



فرزاد فروزان فر 97440190

تمرین سیستم های چند رسانه ای بخش صوت

به نام خدا

مقدمه

استخراج داده

حذف سکوت

استخراج ویژگی

پنجره گذاری

Dtw.plot

DTW

منابع

میخواهیم به کمک روش DTW يك سیستم تصدیق هویت بسازیم. برای این منظور:

- 1 - ابتدا نام خانوادگی خود را 10 بار ضبط کنید.
- 2 - سپس کلمات دیگر را با صدای افراد دیگر نیز ضبط کنید.
- 3 - سپس با استخراج ویژگی از فایلها سعی کنید فاصله آنها را با روش DTW محاسبه نمایید. با این کار سعی کنید سطح استاندارد مناسبی پیدا کنید که سیستم صدای خودتان را در گفتن نام خانوادگی خود بشناسد و تصدیق کند اما در مقابل، صدای افراد دیگر را تایید نکند.
- 4 - تعدادی آزمایشات انجام داده و نتایج را در يك فایل pdf گزارش دهید. همچنین کدهای خود را نیز در گزارش قرار دهید.

در ضمن برای استخراج ویژگی میتوانید از کدهای آماده مانند **voicebox** در متلب یا هر کتابخانه دیگری مانند **librosa** در پایتون استفاده نماید.

مقدمه

استخراج داده

حذف سکوت

استخراج ویژگی

پنجره گذاری

Dtw.plot

DTW

منابع

این پروژه با زبان پایتون در محیط ژوپیتر- نوت بوک برنامه نویسی شده است و از کتابخانه هایی نظیر:

- Librosa
- Dtw
- Dtaidistance
- Matplotlib.pyplot
- Numpy

استفاده شده است که بطور خلاصه اعمالی چون خواندن فایل صوتی , رسم نمودار , سیگنال صوت , محاسبه فاصله بین سیگنال های صدا و استخراج ویژگی ها مورد استفاده قرار گرفته اند.

مقدمه

استخراج داده

حذف سکوت

استخراج ویژگی

پنجره گذاری

Dtw.plot

DTW

منابع

برای خواندن فایل های صوتی بصورت زیر اعمال شده است:

```
my_audio = []
audio = []
for i in range(10):
    path = editFiles[i]
    my_audio.append(sr.AudioFile(path))
    with my_audio[i] as source:
        audio.append(r.record(source))
```

بعد از استخراج داده ها به کمک یکی از کتابخانه های مترجم گوگل توانستم صحت متن هر فایل صوتی را چک کنم که مغایرتی وجود نداشته باشد و اطمینان حاصل شود که فایل به درستی ضبط و ذخیره شده است :

```
In [5]: import speech_recognition as sr
        r = sr.Recognizer()
        for i in range(10):
            r.recognize_google(audio[i], language="fa_IR")
            r.recognize_google(audio[0], language="fa_IR")
```

```
Out[5]: 'فرزاد فروزانفر'
```

مقدمه

استخراج داده

حذف سکوت

استخراج ویژگی

پنجره گذاری

Dtw.plot

DTW

منابع

بخش هایی از فایل صوتی که سیگنال سکوت هستند را حذف می کنیم تا اطلاعات اضافه و بی حاصل حذف شوند که باعث سبکتر و سریع تر شدن محاسبات می شوند.
برای انجام این کارر انرژی در هر فریم سیگنال را محاسبه می کنیم با توان رسیدن سیگنال، اعداد خیلی کوچک، کوچکتر و نزدیک به صفر خواهند شد.

```
for i in range(len(audio)):
    for j in range(len(audio[i])):
        audio[i][j] *= audio[i][j]
```

ویژگی های مورد استفاده :

مقدمه

استخراج داده

حذف سکوت

استخراج ویژگی

پنجره گذاری

Dtw.plot

DTW

منابع

- MFCC
- Mean
- Windowing
- Energy

```
In [6]: train_audio = []
sr_arr = []
mfcc_train = []
for i in range(10):

    #define variable to save data and rate audio
    path = editFiles[i]
    y, sr = librosa.load(path)

    #resize each figure to show better
    fig = plt.gcf()
    fig.set_size_inches(12.5, 19.5)

    #show matplotlib speak
    plt.subplot(10, 1, i+1)
    plt.plot(y)
    IPython.display.Audio(data=y, rate=sr)

    #Add data to train list:
    train_audio.append(y)
    sr_arr.append(sr)

    #Convert the data to mfcc:
    mfccTest = librosa.feature.mfcc(y,sr)
    mfcc_train.append(mfccTest)
```

ضریب کپسترال مل :

مقدمه

استخراج داده

حذف سکوت

استخراج ویژگی

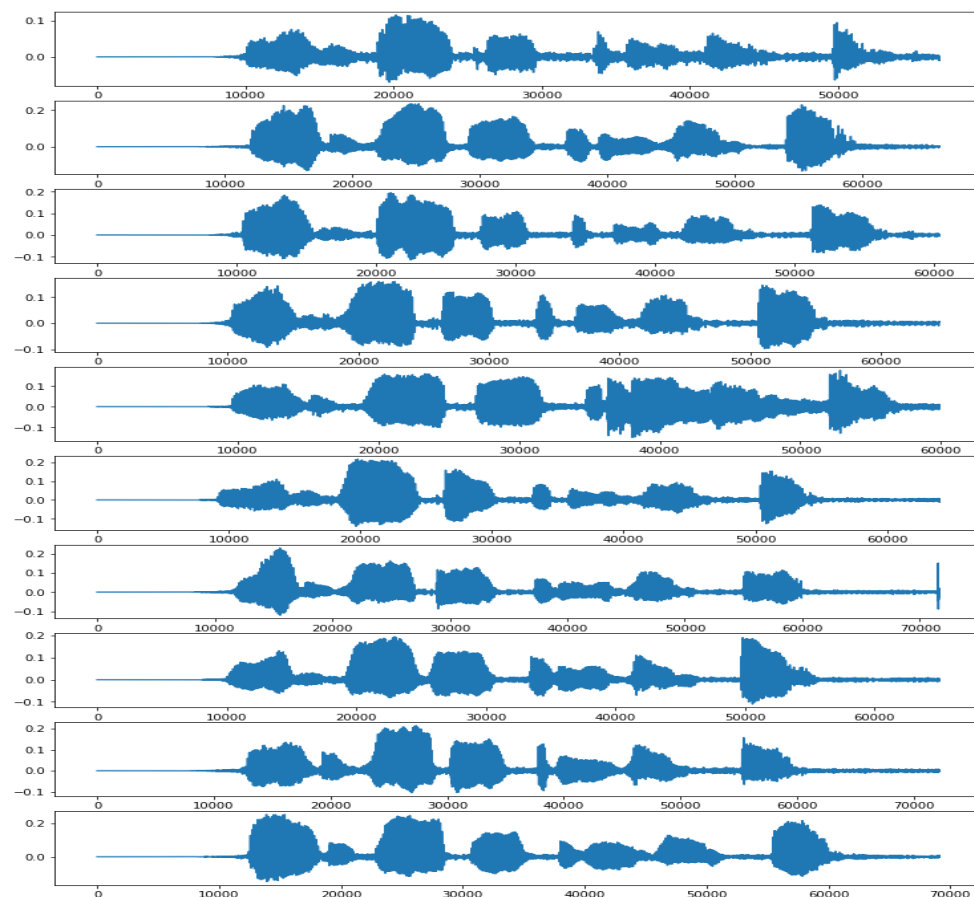
پنجره گذاری

Dtw.plot

DTW

منابع

بعد از استفاده از ویژگی ضریب کپسترال مل و ذخیره آنها در متغیری از جنس `numpy` می‌خواهیم برای درک بهتر از سیگنال های صدا آنها را نمایش داده تا بصورت نموداری قابل فهم تر باشند و حتی میتوان بصورت شهودی شباهت آنها را درک نمود.



مقدمه

استخراج داده

حذف سکوت

استخراج ویژگی

پنجره گذاری

Dtw.plot

DTW

منابع

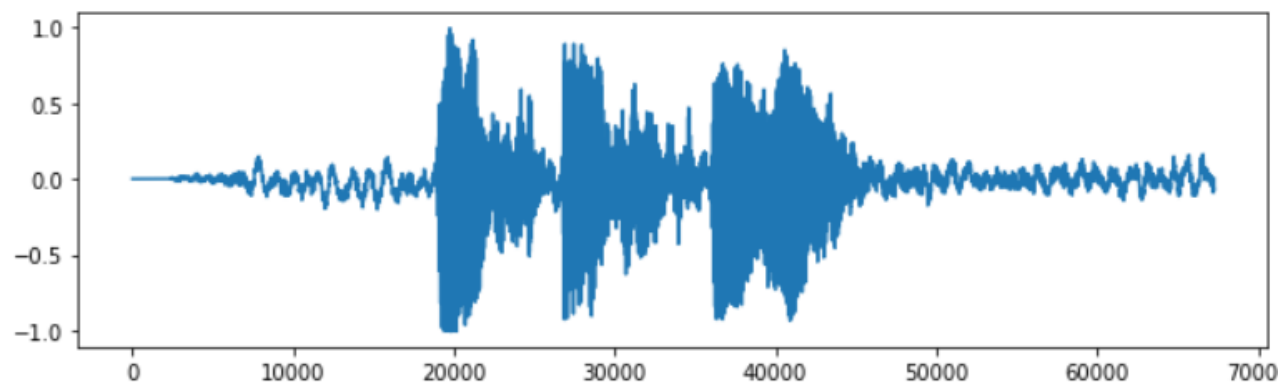
و هم چنین تمام اعمال و پردازش هایی که در اسلاید قبلی توضیح داده شدند را برای صدا شخص دیگری که ضبط شده است را متقابلا انجام دادیم :

```
In [9]: # add test audio
fig = plt.gcf()
fig.set_size_inches(10.25, 2.95)
yTest, srTest = librosa.load('Forouzanfar.wav')
plt.plot(yTest)
```

```
#Convert the data to mfcc:
```

```
mfccTest = librosa.feature.mfcc(yTest, srTest)
mfccTest = preprocess_mfcc(mfccTest)
```

```
C:\Users\win10\AppData\Local\Temp\ipykernel_10444\1072777795.py:8: FutureWarning
... -0.06607325 -0.07110196
0.          ], sr=22050 as keyword args. From version 0.10 passing these as posi
mfccTest = librosa.feature.mfcc(yTest, srTest)
```



مقدمه

استخراج داده

حذف سکوت

استخراج ویژگی

پنجره گذاری

Dtw.plot

DTW

منابع

اکنون که داده های آزمایشی و داده های آموزشی خود را جمع آوری کرده ایم، آماده حل مشکل هستیم. با استفاده از اندازه پنجره ای برابر با طول اولین مثال آموزشی از جمله آزمایشی پنجره می گیریم (از این طول تا حدودی خودسرانه استفاده می کنیم). برای هر پنجره یک فاصله DTW تا داده های آموزشی را محاسبه می کنیم و پنجره ای را با کمترین فاصله تا داده های آموزش انتخاب می کنیم. ما انتخاب می کنیم که از وزن نمایی به عنوان معیار تشابه در DTW استفاده کنیم.

```
In [14]: # have assembled our testing data and training data
from dtw import *
# Windowing
window_size = []
dists = []
dists_secondary = []
for i in range(len(mfcc_final)):
    window_elems = len(mfcc_final[i][-1])
    dist = np.zeros(mfccTest.shape[1] - window_elems)
    window_size.append(window_elems)
    dists.append(dist)

    for j in range(len(dists)):
        mfcc_i = mfccTest[:,j:j+window_size[j]]
        dists_temp = dtw(mfcc_final[j].T, mfcc_i.T, keep_internals=True)
        dists_secondary.append(dists_temp)

    dists_secondary[j].plot(type="threeway")
print(dists_secondary[1])
```

مقدمه

استخراج داده

حذف سکوت

استخراج ویژگی

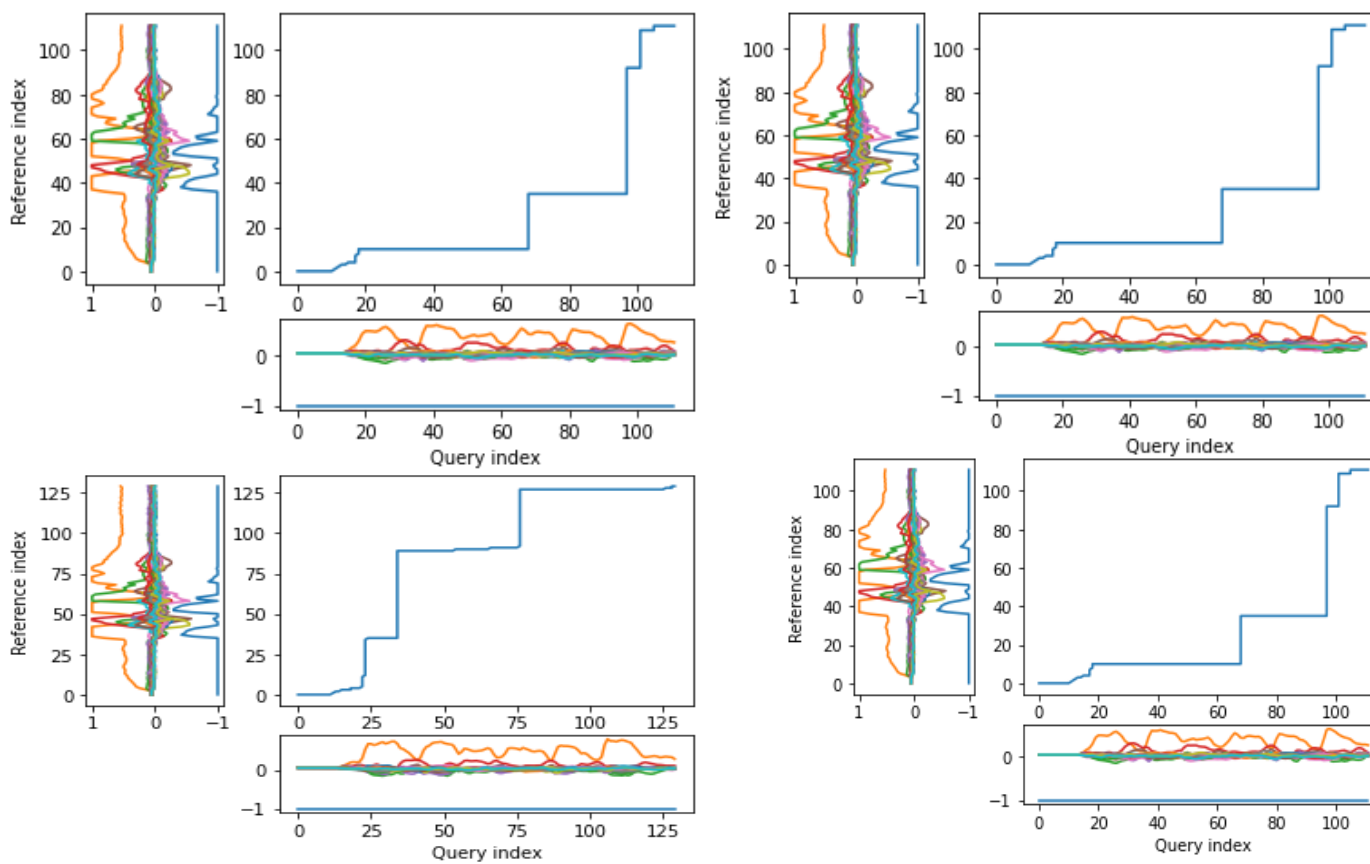
پنجره گذاری

Dtw.plot

DTW

منابع

با استفاده از همین پنجره گذاری و رسم نمودار مقایسه بین صداهاى من و شخص دیگر با استفاده از یکی از متد های dtw.plot می توان مشاهده کرد و نتیجه ای حاصل شود که نشان دهنده شباهت فایل های صدای من با یکدیگر اند و تفاوت کاملاً مشهود آنها با صدای شخص دیگری :



مقدمه

استخراج داده

حذف سکوت

استخراج ویژگی

پنجره گذاری

Dtw.plot

DTW

منابع

برای محاسبه فاصله سیگنال ها با روش dtw از تابع آماده dtw.distance استفاده می کنیم، به این شکل که فقط کافی است دو ماتریس مورد نظر را به عنوان پارامتر به آن بدهیم: در ابتدا فاصله سیگنال های صدای خودم را باهم مقایسه کردم :

```
In [17]: from dtaidistance import dtw
          from dtaidistance import dtw_visualisation as dtwvis
          import numpy as np
          temp = []
          for i in range(len(mfcc_final)):
              for j in range(len(mfcc_final[i])):
                  temp.append(dtw.distance(mfcc_final[i][i], mfcc_final[i][j]))
          path_between_my_voice = np.array(temp)

          path_between_my_voice = np.reshape(path_between_my_voice,(10,20))

          print(path_between_my_voice)
```

سپس مقایسه صداهاى خودم و صدای شخص دیگر با همین روش :

```
In [18]: temp = []
          for i in range(len(mfcc_final)):
              for j in range(len(mfcc_final[i])):
                  temp.append(dtw.distance(mfcc_final[i][i], mfccTest[j]))
          path_between_me_aother_voice = np.array(temp)

          path_between_me_aother_voice = np.reshape(path_between_me_aother_voice,(10,20))
```

مقدمه

استخراج داده

حذف سکوت

استخراج ویژگی

پنجره گذاری

Dtw.plot

DTW

منابع

	1	2	3	4	5	6	7	8	9	10
1	0	1451	1048	1208	1091	1104	1047	1106	1101	1105
2	549	0	342	496	472	484	497	508	464	473
3	487	149	0	560	751	512	857	663	831	870
4	1278	1601	923	0	688	816	855	899	675	736
5	490	389	328	331	0	305	388	350	503	377
6	574	849	369	419	377	0	407	355	391	409
7	781	822	438	291	179	237	0	248	138	122
8	547	917	324	205	186	223	156	0	192	185
9	367	324	205	186	223	156	192	185	0	249
10	983	927	495	301	166	133	197	159	163	0

DTW بین 10 صدای خودم با یکدیگر

List of users

صداهای ضبط شده	1	2	3	4	5	6	7	8	9	10
صدای شخص دیگر	1341	1560	963	1230	1056	1109	1153	1129	1139	1141

DTW بین 10 صدای خودم با صدای شخص دیگر

مقدمه

استخراج داده

حذف سکوت

استخراج ویژگی

پنجره گذاری

Dtw.plot

DTW

منابع

لینک کامل این تمرین را در ریپازیتوری گیت هاب قرار دادم.

[Github](#)

Reference :

- <https://realpython.com/python-speech-recognition/>
- <https://www.simplilearn.com/tutorials/python-tutorial/speech-recognition-in-python>
- <https://pypi.org/project/dtaidistance/>
- <https://dtaidistance.readthedocs.io/en/latest/usage/dtw.html>