = \frac{12 + 11 + 12 + 10}{4} = \frac{45}{4} = 11.25$$

Sada je potrebno dobiti konačni vektor $\hat{\mu}_M$

$\hat{\mu}_M=(\hat{\mu}_{M,V}, \hat{\mu}_{M,T}, \hat{\mu}_{M,D})^T=(5.855,176.25,11.25)^T$

Sada je potrebno napraviti isto za ženski rod:

$$\hat{\mu}_{F,V} = \frac{5 + 5.5 + 5.42 + 5.75}{4} = \frac{21.67}{4} = 5.4175$$

$$\hat{\mu}_{F,T} = \frac{100 + 150 + 130 + 150}{4} = \frac{530}{4} = 132.5$$

$$\hat{\mu}_{F,D} = \frac{6 + 8 + 7 + 9}{4} = \frac{30}{4} = 7.5$$

$$\hat{\mu}_M=(\hat{\mu}_{F,V}, \hat{\mu}_{F,T}, \hat{\mu}_{F,D})^T=(5.4175, 132.5, 7.5)^T$$

Sada je potrebno izračunati matricu kovarijance za žene i muškarce prema formulama (4-14) i (4-15). Po istom principu se dobiju ostale komponente matrice.

$$cov_M^{T,V} = (6 - 5.855) \times (180 - 176.25) + (5.92 - 5.855) \times (190 - 176.25) + (5.58 - 5.855) \times (170 - 176.25) + (5.92 - 5.855) \times (165 - 176.25) = 0.5437 + 0.8937 + 1.7188 - 0.7312 = 2.425$$

Konačna matrica kovarijance (formula 4-14):

$$\hat{\Sigma}_M = \begin{bmatrix} 0.035 & 0.808 & -0.065 \\ 0.808 & 122.917 & 2.917 \\ -0.065 & 2.917 & 0.917 \end{bmatrix}$$

$$\hat{\Sigma}_F = \begin{bmatrix} 0.097 & 6.942 & 0.388 \\ 6.942 & 558.333 & 28.333 \\ 0.388 & 28.333 & 1.667 \end{bmatrix}$$

Uvrštavanjem dobivenog vektora i matrice kovarijance muškaraca i žena u formulu (4-11) dobiva se funkcija vjerojatnosti dviju klasifikacija. Budući da funkcija vjerojatnosti zahtijeva inverznu matricu i determinantu matrice kovarijance, potrebno ih je izračunati.

$$\left|\hat{\Sigma}_M\right| = 2.25 \quad \left|\hat{\Sigma}_F\right| = 0.669$$

$$\hat{\Sigma}_M^{-1} = \begin{bmatrix} 46.826 & -0.418 & 4.651 \\ -0.418 & 0.013 & -0.070 \\ 4.651 & -0.070 & 1.642 \end{bmatrix}$$

$$\hat{\Sigma}_F^{-1} = \begin{bmatrix} 191.004 & -0.847 & -30.104 \\ -0.847 & 0.017 & -0.089 \\ -30.104 & -0.089 & 9.114 \end{bmatrix}$$

Tada je funkcija vjerojatnosti:

$$ln(L_M) = \frac{-1}{2}[ln2.225 + (x - \begin{bmatrix} 5.855 \\ 176.25 \\ 11.25 \end{bmatrix})^T \begin{bmatrix} 46.826 & -0.418 & 4.651 \\ -0.418 & 0.013 & -0.070 \\ 4.651 & -0.070 & 1.642 \end{bmatrix} (x - \begin{bmatrix} 5.855 \\ 176.25 \\ 11.25 \end{bmatrix}) + 3ln(2\pi)]$$

$$ln(L_F) = \frac{-1}{2}[ln0.669 + (x - \begin{bmatrix} 5.4175 \\ 132.5 \\ 7.5 \end{bmatrix})^T \begin{bmatrix} 191.004 & -0.847 & -30.104 \\ -0.847 & 0.017 & -0.089 \\ -30.104 & -0.089 & 9.114 \end{bmatrix} (x - \begin{bmatrix} 5.4175 \\ 132.5 \\ 7.5 \end{bmatrix}) + 3ln(2\pi)]$$

Ukoliko je netko visok 6 ft, težak 130ibs i ima 8 in dugo stopalo, tada je vektor uzorka predviđanja $x=(6, 130, 8)^T$ kojeg je potrebno uvrstiti u $\ln(L_M)$ i $\ln(L_F)$. Uvrštavanjem vektora u jednadžbe dobije se:

$$ln(L_M) = -16.314$$

$$ln(L_F) = -28.730$$

Budući da je ln(L_M) veći od ln(L_F) može se doći do zaključka da je osoba muškog spola.

Kada se izračunava funkcija vjerojatnosti, primjena dekompozicije na singularne vrijednosti će učiniti program sažetijim [6]. Budući da je matrica kovarijance Σ_k simetrična matrica, njezina dekompozicija na singularne vrijednosti je:

$$\Sigma_k = UWV^T = UWU^T \tag{4-16}$$

Gdje W predstavlja dijagonalnu matricu sastavljenu od vlastitih vrijednosti, U je ortogonalna matrica sastavljena od vlastitih vektora sa svojstvom $U^{-1}=U^T$. Inverzna matrica Σ_k^{-1} se tada računa:

$$\Sigma_k^{-1} = (UWU^T)^{-1} = (U^T)^{-1}(UW)^{-1} = (U^{-1})^{-1}W^{-1}U^{-1} = UW^{-1}U^T \tag{4-17}$$

Ukoliko se postavi D=(x-μ_k), onda se izraz $(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)$ može zapisati kao:

$$(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) = D^TUW^{-1}U^TD = (D^TU)W^{-1}(U^TD) = (D^TU)W^{-1}(D^TU)^T \tag{4-18}$$

D^T predstavlja vektor retka, a U kvadratnu matricu. Tada D^TU predstavlja također vektor retka. Element vektora retka su označeni s a_i. $(D^TU)^T$ je vektor stupca, a W je dijagonalna matrica. Elementi na dijagonali matrice su označeni s w_i. W^{-1} je također dijagonalna matrica, ali elementi na dijagonali su $1/w_i$. Primjenjujući prethodno napisano, prethodna formula glasi:

$$(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) =$$

$$(D^TU)W^{-1}(D^TU)^T = \begin{bmatrix} a_1 & a_2 & \dots & a_n \end{bmatrix} \begin{bmatrix} w_1^{-1} & 0 & \dots & 0 \\ 0 & w_2^{-1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w_n^{-1} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} =$$

$$\begin{bmatrix} a_1w_1^{-1} & a_2w_2^{-1} & \cdots & a_nw_n^{-1} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \frac{a_1a_1}{w_1} + \frac{a_2a_2}{w_2} + \cdots + \frac{a_na_n}{w_n} = \sum_{i=1}^{n}\frac{a_ia_i}{w_i} \tag{4-19}$$

Budući da je determinanta jednaka umnošku vlastitih vrijednosti matrice i dobivenu vrijednost $|\Sigma_k|$ karakterizira w_i, tada je njegova determinantna vrijednost:

$$|\Sigma_k| = \Pi_{i=1}^{n} w_i \tag{4-20}$$

5. BAYESOV KLASIFIKATOR U OPENCV BIBLIOTECI

U prethodnom poglavlju je opisan princip po kojem radi Bayesov klasifikator. Bitno je znati da OpenCV implementira normalni Bayesov klasifikator, a ne naivni. U ovom poglavlju su opisane metode koje se nalaze unutar klase normalnog Bayesovog klasifikatora. Implementacija normalnog Bayesovog klasifikatora je dana u prilogu na kraju rada.

Zadani konstruktor CvNormalBayesClassifier klase ima sljedeće parametre:

1. var_count koji označava broj uzoraka za obuku
2. var_all je broj atributa koje svaki uzorak treba imati
3. cls_labels je matrica preslikavanja klasifikacijskih oznaka vrijednosti odgovora
4. count predstavlja broj uzoraka svake kategorije, odnosno broj N_k iz izraza (4-12) i (4-15) iz prethodnog poglavlja
5. sum koji predstavlja sumu vrijednosti svakog karakterističnog atributa klasifikacije, odnosno predstavlja vrijednost $\sum_{j=1}^{N_k} x_{ki}^{(j)}$ iz izraza (4-12)
6. productsum koji predstavlja prvi operator sume iz izraza (4-15)
7. avg predstavlja srednji vektor, odnosno vrijednost izraza (4-12)
8. inv_eigen_values predstavlja vlastite vrijednosti kovarijance matrice Σ_k svake kategorije
9. cov_rotate_mats predstavlja transponiranu matricu vlastitih vrijednosti matrice kovarijance svake kategorije, odnosno U^T izraza (4-16)
10. c koji predstavlja vrijednost $\ln(|\Sigma_k|)$ iz izraza (4-11)
11. default_model_name koji predstavlja ime modela

Svi prethodno opisani parametri su postavljeni na nulu.

Metoda CvNormalBayesClassifier::train koja služi za obuku normalnog Bayesovog klasifikatora prima sljedeće parametre:

1. _train_data koji predstavlja podatke za obuku
2. _responses koji predstavljaju vrijednosti odgovora podataka _train_data

Metoda obučava normalni Bayesov klasifikator. Također, slijedi konvencije opće metode obuke sa sljedećim ograničenjima: samo CV_ROW_SAMPLE raspored podataka je podržan, ulazne varijable su sve poredane, izlazna varijabla je kategorična (tj. _responses elementi moraju biti cijeli brojevi, iako vektor može imati tip CV_32_FC1), a mjerenja koja nedostaju nisu podržana.

Metoda predviđanja CvNormalBayesClassifier::predict prima parametar samples koji predstavlja uzorak koji se želi klasificirati.

Metoda predviđanja može predvidjeti jedan uzorak ili može predvidjeti više uzoraka u isto vrijeme, ali predviđenom uzorku ne može nedostajati niti jedan atribut. Kada je predviđen samo jedan uzorak, povratna vrijednost ove metode je predviđeni rezultat klasifikacije. Prilikom predviđanja više uzoraka, uzorci se spremaju u matricu parametara u obliku redova, a vraćeni rezultati predviđanja spremaju se u rezultate vektora parametara koji se u tom slučaju moraju definirati.

Struktura predict_body prima sljedeće parametre:

1. _c koji predstavlja vrijednost $\ln(|\Sigma_k|)$ iz izraza (4-11)
2. _cov_rotate_mats koji predstavlja U^T iz izraza (4-16)
3. _inv_eigen_values predstavlja vlastite vrijednosti kovarijance matrice Σ_k svake kategorije
4. _avg koji označava procjenu srednjeg vektora svake kategorije
5. _samples koji označava predviđene podatke uzorka
6. _vidx koji označava karakteristične atribute
7. _cls_labels koji predstavlja klasifikacijsku oznaku uzorka
8. _results koji označava predviđanje klasifikacije više uzoraka
9. _value koji označava klasifikaciju jednog uzorka
10. _var_count1 koji označava broj karakterističnih atributa

Struktura predict_body je glavni dio metode CvNormalBayesClassifier::predict jer se glavni dio predviđanja događa unutar nje.

6. PRIMJENA BAYESOVOG KLASIFIKATORA

Bayesov klasifikator implementiran u OpenCV biblioteci ima širok raspon upotrebe - od jednostavne klasifikacije prema spolovima pa sve do razotkrivanja medicinskih bolesti na temelju simptoma. Kako bi se utvrdila točnost izračuna primjera iz poglavlja 4, prikazan je primjer razvrstavanja spolova..

Kod potreban za provjeru spolova na temelju primjera iz 4. poglavlja:

```
#include "opencv2/core/core.hpp"
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/ml/ml.hpp"
#include <iostream>

using namespace cv;
using namespace std;
int main(int argc, char** argv)
{
    float trainingData[8][3] = { {6, 180, 12}, {5.92, 190, 11}, {5.58, 170, 12}, {5.92, 165, 10},{5, 100, 6}, {5.5, 150, 8},{5.42, 130, 7}, {5.75, 150, 9} };
    Mat trainingDataMat(8, 3, CV_32FC1, trainingData);

    float responses[8] = { 'M', 'M', 'M', 'M', 'F', 'F', 'F', 'F' };
    Mat responsesMat(8, 1, CV_32FC1, responses);

    CvNormalBayesClassifier nbc;
    nbc.train(trainingDataMat, responsesMat);

    float myData[3] = { 6, 130, 8 };
    Mat myDataMat(1, 3, CV_32FC1, myData);
    float r = nbc.predict(myDataMat);

    cout << endl << "result:  " << (char)r << endl;

    return 0;

}
```

REZULTAT:

result: M

U polje trainingData su upisani podaci visine (mjerna jedinica ft), težine (mjerna jedinica lbs) i veličine obuće (mjerna jedinica in). Budući da metoda obuke zahtjeva Mat strukturu tipa float, potrebno je podatke trainingData pretvoriti trainingDataMat, kako je i prikazano 12. linijom koda. U polje responses su upisane vrijednosti klasifikacije svakog uzorka navedenog u trainingData. Metoda za obuku također traži Mat strukturu tipa float za vrijednosti klasifikacije stoga ih je također potrebno pretvoriti. Pretvorba vrijednosti klasifikacije je u Mat strukturu je prikazana 15. linijom koda. Za razliku od podataka za obuku gdje se stvorila matrica 8×3, ovdje se stvara matrica 3×1. Bayesov klasifikator nbc je instanciran u 17. liniji te ima postavljene sve vrijednosti na nulu. Na njemu se poziva metoda za obuku kojoj se predaju Mat strukture trainingDataMat i responsesMat. Podaci koji se žele klasificirati su upisani u polje myData u 20. liniji koda. Također, podaci myData se trebaju pretvoriti u Mat strukturu tipa float nazvanu myDataMat. Nakon toga, na klasifikatoru nbc se poziva metoda predviđanja kojoj se predaje Mat struktura myDataMat te varijabla r poprima rezultat klasifikacije. Kada je cijeli postupak proveden, varijabla r se ispisuje, što je i prikazano 24. linijom koda.

7. ZAKLJUČAK

Otvorene biblioteke za strojno učenje služe kako bi se olakšala implementacija različitih algoritama za strojno učenje. Klasifikator koji je implementiran u OpenCV biblioteci nije naivni, nego normalni Bayesov klasifikator. Iako su nazivi vrlo slični, različiti su Bayesovi klasifikatori. Prvi ima ograničenje u korištenju, odnosno karakteristike varijabli moraju biti neovisne jedna o drugoj, dok drugi nema to ograničenje pa je opseg primjene širi. U radu je objašnjen rad normalnog Bayesovog klasifikatora. Podaci se prvo moraju predati metodi za obuku koja vrši sve provjere koje su potrebne kako bi podaci bili pravilno tretirani. Zatim predict metoda vraća rezultat klasifikacije kojeg je predict_body struktura predvidjela. Na taj način je minimizirana vjerojatnost pogrešne klasifikacije. U radu je korišten Bayesov klasifikator na jednostavnom primjeru razvrstavanja spolova, ali klasifikator se može upotrijebiti i na složenijim primjerima. Na temelju simptoma bolesti, može predvidjeti koju bolest neka osoba može imati. Svakako je jedna od korisnijih metoda strojnog učenja.

LITERATURA

[1] M. Huzak i T. Pogány, Uvod u vjerojatnost i statistiku, Osijek: Sveučilište J.J. Strossmayera, Odjel za matematiku, 2014.

[2] R. Galić, Vjerojatnost i statistika, Osijek: ETF, 2004.

[3] *Strojno učenje FER: materijali s predavanja.*

[4] T. Hastie , R. Tibshirani i J. Friedman, The Elements of Statistical Learning, Springer, 2009.

[5] »Naive Bayes classifier,« 20 kolovoz 2021. [Mrežno]. Available: https://en.wikipedia.org/wiki/Naive_Bayes_classifier.

[6] A. Novak i D. Pavlović, »Dekompozicija matrice na singularne vrijednosti i primjena,« *math.e,* prosinac 2013.

SAŽETAK

Strojno učenje je grana umjetne inteligencije koja se bavi oblikovanjem algoritama koji svoju učinkovitost poboljšavaju na temelju empirijskih podataka. Kao jedna od korisnijih metoda strojnog učenja se ističe Bayesov klasifikator koji se bavi problemom vjerojatnosti pojave nekog uzorka. Spomenuti klasifikator je implementiran u OpenCV biblioteci. Jedna od primjena Bayesovog klasifikatora je određivanje spola čovjeka na temelju ulaznih podataka. Da bi se klasifikacija uspješno provela, potrebno je predati poznate podatke metodi za obuku kako bi se kasnije mogao klasificirati spol na temelju proizvoljnih ulaznih podataka. Primjena klasifikatora je prikazana u programskom jeziku C++.

Ključne riječi: Bayesov klasifikator, OpenCV, vjerojatnost

ABSTRACT

Bayesian classifier in OpenCV library

Machine learning is a branch of artificial intelligence that deals with the design of algorithms that improve their efficiency based on empirical data. One of the most useful methods of machine learning is the Bayesian classifier, which deals with the problem of the probability of the occurrence of a sample. The mentioned classifier is implemented in OpenCV library. One of the potential applications of the Bayesian classifier is the determination of human gender based on input data. In order for the classification to be carried out successfully, it is necessary to submit the known data to the training function so that the gender can be classified later on the basis of arbitrary input data. The application of the classifier is shown in the C++ programming language.

Keywords: Bayesian classifier, OpenCV, probability

ŽIVOTOPIS

Jelena Brestovac je rođena 18. svibnja 1999. godine u Slavonskom Brodu. Pohađala je prirodoslovno-matematički smjer Gimnazije Matija Mesić u Slavonskom Brodu. Nakon završene gimnazije, nastavlja svoje obrazovanje na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku.

PRILOG

Zadani konstruktor klase CvNormalBayesClassifier:

```
CvNormalBayesClassifier::CvNormalBayesClassifier()
{
    var_count = var_all = 0;
    var_idx = 0;
    cls_labels = 0;
    count = 0;
    sum = 0;
    productsum = 0;
    avg = 0;
    inv_eigen_values = 0;
    cov_rotate_mats = 0;
    c = 0;
    default_model_name = "my_nb";
}
```

Metoda za izgradnju Bayesovog klasifikatora:

```
bool CvNormalBayesClassifier::train(const CvMat* _train_data, const CvMat* _responses,
    const CvMat* _var_idx, const CvMat* _sample_idx, bool update)
{
    const float min_variation = FLT_EPSILON;  //definira vrlo mali broj
    bool result = false;    //rezultat kojeg metoda  vraća
    CvMat* responses = 0;   //označava rezultat klasifikacije
    const float** train_data = 0;  //označava uzorke za obuku
    CvMat* __cls_labels = 0;   //označava vrijednost odgovora uzroka
    CvMat* __var_idx = 0;   //označava indeks karakterističnog atributa
    CvMat* cov = 0;   //označava matricu kovarijance kategorije

    CV_FUNCNAME("CvNormalBayesClassifier::train");

    __BEGIN__;

    int cls, nsamples = 0, _var_count = 0, _var_all = 0, nclasses = 0;
    int s, c1, c2;
    const int* responses_data;    //pokazivač na vrijednosti odgovora

    CV_CALL(cvPrepareTrainData(0,
        _train_data, CV_ROW_SAMPLE, _responses, CV_VAR_CATEGORICAL,
        _var_idx, _sample_idx, false, &train_data,
```

```
        &nsamples, &_var_count, &_var_all, &responses,
        &__cls_labels, &__var_idx));
//poziva cvPrepareTrainData koja prvo provjerava jesu li ulazni parametri _train_data i _responses točni, nakon toga koristi parametre _var_idx i _sample_idx da bi dobila prave uzorke podataka train_data za obuku i kako bi dobila odgovarajuće vrijednosti odgovora iz parametra _sample_idx. nsapmples označava broj uzoraka u train_data. var_count označava broj uzoraka značajki u train_data, var_all je broj značajki svakog uzroka u train_data. _cls_labels je matrica preslikavanja klasifikacijskih oznaka vrijednosti odgovora. _var_idx je određen parametrom _var_idx iz matrice koja predstavlja masku značajki svakog uzorka odvojenog u var_all.

    if (!update) //ukoliko su podaci nepromijenjeni, gradi se Bayesov klasifikator iz podataka iz uzorka
    {
        const size_t mat_size = sizeof(CvMat*);
        size_t data_size;

        clear();

        var_idx = __var_idx;
        cls_labels = __cls_labels;
        __var_idx = __cls_labels = 0;
        var_count = _var_count;    //broj značajki koji se koristi u stvarnom uzorku za obradu
        var_all = _var_all;   //broj značajki svih uzoraka

        nclasses = cls_labels->cols;    //označava broj kategorija, odnosno predstavlja broj K iz formule
        data_size = nclasses * 6 * mat_size;    //definira veličinu memorijskog prostora za sve potrebne podatke

        CV_CALL(count = (CvMat**)cvAlloc(data_size));    //dodjela prostora
        memset(count, 0, data_size);

        sum = count + nclasses; //count označava broj uzoraka svake kategorije (N_k)
  //sum označava sumu vrijednosti svakog karakterističnog atributa klasifikacije, odnosno predstavlja ∑_{j=1}^{N_k} x_{ki}^{(j)} iz izraza (4-12)

        productsum = sum + nclasses; //productsum predstavlja izraz (4-15)

        avg = productsum + nclasses; //avg označava srednji vektor, odnosno izraz (4-12)
```

```

    inv_eigen_values = avg + nclasses;  //inv_eigen_values predstavlja vlastite vrijednosti kovarijance matrice Σ_k svake kategorije i na kraju pohranjuje recipročne vrijednosti vlastitih vrijednosti (1/w_i) što je prikazano izrazom (4-19)

    cov_rotate_mats = inv_eigen_values + nclasses; //cov_rotate_mats predstavlja transponiranu matricu vlastitih vrijednosti matrice kovarijance svake kategorije, odnosno predstavlja U^T izraza (4-16)

    CV_CALL(c = cvCreateMat(1, nclasses, CV_64FC1)); //kreira matricu c koja predstavlja ln(|Σ_k|) iz izraza (4-11)

    for (cls = 0; cls < nclasses; cls++) //Prolazi kroz sve kategorije, stvara 6 matrica te ih na kraju briše
    {

        CV_CALL(count[cls] = cvCreateMat(1, var_count, CV_32SC1)); //veličina count matrice je K×n

        CV_CALL(sum[cls] = cvCreateMat(1, var_count, CV_64FC1)); //veličina sum matrice je K×n

        CV_CALL(productsum[cls] = cvCreateMat(var_count, var_count, CV_64FC1)); //veličina productsum matrice jest K×n×n

        CV_CALL(avg[cls] = cvCreateMat(1, var_count, CV_64FC1)); //veličina avg matrice je K×n

        CV_CALL(inv_eigen_values[cls] = cvCreateMat(1, var_count, CV_64FC1)); //veličina inv_eigen_values matrice je K×n

        CV_CALL(cov_rotate_mats[cls] = cvCreateMat(var_count, var_count, CV_64FC1)); //veličina cov_rotate_mats je K×n×n
        CV_CALL(cvZero(count[cls]));
        CV_CALL(cvZero(sum[cls]));
        CV_CALL(cvZero(productsum[cls]));
        CV_CALL(cvZero(avg[cls]));
        CV_CALL(cvZero(inv_eigen_values[cls]));
        CV_CALL(cvZero(cov_rotate_mats[cls]));
    }
}
else    //u suprotnom se dodaju novi uzorci za obradu na temelju postojećeg Bayesovog klasifikatora
{
```

```
    //provjera dimenzija novih podataka:
    if (_var_count != var_count || _var_all != var_all || !((!_var_idx && !var_idx) ||
        (_var_idx && var_idx && cvNorm(_var_idx, var_idx, CV_C) < DBL_EPSILON)))
        CV_ERROR(CV_StsBadArg,
            "The new training data is inconsistent with the original training data");

    if (cls_labels->cols != __cls_labels->cols ||
        cvNorm(cls_labels, __cls_labels, CV_C) > DBL_EPSILON)
        CV_ERROR(CV_StsNotImplemented,
            "In the current implementation the new training data must have absolutely "
            "the same set of class labels as used in the original training data");

    nclasses = cls_labels->cols;
}

responses_data = responses->data.i;  //matrica vrijednosti odgovora koja upućuje na uzorak za obuku

CV_CALL(cov = cvCreateMat(_var_count, _var_count, CV_64FC1)); //Kreira se cov matrica koja označava matricu kovarijance

//obrada podataka: prolazi kroz sve uzorke za obuku, izračunava dijelove brojnika i nazivnika izraza (4-12). i prvog Σ dijela u izrazu (4-15)
    for (s = 0; s < nsamples; s++)
    {
        cls = responses_data[s];      //dohvaća vrijednosti odgovora uzoraka za obuku, odnosno klasifikaciju
//Definicija pokazivača za tri matrice: count, sum i productsum
        int* count_data = count[cls]->data.i;
        double* sum_data = sum[cls]->data.db;
        double* prod_data = productsum[cls]->data.db;
        const float* train_vec = train_data[s];      //dohvaća podatke uzorka za obuku
     //Prolazi kroz sve značajke:
        for (c1 = 0; c1 < _var_count; c1++, prod_data += _var_count)
        {

            double val1 = train_vec[c1];  //dohvaća vrijednost c1 značajke, odnosno vrijednost x_ki^(j)
            sum_data[c1] += val1;    //računa brojnik iz izraza (4-12)
            count_data[c1]++;    //računa nazivnik izraza (4-12)

```

```
            for (c2 = c1; c2 < _var_count; c2++) //Prvi $\Sigma$ u izrazu (4-15), odnosno $\sum_{j=1}^{N_k} x_p^{(j)} x_q^{(j)}$. Matrica sastavljena od ove jednadžbe jest simetrična, stoga je dovoljno izračunati pola matrice. For petlja ovdje izračunava gornji desni dio matrice
                prod_data[c2] += train_vec[c2] * val1;
            }
        }
        cvReleaseMat(&responses);
        responses = 0;
    //izračun avg, matrice kovarijance, c:
    //Prolazi kroz sve rezultate klasifikacije, računa izrazu (4-12) i (4-14)., odnosno vektor i matricu kovarijance te $ln(|\Sigma_k|$ u izrazu (4-11)
        for (cls = 0; cls < nclasses; cls++)
        {
            double det = 1; //označava determinantu vrijednosti matrice kovarijance $\Sigma_k$
            int i, j;
            CvMat* w = inv_eigen_values[cls]; //Označava vlastitu vrijednost matrice kovarijance $\Sigma_k$, tj. $w_i$
            //definira matrični brojač, avg i sum pokazivač:
            int* count_data = count[cls]->data.i;
            double* avg_data = avg[cls]->data.db;
            double* sum1 = sum[cls]->data.db;
            cvCompleteSymm(productsum[cls], 0); // poductSum matrica je simetrična. Samo gornji, desni kut matrice je izračunat pa je cvCompleteSymm funkcija pozvana kako bi završila matricu, odnosno gornji desni kut podataka simetrično kopira u donji lijevi kut.
        //Prolazi kroz sve atribute značajki:
            for (j = 0; j < _var_count; j++)
            {
                int n = count_data[j]; //broj j-te značajke trenutne klasifikacije, odnosno $N_K$
                avg_data[j] = n ? sum1[j] / n : 0.; //računa izraz (4-12)
            }
        //Pokazivač ponovno pokazuje na prvu adresu matrice:
            count_data = count[cls]->data.i;
            avg_data = avg[cls]->data.db;
            sum1 = sum[cls]->data.db;
```

```
for (i = 0; i < _var_count; i++) //prolazi kroz sve značajke
{
    double* avg2_data = avg[cls]->data.db;    //pokazivač na vektor

    double* sum2 = sum[cls]->data.db; //pokazivač na $\sum x_{kq}^{(j)}$ iz izraza (4-15)

    double* prod_data = productsum[cls]->data.db + i * _var_count; //pokazivač na productsum matricu, odnosno na $\sum x_{kp}^{(j)} x_{kq}^{(j)}$ u izrazu (4-15)
    double* cov_data = cov->data.db + i * _var_count;    //pokazivač na matricu kovarijance
    double s1val = sum1[i];     //izraz $\sum x_{kp}^{(j)}$ iz izraza (4-15)
    double avg1 = avg_data[i];    //izraz $\hat{\mu}_{kp}$ iz izraza (4-15)
    int _count = count_data[i];     //broj i-te značajke trenutne klasifikacije, odnosno broj $N_k$ iz izraza (4-15)

    for (j = 0; j <= i; j++) //izračunava samo donji lijevi dio matrice kovarijance (formule 4-14.)
    {
        double avg2 = avg2_data[j];    // $\hat{\mu}_{kq}$ iz izraza (4-15)

        double cov_val = prod_data[j] - avg1 * sum2[j] - avg2 * s1val + avg1 * avg2 * _count; //izraz (4-15)

        cov_val = (_count > 1) ? cov_val / (_count - 1) : cov_val; //kovarijanca, odnosno element izraza (4-14)
        cov_data[j] = cov_val;
    }
}


CV_CALL(cvCompleteSymm(cov, 1));      //matrica kovarijance je simetrična. Poznat je samo donji lijevi dio matrice. Funkcija cvCompleteSymm dovršava konačnu simetričnu matricu, odnosno podaci donjeg lijevog kuta se simetrično kopiraju u gornji desni kut.

CV_CALL(cvSVD(cov, w, cov_rotate_mats[cls], 0, CV_SVD_U_T)); //Poziva se funkcija cvSVD za razlaganje pojedinačnih vrijednosti: A=UWVT, gdje je cov A, w je W, a cov_rotate_mats[cls] je transponirana U (zato CV_SVD_U_T) koja je $U^T$ iz izraza (4-16)

```

```
            CV_CALL(cvMaxS(w, min_variation, w)); //uspoređuje vektor vlastitih vrijednosti w sa konstantom min_variation, odabire velike vrijednosti. Uloga ovog koda je ukloniti premale vlastite vrijednosti

            for (j = 0; j < _var_count; j++) //vlastite vrijednosti se množe, potrebno je dobiti element iz izraza (4-20)
                det *= w->data.db[j];

            CV_CALL(cvDiv(NULL, w, w)); //poziva funkciju cvDiv i računa w=1/w kako bi dobili recipročnu vrijednost vlastite vrijednosti, odnosno 1/w_i iz izraza (4-19)

            c->data.db[cls] = det > 0 ? log(det) : -700; //računa ln(|Σ_k|) iz izraza (4-11)
        }

        result = true;    //Identificira varijablu

        __END__;

        if (!result || cvGetErrStatus() < 0)
            clear();
    //oslobađanje memorijskog prostora:

        cvReleaseMat(&cov);
        cvReleaseMat(&__cls_labels);
        cvReleaseMat(&__var_idx);
        cvFree(&train_data);

        return result;
    }
```

Metoda predviđanja:

```
float CvNormalBayesClassifier::predict(const CvMat* samples, CvMat* results) const
{
    float value = 0;    //povratna vrijednost jednog uzorka predviđanja

    if (!CV_IS_MAT(samples) || CV_MAT_TYPE(samples->type) != CV_32FC1 || samples->cols != var_all) //Procjena ispravnosti ulaznih parametara
        CV_Error(CV_StsBadArg,
            "The input samples must be 32f matrix with the number of columns = var_all");

```

```
    if (samples->rows > 1 && !results) //ako se predviđa više uzoraka, moraju se definirati rezultati ulaznih parametara
        CV_Error(CV_StsNullPtr,
            "When the number of input samples is >1, the output vector of results must be passed");

    if (results) //kako bi se procijenila ispravnost rezultata ulaznih parametara, broj elemenata u vektoru mora biti jednak broju predviđenih uzoraka
    {
        if (!CV_IS_MAT(results) || (CV_MAT_TYPE(results->type) != CV_32FC1 &&
            CV_MAT_TYPE(results->type) != CV_32SC1) ||
            (results->cols != 1 && results->rows != 1) ||
            results->cols + results->rows - 1 != samples->rows)
            CV_Error(CV_StsBadArg, "The output array must be integer or floating-point vector "
                "with the number of elements = number of rows in the input matrix");
    }

    const int* vidx = var_idx ? var_idx->data.i : 0; //označava korištene značajke svih atributa uzoraka gdje je vidx 0

    cv::parallel_for_(cv::Range(0, samples->rows),
        predict_body(c, cov_rotate_mats, inv_eigen_values, avg, samples,
            vidx, cls_labels, results, &value, var_count));

    return value;
}
```

Glavna struktura za predviđanje:

```
struct predict_body : cv::ParallelLoopBody {
    predict_body(CvMat* _c, CvMat** _cov_rotate_mats, CvMat** _inv_eigen_values, CvMat** _avg,
        const CvMat* _samples, const int* _vidx, CvMat* _cls_labels,
        CvMat* _results, float* _value, int _var_count1
    )
    {
        c = _c;    //označava ln(|Σk|) iz izraza (4-11) svake kategorije
        cov_rotate_mats = _cov_rotate_mats;  //izraz (4-16)  U^T
        inv_eigen_values = _inv_eigen_values;  //element 1/wi u izrazu (4-19)
        avg = _avg;    //označava procjenu srednjeg vektora svake kategorije
        samples = _samples;    //označava predviđene podatke uzorka
        vidx = _vidx;    //označava karakteristične atribute
        cls_labels = _cls_labels;    //klasifikacijska oznaka uzorka
```

```
        results = _results;    //služi za predviđanje rezultata klasifikacije više uzoraka
        value = _value;    //služi za predviđanje rezultata jednog uzorka
        var_count1 = _var_count1;   //označava broj karakterističnih atributa
    }

    CvMat* c;
    CvMat** cov_rotate_mats;
    CvMat** inv_eigen_values;
    CvMat** avg;
    const CvMat* samples;
    const int* vidx;
    CvMat* cls_labels;

    CvMat* results;
    float* value;
    int var_count1;

    void operator()(const cv::Range& range) const //preopterećeni operator
    {

        int cls = -1;
        int rtype = 0, rstep = 0;
        int nclasses = cls_labels->cols;    //broj kategorija, odnosno broj odgovora
        int _var_count = avg[0]->cols;    //broj karakterističnih atributa

        if (results)
        {
            rtype = CV_MAT_TYPE(results->type);   //tip podataka

            rstep = CV_IS_MAT_CONT(results->type) ? 1 : results->step / CV_ELEM_SIZE(rtype);
        }
        //dodjeljuje memoriju i inicijalizira zaglavlje za izračun

        cv::AutoBuffer<double> buffer(nclasses + var_count1);   //otvara memorijski prostor

        CvMat diff = cvMat(1, var_count1, CV_64FC1, &buffer[0]); //definira matricu diff, rana faza predstavlja x-μk što je D u izrazu 4-18, a kasnija faza predstavlja D^TU u izrazu (4-18)

        for (int k = range.start; k < range.end; k += 1) //prolazi kroz sve retke matrice uzoraka, odnosno prolazi kroz sve predviđene podatke uzorka
        {
```

```
        int ival;
        double opt = FLT_MAX;   //definira velik broj

        for (int i = 0; i < nclasses; i++) //prolazi kroz sve vrijednosti odgovora
        {

            double cur = c->data.db[i];    //dohvaća trenutnu klasifikaciju ln|(|Σ_k|)
            CvMat* u = cov_rotate_mats[i];   //dohvaća transponiranu matricu vlastitih vektora matrice kovarijance trenutne klasifikacije koja je U^T u izrazu (4-16)
            CvMat* w = inv_eigen_values[i];    //dohvaća recipročnu vrijednost vlastite vrijednosti matrice kovarijance trenutne klasifikacije, koja je 1/w_i u izrazu (4-19)

            const double* avg_data = avg[i]->data.db; //pokazivač na srednji vektor trenutne kategorije

            const float* x = (const float*)(samples->data.ptr + samples->step * k);

          //prolazi kroz sve značajke, računa D=x-μ_k u izrazu (4-18) Iako se ovdje računa D=μ_k-x, to ne utječe na konačan rezultat
            for (int j = 0; j < _var_count; j++)
                diff.data.db[j] = avg_data[j] - x[vidx ? vidx[j] : j];
            cvGEMM(&diff, u, 1, 0, 0, &diff, CV_GEMM_B_T); //poziva se funkcija cvGEMM da izvede množenje matrica, odnosno diff=diff*U^T. Budući da je varijabla U^T iz izraza (4-18), izvorna vrijednost U se dobiva nakon dva transponiranja. Konačni diff je D^T U, odnosno [a_1,a_2, ..., a_n]

            for (int j = 0; j < _var_count; j++) //prolaženje kroz atribute značajki, izračunavanje ln(|Σ_k|) + (x − μ_k)^T Σ_k^{-1} (x − μ_k), odnosno dio izraza (4-11). bez nln(2Π) u uglatim zagradama
            {
                double d = diff.data.db[j];    //dohvaća D^T U

                cur += d * d * w->data.db[j];  //d*d*w je iz izraza (4-20)., a početna vrijednost cur jest ln(|Σ_k|)
            }

            if (cur < opt) //Dohvaća minimalnu vrijednost različite klasifikacije
            {
                cls = i;    //Kategorija koja odgovara minimalnoj vrijednosti
                opt = cur;    //Ažuriranje minimalnih vrijednosti
            }

        }

```

```
        ival = cls_labels->data.i[cls];    //vrijednost odgovora koja odgovara klasifikacijskoj oznaci rezultata predviđanja

        if (results) //ako se predviđa više uzoraka, potrebno je rezultat predviđanja staviti u odgovarajući položaj vektora rezultata
        {
            if (rtype == CV_32SC1)
                results->data.i[k * rstep] = ival;
            else
                results->data.fl[k * rstep] = (float)ival;
        }

        if (k == 0) //ako se predviđa uzorak, dodjeljuje se rezultat predviđanja
            *value = (float)ival;
    }
            }
        };
```