

データサイエンス概論I & II データサイエンス総論I & II

データ間の距離と類似度

九州大学 数理・データサイエンス教育研究センター

「データ間の距離」の基本

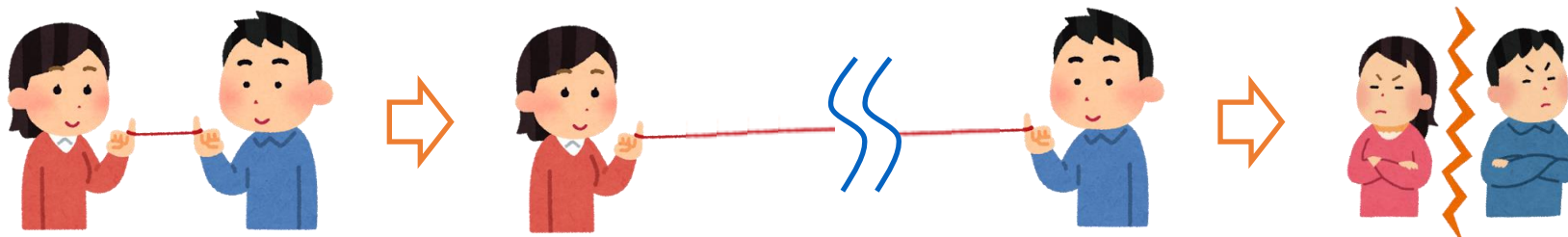


「A君って、タレントの〇〇に似てるよね？」
「えー、全然似てないよ。むしろ△△でしょう。」
「いや、自分では□□似だと思ってるんだけど。」

データ解析における「距離」とは？

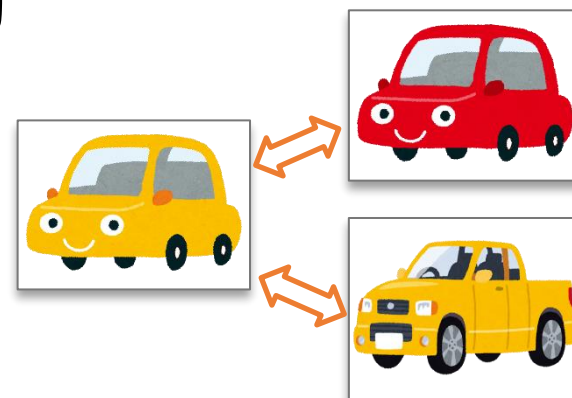
- 日常会話における「距離」

- A地点とB地点がどれくらい離れているか？（単位：mとかkmとか）
- Aさんの気持ちとBさんの気持ちがどれくらい離れているか？



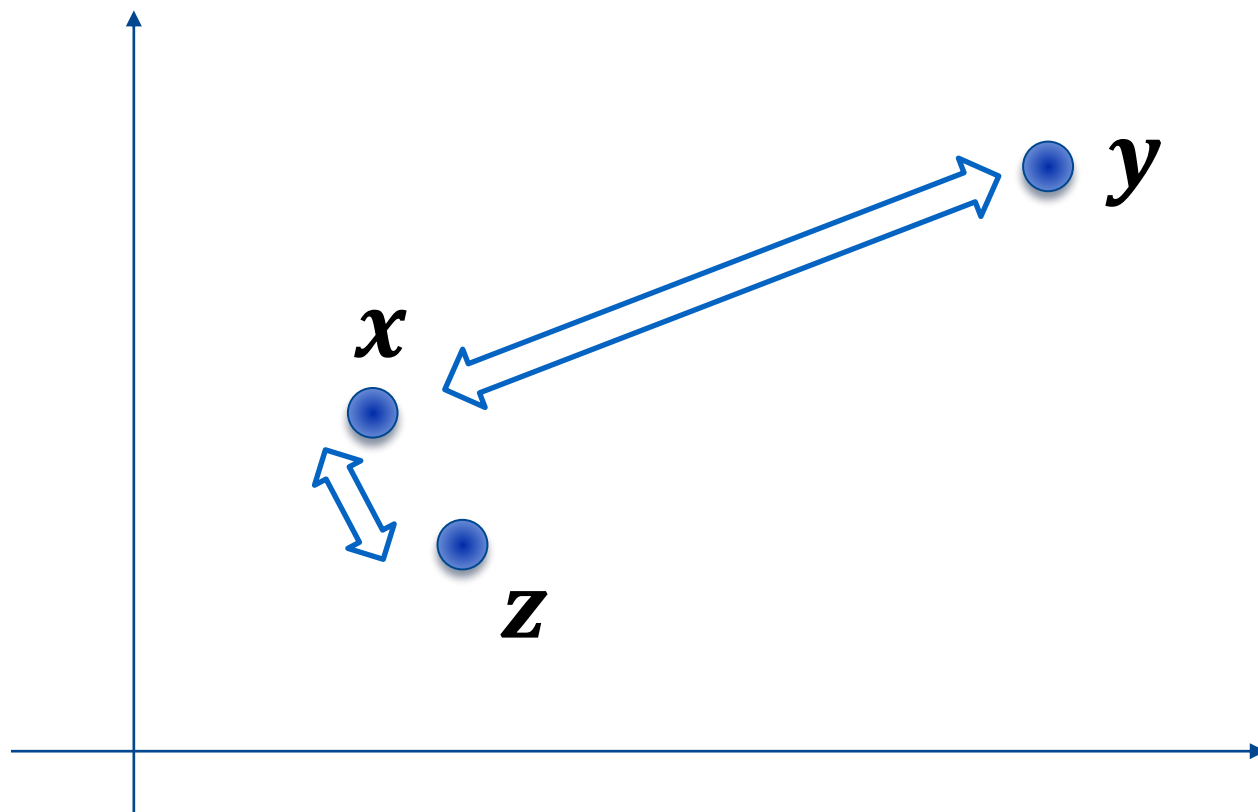
- データ解析における「距離」はもっと一般的

- 要するにデータ間の差異（似てない具合）
- 距離が小さい2データは「似ている」
- 単位がある場合もない場合も



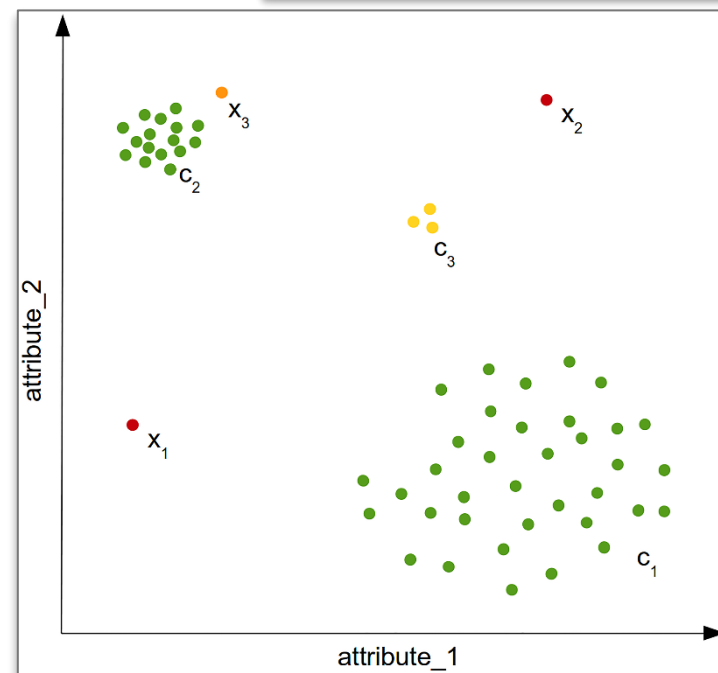
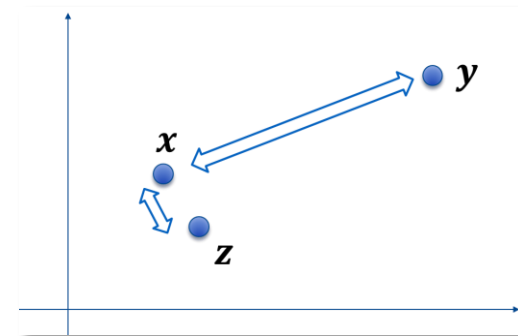
普通に考える「データ間の距離」： 2データがどれくらい違うか？ (=離れているか？)

- x にとって, y は結構違っていて, z は似ている



距離がわかると何に使えるか？ 実は超便利！（1/2）

- データ間の比較が定量的にできる
 - 「 x と y は全然違う/結構似ている」「 x と y は28ぐらい違う」
 - 「 x にとっては、 y よりも z のほうが似ている」
- データ集合のグルーピングができる
 - 「近く」のデータどうしてグループを作る
 - 「クラスタリング」と呼ばれる
- データの異常度が測れる
 - 「近く」にデータがたくさんあれば正常、一つもなければ異常



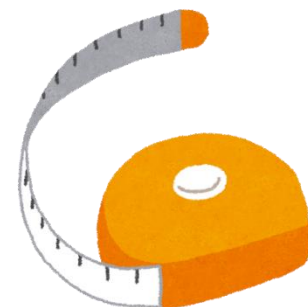
[Goldstein, Uchida, PLoS ONE, 2016]

距離がわかると何に使えるか？ 実は超便利！（2/2）

- データの「認識」ができる
 - 登録されている画像データ中で、画像 x に最も似ているものは「リンゴ」だった
→ 「画像 x はリンゴ」と判断
- 近似精度が測れる
 - データ x を y で近似(代用)した時の誤差 → x と y の距離に等しい
- ... and more!

「距離」の話を通して学んで頂きたいこと

- 距離は「データ解析の基本」である！
- **距離は 1 種類ではない！**
- 距離が変われば，データ解析結果は「まるっきり」変わる
- 解析問題の性質に合致した「距離」を選ぶ必要がある
 - 様々な距離の原理，メリット・デメリットも理解しておこう

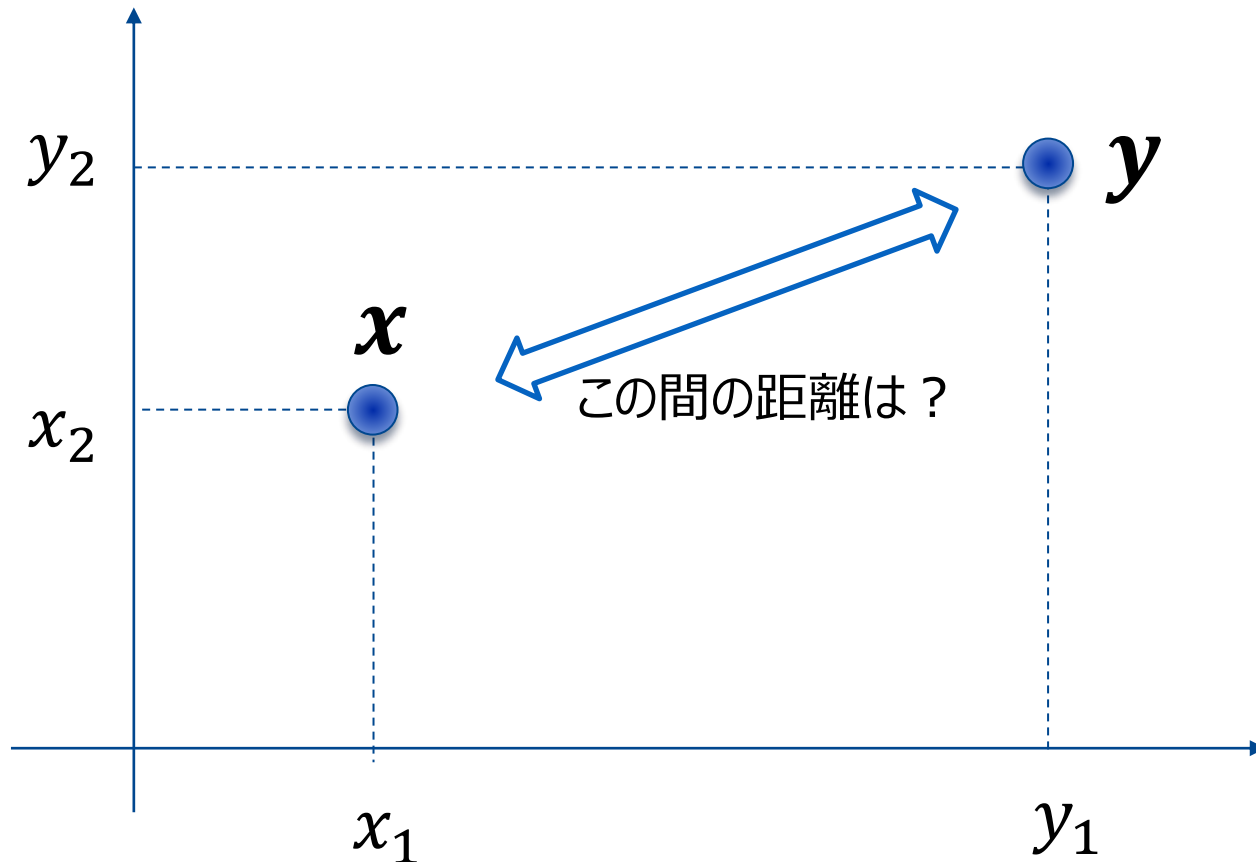


どんな方法も万能ではない！
メリット・デメリットを見極めて、
適切な方法を選択すること！



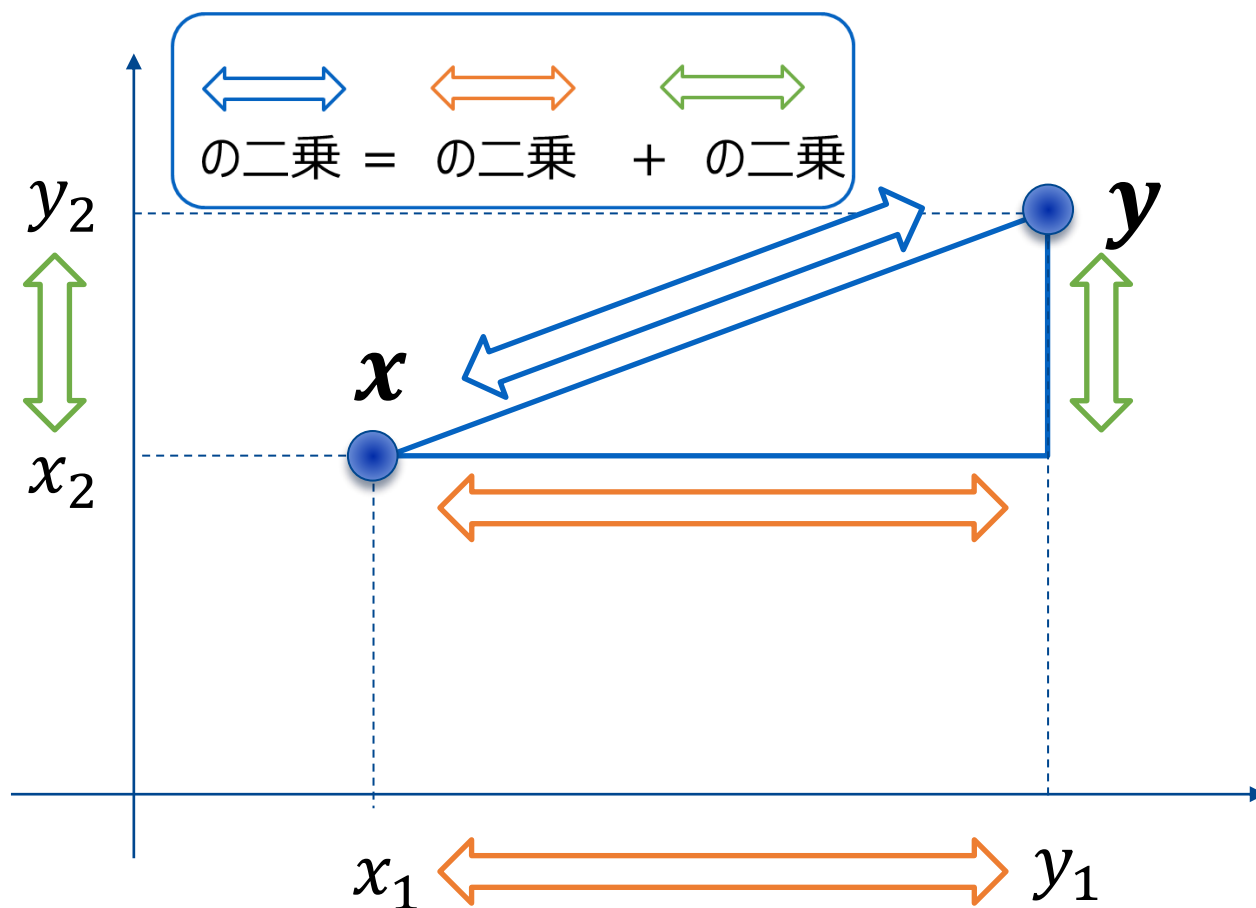
最も代表的な距離：ユークリッド距離 (1)

- 地図上の2点 $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$, $y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$



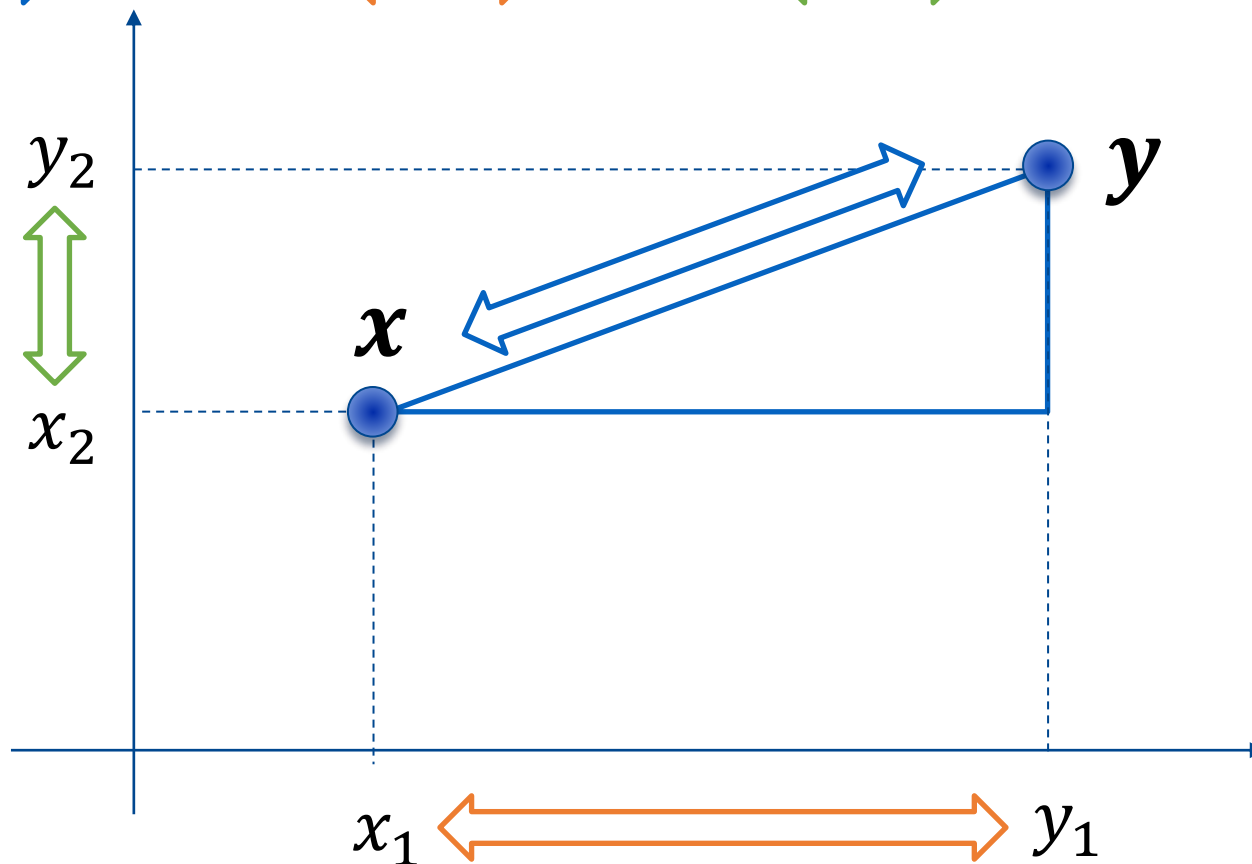
最も代表的な距離：ユークリッド距離 (2)

- ご存じ「三平方の定理」(ピタゴラスの定理)



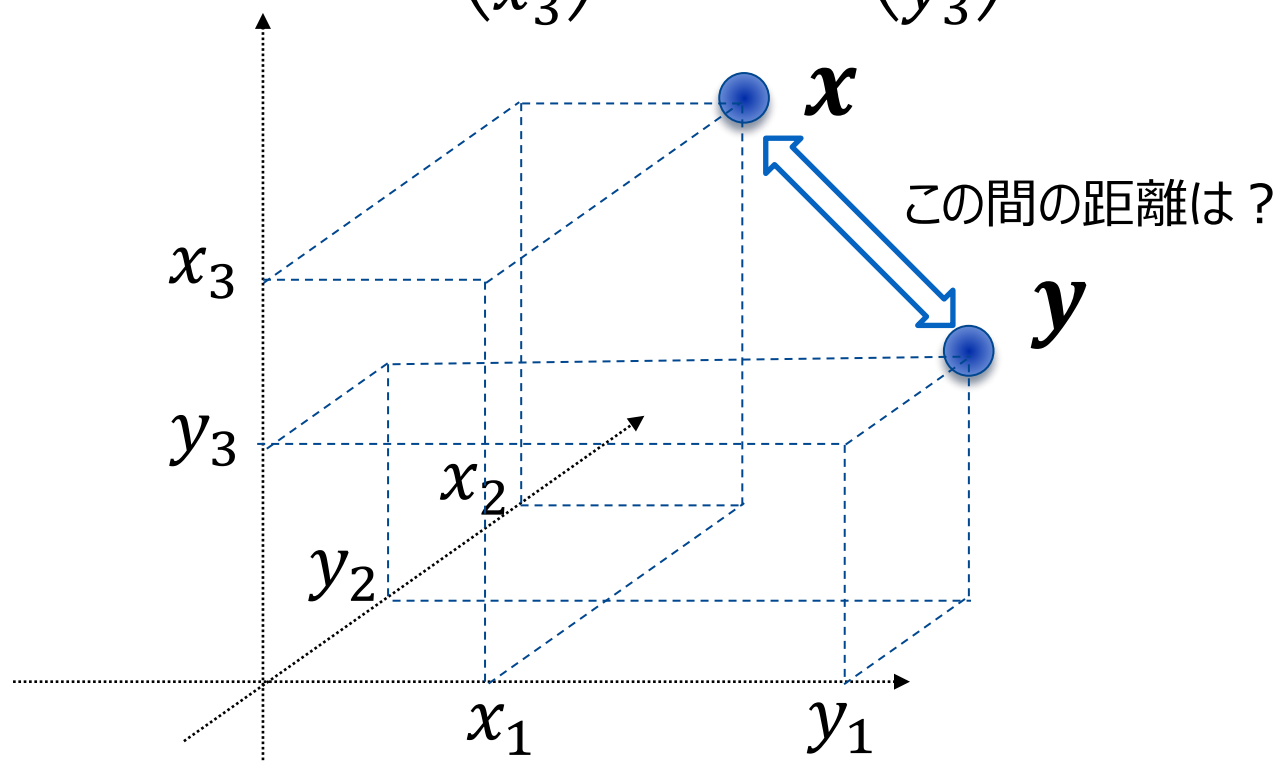
最も代表的な距離：ユークリッド距離 (3)

- x と y の距離の二乗 $= (x_1 - y_1)^2 + (x_2 - y_2)^2$



最も代表的な距離：ユークリッド距離 (4)

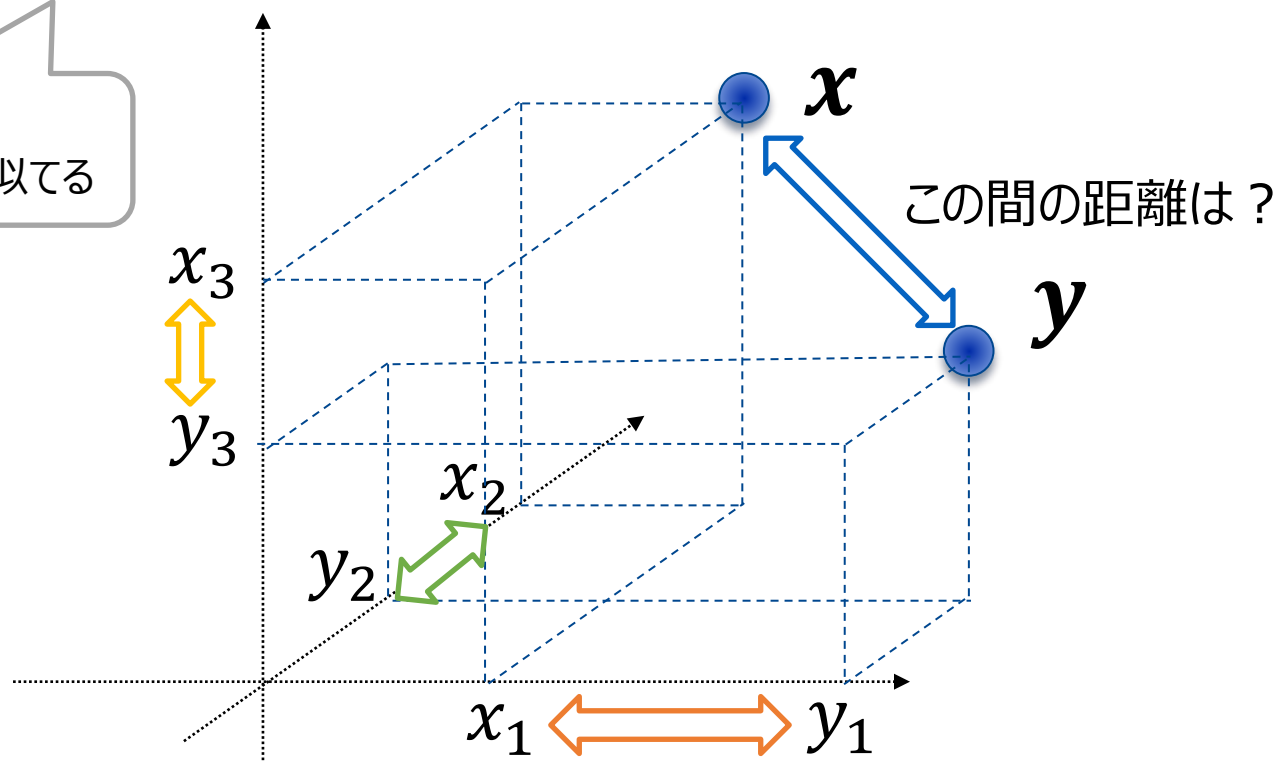
- 3次元だとどうなる？
 $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$



最も代表的な距離：ユークリッド距離 (5)

- \mathbf{x} と \mathbf{y} の距離の二乗 $= (x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2$

なんかやっぱり
ピタゴラスの定理に似てる



最も代表的な距離：ユークリッド距離 (6)

● 2次元の場合

$$\begin{array}{c} \mathbf{x} \quad \mathbf{y} \\ \downarrow \quad \downarrow \end{array}$$

$$\mathbf{x} \text{ と } \mathbf{y} \text{ の距離の二乗} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_1 - y_1 \\ x_2 - y_2 \end{pmatrix}$$

\nearrow 二乗
 $+$
 \searrow 二乗

● 3次元の場合

$$\begin{array}{c} \mathbf{x} \quad \mathbf{y} \\ \downarrow \quad \downarrow \end{array}$$

$$\mathbf{x} \text{ と } \mathbf{y} \text{ の距離の二乗} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} - \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} x_1 - y_1 \\ x_2 - y_2 \\ x_3 - y_3 \end{pmatrix}$$

\nearrow 二乗
 $+$
 \rightarrow 二乗
 $+$
 \searrow 二乗

最も代表的な距離：ユークリッド距離 (7)

- d 次元の場合

$$\begin{array}{c} \text{\textit{x}と\textit{y}の距離の} \\ \text{二乗} \end{array} = \begin{array}{c} \text{\textit{x}} \\ \downarrow \\ \left(\begin{array}{c} x_1 \\ \vdots \\ x_d \end{array} \right) \end{array} - \begin{array}{c} \text{\textit{y}} \\ \downarrow \\ \left(\begin{array}{c} y_1 \\ \vdots \\ y_d \end{array} \right) \end{array} = \begin{array}{c} \text{\textit{x}} - \text{\textit{y}} \\ \downarrow \\ \left(\begin{array}{c} x_1 - y_1 \\ \vdots \\ x_d - y_d \end{array} \right) \end{array} \begin{array}{c} \nearrow \text{二乗} \\ + \\ \vdots \\ + \\ \searrow \text{二乗} \end{array}$$

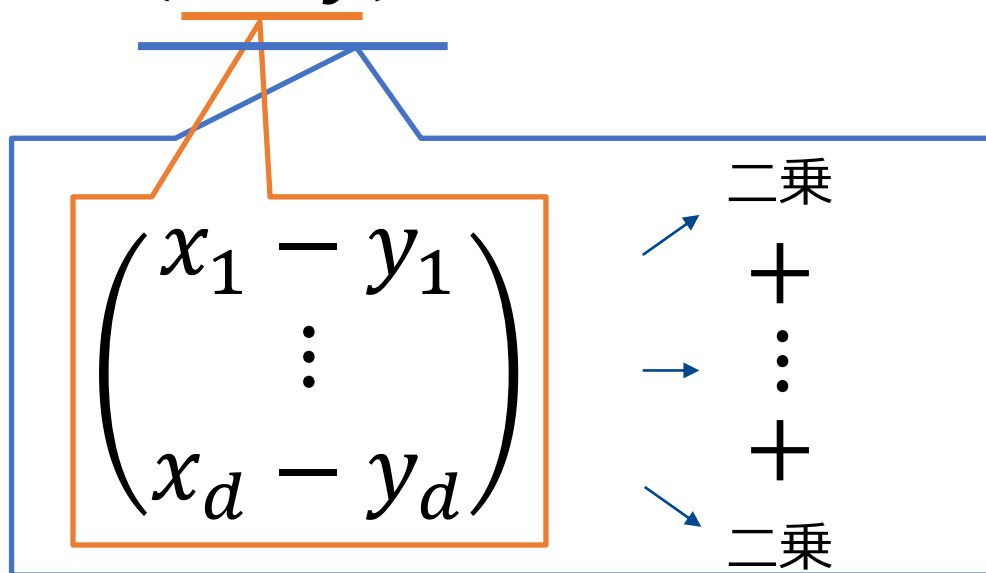
というわけで、何次元ベクトルでも距離は計算可能

もちろん1次元ベクトル(数値)間の距離も計算可能

最も代表的な距離：ユークリッド距離 (8)

- 簡略表現法

$$\mathbf{x} \text{ と } \mathbf{y} \text{ の距離の二乗} = (\mathbf{x} - \mathbf{y})^2$$

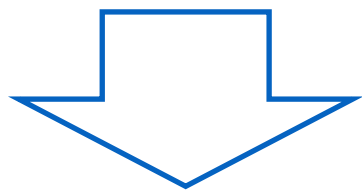

$$\begin{pmatrix} x_1 - y_1 \\ \vdots \\ x_d - y_d \end{pmatrix}$$

二乗
+
:
+
二乗

最も代表的な距離：ユークリッド距離 (9)

- これではようやくユークリッド距離

$$x \text{ と } y \text{ の距離の二乗} = (x - y)^2$$



なんだこの記号？
(次スライド)

$$x \text{ と } y \text{ の距離} = \sqrt{(x - y)^2} = \|x - y\|$$

d 次元ベクトル x と y の間のユークリッド距離

参考：なんだこの二重絶対値 $\|\cdot\|$ は？

- $\|x\|$ はベクトル x の長さを表すんです

- ベクトル x の「ノルム」とも言います！

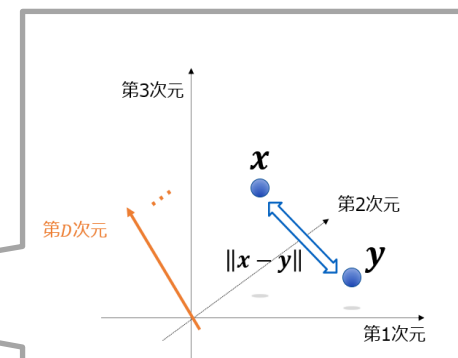
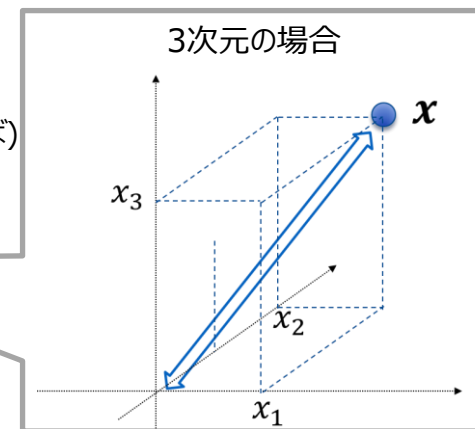
- ベクトル x の長さは(実はノルムにもいろいろあるんですが、そんなことまずは気にせずに考えれば)

$$\|x\| = \sqrt{x_1^2 + \cdots + x_d^2}$$

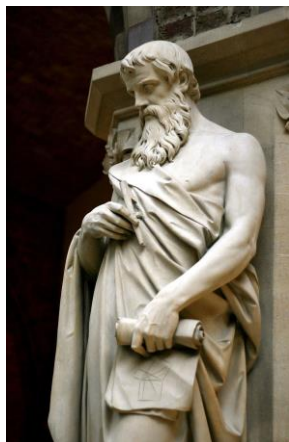
となります

- だから $\|x - y\|$ は x と y の差の長さ、すなわち距離ってわけです

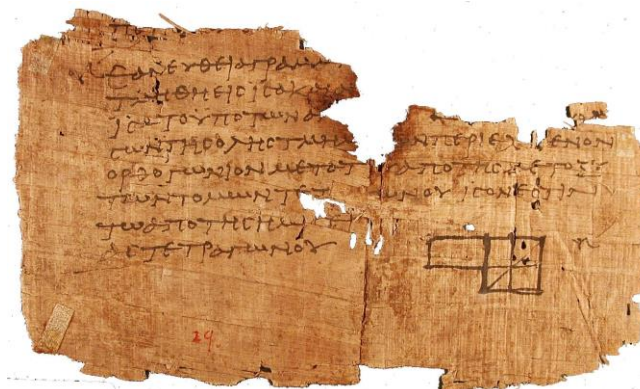
$$\|x - y\| = \sqrt{(x - y)^2}$$



参考：ユークリッド＝幾何学の父



@エジプト
BC330～275年頃？

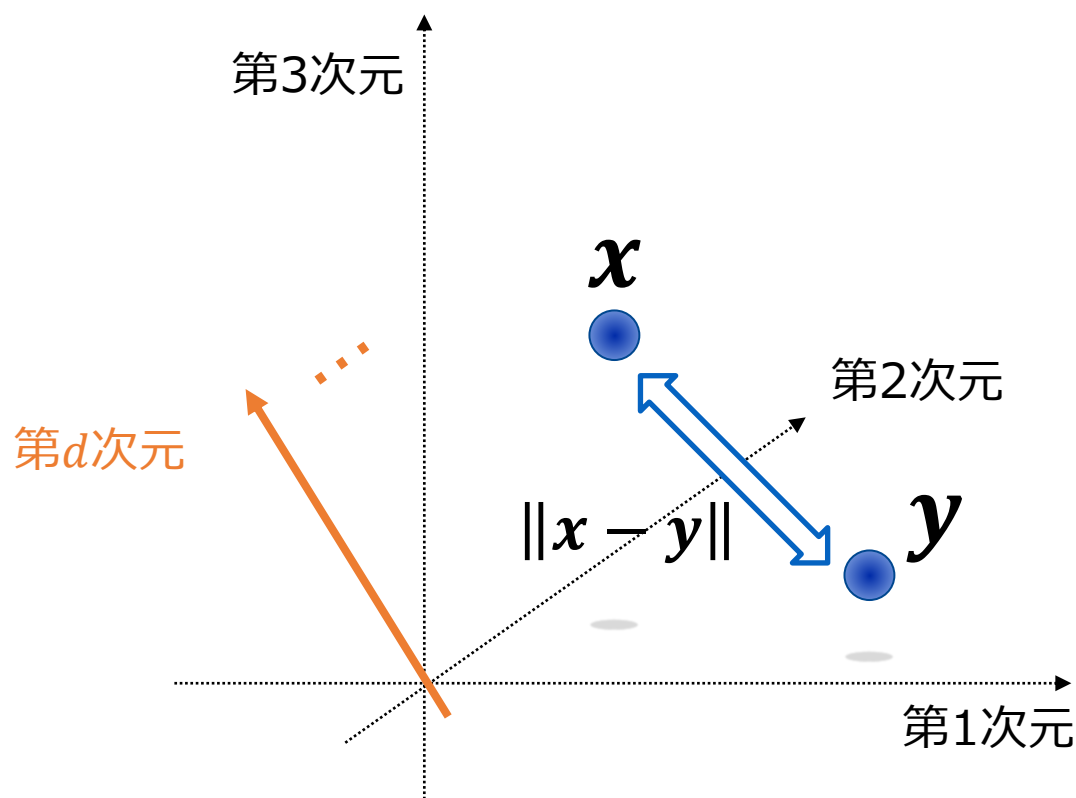


ユークリッド原論

- ユークリッド原論にある5つの公準(≒公理)
 - 第1公準 : 点と点を直線で結ぶ事ができる
 - 第2公準 : 線分は両側に延長して直線にできる
 - 第3公準 : 1点を中心にして任意の半径の円を描く事ができる
 - 第4公準 : 全ての直角は等しい (角度である)
 - 第5公準 : 1つの直線が2つの直線に交わり、同じ側の内角の和が2つの直角より小さいならば、この2つの直線は限りなく延長されると、2つの直角より小さい角のある側において交わる (≒平行線でない2直線は1点で交わる)

最も代表的な距離：ユークリッド距離 (10)

- 図示するとやっぱりこんな感じ



練習：2つのデータ間のユークリッド距離を求めよう

$x = (3), y = (6)$ のとき $\|x - y\|$ は？

$x = \begin{pmatrix} 3 \\ 5 \end{pmatrix}, y = \begin{pmatrix} 6 \\ 1 \end{pmatrix}$ のとき $\|x - y\|$ は？

$x = \begin{pmatrix} 3 \\ 5 \\ 2 \end{pmatrix}, y = \begin{pmatrix} 6 \\ 1 \\ 2 \end{pmatrix}$ のとき $\|x - y\|$ は？

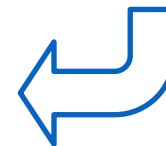
これで画像間の距離(似てる具合)も測れます

どちらも1000x1000画素の画像



100万次元ベクトル x

100万次元ベクトル y



画像間距離
 $\|x - y\|$

お、これで画像
認識AIができそう



尺度の異なる数値が組み合わされた
データに関する距離の計算は要注意

実データ間の距離を測る際の留意点(1/3)

(身長, 体重)データ間のユークリッド距離

$\begin{pmatrix} \text{体重}(kg) \\ \text{身長}(cm) \end{pmatrix}$ とする。このとき

$x = \begin{pmatrix} 60 \\ 150 \end{pmatrix}$ と似ているのは,

$y = \begin{pmatrix} 30 \\ 150 \end{pmatrix}$ と $z = \begin{pmatrix} 60 \\ 153.1 \end{pmatrix}$ のどっち？

それはやはり**Z**さんのほうが似てる(30kg差は大きい)

(身長, 体重)データ間のユークリッド距離

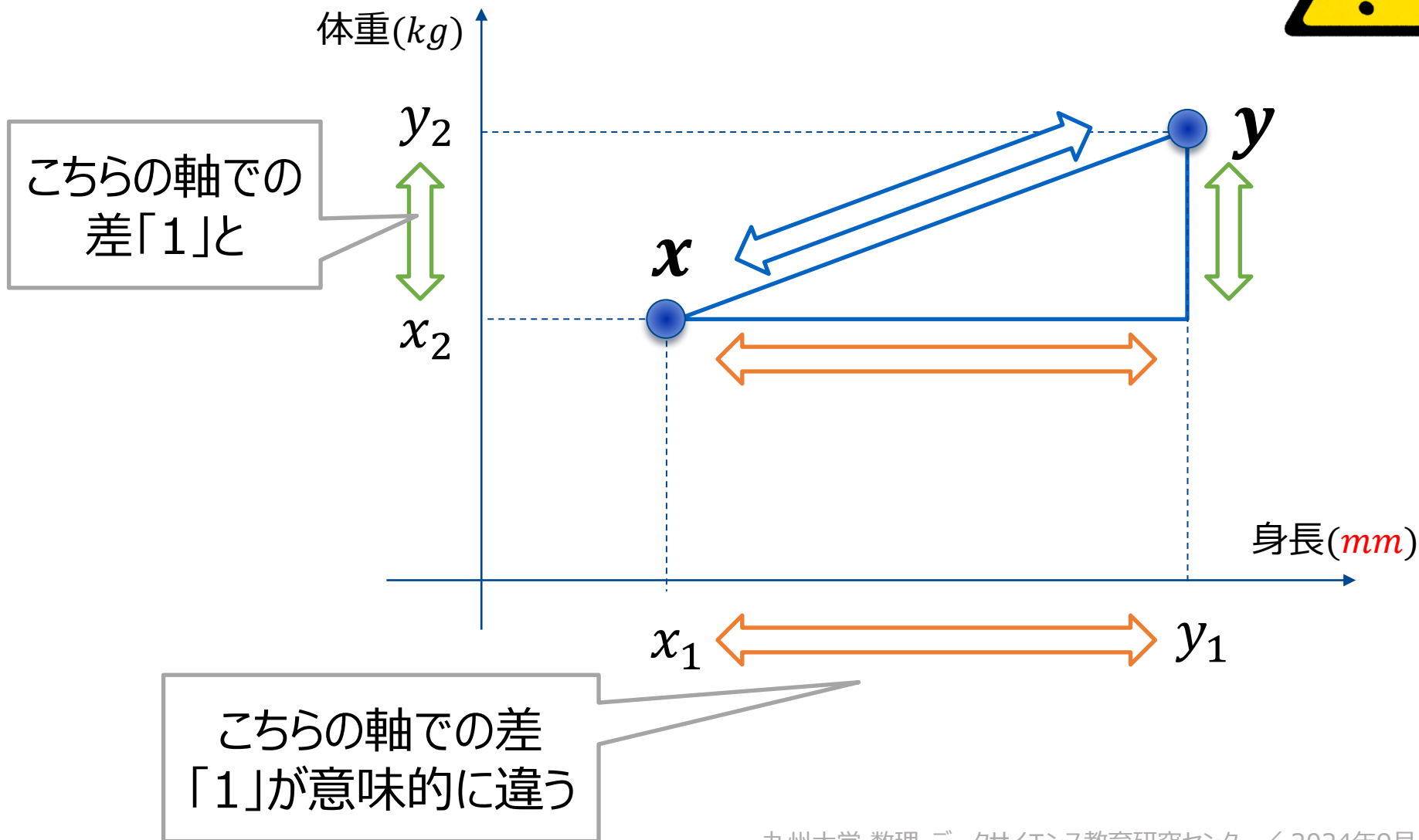
$\begin{pmatrix} \text{体重}(kg) \\ \text{身長}(mm) \end{pmatrix}$ とする. このとき

$x = \begin{pmatrix} 60 \\ 1500 \end{pmatrix}$ と似ているのは,

$y = \begin{pmatrix} 30 \\ 1500 \end{pmatrix}$ と $z = \begin{pmatrix} 60 \\ 1531 \end{pmatrix}$ のどっち?

単位が変わるだけで思ってもないことに..

というわけで、**尺度の異なる数値が組み合わせられたデータ**に関する距離の計算は要注意！



さっきの画像間距離，なぜ大丈夫なのか？



100万次元ベクトル x

100万個の
どの要素も全部
「画素値」



100万次元ベクトル y

100万個の
どの要素も全部
「画素値」

画像間距離
 $\|x - y\|$



安心してそのまま距離計算可能

この辺、結構悩ましい：
(数学点数, 古文点数)データ間のユークリッド距離

(数学の点数(100点満点)
古文の点数(50点満点))
について普通に距離計算していいのか？

解釈1

同じどちらも「点数」
なので、そのまま
計算してOK



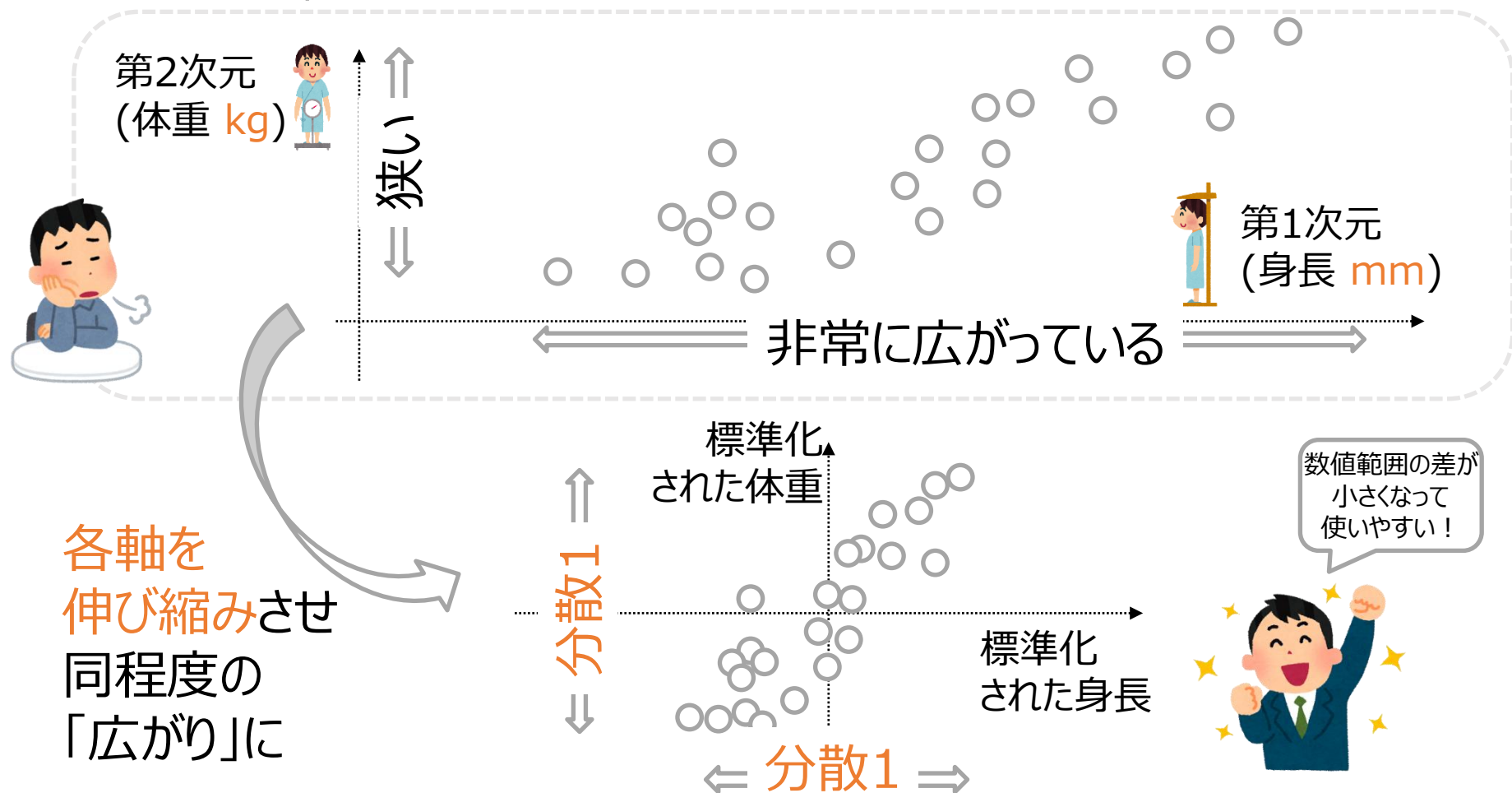
解釈2

古文を2倍にして
どちらも100点満点
にしてから
距離計算すべき

どちらかが「正しい」という証明はできない
(ケースバイケースで使い分けするしかない)

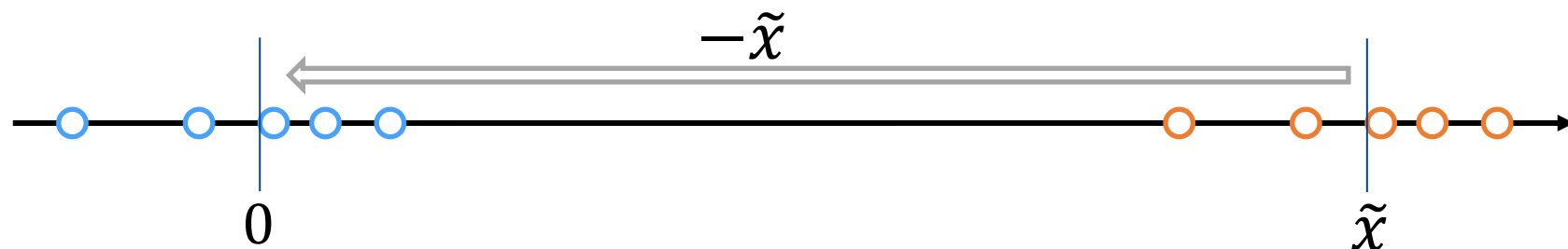
物理的に意味の異なる数値を扱う場合に関して リーズナブルな方法：標準化

- 平均をゼロ，分散を1に「標準化」する！

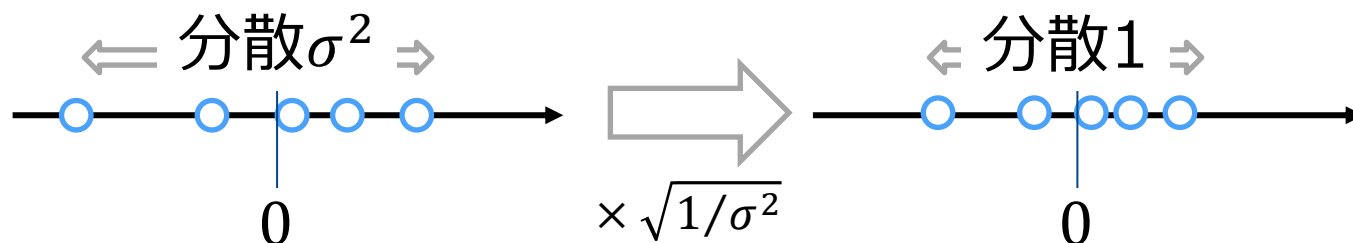


「平均ゼロ，分散を1」にするには？

- まず平均をゼロに
 - 全部の値から現在の平均値を引けば，平均はゼロになる



- 次に分散を1に！
 - 前に「値を x_i から αx_i にすると，分散は $\alpha^2 \sigma^2$ になる」と言いました
 - $\alpha^2 \sigma^2 = 1$ にしたいということは， x_i を $\alpha = \sqrt{1/\sigma^2}$ 倍すればOK!





「見せかけの数値」の距離を測るのは危険

実データ間の距離を測る際の留意点(2/3)

ユークリッド距離が測れる = 差が定義できる

- アンケート結果やランキング(順位)を含むデータについて、ユークリッド距離を測るのは「本当は」間違い

	名称	可能な演算	主な代表値	主な事例
距離が測れる	量的データ	+ - × ÷	各種平均	質量, 長さ, 年齢, 時間, 金額
	間隔データ	+ -	算術平均	温度(摂氏), 知能指数
距離は測れない	質的データ	> =	中央値, 最頻値	満足度, 選好度, 硬度
	カテゴリデータ	度数カウント	最頻値	電話番号, 性別, 血液型

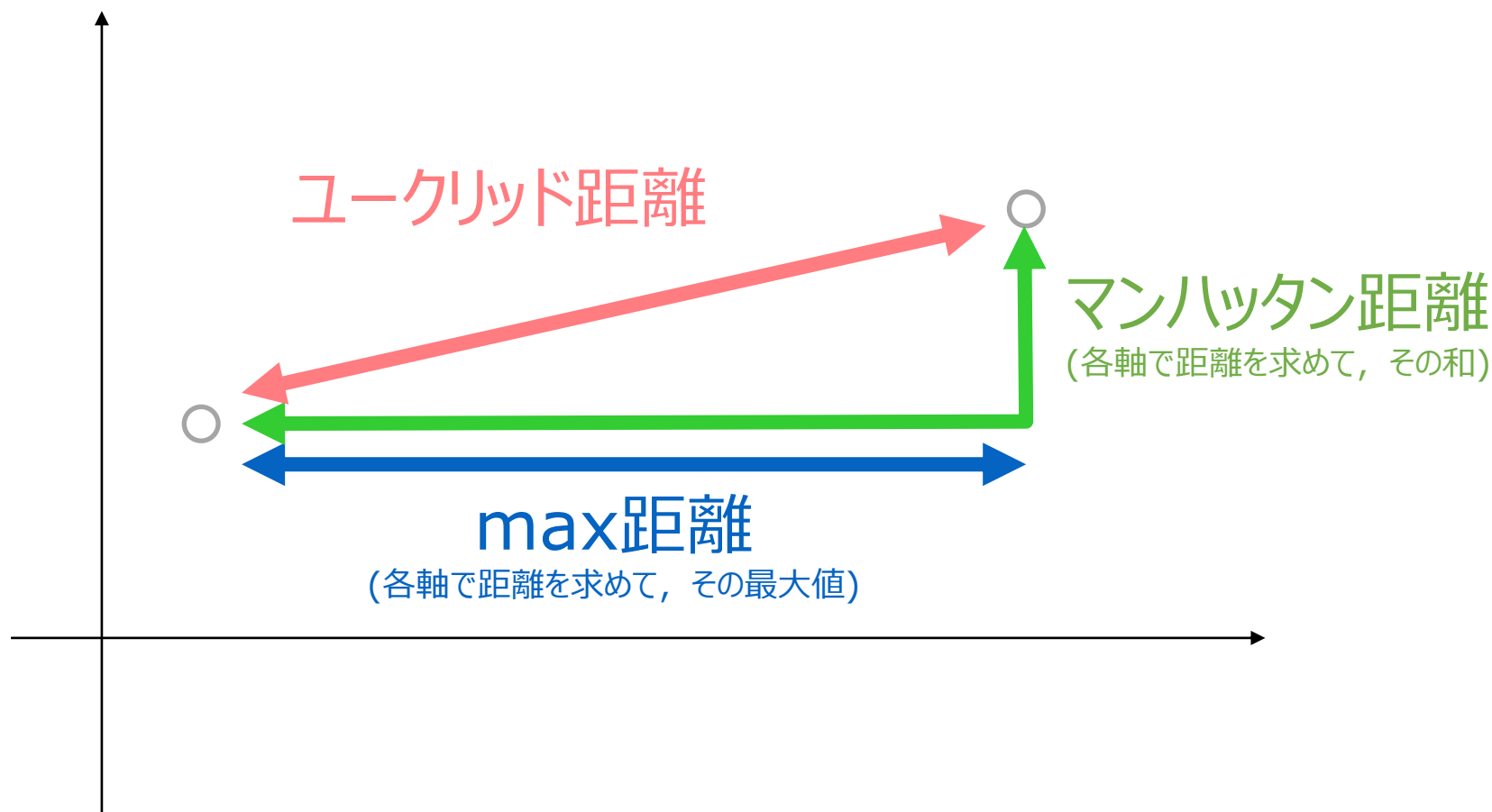
※後述の離散距離なら、カテゴリデータの距離でも測れる



ユークリッド距離だけじゃない

実データ間の距離を測る際の留意点(3/3)

ユークリッド距離だけじゃない： 様々な距離



マンハッタン距離？

- 斜めには行けない街での距離

- 平安京距離
- 平城京距離
- 札幌距離

と呼んでもよいはず

- 「市街地距離」と呼ばれることも



max距離をいつ使う？

- 次の d 次元データ間の距離を考えてみましょう

$$\begin{array}{c}
 \begin{array}{|c}
 \hline
 1 \\
 \vdots \\
 1 \\
 \color{red}{1} \\
 1 \\
 \vdots \\
 1 \\
 \hline
 \end{array}
 \qquad
 \begin{array}{|c}
 \hline
 1 \\
 \vdots \\
 1 \\
 \color{red}{10} \\
 1 \\
 \vdots \\
 1 \\
 \hline
 \end{array}
 \end{array}$$

d 個

- 「1要素でも大きく違ったら、それは結構違うのだ」としたい場合に
 - ただし1要素間でのみの評価になるので、全体的な差異は評価できない

式で書くと.... 実は統一的に書ける

$$L_p \text{ 距離} = \left(\sum_{i=1}^d |x_i - y_i|^p \right)^{1/p}$$

- マンハッタン距離 $\rightarrow L_1$ 距離 (上の式において $p = 1$)
- ユークリッド距離 $\rightarrow L_2$ 距離 (上の式において $p = 2$)
- max距離 $\rightarrow L_\infty$ 距離 (上の式において $p = \infty$)



参考： L_∞ がなぜmax?

$$L_\infty = \left(\sum_{i=1}^d |x_i - y_i|^\infty \right)^{1/\infty} \quad \leftarrow \quad d\text{個の絶対値のうち、一番大きいものが支配的}$$

例($d = 3$)

2乗

10乗

1000乗

$ x_1 - y_1 = 10.1$	$\rightarrow 102.01$	$\rightarrow 1.1 \times 10^{10}$	$\rightarrow 2.1 \times 10^{1004}$
$ x_2 - y_2 = 10.2$	$\rightarrow 104.03$	$\rightarrow 1.2 \times 10^{10}$	$\rightarrow 3.9 \times 10^{1008}$
$ x_3 - y_3 = 10.3$	$\rightarrow 106.09$	$\rightarrow 1.3 \times 10^{10}$	$\rightarrow 6.8 \times 10^{1012}$

あまり差はないが...

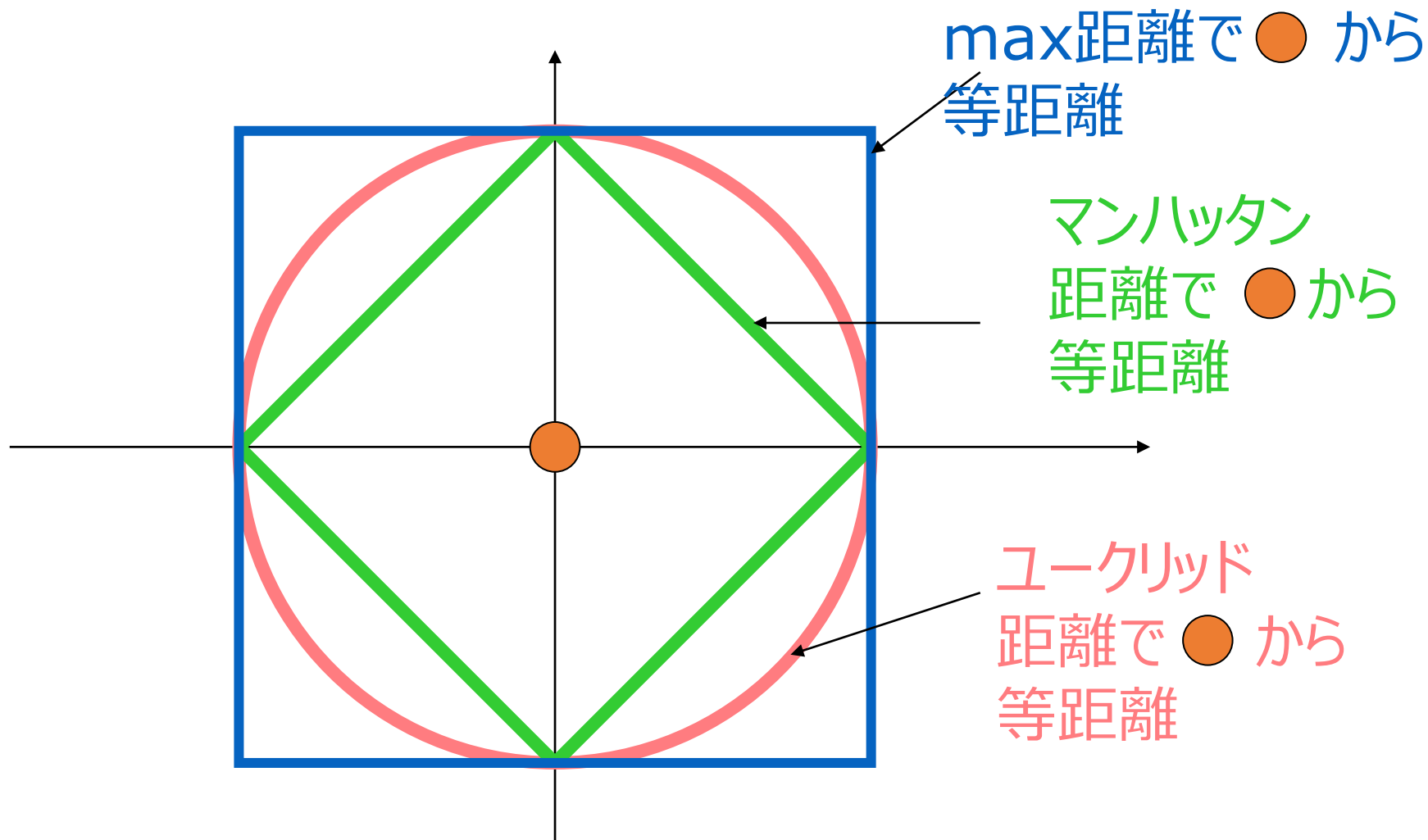
非常に大きな差となった！

最大以外のものは無視できる

よって
$$L_\infty = \left(\sum_{i=1}^d |x_i - y_i|^\infty \right)^{1/\infty} \sim \left(\max_{1 \leq i \leq d} |x_i - y_i|^\infty \right)^{1/\infty} = \max_{1 \leq i \leq d} |x_i - y_i|$$

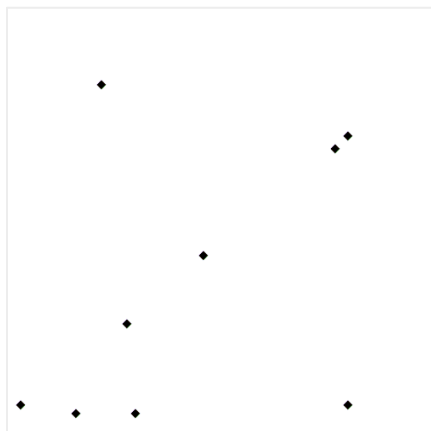


等距離面

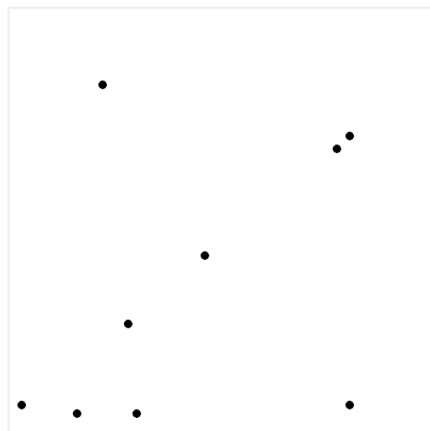


ボロノイ図(縄張り図)を作ると...

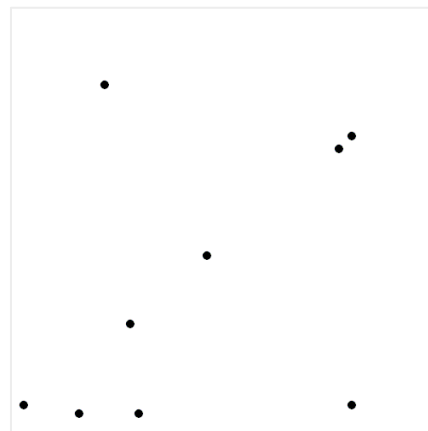
$p = 1$



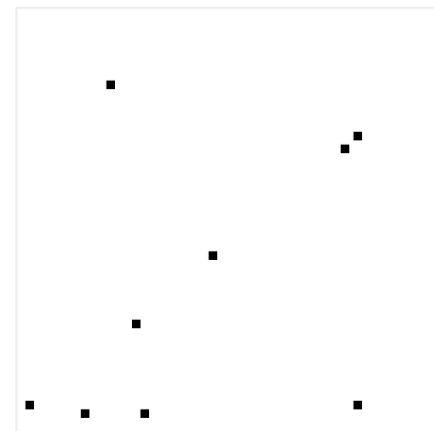
$p = 2$



$p = 3$

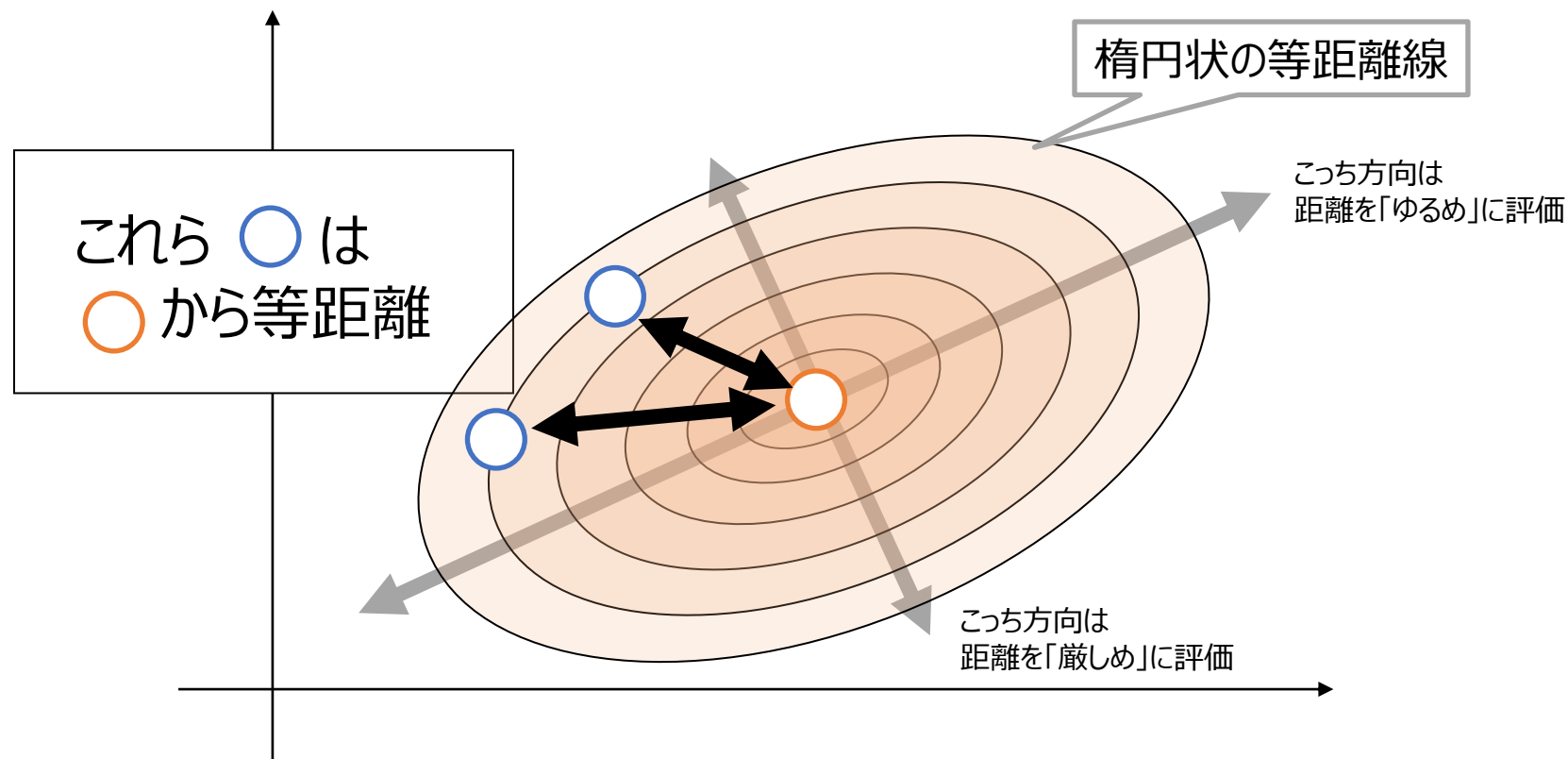


$p = \infty$



by [Jahobr](https://commons.wikimedia.org/wiki/File:Voronoi_growth_minkowski_p1_25.gif) https://commons.wikimedia.org/wiki/File:Voronoi_growth_minkowski_p1_25.gif

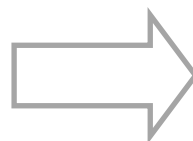
マハラノビス距離 (Mahalanobis distance)



そのうち出てくる「正規分布」と密接に関係

離散距離

- 同じなら0, (少しでも)違うなら 1



距離0



距離1

- どんなデータの距離でも測れます

2番



5番



明らかに意味がない

ユークリッド距離 $\|2 - 5\| = 3$

離散距離 = 1 (違うから)

2つのバスの番号(カテゴリデータ)

ハミング距離 (Hamming distance)

- (長さの同じ)2系列間の距離
- 違う要素の数 = 距離
- 例
 - $100101 \Leftrightarrow 110111 \rightarrow \text{距離}2$
 - “Synchronize” \Leftrightarrow “Simchronise” $\rightarrow \text{距離}3$

編集距離 (edit distance)

- 2系列間の距離。「系列の長さが違って大丈夫」がメリット

- 置換, 挿入, 削除の最小回数
- ハミング距離を一般化
- Levenshtein距離とも

操作回数が最小で済む方法は
自明ではない. 何とかして見つ
ける必要がある!
(操作回数の最小化問題)

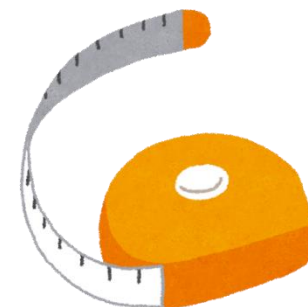
- 例

- "This" \Leftrightarrow "These"
- 置換1回($i \leftrightarrow e$) + 挿入1回(e) \rightarrow 距離 2
- 削除1回(s) + 置換($i \leftrightarrow e$)1回 + 挿入2回(se) \rightarrow 距離 4
- 削除2回(is) + 挿入3回(ese) \rightarrow 距離 5
- 削除4回(This) + 挿入5回(These) \rightarrow 距離 9
-

こんな調子で距離には膨大な種類がある！
そこで大事なことをもう一度！



- 距離は「データ解析の基本」である！
- 距離は1種類ではない！
- 距離が変われば，データ解析結果は「まるっきり」変わる
- 解析問題の性質に合致した「距離」を選ぶ必要がある
 - 逆に言うと，「自分で好きな距離を選べる」ともいえる



データ解析の基本中の基本が
自分のチョイスに任されているなんて…



数学は「だれが解いても同じ
答えが出る」と思っていたのに…

高校までの「ドリル」の世界の話。
数学の本質は「自由」！

そう、この自由さがあるからこそ データ分析をしっかりと学ぶ必要があるんです！

- 「これだけが唯一の方法」というものはない！
 - データ分析に「絶対的『真』」は存在しない
- だからこそ自分で考えないと！
 - 「このデータを、という手法で分析するのが妥当か？」
- だからこそ疑わないと！
 - 「この論文の分析結果は、どのような手法で得られたものか？」
 - 「また、なぜその手法をつかっているのか？」
 - 「都合の良い結果が得られるような、恣意的な手法はつかっていないか？」

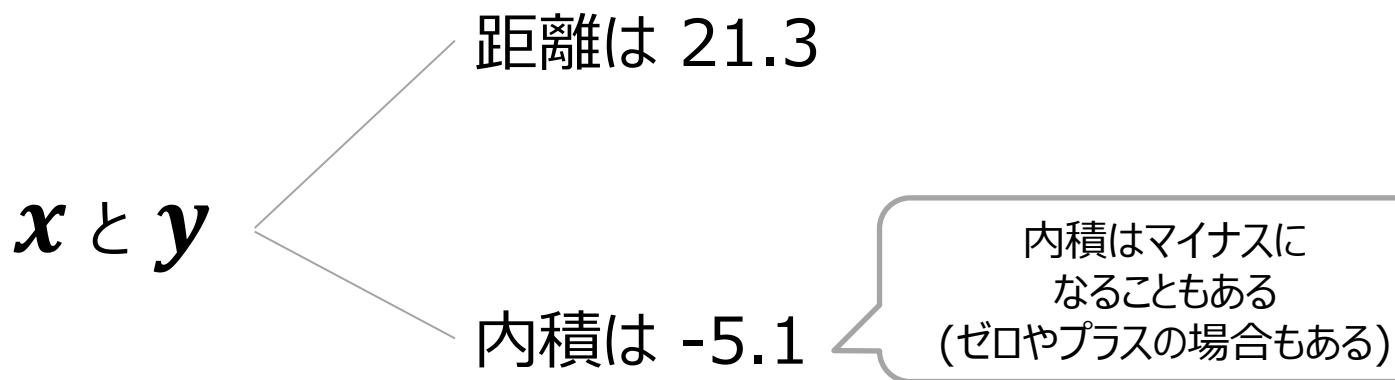


突然ですが…
ベクトル間の内積

高校の時に習った内積. 人生の役に立つことなんてないと思ってました？
データ解析やAIでは内積が超重要 (皆さんの脳内でも常に内積計算が？)

内積？

- 距離と同様，2つのベクトルの関係を，1つの数字で表現
 - どちらが良いとか悪いとかいう話ではない．単に違うモノ



- なんでそんなもんが必要？
 - ベクトル(データ)間の類似度^{類似度}に使える
 - 類似度は「似てる具合」．距離は「似てない具合」．
 - そのうち非常に重要になってくる
 - 主成分分析とかディープラーニング(深層学習)とかはバリバリ

内積の計算法

内積の書き方4種(どれも同じ)

$$\mathbf{x} \cdot \mathbf{y}$$

$$(\mathbf{x}, \mathbf{y})$$

$$\langle \mathbf{x}, \mathbf{y} \rangle$$

$$\mathbf{x}^T \mathbf{y}$$

- 習うより慣れよう。こんな感じ。

$$\mathbf{x} = \begin{pmatrix} 3 \\ 5 \end{pmatrix}, \mathbf{y} = \begin{pmatrix} 6 \\ 1 \end{pmatrix} \text{の内積} \rightarrow \mathbf{x} \cdot \mathbf{y} = 3 \times 6 + 5 \times 1 = 23$$

- 要するに、「要素どうしの積をとって、全部足す」(小学生でも計算できる)

- その原理で、何次元ベクトルでも計算可能

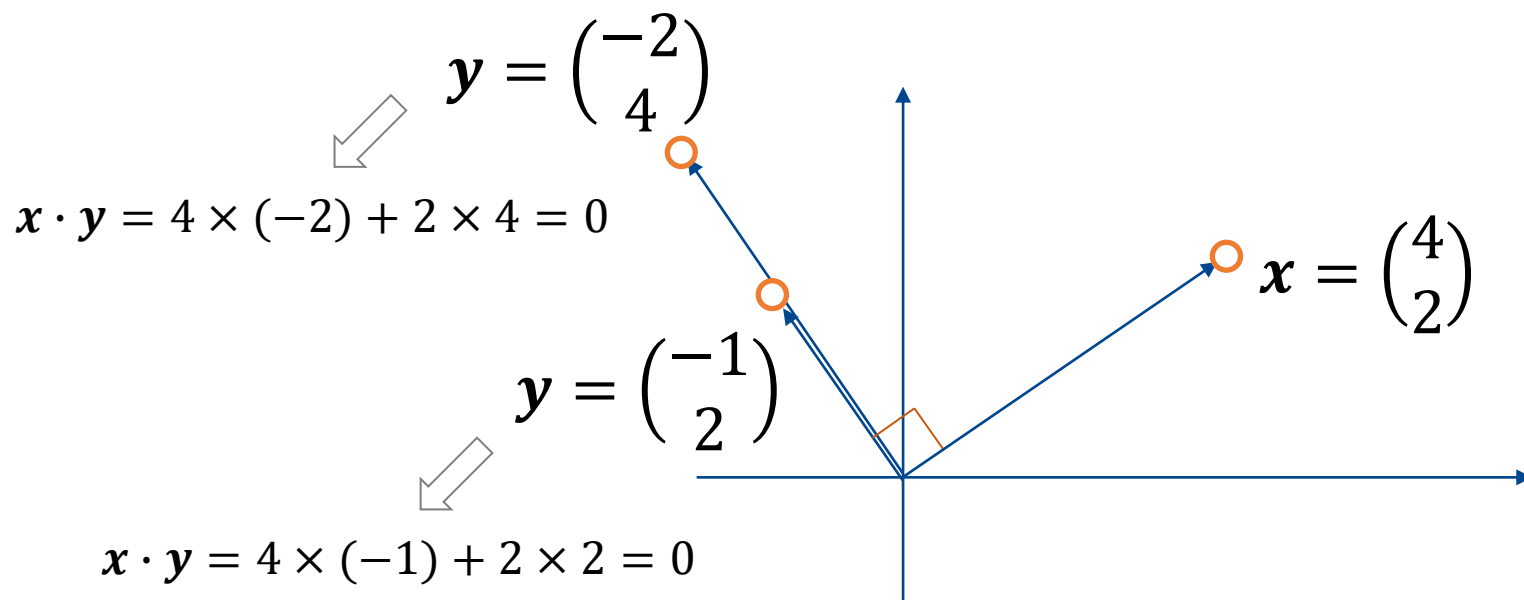
$$\begin{pmatrix} 3 \\ 5 \end{pmatrix} \text{と} \begin{pmatrix} 6 \\ 1 \end{pmatrix} \text{の内積} \Rightarrow \left. \begin{array}{l} \begin{pmatrix} 3 \\ 5 \end{pmatrix} \times \begin{pmatrix} 6 \\ 1 \end{pmatrix} = 18 \\ \phantom{\begin{pmatrix} 3 \\ 5 \end{pmatrix}} \times \phantom{\begin{pmatrix} 6 \\ 1 \end{pmatrix}} = 5 \end{array} \right\} 18 + 5 = 23$$

$$\begin{pmatrix} 3 \\ 5 \\ 2 \end{pmatrix} \text{と} \begin{pmatrix} 6 \\ 1 \\ 2 \end{pmatrix} \text{の内積} \Rightarrow \left. \begin{array}{l} \begin{pmatrix} 3 \\ 5 \\ 2 \end{pmatrix} \times \begin{pmatrix} 6 \\ 1 \\ 2 \end{pmatrix} = 18 \\ \phantom{\begin{pmatrix} 3 \\ 5 \\ 2 \end{pmatrix}} \times \phantom{\begin{pmatrix} 6 \\ 1 \\ 2 \end{pmatrix}} = 5 \\ \phantom{\begin{pmatrix} 3 \\ 5 \\ 2 \end{pmatrix}} \times \phantom{\begin{pmatrix} 6 \\ 1 \\ 2 \end{pmatrix}} = 4 \end{array} \right\} 18 + 5 + 4 = 27$$

※この調子で、4次元でも、100万次元でも可能

内積の性質： ベクトルのなす角が90度のときに，内積はゼロ！

- なんと美しい性質！

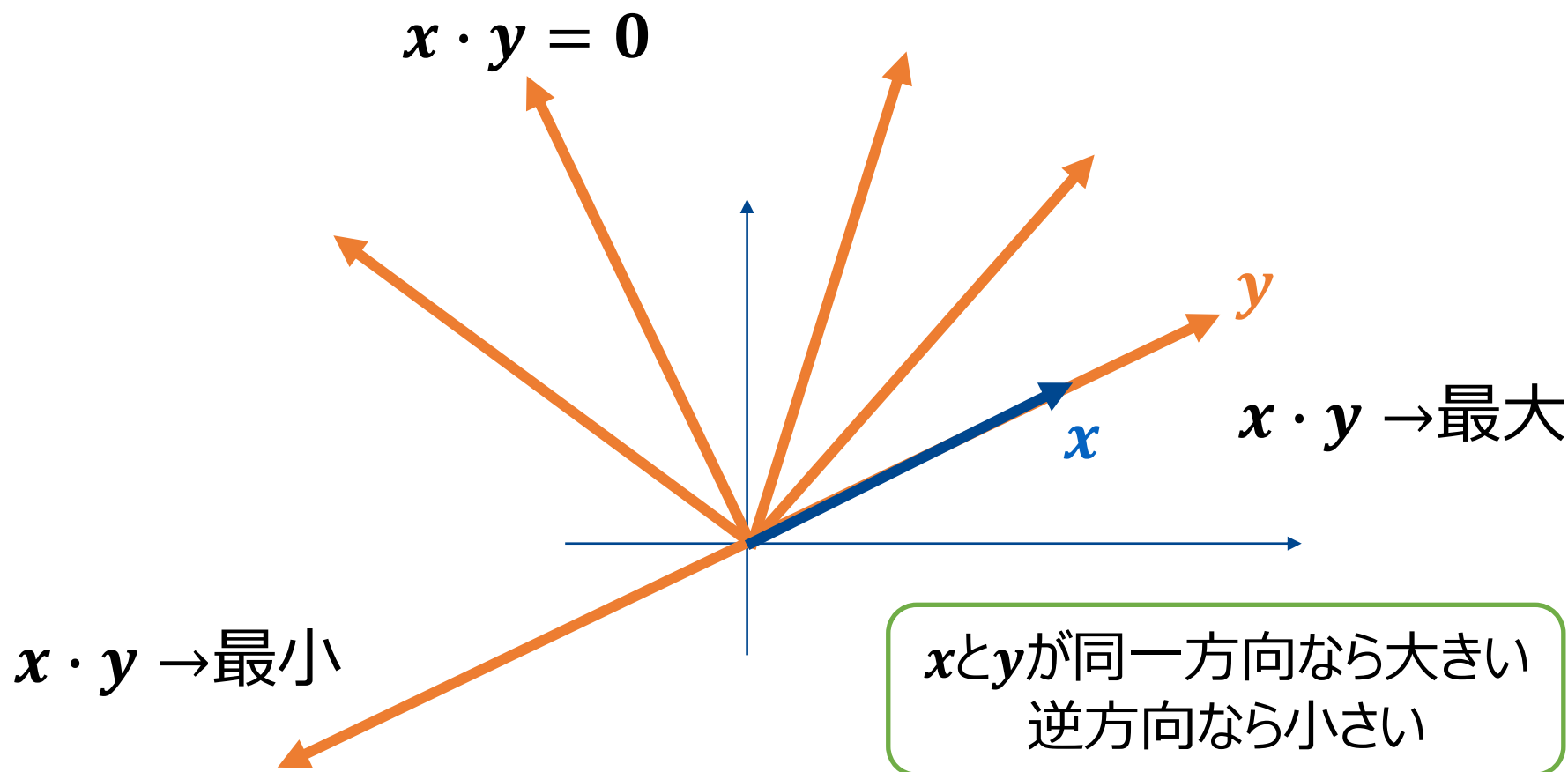


ちなみに2次元以外でも
成り立つ…



内積の性質： ベクトルのなす角 θ が変わると内積も変わる

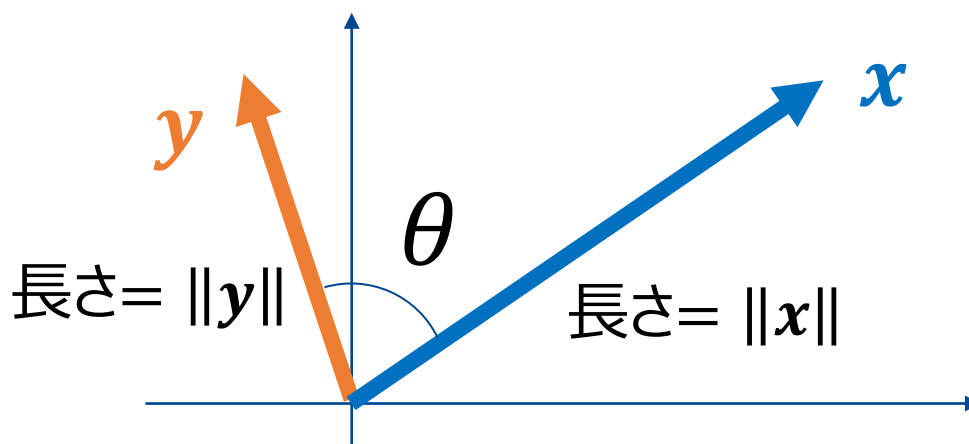
- 一定の大きさのベクトル y を回転させながら x と内積を取ると...



内積の性質： だったら、ベクトルのなす角 θ を使って書けるんじゃない？

- $\cos \theta$ を使って書けます

$$x \cdot y = \|x\| \|y\| \cos \theta$$

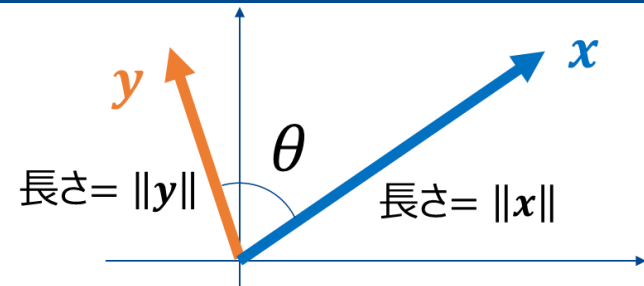


- なんか不思議ですね
 - 要素を掛け算して足したもの = ベクトルの長さとその間の角度で計算されたもの
 - さらに「掛け算と足し算」が「三角関数」と関係するなんて…!?

※この後 \cos はほとんど出ないので、わからない人も心配しないで...

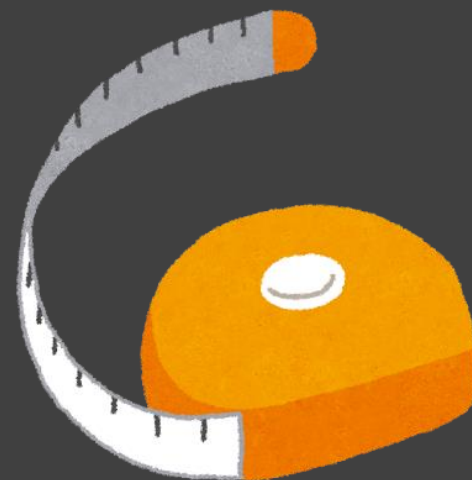
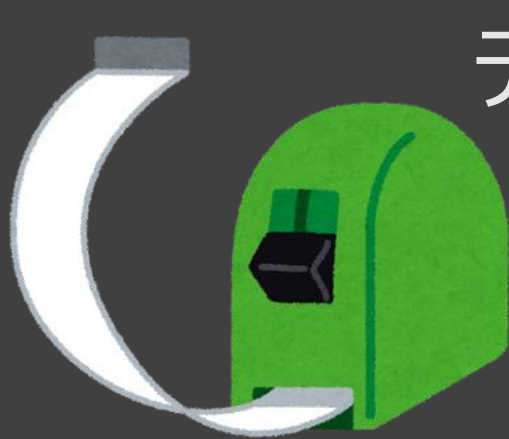
内積の性質：まとめると

$$\mathbf{x} \cdot \mathbf{y} = \|\mathbf{x}\| \|\mathbf{y}\| \cos \theta$$



- ベクトルが長いほど(= $\|\mathbf{x}\|$ と $\|\mathbf{y}\|$ が大きいほど)内積値は大きくなりやすい
- $-1 \leq \cos \theta \leq 1$ だから、内積値は…
 - $\theta = 0^\circ$ の時に最大となって $\|\mathbf{x}\| \|\mathbf{y}\|$,
 - $\theta = 180^\circ$ の時に最小になって $-\|\mathbf{x}\| \|\mathbf{y}\|$
- そして $\cos \theta = 0$ の時、すなわち $\theta = 90^\circ$ のとき、内積値はゼロ、

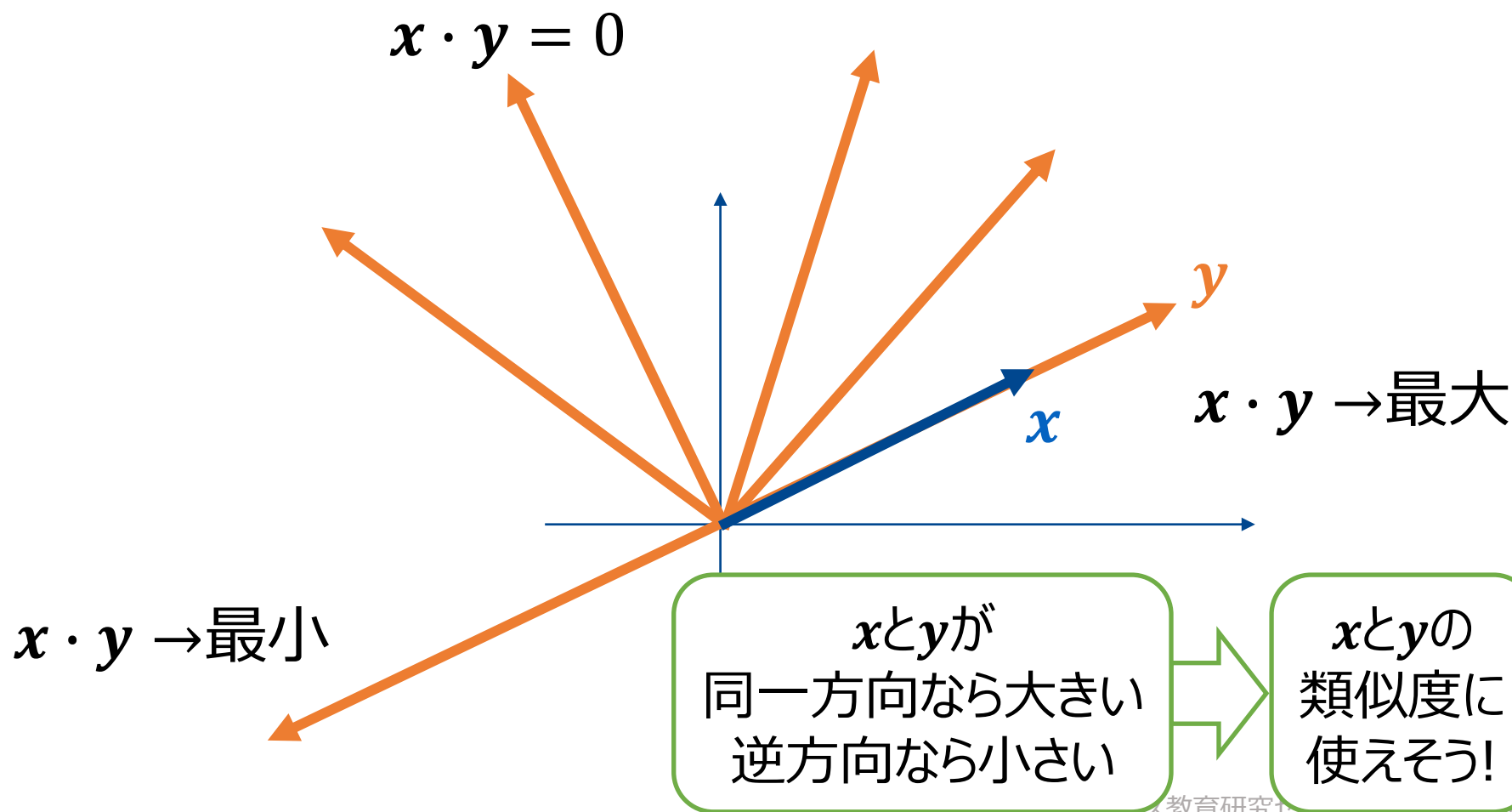
データ間の類似度



距離と逆で，似てると大きくなる！
内積をつかって計算できる！

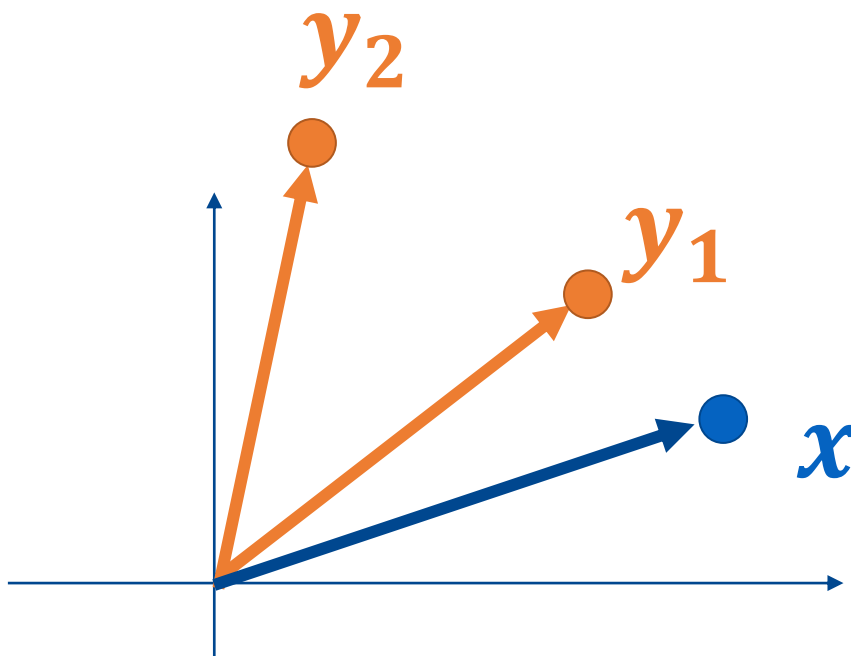
というわけで： ベクトルのなす角 θ が変わると内積も変わる

- 一定の大きさのベクトル y を回転させながら x と内積を取ると...



内積は類似度として使える？ (1/2)

- 使えるかも！
- 次の図なら, $\mathbf{x} \cdot \mathbf{y}_1 > \mathbf{x} \cdot \mathbf{y}_2$ (\mathbf{x} にとって, \mathbf{y}_1 のほうが似ている(近い))



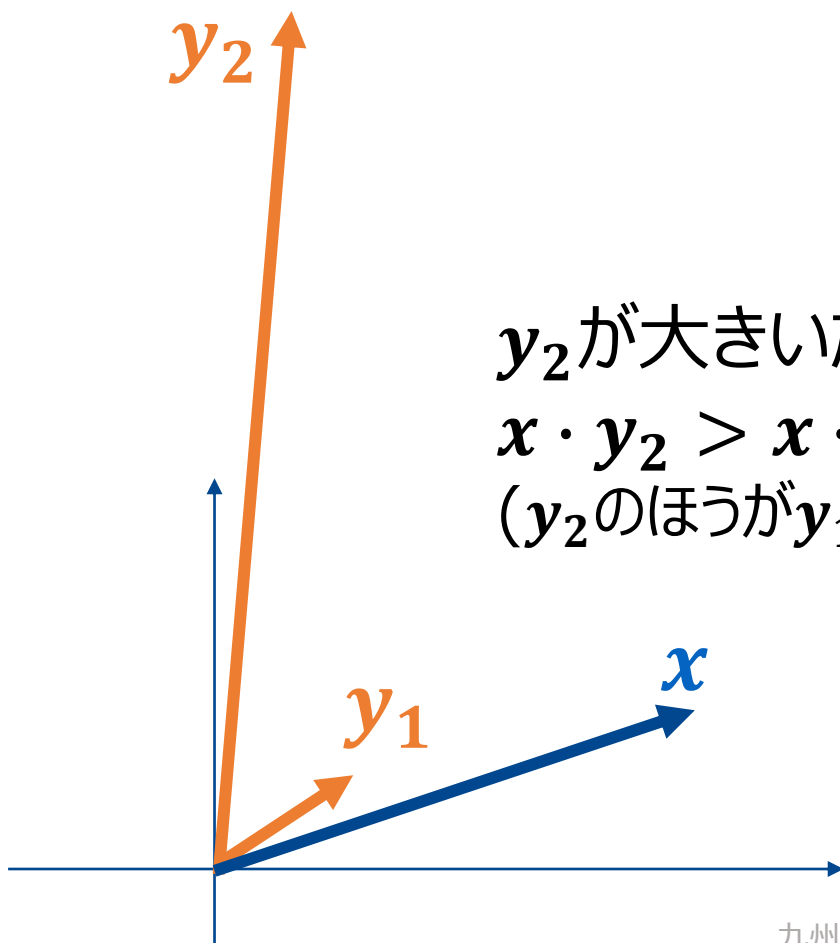
※実は機械学習やAIでは内積を類似度的に使います

内積は類似度として使える？(2/2)

- でも内積値はベクトルの大きさに依存するので...

$$x \cdot y = \|x\| \|y\| \cos \theta$$

y_2 が大きいために
 $x \cdot y_2 > x \cdot y_1$ となってしまう
(y_2 のほうが y_1 よりも x に類似!?)



最も代表的な類似度：正規化相関

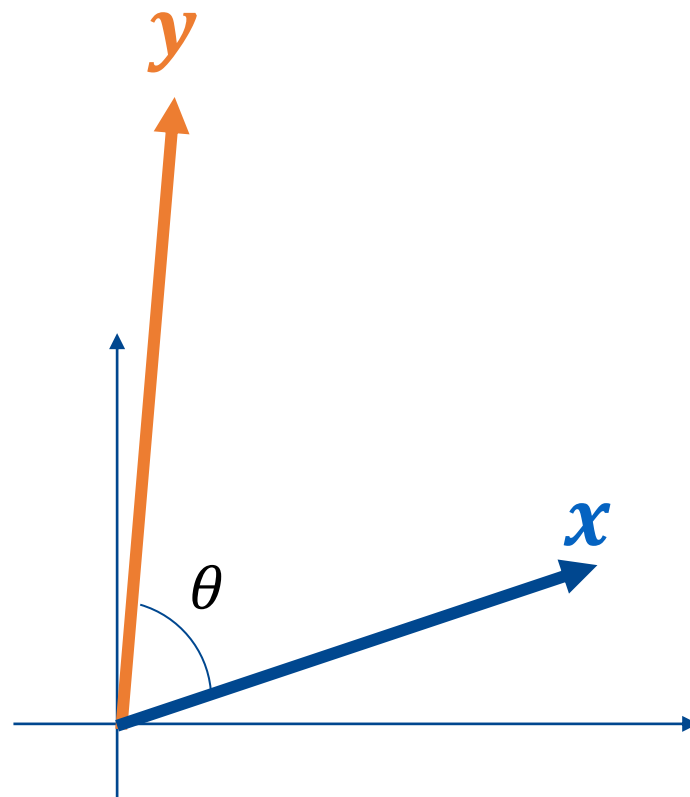
- ベクトルの大きさの影響が問題なら,
- 大きさに影響されない $\cos \theta$ を使えばいいのでは？

$$x \cdot y = \|x\| \|y\| \cos \theta$$



$$\cos \theta = \frac{x \cdot y}{\|x\| \|y\|}$$

結構簡単に求まる...



【付録 1】 距離とは何か？

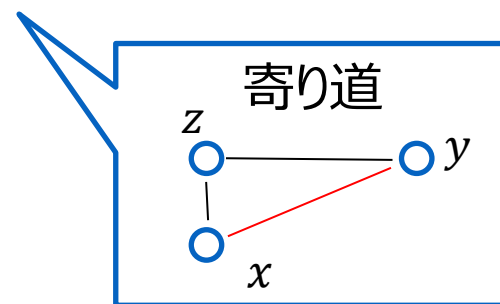


数学の本質はその自由さにあり！
答えが唯一の「ドリル」が数学ではない。
きちんと定義さえ守れば何でもあり！それが数学の本当の姿

そもそも距離ってなんだ？ (1/2)

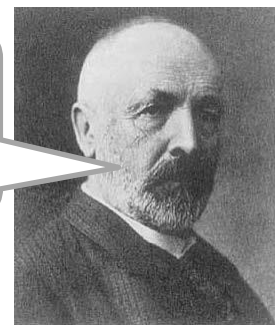
- 数学的には、次の3条件を満たす $d(x, y)$ を x, y の「距離」と呼ぶ
 - 非退化性(同じものだけ距離がゼロ) : $x = y \Leftrightarrow d(x, y) = 0$
 - 対称性(「 x から y へ」と「 y から x へ」の距離は同じ) : $d(x, y) = d(y, x)$
 - 三角不等式(寄り道したら遠くなる) : $d(x, z) + d(z, y) \geq d(x, y)$

↑ 「**距離の公理**」と呼ばれる
(公理 = 決めごと)



- 条件を満たすなら、何でも「距離」
 - 山本君が「山本距離」を勝手に作ってもOK
 - ルールさえ満たせば、何作ってもOK!

数学の本質は
その自由さにある
The essence of mathematics
is its freedom.



G. Cantor (1845-1918)

そもそも距離ってなんだ？ (2/2)

- 実用上は上記条件を満たさない $d(x, y)$ を使うことがある
 - 正確には「**擬**距離」(pseudo-distance)と呼ばれる
 - 対称性を満たさない場合が多い
 - Ex. 2地点間の距離を, 「所要時間」で測ると...

登りと降りで
距離が違う!?



参考：数学が自由さを見せる例 ブール代数($1+1=1$ の世界)

- 0と1しかない世界で定義された数学
- 和と積しかない
- ルール

- $0 + 0 = 0$

- $0 \cdot 0 = 0$

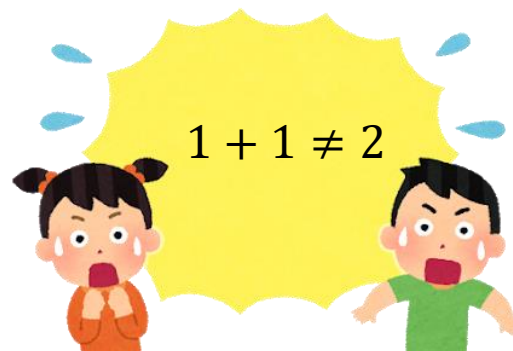
- $1 + 0 = 1$

- $1 \cdot 0 = 0$

- $1 + 1 = 1$

- $1 \cdot 1 = 1$

小学校1年で習った $1 + 1 = 2$ という常識が
通用しない世界！



参考：数学が自由さを見せる例 ブール代数(1+1=1の世界)

- 1=1個ではなく「1=on, 0=off」と考える
- 和 = 並列つなぎ, 積 = 直列つなぎ
- ルール

- $0 + 0 = 0$

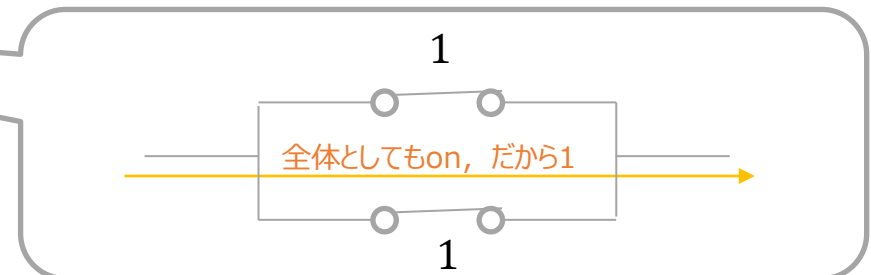
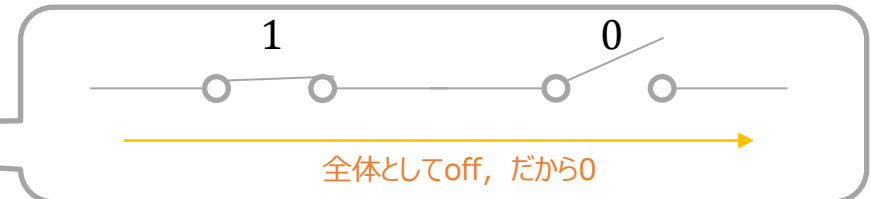
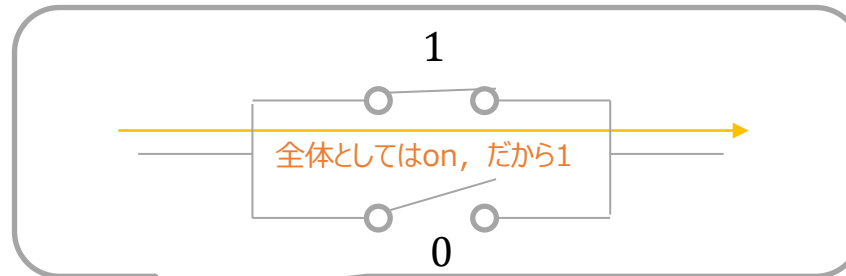
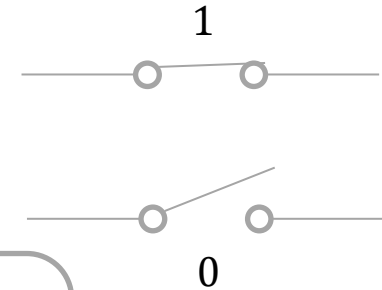
- $0 \cdot 0 = 0$

- $1 + 0 = 1$

- $1 \cdot 0 = 0$

- $1 + 1 = 1$

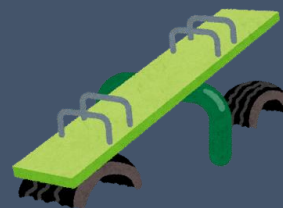
- $1 \cdot 1 = 1$



要するに{0,1}や{+,·}の意味が, 小学校の時と違う!

【付録 2】 ユークリッド距離と内積の関係

実は関係してます



ユークリッド距離と内積の関係

- ユークリッド距離の計算を展開すると内積が出てくる

$$\|x - y\|^2 = (x - y)^2 = \|x\|^2 - \underbrace{2x \cdot y}_{\text{内積}} + \|y\|^2$$

ユークリッド距離(の二乗)

- なので、両者は無関係ではない！
- 特に $\|x\|$ や $\|y\|$ が一定なら、
ユークリッド距離大 \rightarrow 内積小

