

## 14 相互相関

### 14.1 ベクトルの類似度

ベクトルの類似度を計算する方法はいくつかある。計算方法の特徴を考察するため、最も単純なベクトルとして2次元ベクトルを考える。2次元空間での2次元ベクトルは平面上の点を表わす。

ここで3つのベクトル  $w = (1, 1)$ ,  $x = (3, 3)$ ,  $y = (-1, 1)$  を考える。

2つの点の距離は、2つの点の近さを表わす。2点のユークリッド距離は次のように計算できる。

```
>> w=[1 1];
>> x=[3 3];
>> y=[-1 1];
>> norm(w-y)

ans =

     2
```

次に、 $w = (1, 1)$ ,  $x = (3, 3)$ ,  $y = (-1, 1)$  を位置ベクトルだと考える。位置ベクトルには向きがある。向きがどのくらい似ているか、を考えることもできる。

ベクトル間の角度は内積を用いて計算できる。 $a$  と  $b$  の内積は  $a \cdot b$  とあらわすことにする。内積は次のように計算できる。

```
>> dot(w,x)

ans =

     6

>> w*x'

ans =

     6
```

線型代数では、通常、ベクトルは列ベクトルなので、列ベクトルをそのまま、行ベクトルは列ベクトルを転置したものの、と表記することになると、上記の式は次のようになる。

$$\mathbf{a} \cdot \mathbf{b} = \mathbf{a}^T \mathbf{b} \quad (17)$$

内積と、2つのベクトルの角度  $\theta$  の関係は次のようになる。

$$\cos \theta = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|} \quad (18)$$

$\cos \theta$  の値は、 $0 \leq \theta \leq \pi$  では、 $\theta$  が大きくなるにつれて小さくなる。つまり、向きが同じときに最大で、似てないほど  $\cos \theta$  の値は小さくなり向きが逆のときに最小になる。したがって、類似度としては、 $\theta$  よりも  $\cos \theta$  が適している。

他にも様々な距離の計算方法がある。これらの中からどれを選べばよいかは、目的によって変わる。

系列の類似度の計算方法を考えるために、もう少し点数を増やして4点のベクトルを考える。

```
>> a=[1 5 -1 3];
>> b=[4 20 -4 12];
>> c=[-3 1 5 1];
>> plot(a); hold on; plot(b,'r'); plot(c,'g')
```

プロットを見ればわかるように、b は、a を 4 倍したものである。波として見れば、振幅が異なるだけで形は同じである。一方、c は a,b とは違う形である。これらは、ドット積とユークリッド距離では、どの組合せが近いかが変わる。

統計学で 2 つの確率変数の類似度の指標となる相関係数は、同じ長さのベクトル  $x, y$  の  $x_i, y_i$  が組になっていると見なしたときに、次のように計算される。

$$\frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (19)$$

ただし、 $\bar{x}, \bar{y}$  はそれぞれ  $x, y$  の (相加) 平均である。この式をよく見てみると、式 (18) と同じ形であることがわかる。

## 14.2 相互相関関数

長さを同じにそろえた系列どうしであれば、内積やユークリッド距離を簡単に計算できる。しかし、実際に信号が似ているかどうかを調べたいときには、長さがまちまちであったり、一部だけが似ているということが多い。

例えば、ある CM が、あるテレビ局が放映した番組で流されたかどうかを調べたいとする。すると、 $x$  を CM (の音声)、 $y$  をテレビ番組 (の音声) とすると  $y$  のどの部分に  $x$  (と似た信号) が含まれているかを調べる問題になる。

このようなときに利用できるのが相互相関関数 `xcorr` である。

ソースコード 28: 相互相関の計算例

```

1 >> y=[8 8 -3 4 -6 -10];
2 >> x=[8 -3 4 -6];
3 >> xc=xcorr(x,y)
4 xc =
5
6   -80.0000   -18.0000    10.0000    0.0000   125.0000    4.0000    26.0000   -16.0000   -48.0000    0.0000
7   0.0000
8 >> plot(xc)

```

この例では、短い  $x$  が長い  $y$  の 2 番目から 5 番目に含まれている。

この `xcorr` は、長さが異なるベクトルの場合には、長さをそろえてから、ずらしながら内積を計算している。`help` コマンドで確認できるように、長さが異なる場合には、短いベクトルに 0 を付加する。

この場合、 $x$  を  $[8 -3 4 -6]$  から  $[8 -3 4 -6 0 0]$  とする。

以後、ずらしながら内積を計算する。まず、次のようにずらして対応をとる。

```

8 8 -3 4 -6 -10
   8 -3 4 -6 0 0

```

(-10 と 8 の対応がとられている)。重なっているところをベクトルとしてみると  $[-10]$  と  $[8]$  となる。

```

>> dot([-10],[8])
ans =
   -80

```

であるので、`xcorr(x,y)` の 1 番目の要素は -80 となっている。

以後、1 つずつずらして同様に計算する。2 番目の要素を計算するときには、次のようになる。

```

8 8 -3 4 -6 -10
   8 -3 4 -6 0 0

```

したがって、`dot([-6 -10],[8 -3]) = -18` となる。

最後は、次のようになる。

```
8 8 -3 4 -6 -10
8 -3 4 -6 0 0
```

したがって、0 となる。

このような計算なので、値の大きなところが、最も内積が大きくなるところで、ソースコード 28 では、`xc(5)` である。これは、最初から 5 番目というよりは、真ん中の 6 番目から 1 つ左にずらしたと考えるのがよい。(真ん中は、両方の最初の要素を揃えた場合に対応する。) つまり、 $y$  を 1 つ左にずらしたときに  $x$  と一致することを反映している。

内積を用いているため、`xcorr` は全く同一の系列でなくても、似ている部分があるかどうか、どこにあるかを調べることができる。

## 15 自己相関

ある関数  $x$  が周期関数であるということは、周期が  $T$  秒とすると、次の式であらわされる。

$$x(t) = x(t + T) \quad (20)$$

離散データの場合は、 $N$  点が周期だとすると、次の式になる。

$$x[n] = x[n + N] \quad (21)$$

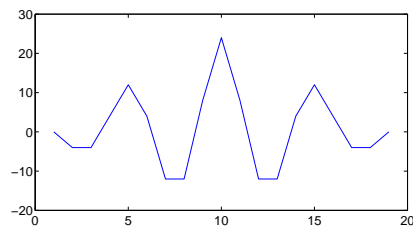
この式は、部分が似ている、という視点で見ると、周期関数はその信号自身のある部分と、いくつかずらした部分が一致する、といえる。

この考えに基づくと、前節の相互相関を同じベクトルに対して計算して、ずらして最も値が大きくなるところが周期をあらわすと推定できる。

相互相関関数を用いて、同じ信号の相互相関を見してみる。

```
>> y=[0 2 2 0 -2 0 2 2 0 -2];
>> yc=xcorr(y,y);
>> plot(yc)
```

プロットされるグラフは次のようになる。



真ん中の一番高いところは、開始点を揃えて比較した場合である。このグラフは左右対称になるので、右側だけ見てみる。真ん中から右に 5 点目が真ん中の次に大きな値をとっている。これは、5 点ずらしたとき最も似ているということである。 $y$  の系列を見ても、周期が 5 点であるという推測は妥当であろう。

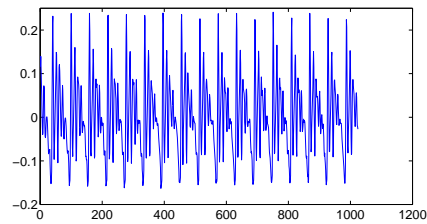
実は、MATLAB では、`xcorr(y,y)` は `xcorr(y)` と書ける。この形式であらわされる、このような同じ信号に対する相互相関関数を特に「自己相関関数」とよぶ。ところで、自己相関では、中央の値が常に最大になる。したがって、その値が 1 になるように、その値 (最大値) で全体を正規化することが多い。

自己相関関数で調べられる周期は基本周波数 ( $f_0$ ) と呼ばれる。基本周波数は音の高さを調べるのに使われることが多い<sup>4</sup>。

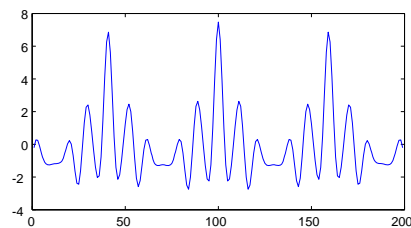
#### ソースコード 29: 母音「あ」の時間波形の拡大プロット

```
1 >> [y, fs]=wavread('a-.wav');
2 >> plot(y)
3 >> a=y(2001:3024);
4 >> plot(a)
5 >> ac=xcorr(a);
6 >> plot(ac(1024-99:1024+99))
```

2行目のプロットを拡大し、周期性がありそうな部分を抽出する (3行目)。4行目のプロットは以下ようになる。



プロットから、ほぼ一定の周期があることが確認できる。この部分の自己相関をとった結果のプロットの中央部を拡大する (6行目) と次のようになる。



このプロットを拡大すると159点目にピークがあることがわかる。100点目が中央なので、周期は59点であることがわかる。

<sup>4</sup>音声ファイルは授業支援システムからダウンロードしておくこと

## 16 時間波形のフレーム処理

単語や文章をしゃべった音声は、イントネーションやアクセントに起因して、周期が刻一刻と変化する。このようなときには、8章で述べたフレーム処理をするのがよい。

MATLAB の特徴を用いると、for ループを利用せずに音声をフレーム分割できる。

ソースコード 30: 行列によるインデクス指定

```
1 >> x=(1:9)*2
2
3 x =
4
5     2     4     6     8    10    12    14    16    18
6
7 >> ind=[1 3 5 7; 2 4 6 8; 3 5 7 9]
8
9 ind =
10
11     1     3     5     7
12     2     4     6     8
13     3     5     7     9
14
15 >> x(ind)
16
17 ans =
18
19     2     6    10    14
20     4     8    12    16
21     6    10    14    18
22
23 >>
```

`x` は偶数の行ベクトルである。7行目で3行4列の行列 `ind` を設定している。15行目では、この `ind` を使って、ベクトル `x` にアクセスしている。すると、返り値は、`ind` と同じ大きさの行列になり、`ind` の値に対応した値が代入される。

よく見ればわかるように、`ind` を用いると、フレーム幅3、フレームシフト(ずらし幅)2のフレームを生成できる。

この `ind` のような行列は、`repmat` という関数を利用して簡単に作ることができる。

```
>> W=3;
>> N=4;
>> SP=2;
>> repmat((1:W)',1,N)+repmat((0:(N-1))*SP,W,1)

ans =

     1     3     5     7
     2     4     6     8
     3     5     7     9
```

`repmat` がどのような関数であるか `help` コマンドで調べること。

この考え方をういて音声信号をフレームに分割して処理するための関数を生成する。MATLAB のメニューの [ファイル]-[新規作成]-[ファンクション M-ファイル] を選び、次のように入力して、frameindex.m という名称で保存する。

#### ソースコード 31: 関数 frameindex

```
1 function findex = frameindex(framelength, noverlap, signallength)
2 %FRAMEINDEX 音声信号をフレーム処理用に変形するためのインデクスを出力
3 % FINDEX=FRAMEINDEX (FRAMELENGTH,NOVERLAP,SIGNALLENGTH) は
4 % 音声信号をフレーム幅 FRAMELENGTH, ずらし幅 NOVERLAP で
5 % フレームに分割するためのインデクスを生成する。
6 % 音声信号の長さ(点数)を SIGNALLENGTH で与える。
7 n=fix((signallength-framelength)/noverlap);
8 findex= repmat( (1:framelength) ',1,n)+ repmat( (0:(n-1))*noverlap, framelength, 1);
9 end
```

関数 frameindex は、例えば、次のようにフレーズの周期の変化をプロットするのに利用できる<sup>5</sup>。

#### ソースコード 32: フレーズの周期のプロット

```
1 [y,fs]=wavread('aiueo8.wav');
2 yframe=y(frameindex(256,64,length(y)));
3 [f,nframe]=size(yframe);
4 for i=1:nframe,
5     xc=xcorr(yframe(:,i));
6     [pks,loc]=findpeaks(xc(257:end),'minpeakheight',xc(256)*0.5);
7     if isempty(loc),
8         f0(i)=nan;
9     else
10        [v,maxindex]=max(pks);
11        f0(i)=loc(maxindex);
12    end
13 end
14 plot(f0);
```

isempty や max の使い方は help など各自調べること。8 行目の nan は不定値を示す。不定値とは  $0/0$  や  $\infty - \infty$  など数学的に定義されないような計算結果として使われる。通常、文章中では NaN などと表現されることが多い。

```
>> 0/0
ans =
     NaN
```

Java など多くの言語では、このような演算が起きるとエラーが出ることが多いが、MATLAB ではエラーを出さない。つまり、多くの関数が、入力の一部に NaN を含んでいても、その部分は無視して処理できるようになっているので注意が必要である。

```
>> x=[5 nan];
>> x'
ans =
     5
     NaN
```

<sup>5</sup>音声ファイルは授業支援システムからダウンロードしておくこと