Izveštaj
Domaći 2

Ljiljana Popović, EE72/2014, biskvitcokolada@gmail.com

I. ZADATAK 1: PCA ANALIZA

A. *Baza podataka*

Baza podataka pod nazivom „USArrests“ predstavlja podatake koji su prikupljeni za 50 država u Americi o broju hapšenja na 100 000 stanovnika za ubistva, napade i silovanja. Dat je takođe i podatak o procentu populacije u urbanim zonama. Svi atributi su numerički. Baza podataka je potpuna, nema ndostajućih podataka.

Ovakva baza može poslužiti na primer, za lakše praćenje u kojim državama u Americi je porasla ili opala stopa kriminala i slično.

B. *PCA analiza*

Kako broj dimenzija prostora raste, javlja se problem analize multivarijabilnih podataka. Ljudi imaju izvanrednu sposobnost da razlikuju oblike u 1,2 ili 3 dimenzije, ali ova sposobnost drastično opada za 4 (u našem slučaju) ili više dimenzija. Problem se može rešiti nekim od postupaka *smanjenja dimenzionalnosti*. Postoje dva osnovna pristupa smanjenju dimenionalnosti. To su: odabir (selekcija) obeležja i izdvajanje obeležja. U okviru linearnog izdvajanja obeležja najčešće se koriste dve tehnike: PCA i LDA. Ovde ćemo pričati o PCA tehnici.

Razlaganje na glavne komponente (eng. *principal component analysis* - PCA) koristi kriterijum reprezentacije. Cilj je smanjenje dimenzionalnosti prostora uz očuvanje rasutosti (varijanse) podataka u višedimenzionalnom prostoru, s obzirom da je varijansa nosilac informacije. PCA vrši samo rotaciju koordinatnog prostora tako što poravnava ose rotiranog prostora sa pravcima maksimalne vrednosti.

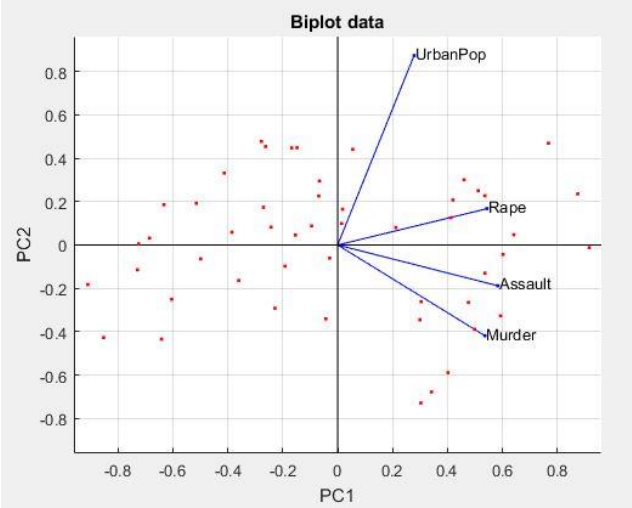
Algoritam PCA:

1. centriranje uzoraka d-dimenzionalnog prostora,
2. pravljenje kovarijansne matrice,
$$\hat{\sigma}_{pk} = \frac{1}{n}\sum_{i=1}^{n}(x_{ip} - \hat{\mu}_p)(x_{ik} - \hat{\mu}_k)$$
3. odredjivanje karakterističnih vektora i vrednosti,
4. sortiranje u opadajućem redosledu po veličini karakterističnih vrednosti,
5. izračunavanje udela komponenti u objašnjenju varijanse
$$udeoKomponente\_i = \frac{\Lambda_i}{\sum_{i=1}^{d}\Lambda_i}$$
6. izdvajanje podskupa od M sopstvenih vektora,
7. redukcija dimenzionalnosti uzoraka na M-dimenzioni prostor.

U ovom delu zadatka primenjena je ugradjena MATLAB-ova funkcija *pca* koja izvršava pomenuti algoritam i to nad normalizovanim podacima, kao što zahteva stavka 1 algoritma.

C. *Vizuelizacija podataka*

Za vizuelizaciju podataka je takođe iskorišćena ugrađena funkcija *biplot* i dobijen je sledeći grafik.



Sl. 1. Grafički prikaz podataka u redukovanom prostoru

Biplot je poboljšani scatterplot koji koristi tačke i vektore za prikaz podataka. Tačke koristi da predstavi rezultate (score-ove) posmatranja glavnih komponenti i koristi vektore da predstavi koeficijente varijabli na glavnim komponentama. Na biplotu su prikazane prve dve PCA komponente.

Ugao između dva vektora prikazuje njihovu korelaciju, što je manji ugao veća je korelacija. Na ovom grafiku vidimo da su atributi *Assault* i *Murder* najviše korelisani i da zajedno sa atributom *Rape* više doprinose prvoj komponenti. *Urbanpop* je manje korelisan sa ostale tri i on više doprinosi drugoj PCA komponenti. Stoga prva komponenta približno odgovara meri ukupne stope teških zločina.

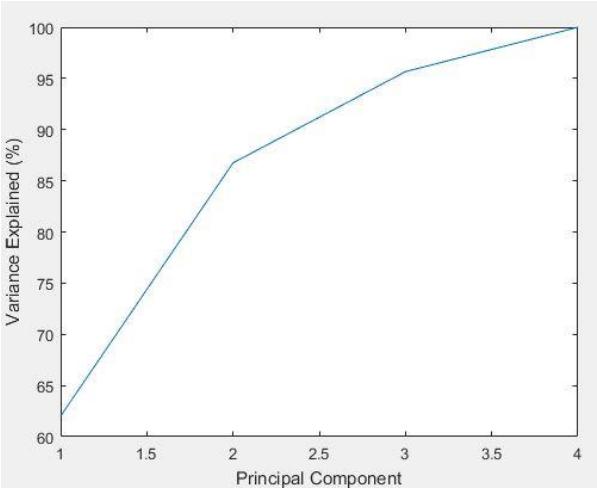
U narednoj tabeli su prikazani koeficijenti prve dve komponente PCA.

TABELA 1: KOEFICIJENTI PRVE DVE PCA KOMPONENTE

	PC1	PC2
Murder	0.5359	-0.4182
Assault	0.5832	-0.1881
UrbanPop	0.2782	0.8728
Rape	0.5434	0.1673

D. Objašnjena varijansa

Procenat objašnjene varijanse je odnos varijanse te komponente i ukupne varijanse, dok je ukupna varijansa suma varijansi svih glavnih komponenti. Broj osnovnih komponenti biramo na taj način da zadržimo veći deo varijanse. Kriterijum se svodi na to da biramo tako da zadržimo 99% varijanse. Na slici 2 je prikazana zavisnost dela objašnjene varijanse od broja PCA komponenti. Brojne vrednosti za procenat objašnjene varijanse su takođe date u tabeli 2.



Sl. 2. Zavisnost dela objašnjene varijanse od broja PCA komponenti.

TABELA 2: PROCENTI OBJAŠNJENE VARIJANSE PO KOMPONENTAMA I UKUPNO

komponente	PC1	PC2	PC3	PC4
objašnjena varijansa po komponentama [%]	62.006	24.7441	8.9141	4.3358
ukupna objašnjena varijansa [%]	62.006	86.7501	95.6642	100

Vidimo da je prva komponenta objasnila 62% varijanse, druga 24,7%, treća 8,9% i četvrta približno 4,3%. Može se zaključiti da dve komponente nisu dovoljne jer objašnjavaju samo 86,75%, što je manje od zahtevanog kriterijuma, pa bi tek tri komponente bile dovoljne za dobru aproksimaciju.

Glavno ograničenje PCA je da ne razmatra separabilnost između klasa jer ne uzima u obzir oznake klasa pojedinih uzoraka. Nema garancije da će pravci sa maksimalnom varijansom biti dobri za diskriminaciju.

II. ZADATAK 2: KNN KLASIFIKATOR

A.Baza podataka

U ovom zadatku baza podataka je pod nazivom „recepti_train“ i sadrži 133 atributa koji predstavljaju različite sastojke za pravljenje kolača, peciva ili pice. Prikupljeno je 1738 uzoraka i oni predstavljaju recepte, gde je za svaki naznačeno prisustvo (1) ili odsustvo (0) svakog od 133 sastojaka.

Baza je podeljena u 3 klase prema oznaci labele. Utvrđeno je da nema nedostajućih podataka, što govori da ne treba brisati neke podatke i da je baza potpuna.

Analiziranje ovih podataka bi moglo poslužiti kao osnova za projektovanje klasifikatora koji će na osnovu prisustva/odsustva određenih sastojaka svaki novi recept svrstati u odredjenu grupu recepata. Takav klasifikator bi mogao biti koristan svakome prilikom pravljenja pomenutih poslastica.

B. kNN klasifikacija

kNN pravilo klasifikacije je veoma intuitivan metod koji klasifikuje neobeležene primerke na osnovu njihove sličnosti sa primercima iz trening skupa, odnosno, odluka o pripadnosti klasi donosi se na osnovu preglasavanja među k uzoraka koji su najbliži (po nekoj meri najsličniji) test uzorku.

Za realizaciju ovog klasifikatora prvo je neophodno odvojiti deo podataka za testiranje, jer nije dobra ideja da se podaci za trening skupa iskoriste i za izbor optimalnog modela i procenu stvarne stope greške. Krajnji model bi u tom slučaju bio nadprilagođen trening podacima i ne bi bio u stanju da generalizuje svoje odlučivanje za nove podatke. Takođe, u tom slučaju bi estimacija stvarne stope greške bila suviše optimistična. Stoga je izdvojeno 15% (po zahtevu zadatka) podataka za testiranje. Prilikom izdvajanja, paženo je da od svake klase postoji dovoljno uzoraka u test skupu. To je postignuto *Holdout* metodom sa parametron 0.15 i to pomoću funkcije *cvpartition*. Na ovaj način skup raspoloživih podataka je podeljen na dva podskupa: trening skup (koji dalje koristimo za obuku) i test skup (koji dalje služi za procenu stvarne stope greške obučenog modela).

Zatim je potrebno varirati broj najbližih suseda k od 1 do 10 da bi se uočili različiti modeli. Izbor parametra k je heuristički, i treba imati na umu da za malo k predikcija ima veću varijansu (manja stabilnost), dok za veće k predikcija ima veću pristrasnost (suviše udaljeni uzorci utiču na predikciju pa je ona najčešće manje tačna). Ukoliko je k=1 test uzorak će pripadati onoj klasi kojoj pripada njemu najbliži trening uzorak.

Izbor mere udaljenosti je takođe veoma bitan činilac i od njega će zavisiti tačnost modela. Mogu se koristiti različite mere među kojima su sledeći nazivi: *euclidean*, *correlation*, *cosine*, *cityblock*, *hamming*, *mahalanobis*, *minkowski*, *chebychev* i druge.

Variranjem broja najbližih suseda dobijeni su različiti modeli. Takođe je variran i parametar rastojanja. Isprobane su različite metrike i kao najbolja se pokazala kosinusna. Najbolji model je izabran metodom krosvalidacije. Takodje su varirane krosvalidacione tehnike i kao najbolja se pokazala *Leaveout* ali zbog velike količine podataka ova tehnika je veoma zahtevna u pogledu vremena koje je potrebno za obradu podataka. Stoga je rađena *K-fold* krosvalidacija sa parametrom 10 gde se trening skup deli na

K (u opštem slučaju) jednakih disjunktih podskupova, a obuka se vrši K puta, svaki put koristeći jedan od K podskupova za testiranje, a ostale za obuku. Prednost ove krosvalidacije je što se svi primerci koje imamo na raspolaganju koriste i za obuku i za trening. Estimacija stvarne stope greške određuje se kao prosečna vrednost pojedinačnih procena:

$$E = \frac{1}{K}\sum_{i=1}^{K} E_i$$

U tabeli 3 su prikazane greške modela dobijene variranjem broja najbližih suseda.

TABELA 3: GREŠKE MODELA ZA SVAKO K (1-10)

k	loss
1	0.1008
2	0.1333
3	0.1346
4	0.1414
5	0.1381
6	0.1536
7	0.1414
8	0.1421
9	0.1509
10	0.1448

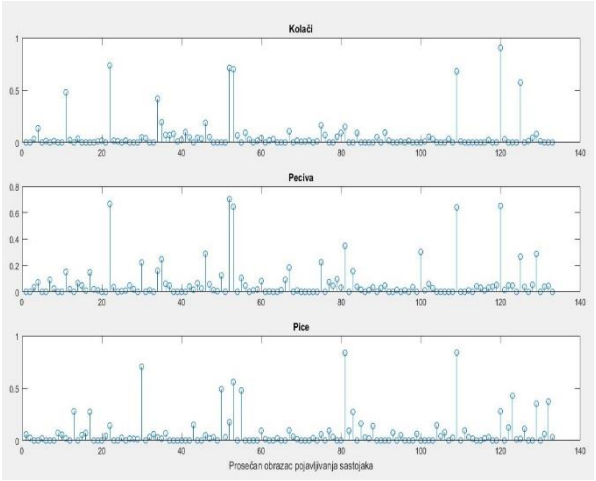
Ispostavilja se da je model sa k=1 najbolji po pitanju greške i on se koristi za obuku nad svim trening podacima. Ovo predstavlja poseban slučaj i naziva se *metoda najbližeg suseda* (eng. *nearest neighbour*), i predstavlja izuzetno jednostavan pristup, koji uz dovoljno veliki skup za obuku ipak postiže solidne rezultate.

Dalje je zahtevano da se nad test podacima izračunaju mere poput *sličnosti, tačnosti, osetljivosti* i *preciznosti* za svaku od klasa, kao i prosečne mere za celokupan klasifikator. Ovo je postignuto uz pomoć matrice konfuzije iz koje je lako odrediti pomenute mere. Dobijeni rezultati su prikazani u tabeli 4.

TABELA 4: MERE ZA SVAKU OD KLASA I PROSEK ZA CELOKUPAN KLASIFIKATOR

	osetljivost	preciznost	specifičnost	tačnost
kolači	0.9266	0.9182	0.9404	0.9346
peciva	0.8696	0.9302	0.9643	0.9308
pice	0.9831	0.9063	0.9701	0.9731
prosečne	**0.9264**	**0.9182**	**0.9583**	**0.9462**

Potom je potrebno nad čitavim setom podataka za svaku od klasa napraviti prosečan obrazac pojavljivanja sastojaka. Podaci su usrednjeni i rezultati su ilustrovani na sledećoj slici.



Sl. 3. Prosečan obrazac pojavljivanja sastojaka

Na grafiku se može uočiti na primer, da je sastojak sa rednim brojem 53 približno sadržan u sve tri klase i to je brašno, što i intuitivno odgovara našim pretpostavkama. Slično se dobija i za 109. sastojak-so. Za razliku od toga vidi se da se neki sastojci pojavljuju samo kod peciva i pice dok u kolačima ne postoje. Takav je na primer, 83. sastojak-luk.

Iako kNN klasifikator zahteva veliki skladišni prostor i obimna izračunavanja prilikom klasifikacije, on je vrlo jednostavan za implementaciju, visoko prilagodljiv i neosetljiv na složenost stvarnih raspodela po pojedinim klasama.