

内積で近さを測る

ベクトル同士の関係性を知る方法

集合だけだと身動きできないが、演算によって互いに行き来できるようになった

ただし、2つのベクトルを取り出したときに、それらが似ているかどうかを議論するためには道具が少し必要となる

それがベクトルの内積である

矢印で考えた場合、内積は次のように定義された

- 1. \vec{x} と \vec{y} のなす角を θ とする
- 2. ベクトル \vec{x} の大きさを $|\vec{x}|$ 、ベクトル \vec{y} の大きさを $|\vec{y}|$ とする
- 3. \vec{x} と \vec{y} の内積を $\vec{x} \cdot \vec{y} = |\vec{x}| |\vec{y}| \cos \theta$ とする

矢印で記述できる場合にはこれでも大丈夫だが、想像できないような4次元以上の高次元では、角度 θ から出発するわけにはいかない
そのため、順番を逆にして定義していく

* * *

内積はスカラー値を与える関数

内積を「二つのベクトルを引数にとり、スカラー値を返す関数」として捉えてみる

ただし、どんな関数でもよいわけではなく、いくつかの性質を満たす必要がある

2つのベクトル \mathbf{a}_1 と \mathbf{a}_2 を考える

これらは D 次元空間内の矢印だとし、それぞれのベクトルを成分に分けて以下のように書くことにする

$$\mathbf{a}_1 = \begin{bmatrix} a_{1,1} \\ a_{1,2} \\ \vdots \\ a_{1,D} \end{bmatrix}, \quad \mathbf{a}_2 = \begin{bmatrix} a_{2,1} \\ a_{2,2} \\ \vdots \\ a_{2,D} \end{bmatrix}$$

1つ目の添え字はどちらのベクトルかを指定するもので、2つ目の添字が空間の次元を示す

これら2つのベクトルの内積を以下のように定義する

$$\mathbf{a}_1 \cdot \mathbf{a}_2 = \sum_{d=1}^D a_{1,d} a_{2,d}$$

ベクトルの要素ごとにかけ算をして足し合わせる、というだけ

* * *

内積の「書き方」は一つではない

内積の記法はいくつかある

- $\mathbf{a}_1 \cdot \mathbf{a}_2$
- $(\mathbf{a}_1, \mathbf{a}_2)$
- $\mathbf{a}_1^\top \mathbf{a}_2$
- $\langle \mathbf{a}_1 | \mathbf{a}_2 \rangle$

$\mathbf{a}_1 \cdot \mathbf{a}_2$ の記法は、本書では今後は使わない

$(\mathbf{a}_1, \mathbf{a}_2)$ では括弧が閉じていて、2つのベクトルを用いていることがわかりやすい

これは数学でよく使う

$\mathbf{a}_1^\top \mathbf{a}_2$ は、すでに行列について学んだ人にはわかりやすい表記

ただし、本書では先にこの記法を導入しておく
縦向きにならんだベクトルを横向きに転置した \mathbf{a}_1^\top を左側に置き、右側のベクトル \mathbf{a}_2 とならべて書いたときに、「要素ごとの積の総和」を意味することにする

$\langle \mathbf{a}_1 | \mathbf{a}_2 \rangle$ の記法は物理学、特に量子力学の分野でよく用いられるもの

* * *

内積の「定義」ですら一つではない

実は内積と呼ばれる量はこれだけに限らない

たとえば物理学の一般相対性理論では曲がった空間を考える

すると、ベクトル同士の関係性が、空間の曲がり方によって変わる

内積は関係性を議論するための道具なので、曲がった空間には曲がった空間なりの関係性、つまり内積が定義される

形式的な定義を与えたとき、その具体的な可能性はいろいろとあり得るのが数学のよいところ

答えや手段が一つに決まらないのは不安かもしれないが、逆に言えば、たくさんの可能性のなかから目的にあったものを選び取れるということ

* * *

内積の形式的な定義

ここでは \mathbb{R} 上の線形空間 V を考える

このとき、2つのベクトルを引数にとり、実数を返す関数 $(\cdot, \cdot) : V \times V \rightarrow \mathbb{R}$ として、次の性質を満たすものを **内積** と呼ぶ

なお、ここでは $\mathbf{u}, \mathbf{v}, \mathbf{w} \in V$ 、 $c \in \mathbb{R}$ とする

1. $(\mathbf{u}, \mathbf{v}) = (\mathbf{v}, \mathbf{u})$
2. $(c\mathbf{u}, \mathbf{v}) = (\mathbf{u}, c\mathbf{v}) = c(\mathbf{u}, \mathbf{v})$
3. $(\mathbf{u} + \mathbf{v}, \mathbf{w}) = (\mathbf{u}, \mathbf{w}) + (\mathbf{v}, \mathbf{w})$, $(\mathbf{u}, \mathbf{v} + \mathbf{w}) = (\mathbf{u}, \mathbf{v}) + (\mathbf{u}, \mathbf{w})$
4. $(\mathbf{u}, \mathbf{u}) \geq 0$, $(\mathbf{u}, \mathbf{u}) = 0 \Leftrightarrow \mathbf{u} = 0$

1 番目は対称性を意味しており、順番を変えても結果が変わらない

2 番目と 3 番目の性質は **双線形性** と呼ばれるもの

4 番目は、自分自身との内積は負の値にならないことを意味している

* * *

線形代数という言葉にも使われている **線形性** についても触れておこう

関数 $f : V \rightarrow \mathbb{R}$ が線形であるとは、以下の 2 つの性質を満たす場合を言う

ここでは $\mathbf{u}, \mathbf{v} \in V$ 、 $c \in \mathbb{R}$ とする

1. $f(c\mathbf{u}) = cf(\mathbf{u})$
2. $f(\mathbf{u} + \mathbf{v}) = f(\mathbf{u}) + f(\mathbf{v})$

つまり、スカラー倍や和などの演算をしてから f に入れるのと、先に f に入れてから演算をするのは同じ、ということ

関数を使うタイミングと演算をするタイミングを入れ替えられるので、計算がとても楽になる

先ほどの双線形性は、引数が 2 つの場合なので「双」がつく

* * *

これらを満たせばすべて内積なので、今後、内積を使った議論が出てきた場合には、自分好みの内積を定義して当てはめることができる

* * *

ブラケット記号は「閉じた」形

縦方向に数が並んだベクトルに対応する記号として $|\mathbf{a}_1\rangle$ を導入する

これを **ケットベクトル** と呼ぶ

本書では単に **ケット** と呼ぶこともある

たとえば以下のようなもの

$$|\mathbf{a}_1\rangle = \begin{bmatrix} 0 \\ 1 \\ 7 \end{bmatrix}, \quad |\mathbf{a}_2\rangle = \begin{bmatrix} 1 \\ 0 \\ 4 \end{bmatrix}$$

そしてこれらを横倒しに転置したものが **ブラベクトル**

$$\langle \mathbf{a}_1| = \begin{bmatrix} 0 & 1 & 7 \end{bmatrix}, \quad \langle \mathbf{a}_2| = \begin{bmatrix} 1 & 0 & 4 \end{bmatrix}$$

本書では単に **ブラ** と呼ぶこともある

英語で括弧のことをブラケット (bracket) と言う
左側に来る $\langle \mathbf{a}_1 |$ などがブラ (bra)、右側に来る $|\mathbf{a}_2\rangle$
がケット (ket)、つなぎとしてアルファベットの c
を追加してあげれば、「bracket」の完成

この表記だと括弧が閉じるので、ブラベクトルと
ケットベクトルがセットになることもわかりや
すい

内積はスカラー、つまり単なる数を与えるので、
 $\langle \mathbf{a}_1 | \mathbf{a}_2 \rangle$ が出てきたらスカラーとして扱える

今は具体的なベクトルを考えたが、記法を変えた
のもう少し抽象的なものとして捉えることがで
きる

「無限個の数字がならんだベクトル」を扱うとき
に、この表記が便利

大きさ、距離、さらなる解釈

自分自身の大きさは自分自身との内積

ほかのベクトルとの関係性を見る前に、自分自身
との関係性を見てみよう

つまりベクトルの大きさである

ベクトルの大きさは、要素ごとの 2 乗を計算し、和
をとって、その平方根をとったもの

D 次元ベクトル \mathbf{a}_1 の場合には、

$$\|\mathbf{a}_1\| = \sqrt{a_{1,1}^2 + a_{1,2}^2 + \cdots + a_{1,D}^2} = \sqrt{\sum_{d=1}^D a_{1,d}^2}$$

と書ける

ここで、 $\|\mathbf{a}_1\|$ の記号は **ノルム** と呼ばれる

わざわざ新しい言葉を導入したのは、今後を見す
えて概念を広く捉えるため

また、上式の形のノルムを特に $\|\mathbf{a}_1\|_2$ と書くことも
ある

要素ごとの 2 乗を考えているので右下添字として
2 をつけた、と捉えられる

ちなみに、上式を次のように書き換えられる

$$\|\mathbf{a}_1\| = \sqrt{\langle \mathbf{a}_1 | \mathbf{a}_1 \rangle}$$

内積を使って、ノルムを定義できるということに
なる

* * *

ノルムの定義も一つではない

以下の 3 つの性質を満たすものはすべてノルム

1. $\|\mathbf{u}\| \geq 0$ であり、また $\|\mathbf{u}\| = 0 \Leftrightarrow \mathbf{u} = \mathbf{0}$
2. $c \in \mathbb{R}$ に対して $\|c\mathbf{u}\| = |c|\|\mathbf{u}\|$ ($|c|$ は通常の絶
対値)
3. $\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|$

1 番目は、「大きさがゼロのベクトルは、ゼロベク
トル」であることを意味している

* * *

互いがどれだけ離れているかを測る

ここで導入する関係性は **距離**

距離としては、離れているものほど大きな値を、近
いほど小さな値を返すような関数を考えればよい
また、負の距離というのは不自然なので、ゼロ以
上の値を返してほしい

イメージをもつために矢印で考えると、ベクトル
 $\mathbf{c} = \mathbf{a}_1 - \mathbf{a}_2$ の大きさが、まさに距離としての性質
を備えている

$$d(\mathbf{a}_1, \mathbf{a}_2) = \|\mathbf{a}_1 - \mathbf{a}_2\|$$

このノルムで定義された関数 $d(\cdot, \cdot) : V \times V \rightarrow \mathbb{R}$
が、2つのベクトルの距離を与える
もし自分自身との距離を考えると、距離がゼロになることもわかる
ノルムの性質から負の値を返さないこともわかる

* * *

距離もやはり…

以下の性質を満たすような集合 V 上の関数 $d :$
 $V \times V \rightarrow \mathbb{R}$ はすべて距離である
なお、 $u, v, w \in V$ とする

1. $d(u, v) \geq 0$ 、また、 $u = v$ ならば $d(u, v) = 0$ (距離はゼロ以上、また、同じベクトルであれば距離はゼロ)
2. $d(u, v) = d(v, u)$ (対称性があり、どちらから測っても距離は同じ)
3. $d(u, v) \leq d(u, w) + d(w, v)$ (三角不等式：三角形の2辺の長さを足すと、もう1辺の長さと等しい、もしくは大きくなる)