

Pythonでの行列とベクトルの積

ここまで、プログラミング言語における数学表現は必ずしも厳密に数学の規則に従っているとは言えない例をいくつか見てきましたが、行列とベクトルの積についてもPythonでの計算には厳密数学とは異なる点があります。まず、 m 行 n 列の行列と n 次ベクトルとの積の定義について再度確認します。

$$\text{行列 } \mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \text{ と列ベクトル } \mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} \text{ との積 } \mathbf{A}\mathbf{v} \text{ は下記のようになります。}$$

$$\mathbf{A}\mathbf{v} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} = \begin{pmatrix} \sum_{k=1}^n a_{1k}v_k \\ \sum_{k=1}^n a_{2k}v_k \\ \vdots \\ \sum_{k=1}^n a_{mk}v_k \end{pmatrix}$$

この定義は n 次ベクトルを n 行1列の行列と見なすことで行列の積の理論の下に統一されています。

具体的な例を見てみましょう。次の行列 \mathbf{A} とベクトル \mathbf{v} について積をとってみましょう。

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}, \mathbf{v} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

行列の積の定義に沿って計算します。

$$\mathbf{A}\mathbf{v} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 1 \times 1 + 2 \times 2 + 3 \times 3 \\ 4 \times 1 + 5 \times 2 + 6 \times 3 \\ 7 \times 1 + 8 \times 2 + 9 \times 3 \end{pmatrix} = \begin{pmatrix} 1 + 4 + 9 \\ 4 + 10 + 18 \\ 7 + 16 + 27 \end{pmatrix} = \begin{pmatrix} 14 \\ 32 \\ 50 \end{pmatrix}$$

この計算をPythonで実行するためにNumPyをインポートします。

```
import numpy as np
```

```
In [1]: import numpy as np
```

数学的に正しい定義にそって、 \mathbf{A} と \mathbf{v} を与えて計算します。

```
A = np.array([[1,2,3],[4,5,6],[7,8,9]])
v2 = np.array([[1],[2],[3]])
A.dot(v2)
```

ここでベクトル変数の名前を $v2$ とします。

```
In [2]: A = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])  
v2 = np.array([[1], [2], [3]])  
A.dot(v2)
```

```
Out[2]: array([[14],  
              [32],  
              [50]])
```

この例で注目することは列ベクトル \mathbf{v} の定義が二重配列で3行1列の行列として定義されていることです。これは数学的定義に全く忠実なプログラムになっています。 Jupyter notebookのCodeセルにて \mathbf{v} を表示させてみます。

```
In [3]: v2
```

```
Out[3]: array([[1],  
              [2],  
              [3]])
```

このプログラムは全く正しいです。しかしPythonでのベクトルの取扱い方法は単純配列としていました。

```
v1 = np.array([1,2,3])  
v1
```

```
In [4]: v1 = np.array([1, 2, 3])  
v1
```

```
Out[4]: array([1, 2, 3])
```

それでは、このように定義して行列とベクトルの積を計算しましょう。

```
A.dot(v1)
```

```
In [5]: A.dot(v1)
```

```
Out[5]: array([14, 32, 50])
```

数学的定義と厳密には合っていないですが、Pythonの中でこのように使われています。数学の理論を理解する上では厳密性が重要になりますが、プログラミングにおいては柔軟性を受け入れることをお勧めします。