音声処理実験
作業日報
第5回　SP-3

学生番号:09430509

氏名:今田将也

提出日:2020年11月17日

SP-3の概要

1. 問題4 （優先発展♫）
    1. 記録した6つの音声（グー，チョキ，パーそれぞれ2つ）のスペクトログラムを図示しなさい．
2. 問題 5
    1. 参照パターンと入力音声のスペクトル距離の比較に基づく単語音声認識について，考察せよ．
3. 問題 6
    1. 自身で実装した自動音声認識器を用いて，音声インタフェースを作成し，考察せよ．

問題4

じゃんけんの3手（「グー」「チョキ」「パー」）の単語発声を音声ファイルとして録音し，今後の音声認識の実験で利用できるよう準備せよ．

- いずれの音声ファイルも，サンプリング周波数 16 kHz，信号長 600 ms とすること．
- 各パターンにつき，少なくとも2つは収録すること．

問題4の小問題

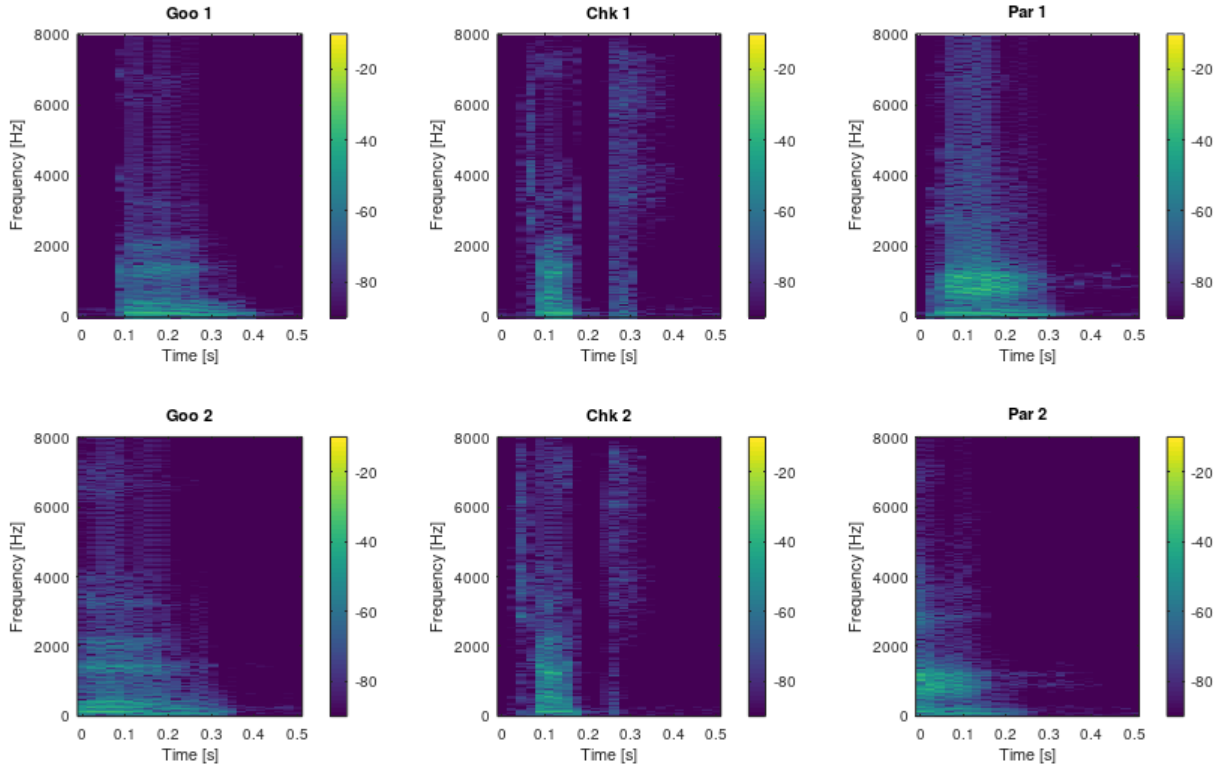
小問題（優先発展♫）

- 記録した6つの音声（グー，チョキ，パーそれぞれ2つ）のスペクトログラムを図示しなさい．

記録した6つの音声（グー，チョキ，パーそれぞれ2つ）のスペクトログラムを図示

グー，チョキ，パー，それぞれ2個ずつのスペクトログラムを表示した．

それぞれ縦はHzで，横は時間である．



問題4のまとめ

問題

じゃんけんの3手（「グー」「チョキ」「パー」）の単語発声を音声ファイルとして録音し，今後の音声認識の実験で利用できるよう準備せよ.

まとめ

スペクトログラムを図示，スペクトログラムの表示手順を確認した.

問題 5

1. 参照パターンと入力音声のスペクトル距離の比較に基づく単語音声認識について，考察せよ．

問題5の小問題

- どのようなアプローチで音声認識を実現するのか，簡単に説明しなさい．
- 問題4で収録した3+3=6つの音声の（対数）パワースペクトルを用いて，3x3=9種類の組み合わせで距離を計算し，その結果を表としてまとめなさい．

5-1.どのようなアプローチで音声認識を実現するのか，簡単に説明しなさい．

元となる教師データと，識別したいがどれだけ似ているか2音声の距離計算にスペクトル距離を用いる

ベクトル$x = [x_1\ x_2\ \cdots x_N\ ]^T$とベクトル$\mathrm{y} = [y_1\ y_2\ \cdots y_N\ ]^T$の距離$D(x, y)$を以下のように計算する

$$D(x, y) = \sqrt{\sum_{n=1}^{N} (x_n\ - y_n\ )^2}$$

5-2.問題4で収録した6つの音声のパワースペクトルを用いた，9種類の距離の結果表

	Goo2	Chk2	Par2
Goo1	888.69	1365.46	1148.65
Chk1	1313.32	737.83	1146.56
Par1	1263.56	1110.19	767.11

5-2.問題4で収録した6つの音声のパワースペクトルを用いた，9種類の距離の結果表

【表からわかること】

グーとグー同士などの同じ音同士の距離は4桁まで大きくはならないぐらいの距離になった

異なる音同士では，距離が大きくなっていて，それだけ似ていないということがわかる

問題5のまとめ

問題

じゃんけんの3手（「グー」「チョキ」「パー」）の単語発声を音声ファイルとして録音し，今後の音声認識の実験で利用できるよう準備せよ.

まとめ

どのように音声認識をするのかのアプローチについて式を提示しつつ，距離について見ることについて説明した

9種類の音の組み合わせについて距離の大きさを表にした

問題6

自身で実装した自動音声認識器を用いて，音声インタフェースを作成し，考察せよ．

- 単語認識ができればよい．
- マイクから音声を入力して，結果を画面に出力するまでの一連の流れを，MATLAB/Octaveのスクリプト（あるいは，関数）として実行できればよい．

問題6の小問題

1. MATLAB/Octave等のソースコードを用いて，実装と動作例を示しなさい．
2. 音声認識器の”良さ”を，何らかの客観的な数値を示すことで，評価しなさい．

6-1.実装手順

以下の流れでコマンド群を実行していく

1. 初期化
2. 参照パターン用の音声ファイルを読み込む
3. 読み込んだ音声ファイルのデータを対数パワースペクトルに変換する
4. 入力パターンとして音声を録音する
5. 録音した音声を対数パワースペクトルに変換する
6. 入力パターンと参照パターンそれぞれの距離を計算して，最短の距離を持つ参照パターンを見つける
7. 参照パターンに相当する文字列を表示する

6-1.実装(1/2)

%0 初期化
- すべての変数をクリアし，録音のための変数，結果表示用文字列の配列を用意する

```
%% 0. Initialization
clear;
rec = audiorecorder(16000, 16, 1);  % 16000 Hz, 16 bit, 1 channel
fft_len = 16384;
result_string_table = {'Goo', 'Chk', 'Par'};
```

%1 参照パターン用の音声ファイルを読み込む
- スクリプト化してあるため簡略的に読める

%2 読み込んだ音声ファイルのデータを対数パワースペクトルに変換
- スクリプト化してあるため簡略的に読める

```
% 1. Load waveform from WAV files
JAN_LOAD_WAVEFILES;

%% 2. Convert them to power spectrums
Jan_calc_powerspecs;
```

%3 入力パターンとして音声を録音
- マイクから入力された，0.6秒の音声は，変数xに格納

```
%% 3. Record an input waveform
disp('3'); pause(1); disp('2'); pause(1); disp('1'); pause(1); disp('Go!'); % count down
recordblocking(rec, 0.6);
x = getaudiodata(rec);
```

6-1.実装(2/2)

%4　録音した音声を対数パワースペクトルに変換

- 以前作成したスクリプトを利用

```
%% 4. Convert the input waveform to power spectrum
[PowX_dB, PowX] = calc_powerspec(x, fft_len);
```

%5-1　距離計算

```
%% 5-1. Calculate distance between the input pattern and every reference patterns
Dist(1) = sqrt( sum( (PowX_dB(:,1) - Jan_Goo_PowX_dB(:,1) ) .^2 ) );
Dist(2) = sqrt( sum( (PowX_dB(:,1) - Jan_Chk_PowX_dB(:,1) ) .^2 ) );
Dist(3) = sqrt( sum( (PowX_dB(:,1) - Jan_Par_PowX_dB(:,1) ) .^2 ) );
disp(Dist) % for debug
```

%5-2　最短距離の参照パターンの探索

- 「配列の中で最小値を持つ配列のインデックスを得る」ことで実現

```
%% 5-2. Select the pattern that has a minimum distance
[~, idx] = min(Dist);
```

%6　参照パターンに相当する文字列を表示

```
%% 6. Display the result stiring!
disp(result_string_table{idx});
```

6-2.動作例

- ASR.mというファイルで作成をしたため，ASRというスクリプトで実行する.
- 3 2 1 Go! という表示のあとに録音をする
- 録音されたものが参照パターンとどれだけ似ているか距離を表示する
- そのなかで最も小さいものを認識結果として表示
- 認識率は自分の声なので100%だった

```
>> ASR

3
2
1
Go!
    937.65    1313.69    1235.70
Goo


>> ASR
3
2
1
Go!
   1533.20     829.29    1252.95
Chk
>> ASR
3
2
1
Go!
   1191.90    1116.14     775.40
Par
```

6-3.客観的に良さを示す(1/2)

- ✓友人２人に協力してもらって，グー・チョキ・パーそれぞれの音データをもらい，自分以外のデータでも認識できるのか試す．
- ✓2x6ファイル分のループを処理するコードを書いて行った．
- ✓グー1・2，チョキ1・2，パー1・2の順で読み込んだ

それぞれの声を正しく認識できていればこの識別器は良いと言えるだろう

```
for k = 1:length(pararent_files)
    for i = 1:length(filenames)

      filename = strcat(strcat(pararent_files(k)),filenames(i));
      x = audioread(char(filename));

      %% 4. Convert the input waveform to power spectrum
      [PowX_dB, PowX] = calc_powerspec(x, fft_len);

      %% 5-1. Calculate distance between the input pattern and every reference patterns
      Dist(1) = sqrt( sum( (PowX_dB(:,1) - Jan_Goo_PowX_dB(:,1) ) .^2 ) );
      Dist(2) = sqrt( sum( (PowX_dB(:,1) - Jan_Chk_PowX_dB(:,1) ) .^2 ) );
      Dist(3) = sqrt( sum( (PowX_dB(:,1) - Jan_Par_PowX_dB(:,1) ) .^2 ) );
      disp(Dist); % for debug

      %% 5-2. Select the pattern that has a minimum distance
      [~, idx] = min(Dist);

      %% 6. Display the result stiring!
      disp(result_string_table{idx});
    end
end
```

6-3.客観的に良さを示す(2/2)

結果は右図のようになった

✓一人目の2回目のグーが正しく認識できない

✓二人目の音声についてはグーのみ正しく認識できていない

➢参照パターンが私のみであるから，ここの学習データを複数人の別々のデータを用いれば精度があがるのだろうと推測した

➢時間のずれについて考慮されていないので，参照パターンとのずれがあるため上手く認識できていないのかもしれない

```
>> ASR2

   1111.4   1370.7   1196.1
Goo
   1172.8   1253.6   1154.3
Par
   1911.4   1158.0   1385.6
Chk
   1961.9   1172.1   1479.3
Chk
   1505.0   1442.7   1116.8
Par
   1493.4   1503.7   1115.5
Par
   1990.0   1430.0   1518.6
Chk
   1923.9   1509.8   1450.4
Par
   2872.9   1996.5   2342.0
Chk
   2475.8   1678.5   1927.1
Chk
   1956.5   1606.2   1284.7
Par
   1874.7   1524.5   1216.5
Par
```

問題6のまとめ

問題

自身で実装した自動音声認識器を用いて，音声インタフェースを作成し，考察せよ.

まとめ

ソースコードをしめしながら，実装について説明し，実際に動作させて動作例を示した

友人に協力してもらい，自分以外のデータを識別させて，識別させた数値を示し考察をした

感想

正弦波のときのスペクトログラムは何も変化無くておもしくなかったけど，グーチョキパーでスペクトログラムを見ると，きちんと時間が経過しても周波数成分の特徴が保持されていることがわかり，色も変化していて視覚的におもしろかった.

識別器も自分の声はそりゃ認識できるけど，友人の声だと認識できていなくてちょっと残念だったけど，なぜ認識できないのかということを考えられたので良かった.