Experimentation

September 21, 2022

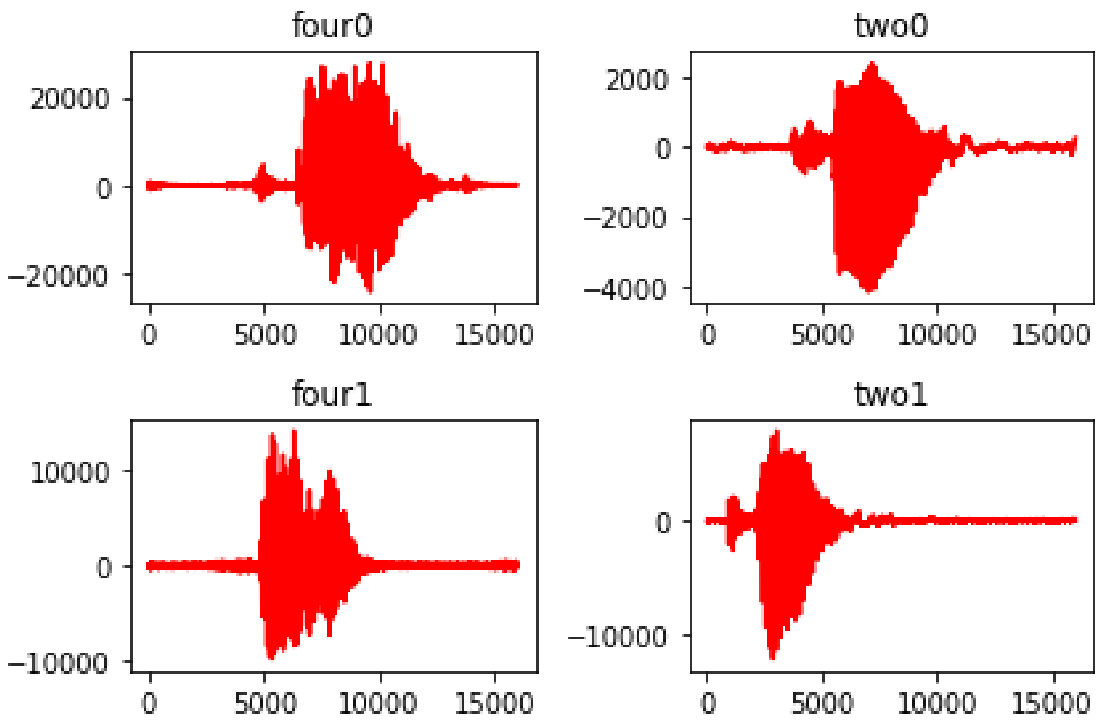
[ ]:
```
# Compile library with different flags
!rm -f libspeechr*.so
!cd .. && make clean && make AVX=false && make AVX=true
!cp ../build/libspeechr*.so ./
```

```
rm -rf ./build
rm -f ./build/libspeechr.so
rm -f ./build/test
Compiling…
gcc flags are: -Wall -Wextra -pedantic -Wdouble-promotion -fPIC
nasm flags are: -f elf64 -F DWARF -Wall
Compiling ./build/libspeechr.so…
Done!
Compiling…
gcc flags are: -Wall -Wextra -pedantic -Wdouble-promotion -fPIC -D__VECTOR_AVX__
nasm flags are: -f elf64 -F DWARF -Wall
Compiling ./build/libspeechr-avx.so…
Done!
```

[ ]:
```
from numpy.ctypeslib import *
from speechrlib import *
import wave
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = (10, 5)
```

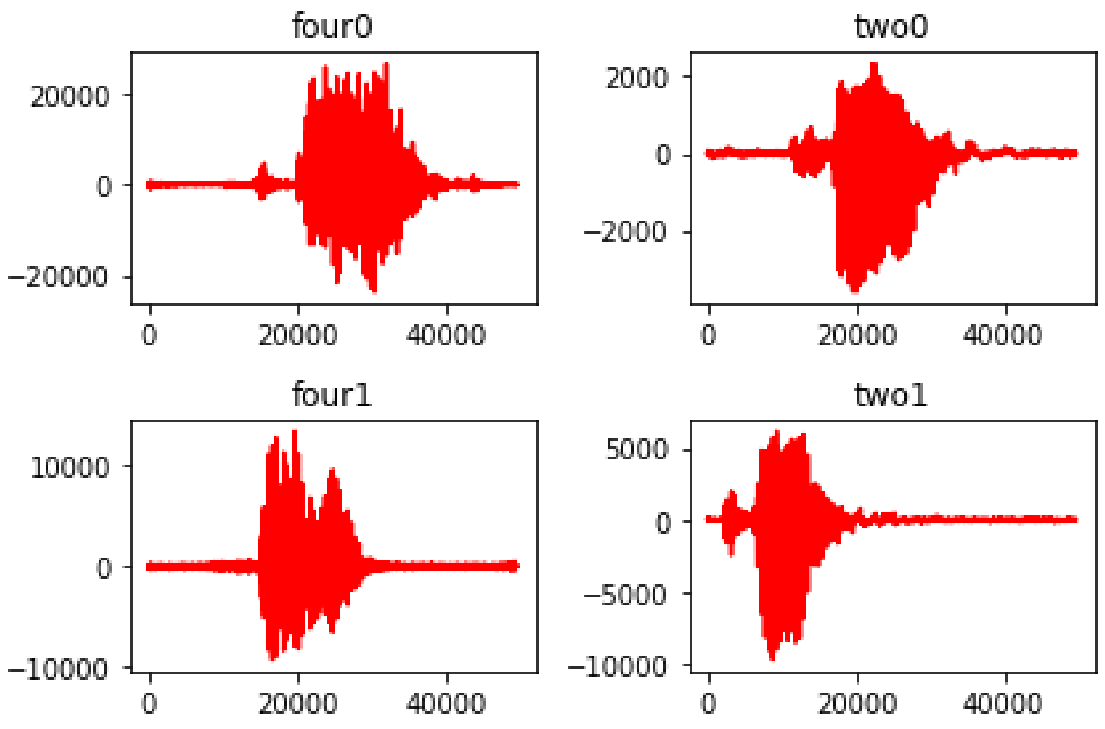
[ ]:
```
sample_names = ['four0', 'two0', 'four1', 'two1']
audio_list = []
samplerate = 16000
for i in range(0,len(sample_names)):
    audio = wave.open("../data/samples/" + sample_names[i] + ".wav")
    length = audio.getnframes()
    audio_list.append(np.frombuffer(audio.readframes(length), dtype=np.int16).
 ↪astype(c_float))
```

[ ]:
```
fig = plt.figure()
for i in range(0,4):
    fig.add_subplot(2,2,i+1)
    plt.plot(audio_list[i], c='red')
    plt.title(sample_names[i])
fig.tight_layout()
```

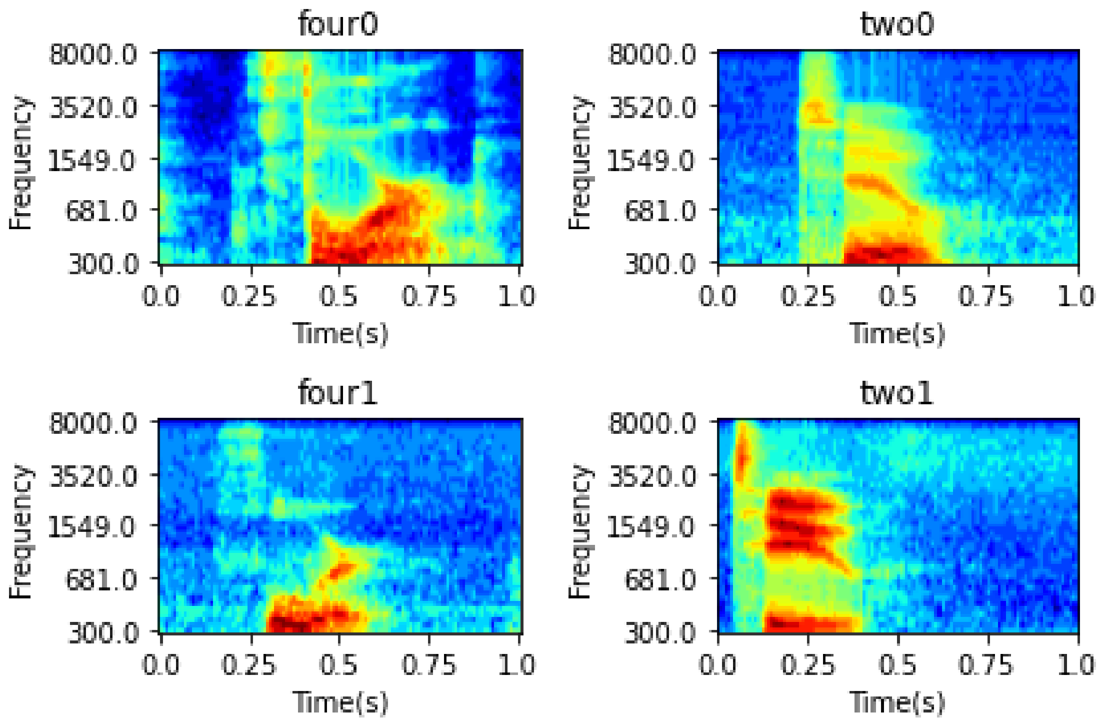


[ ]:
```
from speechrlib import *
speechr = load_speechrlib("./libspeechr.so")
```

[ ]:
```
fig = plt.figure()
for i in range(0,4):
    ax = fig.add_subplot(2,2,i+1)
    framed_audio = speechr.frame(audio_list[i].ctypes.data_as(c_float_p),␣
 ↪len(audio_list[i]), samplerate).contents
    data = as_array(framed_audio.data, [framed_audio.size])
    signal = [i[0] for i in data]
    ax.plot(signal, c='red')
    plt.title(sample_names[i])
fig.tight_layout()
```

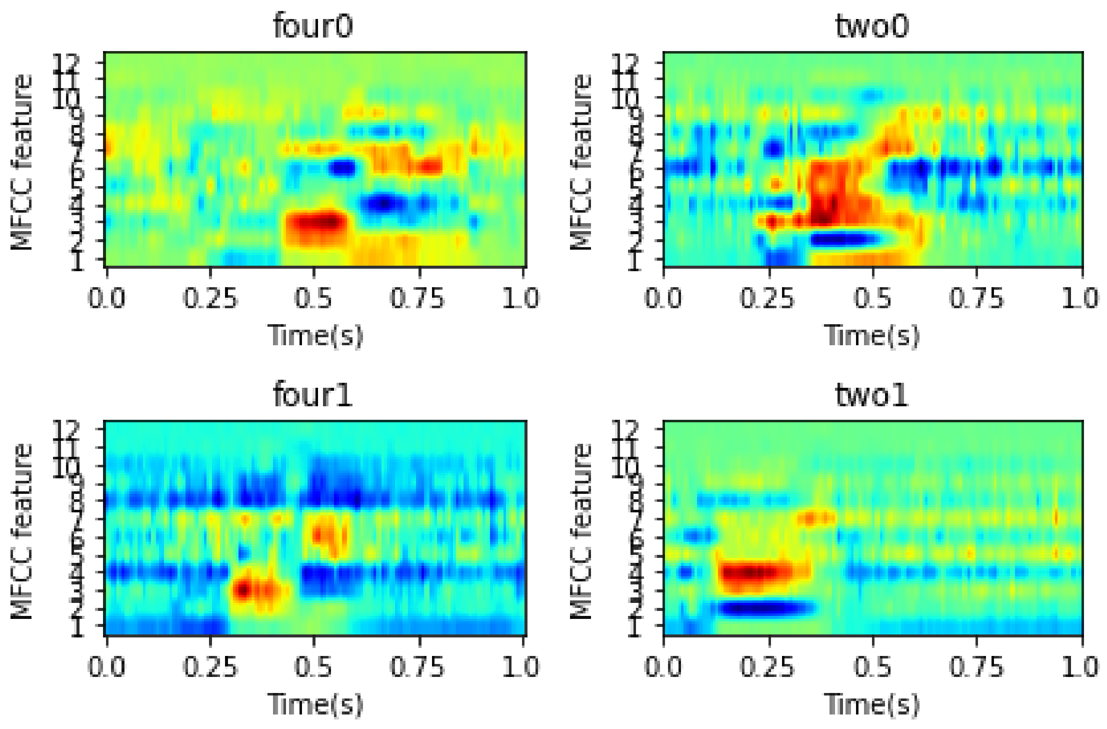


```
[ ]: fig = plt.figure()
     for i in range(0,4):
         ax = fig.add_subplot(2,2,i+1)
         spectrogram_matrix = speechr.mel_spectrogram(audio_list[i].ctypes.
      ↪data_as(c_float_p), len(audio_list[i]), samplerate)
         data = matrixf_as_array(spectrogram_matrix)
         ax.imshow(data.T, origin='lower', cmap='jet', aspect='auto')
         plt.xticks(np.linspace(0, data.shape[0]-1, num=5), np.linspace(0, 1, num=5))
         plt.yticks(np.linspace(0, data.shape[1]-1, num=5),np.trunc(np.logspace(np.
      ↪log2(300),np.log2(8000), base=2, num=5)))
         plt.title(sample_names[i])
         plt.xlabel("Time(s)");
         plt.ylabel("Frequency");
     fig.tight_layout()
```

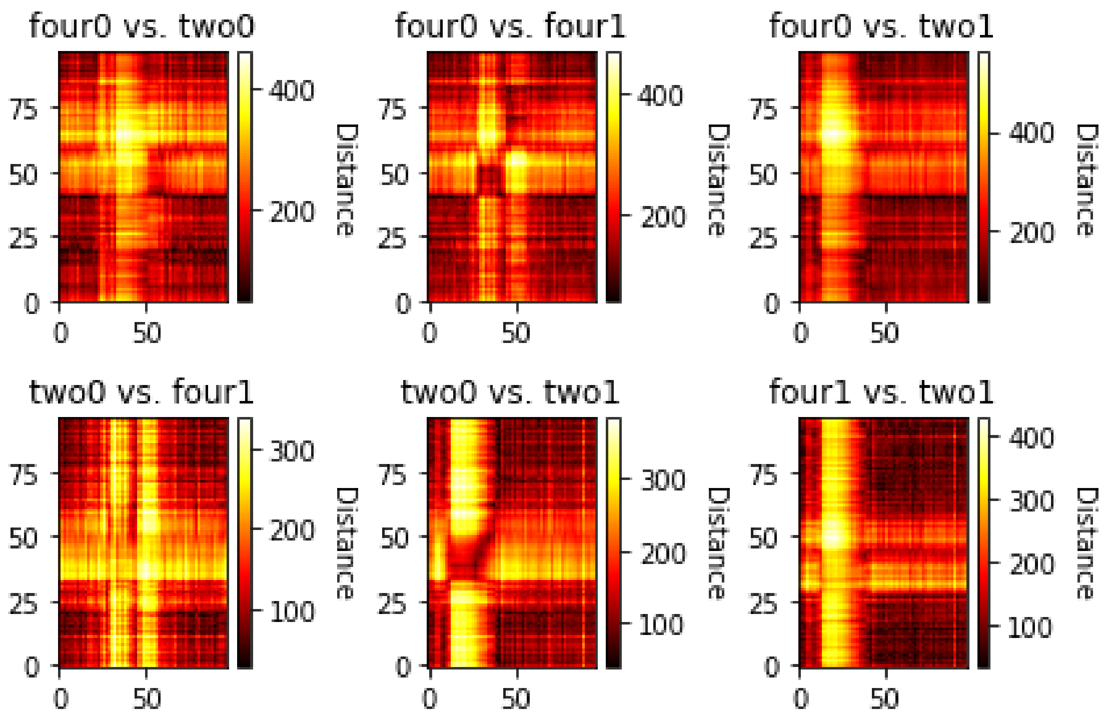


```
[ ]: feature_matrix_list = []
     for i in range(0, len(audio_list)):
         mfcc_matrix = speechr.mfcc(audio_list[i].ctypes.data_as(c_float_p),␣
      ↪len(audio_list[i]), samplerate)
         feature_matrix_list.append(mfcc_matrix)
```

```
[ ]: fig = plt.figure()
     for i in range(0,4):
         ax = fig.add_subplot(2,2,i+1)
         data = matrixf_as_array(feature_matrix_list[i])
         ax.imshow(data.T, origin='lower', cmap='jet', aspect='auto')
         plt.xticks(np.linspace(0, data.shape[0]-1, num=5), np.linspace(0, 1, num=5))
         plt.yticks(np.arange(0,12,1), np.arange(1,13,1))
         plt.title(sample_names[i])
         plt.xlabel("Time(s)");
         plt.ylabel("MFCC feature");
     fig.tight_layout()
```



[ ]:
```
from scipy import spatial
fig = plt.figure()
subplot_num = 1
for i in range(0,4):
    for j in range(i+1,4):
        fig.add_subplot(2,3,subplot_num)
        subplot_num = subplot_num + 1
        mfcc1 = matrixf_as_array(feature_matrix_list[i])
        mfcc2 = matrixf_as_array(feature_matrix_list[j])
        dmatrix = spatial.distance_matrix(mfcc1, mfcc2)
        im = plt.imshow(dmatrix, aspect='auto', interpolation='none',␣
 ↪cmap='hot', origin='lower')
        ax = fig.gca()
        plt.title(sample_names[i] + ' vs. ' + sample_names[j])
        cbar = ax.figure.colorbar(im, ax=ax)
        cbar.ax.set_ylabel('Distance', rotation=-90, va="bottom");
fig.tight_layout()
```



```
[ ]: similarity_estimations = pd.DataFrame(columns=['sample_name'] + sample_names)
     similarity_estimations['sample_name'] = sample_names
     for i in range(0,4):
         for j in range(0,4):
             value = 0 if i == j else speechr.dtw(feature_matrix_list[i],␣
      ↪feature_matrix_list[j])
             similarity_estimations[sample_names[i]][j] = value
     print(similarity_estimations)
```

```
  sample_name         four0         two0        four1         two1
0       four0             0  19753.121094  17010.320312  23419.443359
1        two0  19753.121094             0  14626.361328  13256.773438
2       four1  17010.320312  14626.361328             0  15375.027344
3        two1  23419.443359  13256.773438  15375.027344             0
```