Sveučilište u Rijeci
Tehnički fakultet
Diplomski studij računarstva

Klasifikacija segmenata signala govora na zvučne/bezvučne

Kolegij: Računalna obrada govora i jezika
Student: Sonia Žuškin
Ak. god.: 2018./2019.

Rijeka, srpanj 2019.

Sadržaj

1. Uvod

Glas je u fonetici zvuk koji nastaje vibracijom glasnica. Svi se samoglasnici normalno izražavaju, odnosno zvučno, dok suglasnici mogu biti ili zvučni ili bezvučni. [1] Glasovi se dijele s obzirom na različite kriterije. S obzirom na slobodu prolaska zračne struje dijele se na samoglasnike i suglasnike. Kriterij koji se obrađuje tokom ovog projektnog zadatka jest zvučnost glasa. S obzirom na to titraju li glasnice pri nastanku glasa, glasovi se dijele na zvučne i bezvučne. [2] Hrvatski je pravopis fonološki, što znači da se jednačenja (po zvučnosti i mjestu tvorbe) vrše u govoru i pisanju.

Mel-frekvencijski cepstrum (u daljnjem tekstu MFC, engl. Mel-frequency cestrum) prikaz je kratkotrajnog spektra snage zvuka, koji se temelji na linearnoj kosinusovoj transformaciji logaritamskog spektra na naelinearnoj mel-skali frekvencije. Mel-frekvencijski cepstralni koeficijenti (u daljnjem tekstu MFCC, engl. Mel-frequency cepstra coefficients) su koeficijenti koji zajedno čine MFC. Kod MFC-a su frekvencijski pojasevi jednako razmaknuti na mel-skali (koja približno odgovara odzivu ljudskog slušnog sustava). MFCC-ovi se izvode na slijedeći način:

1) Uzima se Fourierova transformacija signala dijela signala.
2) Mapira se snaga dobivenoga spektra na mel-skalu koristeći trokutaste preklapajuće prozore.
3) Spremaju se dobivene vrijednosti na svakoj mel-frekvenciji.
4) Uzima se diskretna kosinusova transformacija iz liste dobivenih u prethodnome koraku.
5) Na kraju se dobivaju MFCC-ovi koje su amplitude dobivenoga spektra. [3]

Zadatak je ovog projektnog zadatka pomoću mel-kepstralnih koeficijenata kratkotrajnog segmenta signala govora realizirati postupak klasifikacije pojedinih segmenata signala u kategorije zvučni/bezvučni glas. U daljnjem tekstu biti će opisani postupak dobivenoga signala.

2. Projektni zadatak

Kako je prije navedeno, zadatak ovoga projekta je pomoću mel-kepstralnih koeficijenata kratkotrajnog segmenta signala govora realizirati postupak klasifikacije pojedinih segmenata signala u kategorije zvučni/bezvučni glas. Osim uobičajenih knjižica, dodatno se koristila knjižica Librosa za obradu audio datoteka (odnosno .wav datoteka) i knjižica pandas pomoću koje smo baratali s podatcima. Koristio se skriptni jezik python, a u vrijeme pisanja ovoga projekta koristio se python 3. Cijeli projekt je pisan u Jupyter Notebook-u.

Jupyter Notebook postoji za razvoj softvera otvorenoga koda, otvorenih standarda i usluga za interaktivno računarstvo preko desetaka programskih jezika. JupyterLab je interaktivno web sučelje koje pruža razvojno sučelje za Jupyter Notebook, kod i podatke. Ekstenzibilan je i modularan što znači da je moguće pisati ekstenzije koje dodaju nove komponente i integriraju ih sa postojećima.

Jupyter Notebook je open-source web aplikacija koja omogućuje stvaranje i dijeljenje dokumenata koje sadrže kodove, jednadžbe, vizualizacije i narativni tekst. Cijela ta upotreba uključuje sljedeće: čišćenje i transformaciju podataka, numeričku simulaciju, statističko modeliranje, strojno učenje i još mnogo toga. Čini ga vrlo prikladnim za razvoj ovoga projekta zato što pomoću naredbe pip u linux terminalu omogućuje lako

dodavanje potrebnih knjižica i ekstenzija za razvoj ovoga projekta. Verzija Jupyter Notebook-a koja se koristila u ovome projektu je 7.2.0.

Bitne knjižice za razvoj ovoga projekta su nam bile pandas, librosa i sklearn.

Pandas je knjižica s BSD (engl. Berkley Software Distribution) licencama otvorenog izvornog koda koja pruža visokoučinkovite, jednostavne strukture podataka i alate za analizu podataka za Python.

Librosa je Python paket za analizu muzike i audio datoteka. Pruža graditivne elemente koji su potrebni za stvaranje glazbenih sustava za pronalaženje informacija.

Scikit-learn je knjižica otvorenog koda sa BSD dozvolom te pruža alate za rudarenje podataka i analizu podataka. Građena je na NumPy, SciPy i matplotlib knjižicama. Ovu knjižicu koristimo kada nam je potrebno koristiti neki algoritam strojnoga učenja. Iz ove knjižice u ovome projektu se koristi K-Means klasteriranje.

Stvaranje baze podataka

Za stvaranje podataka kojima će se kasnije baratati iz 168 .wav i .lab datoteka su izvučene potrebne značajke i spremljene u drugu .csv datoteku. Datoteka s .lab ekstenzijom je datoteka koja ima informaciju kada započinje i završava pojedini vokal u prikladnoj .wav datoteci. Ono što nam je iz .lab datoteke bilo potrebno je broj vokala kojih ima prikladna .wav datoteka. Prilikom učitavanja .lab datoteke vršila se podjela vokala na zvučne i bezvučne. Također je postavljena kategorija "undefined" (nedefiniran) gdje su se uvrštavali vokali koji nisu bili uvršteni u jednu od kategorija zvučni/bezvučni. Također je u .lab datoteci bila oznaka "sil", što na engleskom označava tišinu (engl. Silence). "Sil" se također uvrštavao u kategoriju "undefined" kako nam neće biti potrebna za daljnju obradu podataka.

Za upravljanje podatcima koristio se `pandas.DataFrame` koji je dvodimenzionalna veličina promjenjive tablične strukture podataka s označenim osima (redci i stupci). U radu sa DataFrame-om aritmetičke se operacije poravnavaju na oznakama retka i stupca što ga čini povoljnim za daljnji rad.

Pomoću librose smo otvorili datoteke sa .wav ekstenzijama te ovisno o broju učitanih znakova iz .lab datoteke smo izračunavali MFCC koeficijente .wav datoteke:

```
mfcc = librosa.feature.mfcc(y=y, sr=sr, n_mfcc = len(df))
```

Kako za svaki vokal dobivamo vektor, izračunavamo njihovu prosječnu vrijednost:

```
mfcc = mfcc.mean(axis=1)
```

Bio je napravljen pokušaj prikaza MFFC-ova u ovisnosti o vremenu. Gotovo je bio linearni prikaz pri čemu se zaključuje da vremenska domena nije bila najbolja opcija za provjeru vokala da li su zvučni ili bezvučni. Stoga se uvela nova komponenta: delta. Pomoću te značajke računale su se delta značajke MFCC-va pomoću Savitsky-Golay fiteriranja.

```
mfcc_delta = librosa.feature.delta(mfcc)
```

Nakon provedenih koraka podatke smo spremili u .csv datoteku radi daljnje obrade.

K-Means klasteriranje

K-Means klasteriranje je jedan od najčešće korištenih algoritama strojnoga učenja bez nadzora koji formira klastere podataka na temelju sličnosti između podataka. Za njegovu funkcionalnost potrebno je unaprijed definirati broj klastera. K u K-Means nazivu se odnosi na broj klastera koji će se koristiti. Kako mi tražimo dvije značajke, odnosno kako želimo raspodijeliti vokale na zvučne ili bezvučne, definirali smo broj klastera K=2. K-means algoritam funkcionira na način da nasumično odabire vrijednosti centroida za svaki klaster. Nakon toga algoritam iterativno izvršava tri koraka:

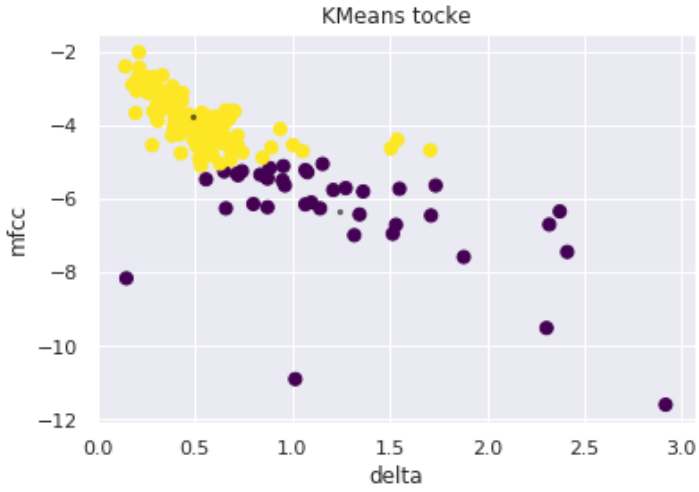
1) Pronalazi euklidske udaljenosti između svakog podatka i centroida svih klastera.
2) Dodijeli instance podatka klasteru centroida najbliže udaljenosti.
3) Izračunava nove vrijednosti centroida na temelju srednjih vrijednosti koordinata svih podataka iz odgovarajućih klastera. [4]

U projektu se koristila funkcija `KMeans` iz knjižice `sklearn.cluster` te funkcionira na gore opisani način.

Nakon što smo ekstrahirali potrebne značajke iz 168 datoteka, spremili smo ih u .csv datoteku. Kako su za `KMeans` funkciju potrebni ulazni podatci u obliku matrice, MFCC i delta vrijednosti MFCC-ova spremamo u matricu. Izračunavamo centroide i prilagodimo naše podatke (X matrica) postojećoj funkciji:

```
# Kreiranje i treniranje modela
kmeans = KMeans(n_clusters=2)
# Kalkuriranje cluster centre
kmeans.fit(X)
```

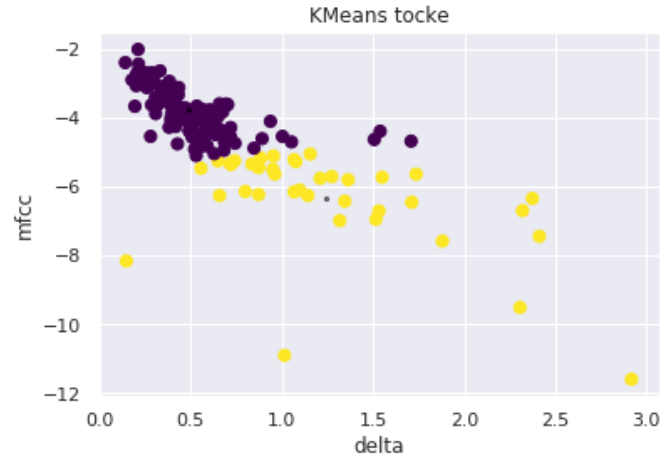
Nakon izračuna prikazujemo podatke na grafu (slika 1):




```
[[ 1.24539467 -6.34975181]
 [ 0.48941176 -3.78884412]]
```

Slika 1: Prikaz izračunatih dva centra pomoću KMeans algoritma strojnoga učenja

Treba uzeti u obzir da nakon svakog pokretanja KMeans algoritma, algoritam ima tendenciju mijenjanja središta klastera što je vidljivo na slici 2:




```
[[ 0.48941176 -3.78884412]
 [ 1.24539467 -6.34975181]]
```

Slika 2: Drugo pokretanje KMeans algoritma

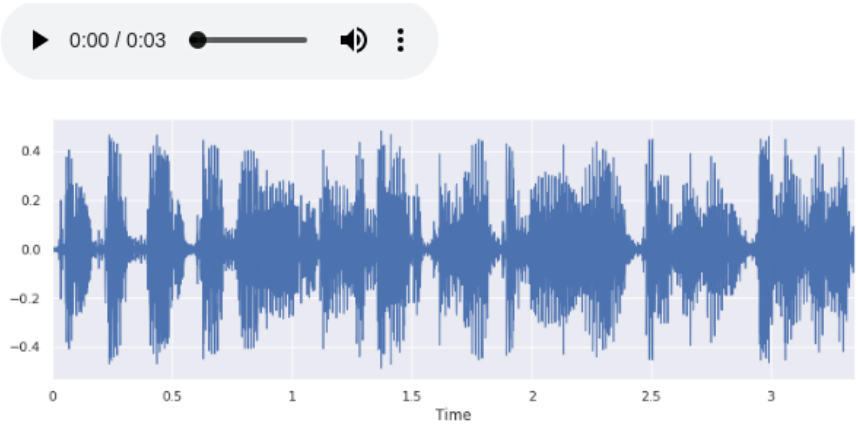
Ispod oba grafa vidljive su izračunate točke središta klastera, iako su vrijednosti iste, treba pripaziti koje vrijednosti uzeti. Kako je broj klasificiranih zvučnih vokala daleko veći nego broj bezvučnih vokala, kao središte klastera za zvučne vokale prema slici 1 se odabire žuto osjenčano područje, odnosno točka s vrijednosti (0.48941176, -3.78884412, odnosno prvi vektor). Dok sa slike 2 bi u tome slučaju uzeli drugi vektor.

Provjera podudarnosti

Za provjeru podudarnosti uzeli smo kao primjer sm04300104354.wav datoteku. Učitavamo navedenu datoteku pomoću knjižice librose:

```
plt.figure(figsize=(12,4))
data,sample_rate = librosa.load(filename)
_ = librosa.display.waveplot(data,sr=sample_rate)
ipd.Audio(filename)
```

te prikazujemo signal grafički kao na slici 3:



Slika 3: Grafički prikaz signala za sm04300104354.wav audio datoteku

Kako bi se utvrdila točnost klasificiranja vokala na zvučne/bezvučne, potrebno je napraviti provjeru koliko točno algoritam klasificira vokale. Napravljena je provjera na nekoliko datoteka sa .wav ekstenzijama. Prvo se iz pojedinačne .lab datoteke izvlače podatci i raspodjeljuju se na zvučne ili bezvučne gdje se spremaju u `pandas.DataFrame`:

```
if vocal in voiced:
        # z za zvucni
        type_vocal.append('z')
    elif vocal in voiceless:
        # b za bezzvucni
        type_vocal.append('b')
    else:
        type_vocal.append('undefined')

df = pd.DataFrame({'start': start_ticks, 'stop': end_ticks, 'vocals': vocals, 'type':
type_vocal})
```

Kako je i prije bilo napravljeno prilikom ekstrakcija potrebnih podataka za stvaranje baze, za ovaj audio file sa .wav ekstenzijom se izvlače potrebne značajke za provjeru rezultata. Za svaki vokal izračunavaju se njegove MFCC i delta vrijednosti. Također se miču tišine i nesvrstani vokali:

```
 df = df[df.type != 'undefined']
```

Uzimaju se izračunati centroidi:

```
# centar za zvucne za trenutno pokrenuti KMeans algoritam
cz = centers[:, 0]
# centar za bezzvucne vokale
cb = centers[:, 1]
```

Te se za svaki izračunati vokal izračunavaju njegove euclidove udaljenosti od svakoga središta:

```
#racunanje euclidove udaljenosti od sredista zvucne tocke
for i in range(len(df)):
    distance = plt.mlab.dist(cz, X[i])
    distance_voiced.append(distance)

#racunanje euclidove udaljenosti od sredista bezvucne tocke
for i in range(len(df)):
    distance = plt.mlab.dist(cb, X[i])
    distance_voiceless.append(distance)
```

Na kraju se vrši provjera na gdje se uspoređuju udaljenosti i razvrstavaju se zvučni i bezvučni vokali:

```
for i in range(len(df)):
    if distance_voiced[i] <= distance_voiceless[i]:
        predicted_class.append('z')
    elif distance_voiced[i] > distance_voiceless[i]:
        predicted_class.append('b')
```

Provjera algoritma testirana je na nekoliko datoteka, ali prije provjere provjerimo kako izgleda dio strukture dobivenih podataka na slici 4:

	start	stop	vocals	type	mfcc	delta	predicted
1	240000	880000	u	z	168.380678	3.657116	z
2	880000	1360000	s	b	-54.487427	3.657116	b
3	1360000	2160000	k	b	56.731675	3.657116	z
4	2160000	2480000	l	z	-17.906306	3.657116	b
6	3360000	3920000	p	b	-4.156830	-0.283718	b
7	3920000	4160000	u	z	13.443527	-5.300418	z
8	4160000	4960000	i	z	-14.101237	-1.195114	b
9	4960000	5600000	s	b	-5.747785	-3.750464	b
10	5600000	6160000	t	b	-9.701678	-0.184933	b
11	6160000	6720000	r	z	-8.677394	-0.326710	b
12	6720000	6960000	a	z	0.581303	1.190790	z
13	6960000	7760000	g	z	-20.068628	0.609728	b
14	7760000	8800000	e	z	11.594427	1.339247	z
16	10160000	11120000	s	b	-3.746074	-0.345044	b
17	11120000	11440000	a	z	-5.090945	0.395830	b
18	11440000	12080000	m	z	6.378514	-1.332806	z
20	12320000	12960000	u	z	-11.830260	-0.536516	b
21	12960000	13520000	b	z	1.483030	-0.060669	z
22	13520000	13760000	i	z	-9.521747	-0.371064	b
23	13760000	14400000	l	z	-6.591371	0.692706	b
24	14400000	14720000	a	z	-4.912349	0.442307	b
25	14720000	15360000	c	b	1.539279	0.074083	z
26	15360000	16080000	k	b	-7.251923	0.547054	b
27	16080000	16320000	i	z	-1.516523	0.213819	z
28	16320000	16800000	m	z	-6.113564	0.071525	b
29	16800000	17200000	n	z	-1.914390	-0.320252	z
30	17200000	18480000	a	z	-3.459116	0.503956	z
31	18480000	19040000	p	b	-3.031763	-0.229756	z
32	19040000	19280000	a	z	-2.208128	0.144490	z
33	19280000	19920000	d	z	-6.953338	0.062275	b
34	19920000	20320000	i	z	2.711648	0.379969	z
35	20320000	21280000	m	z	-9.375045	0.464842	b
36	21280000	21680000	a	z	-0.455707	0.249894	z
37	21680000	22800000	u	z	-1.291133	0.625967	z
38	22800000	23360000	i	z	0.664157	0.088541	z
39	23360000	24160000	s	b	-0.252617	0.401196	z
40	24160000	24720000	t	b	-4.482494	-0.293980	b

Slika 4: Dio DataFrames strukture za sm04300104354 datoteku

Na slici su prikazane izračunate vrijednosti za sm04300104354.wav datoteku. Na kraju postotak točno klasificiranih vokala za navedenu datoteku iznosi kako je prikazano na slici 5:

```
Postotak pogodka je: 0.6153846153846154 posto.
```

Slika 5: Postotak pogodtka za sm04300104354.wav datoteku

Također je napravljena provjera na drugim audio datotekama, a izračunata podudarnost prikazana je na slikama 6, 7, 8 i 9:

```
100  60960000  61360000      b    z   -0.319411  0.154912         z

[89 rows x 7 columns]
Postotak pogodka je: 0.7078651685393258 posto.
```

Slika 6: Postotak podudarnosti za sm04111204115.wav audio datoteku

```
65  43040000  44080000      i    z   -1.925188  -0.091414         z
Postotak pogodka je: 0.7627118644067796 posto.
```

Slika 7: Postotak podudarnosti za sm04170105151.wav audio datoteku

```
51  35760000  36480000      t    b   -0.046985   0.167089         z
Postotak pogodka je: 0.7142857142857143 posto.
```

Slika 8: Postotak podudarnosti za sm04241204444.wav audio datoteku

```
82  50000000  50640000      e    z   -0.335399  0.137957         z

[74 rows x 7 columns]
Postotak pogodka je: 0.6756756756756757 posto.
```

Slika 9: Postotak podudarnosti za sm0416120422.wav audio datoteku

Sa slika je vidljivo da je postotak pogotka poprilično zadovoljavajući.

Napomene

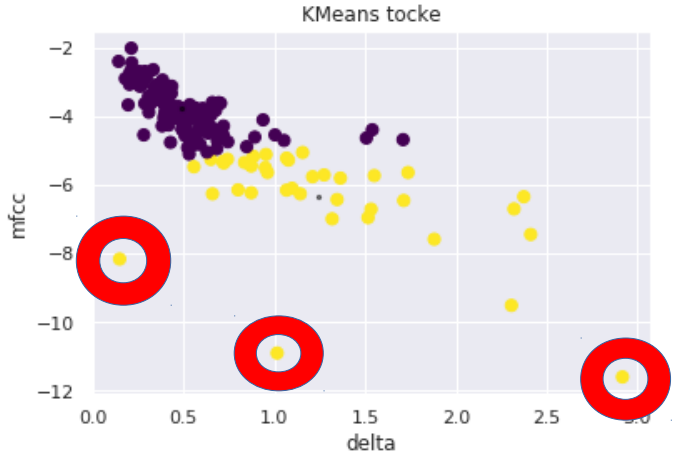
Treba napomenuti korak prilikom izračunavanja MFCC koeficijenata, odnosno:

```
mfcc = librosa.feature.mfcc(y=y, sr=sr, n_mfcc = len(df))
```

MFCCovi se izračunavaju pomoću funkcije koja pruža paket librosa. Pomoću parametra `n_mfcc` postavlja se broj MFCC koeficijenata koje želimo izračunati. Maksimalni broj MFCC koeficijenata koje je moguće izračunati pomoću ove funkcije je za 128 vokala. Stoga ako se pokuša izračunati broj MFCC koeficijenata za više od 128 vokala, skripta pada, odnosno više ne funkcionira.

Također treba obratiti pažnju prilikom izračunavanja MFCC koeficijenata. Vidimo ponovno izračunate koeficijente sa slike 4. Vidljivo je da prvih tri izračunatih MFCC koeficijenta daleko odstupaju od ostalih izračunatih koeficijenata. Kako je opisano u radu [5], uobičajeno se uzima 10 do 12 MFCC koeficijenata, od mogućih 13. Najveća mana korištenja MFCC-ova je njegova osjetljivost na šum zbog njegove osjetljivosti na

spektralnu formu. Prilikom ekstrahiranja potrebnih značajki za stvaranje baze podataka koji su se koristili prilikom izračuna kepstralnih centroida kod K-klasteringa se navedena mana u većini slučajeva može zanemariti. Naime, na slici 10 vidljiva su odskakanja pojedinih vrijednosti od klastera koji su razvrstani kao bezvučni u ovome slučaju:




```
[[ 0.48941176 -3.78884412]
 [ 1.24539467 -6.34975181]]
```

Slika 10: Odstupanja od klastera

3. Zaključak

Zadatak ovog projektnog zadataka je bilo klasificirati pojedine segmente signala govora pomoću mel-kepstralnih koeficijenata na kategoriju zvučni/bezvučni. Iz nekoliko audio datoteka su bile izvučene mel-kepstralni koeficijenti i delta značajke izračunatih MFCC koeficijenata. Taj se set podataka uvrstio u algoritam strojnoga učenja bez nadzora: K-klasteriranje. Pomoću K-klasteriranja izračunata su dva centroida za klasifikaciju vokala, odnosno za zvučni i bezvučni. Nakon toga se za svaki vokal izračunavala euklidova udaljenost od centroida, gdje se zatim ovisno o udaljenosti pojedini segmenti signala uvrštavali u kategorije zvučni/bezvučni. Neki rezultati pokazuju podudarnost do čak 76%, dok je najmanja podudarnost bila oko 60%. Stoga se može zaključiti da navedeni postupak klasifikacije segmenata signala na zvučne/bezvučne pomoću MFCC koeficijenata dobro funkcionira. Smatra se da bi se točnost algoritma vjerojatno popravila još većim povećanjem bazi značajki, izbacivanjem koeficijenata koji daleko odstupaju od prosjeka te dodavanjem novih značajki.

4. Literatura

[1] M. Sampaolo, (2017) National Library of Medicine – Voice Disorders, [online], https://www.britannica.com/topic/voice-phonetics
[2] [online], http://gramatika.hr/pravilo/podjela-glasova/1/
[3] [online], https://en.wikipedia.org/wiki/Mel-frequency_cepstrum
[4] S. Robinson. (2018), “K-Means Clustering with Scikit-Learn”, [online], https://stackabuse.com/k-means-clustering-with-scikit-learn/
[5] U. Shrawankar, “Techniques for feature extraction in speech recognition system: A comparative study”, [online], https://pdfs.semanticscholar.org/2c19/496a8910ad30e1f0848969ae1402ec6c39e6.pdf

5. Popis slika