

特異値分解

特異値分解を求める関数 SVD の使用例をリスト 1 に示す。

リスト 1 特異値分解のための関数 SVD の使用例。

```
#include "stdafx.h"
#include <iostream>
#include "myMat.h"
#include "myEigen.h"

using namespace System;
using namespace std;
using namespace myLib;

int main(array<System::String ^> ^args)
{
    myMat a(2, 3);
    a[0][0] = 2.0; a[0][1] = 1.0; a[0][2] = 0.0;
    a[1][0] = 0.0; a[1][1] = 1.0; a[1][2] = 2.0;

    cout << "a =" << endl;
    WriteMat( 7, a );

    myMat U(2, 2);
    myMat V(3, 3);
    myMat lambda(3, 1);
    int rnk;

    SVD( a, rnk, U, V, lambda );

    Console::WriteLine("Root of 6 = " + Math::Sqrt(6.0));
    Console::WriteLine("1/Root(2) = " + 1.0/Math::Sqrt(2.0));
    Console::WriteLine("1/Root(3) = " + 1.0/Math::Sqrt(3.0));

    cout << "rank = " << rnk << endl;
    for (int i = 0; i < rnk; i++) {
        cout << "Lambda-" << (i+1) << " = " << lambda[i][0] << endl;
        for (int j = 0; j < a.m; j++) cout << " u/" << U[j][i];
        cout << endl;
        for (int j = 0; j < a.n; j++) cout << " v/" << V[j][i];
        cout << endl;
    }

    myMat a0(2, 3);
    cout << endl << "a0 =" << endl;
    WriteMat( 7, a0 );
    SVD( a0, rnk, U, V, lambda );
    cout << "rank = " << rnk << endl << endl;

    myMat a1(3, 2);
```

```

a1 = matTransp( a );
cout << "a1 =" << endl;
WriteMat( 7, a1 );

myMat U1(3,3);
myMat V1(2,2);
myMat lambda1(3,1);
int rnk1;

SVD( a1, rnk1, U1, V1, lambda1 );

Console::WriteLine("Root of 6 = " + Math::Sqrt(6.0));
Console::WriteLine("1/Root (2) = " + 1.0/Math::Sqrt(2.0));
Console::WriteLine("1/Root (3) = " + 1.0/Math::Sqrt(3.0));

cout << "rank1 = " << rnk1 << endl;
for (int i = 0; i < rnk1; i++){
    cout << "Lambda1-" << (i+1) << " = " << lambda1[i][0] << endl;
    for (int j = 0; j < a1.m; j++) cout << " u1/" << U1[j][i];
    cout << endl;
    for (int j = 0; j < a1.n; j++) cout << " v1/" << V1[j][i];
    cout << endl;
}

Console::WriteLine();
Console::WriteLine("Enter キーを押して終了");
Console::ReadLine();
return 0;
}

```

リスト 1 では、特異値分解をヘッダーファイル `myEigen.h` に宣言されている関数 `SVD` によって行っている。行列 `a` の特異値分解は

$$\text{SVD}(a, \text{rnk}, U, V, \text{lambda});$$

と関数 `SVD` を呼び出すと、`rnk`, `U`, `V`, `lambda` にそれぞれ `a` の階数、左特異ベクトル、右特異ベクトル、特異値が返される。

リスト 1 において `#include` の対象となっているヘッダーファイル `mymat.h` および `myEigen.h` を利用するときは、それらのファイルで宣言されているものの定義が置かれているファイル `mymat.cpp` および `myEigen.cpp` も他の `*.cpp` ファイルや `*.h` ファイルと同じフォルダにコピーして、メニュー「プロジェクト | 既存項目の追加」により追加しておく。ヘッダーファイルは `#include` により挿入されているが、`*.cpp` ファイルはこの「プロジェクト | 既存項目の追加」によって追加しておかないとビルド時にエラーとなる。

リスト 1 のプログラムの実行例を図 1 に示す。

```

a =
    2    1    0
    0    1    2
Root of 6 = 2.44948974278318

```

```

1/Root(2) = 0.707106781186547
1/Root(3) = 0.577350269189626
rank = 2
Lambda-1 = 2.44949
  u/0.707107  u/0.707107
  v/0.57735  v/0.57735  v/0.57735
Lambda-2 = 2
  u/-0.707107  u/0.707107
  v/-0.707107  v/0  v/0.707107

a0 =
      0      0      0
      0      0      0
rank = 0

a1 =
      2      0
      1      1
      0      2
Root of 6 = 2.44948974278318
1/Root(2) = 0.707106781186547
1/Root(3) = 0.577350269189626
rank1 = 2
Lambda1-1 = 2.44949
  u1/0.57735  u1/0.57735  u1/0.57735
  v1/0.707107  v1/0.707107
Lambda1-2 = 2
  u1/-0.707107  u1/0  u1/0.707107
  v1/-0.707107  v1/0.707107

Enter キーを押して終了

```

図1 リスト1のプログラムの実行例。

プログラムを実行すると、まず特異値を行う1番目の行列 a の要素が出力される。この行列の固有分解は次のようになっている。

$$\begin{pmatrix} 2 & 1 & 0 \\ 0 & 1 & 2 \end{pmatrix} = \sqrt{6} \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \begin{pmatrix} 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} \end{pmatrix} + 2 \begin{pmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \begin{pmatrix} -1/\sqrt{2} & 0 & 1/\sqrt{2} \end{pmatrix}$$

この固有分解に図1の結果は対応している。

2番目の行列 a0 は

```
myMat a0(2, 3);
```

と宣言されている。クラス型 myMat の行列はすべての要素が0である状態で初期化されている。したがって、

```
SVD( a0, rnk, U, V, lambda );
```

と関数 SVD を呼び出して返される rnk の値は0である。

3番目の行列 a1 は

```
a1 = matTransp( a );
```

で与えられ、 a の転置行列である。 a_1 の特異値分解は a の特異値分解において左特異ベクトルと右特異ベクトルを入れ替えた形になっている。