情報科学
【AI・データサイエンス】

第5回
ベクトル・距離・類似度

ベクトルによるデータ表現
距離・類似度

ベクトルによるデータ表現

ベクトルとは何か？

高校数学等の先入観はとりあえずおいといて，
気楽に考えましょう． 単に数字の組です．

ベクトルとは?

- 複数の数値を「組」にしたもの
  - 組にした数値の数を「次元」という

- ( ) の中にカンマで区切って書く
  ※他にも書き方はあります

英数国理社の点数のベクトル

英　数　国　理　社
(50, 89, 77, 90, 40)

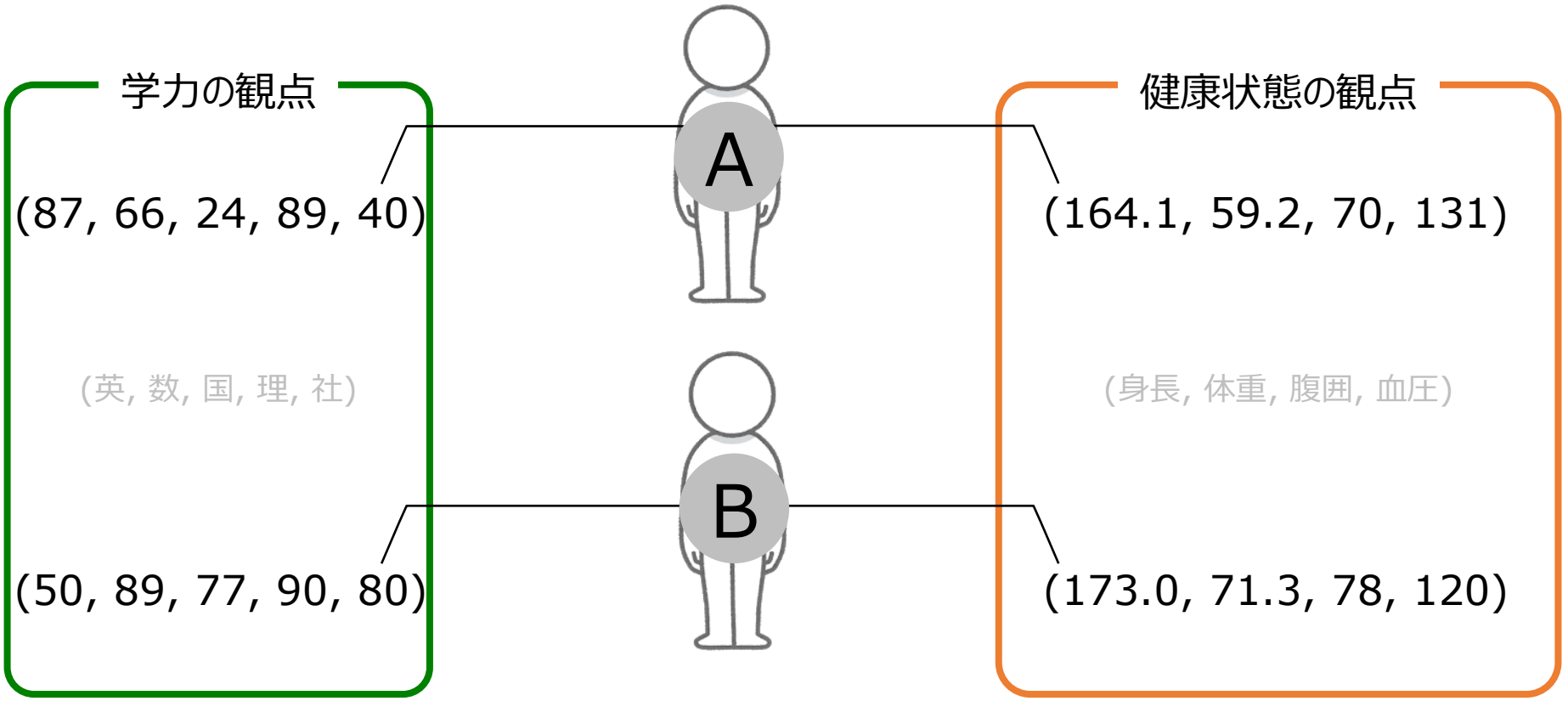
5つの数字の組だから
5次元ベクトル

身長・体重・腹囲・血圧のベクトル

身長　体重　腹囲　血圧
(173.0, 71.3, 78, 120)

4つの数字の組だから
4次元ベクトル

表現としてのベクトル



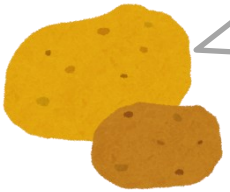
特定の観点から人をベクトルとして表現できる

突然ですが，料理を「分析」すると

材料の観点
から分析

玉ねぎ 80g

ニンジン 70g



ジャガイモ
50g

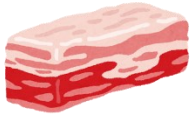
肉150g

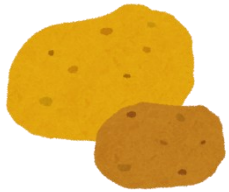
カレー粉 10g

色々なものが混ざっているので
パッと見ただけでは
どんな料理かわからない

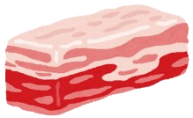
料理を「材料で表してみる」ことで分析する

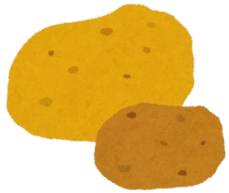
肉じゃが



カレー

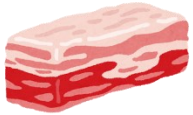
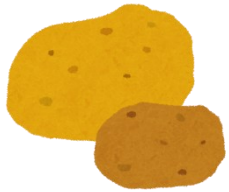



牛丼


何がどれぐらい混ざっているかわかったら，
どんな料理かクリアになる！

料理の分析結果：
料理をベクトルで表現すると，よくわかる！

						
肉じゃが	(60,	35,	35,	100,	0,	9)
カレー	(50,	80,	70,	70,	10,	5)
牛丼	(100,	50,	0,	0,	0,	18)

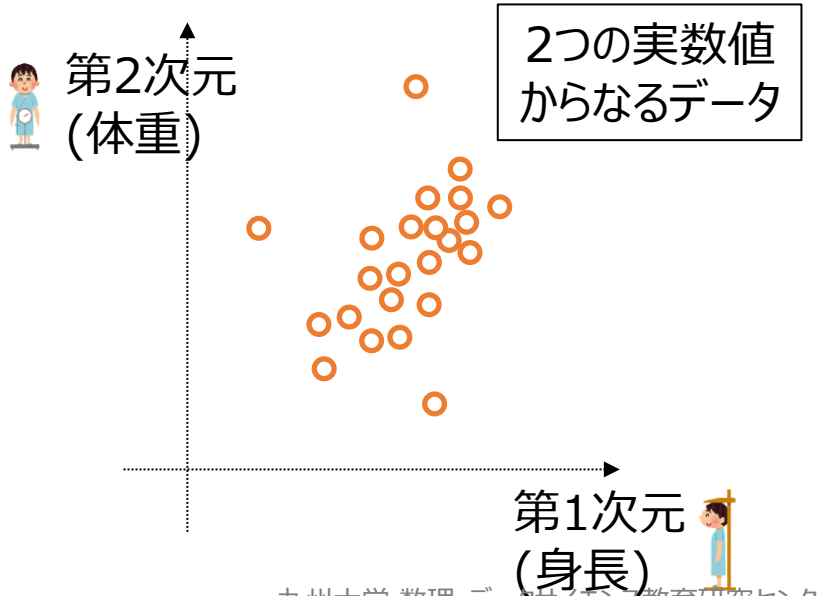
数値はグラム

体格をベクトルで表現してみる

身体計測データ

氏名	性別	身長	体重	…	測定日時
田中 太郎	男	(171.1	,62.2	)	2019-04-16 10:30:29
鈴木 次郎	男	(160.8	,55.5	)	2019-04-17 11:42:54
佐藤 葵	男	(165.0	,57.9	)	2019-04-17 15:21:11
⋮	⋮	⋮	⋮		

男＝０，女＝1とすれば，性別を
含めた多次元ベクトルとしても
表現できる



文書をベクトルで表現してみる

- 文書（単語の並び）もベクトルで表現できる
  - 「どんな単語がどのくらい使用されているか」に着目
  - 単語の順序は無視
    - 文書の部分の情報は失われる
    - 文書全体の大まかな情報のみ保持
    - どんな話題かぐらいは分かる

「僕」の出現回数　「犯人」の出現回数　「福岡」の出現回数　「東京」の出現回数　← 小説の成分

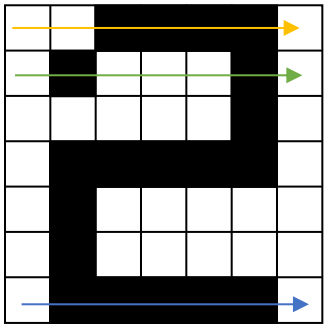
小説A  ➡ (100, 0, 1, 22, …)

小説B  ➡ (87, 97, 55, 10, …)

画像をベクトルで表現してみる (1/2)

画像の成分

- 画像（ピクセルの並び）もベクトルで表現できる
- 白(1)か黒(0)の2値画像の例

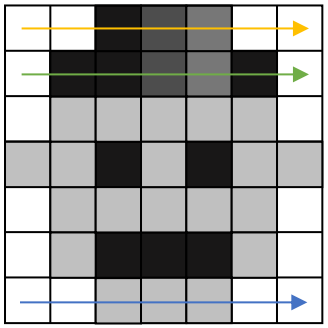


7×7画素

$$(1,1,0,0,0,0,1,1,0,\ldots,0,1)$$

49次元ベクトル

- 灰色を含むグレースケール画像の例(255=真っ白，0=真っ黒)

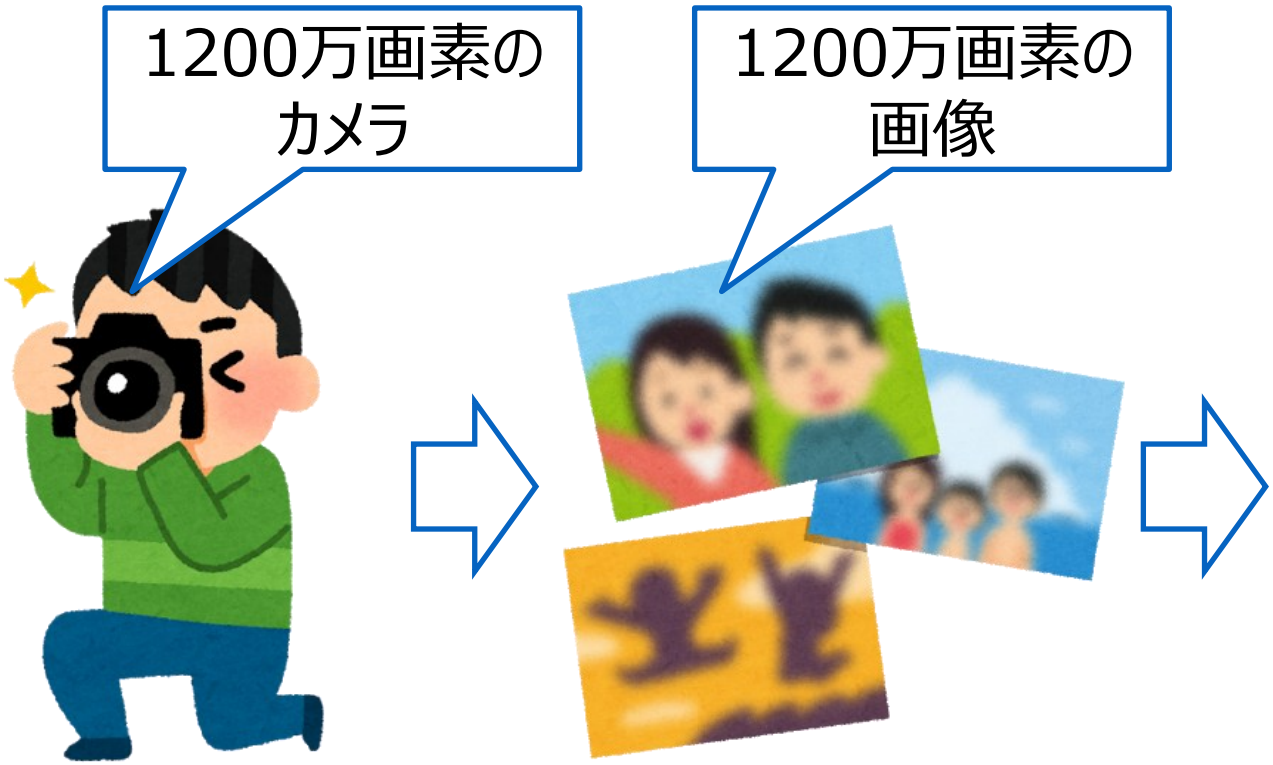


7×7画素

$$(255,245,10,35,92,231,254,\ldots,249)$$

49次元ベクトル

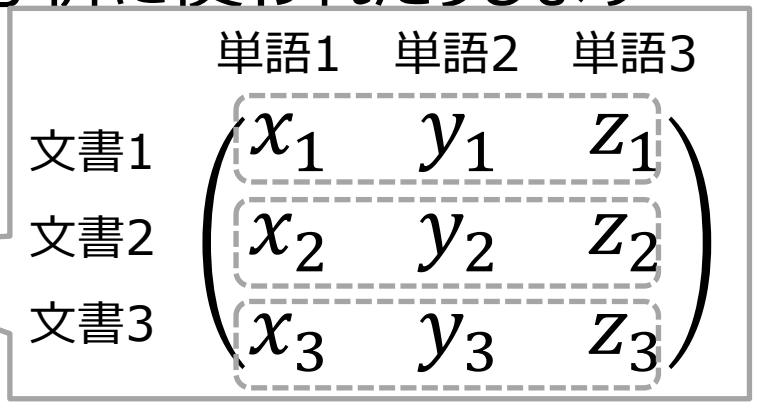
画像をベクトルで表現してみる (2/2)



- 皆さんのスマホ・デジカメ・コンピュータは，いつも超高次元ベクトルを扱っている
  - シャッター押した瞬間に1200万次元ベクトルが一つ生まれている

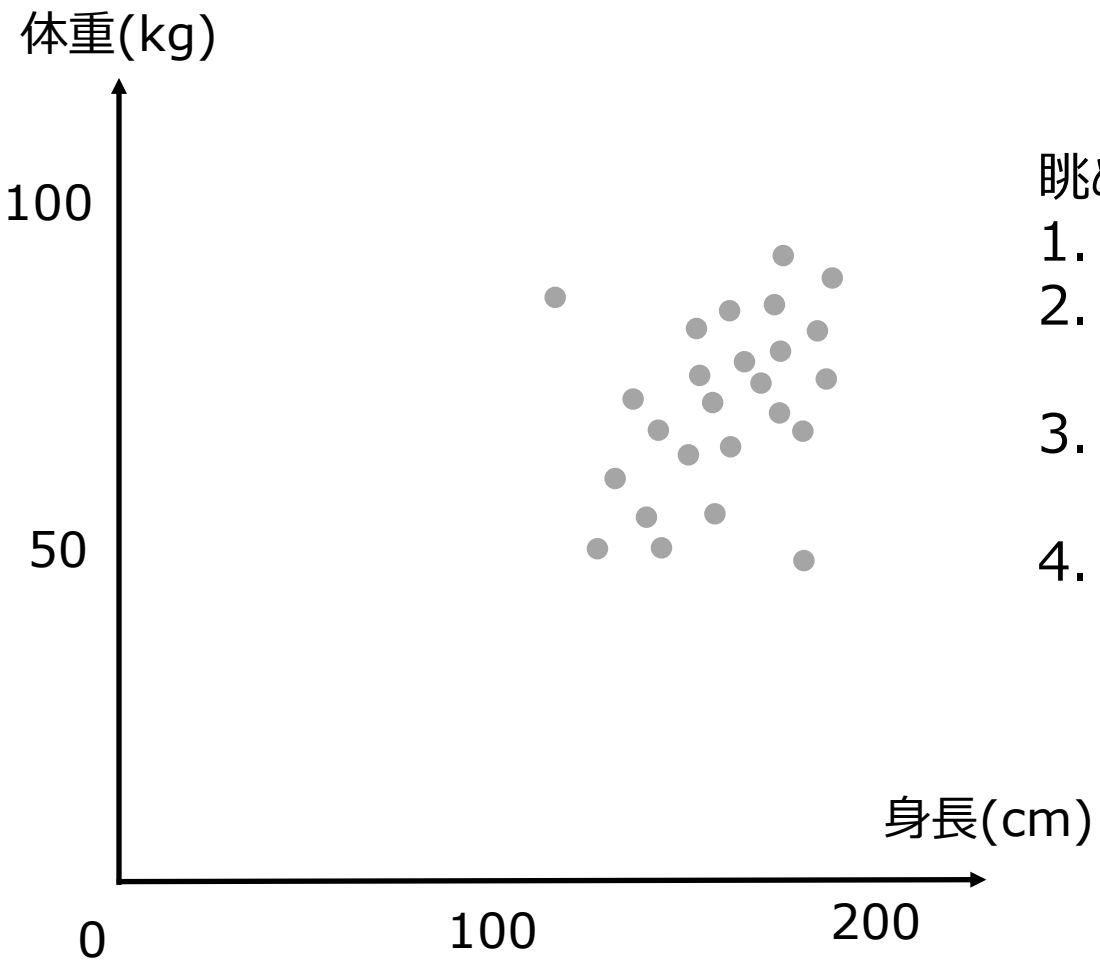
参考：線形代数

- ベクトルは，線形代数という科目でも習ったのでは？
- 線形代数で出てくる**行列**もデータ分析に使われたりします
  - 行列＝数字が縦横に並んだもの
  - 行列の使われ方
    - ベクトル(データ)の集まりとしての行列
    - 対応関係（ネットワーク）の表現
    - ベクトルを別のベクトルに変換する



- 要するに，「データ分析をより深く学びたい人は，線形代数も学ぼう！」

多数のベクトルデータ（体格データ）を眺めてみる



眺めることでわかること：

1. 身長が高いと体重が重い
2. ただしそれに従わない人（痩せすぎ・太りすぎ）も数名
3. 平均（分布の中心）は，160cm, 60kgぐらい
4. 身長100cm以下や，体重40kg以下の人はいない

アヤメの測定データ（1/3）

- 3つのアヤメの種の個体データ

Setosa

出典: CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=170298

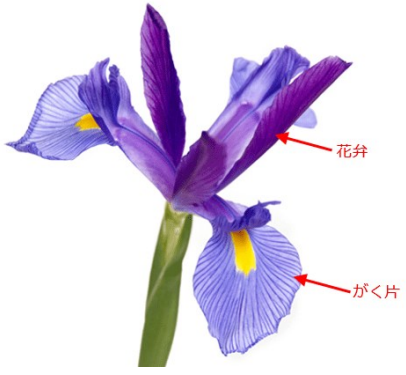
Versicolor

出典: By D. Gordon E. Robertson - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=10227368

Virginica

出典: By Frank Mayfield - originally posted to Flickr as Iris virginica shrevei BLUE FLAG, CC BY-SA 2.0, https://commons.wikimedia.org/w/index.php?curid=9805580

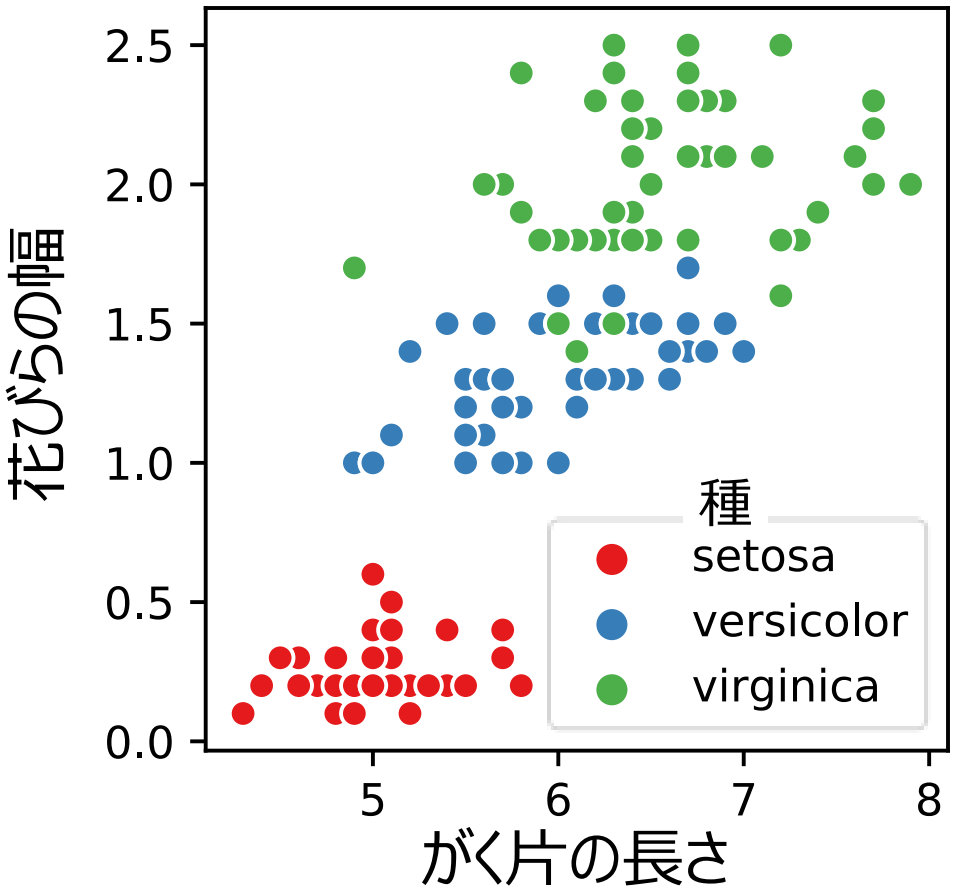
- 種ごとに50個体ずつ
- 「花弁」や「がく片」を測定



https://atmarkit.itmedia.co.jp/ait/articles/2003/24/news016.html

アヤメの測定データ（2/3）

- (がく片の長さ，花びらの幅)の2次元ベクトルで全データを眺めてみる



眺めることでわかること：

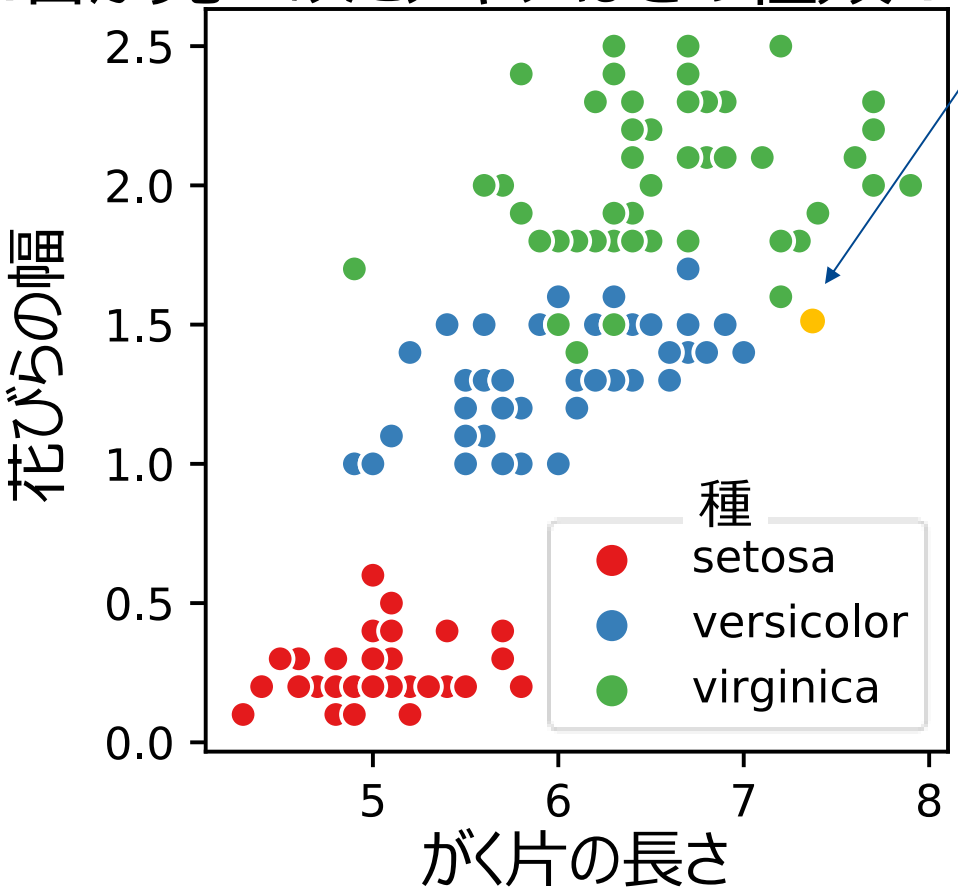
1. がく片が長いと，花びらの幅も広い
2. **3種がある程度分かれて分布している**

↓

「がく片の長さ」「花びらの幅」がわかれば，どの種類のアヤメか，「識別」できるのでは！？

アヤメの測定データ（3/3）

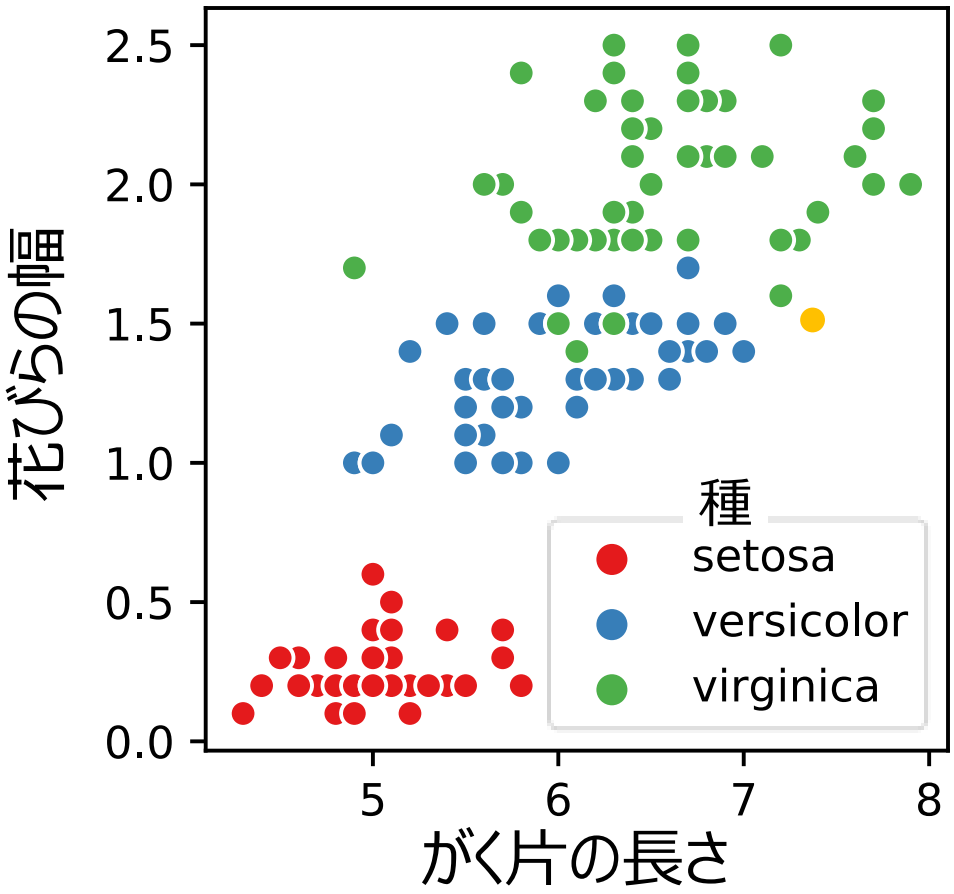
- A君が見かけたアヤメを測定したら黄色の点になった
- A君が見つけたアヤメはどの種類？



「黄色の点」の
最も近くにあるのは
「緑の点」だから
Virginicaでは？

「最も近く」！？

- 感覚的にはわかるけど「データ間の近さ」ってどう測る？
  - 次のセクションでは，それを学ぼう！



「黄色の点」の
最も近くにあるのは
「緑の点」だから
Virginicaでは？

距離・類似度

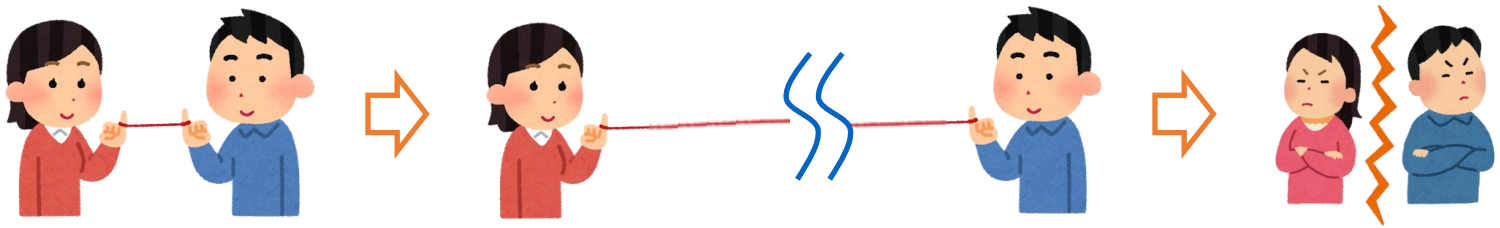
距離 = 2つのデータがどれぐらい似てないか？
類似度 = 2つのデータがどれぐらい似ているか？

まぁ，
同じようなもの

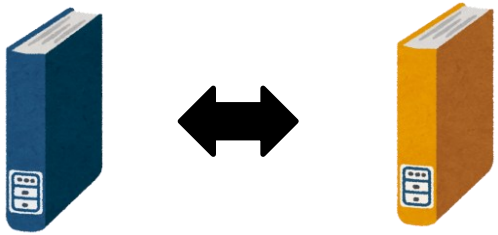
距離や類似度とは何か？

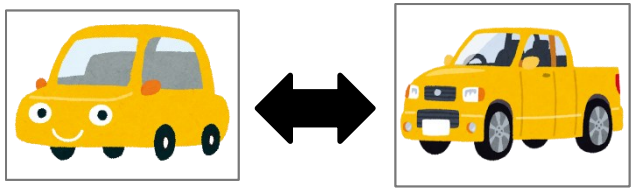
距離

- 日常会話における「距離」
  - A地点とB地点がどれぐらい離れているか？
    （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ぐらい離れているか？



- データ解析における「距離」はもっと自由
  - 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



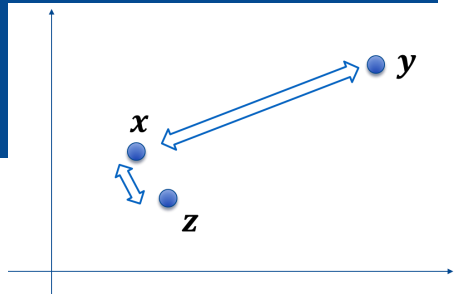


類似度＝距離の逆

- 距離は「似てない具合」，類似度は「似てる具合」
  - まぁ，同じようなもの…
  - 大雑把に言えば，
    - 類似度が大きいものは，似てる．だから距離は小さい
    - 類似度が小さいものは，似てない．だから距離は大きい

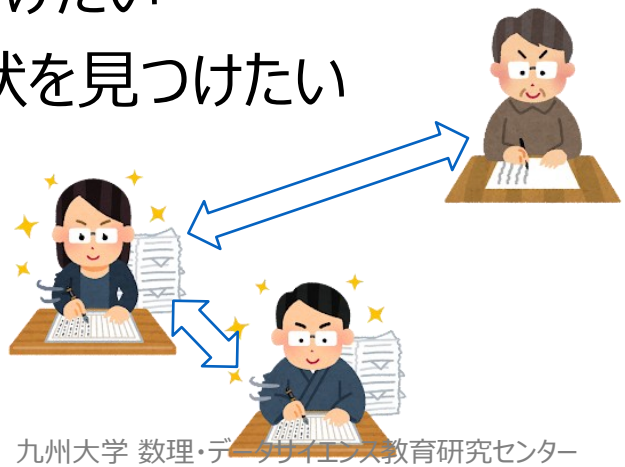
双子は…
類似度高い，
すなわち距離小さい

距離や類似度は何に使えるか？ 実は超便利！（1/2）



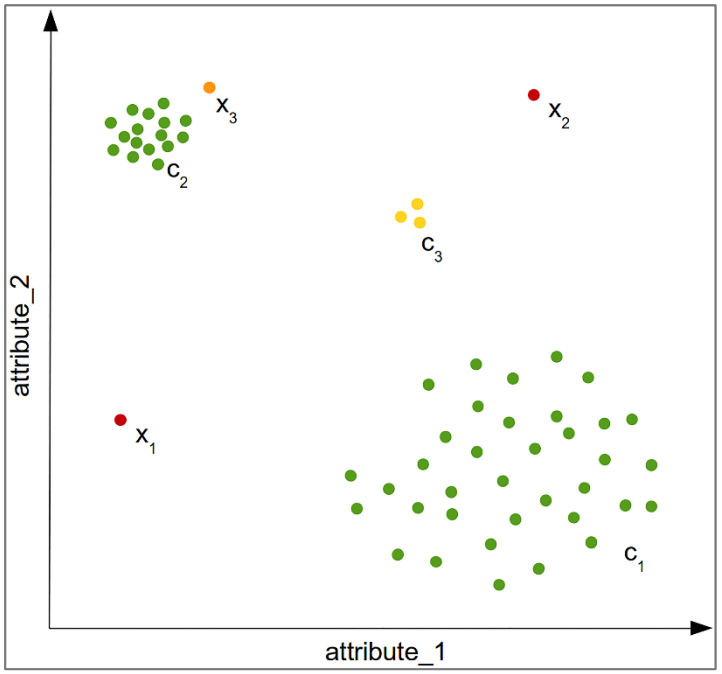
- データ間の比較が定量的にできる
  - 「xとyは全然違う/結構似ている」「xとyは28ぐらい違う」
  - 「xにとっては，yよりもzのほうが似ている」

- 応用例
  - 好きな曲（小説）と似てる曲（小説）を知りたい
  - ある性質をもつ化合物と近い化合物を見つけたい
  - 診察した患者さんに似た既知の病変や症状を見つけたい
  - 作風が似ている別の作家を見つけたい
  - 2つの細菌が近縁種かどうか知りたい



距離や類似度は何に使えるか？ 実は超便利！（2/2）

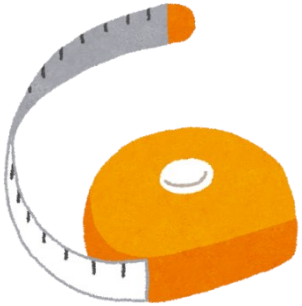
- データ集合のグルーピングができる
  - 「近く」のデータどおしでグループを作る
  - 「クラスタリング」と呼ばれる(後述)
- データの識別（認識）ができる
- データの異常度が測れる
  - 「近く」にデータがたくさんあれば正常，一つもなければ異常（異常検出→付録）
- … and more!



[Goldstein, Uchida, PLoSONE, 2016]

「距離・類似度」の話を通して学んで頂きたいこと

- 距離や類似度は「データ解析の基本」である！
- 距離や類似度は１種類ではない！
- 距離や類似度が変われば，データ解析結果は「まるっきり」変わる
- データや解析問題の性質に合った「距離・類似度」を選ぶ必要がある
  - 様々な距離・類似度の原理，メリット・デメリットも理解しておこう

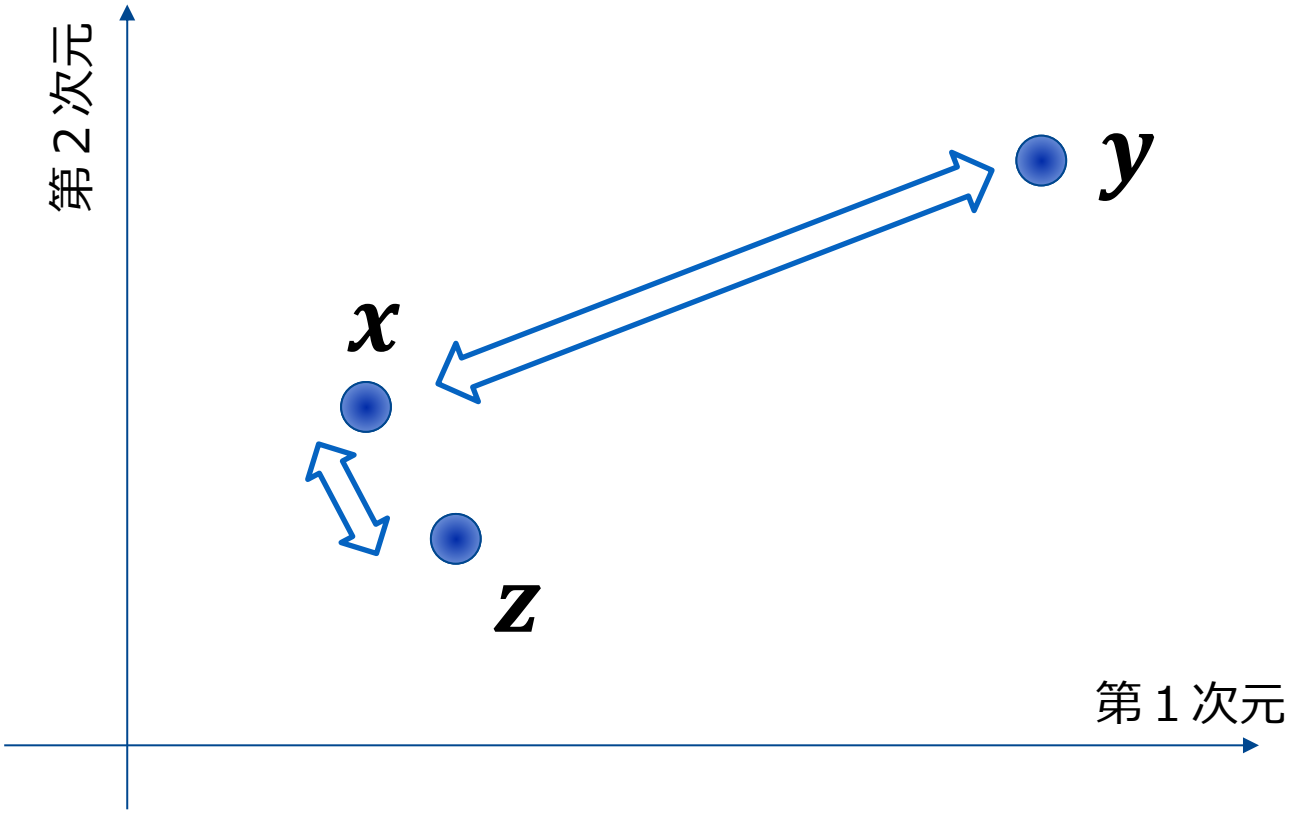


どんな方法も万能ではない！
メリット・デメリットを見極めて，
適切な方法を選択すること！

最も基本的な距離：ユークリッド距離

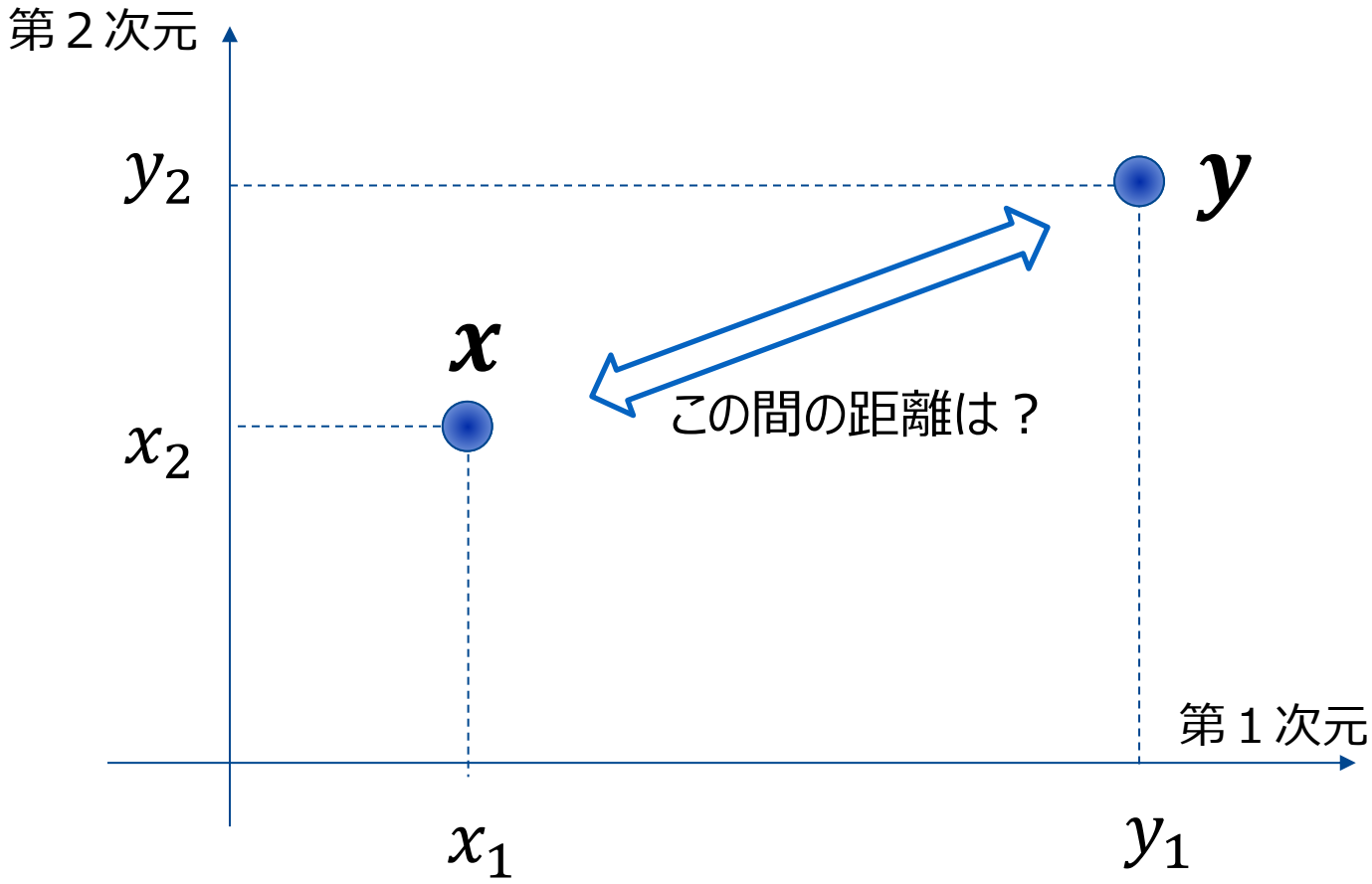
普通に考える「データ間の距離」

- 2データがどれぐらい違うか（＝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



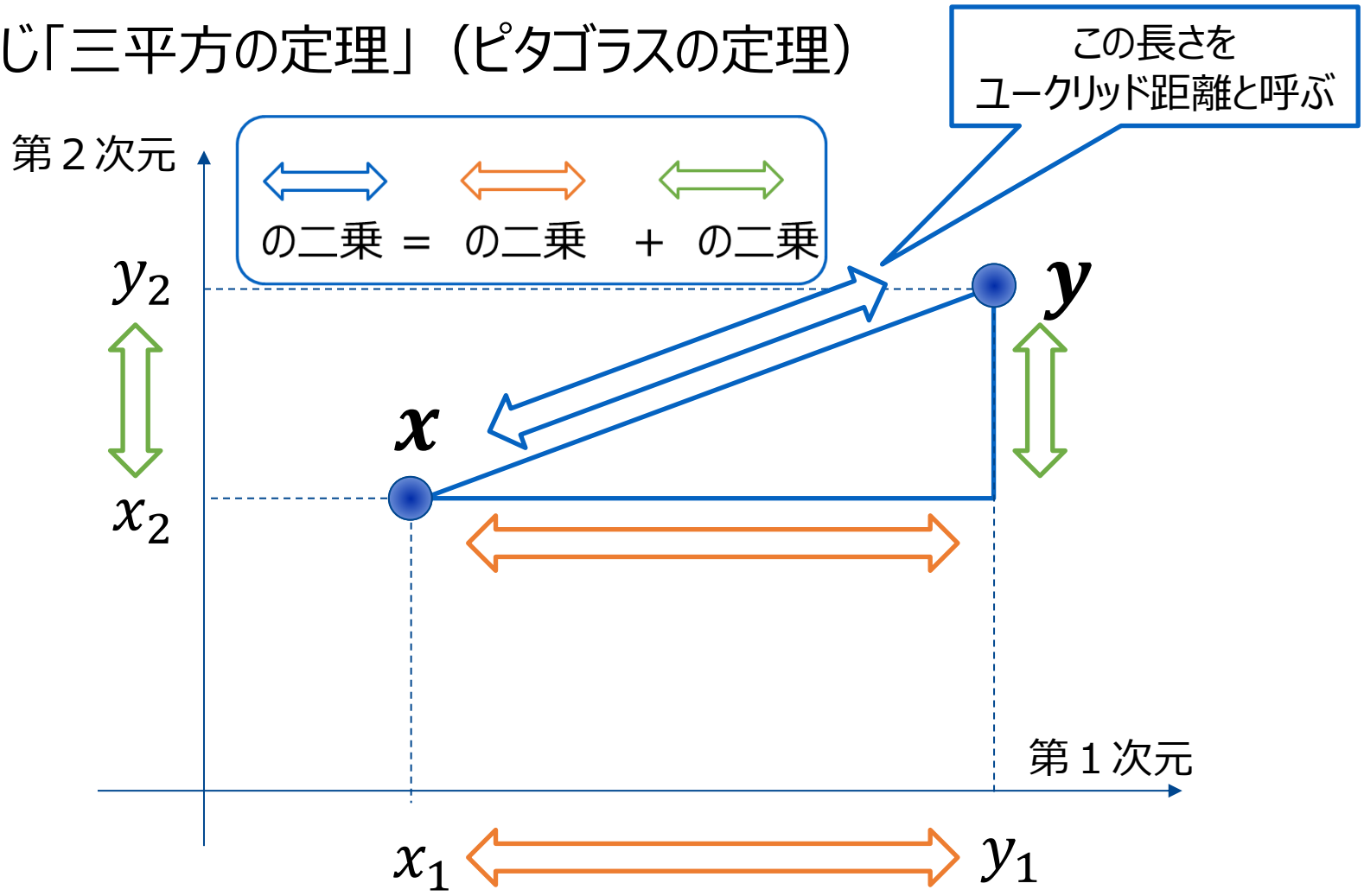
最も代表的な距離：ユークリッド距離 (1/2)

- 2つのベクトル $\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad \boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$



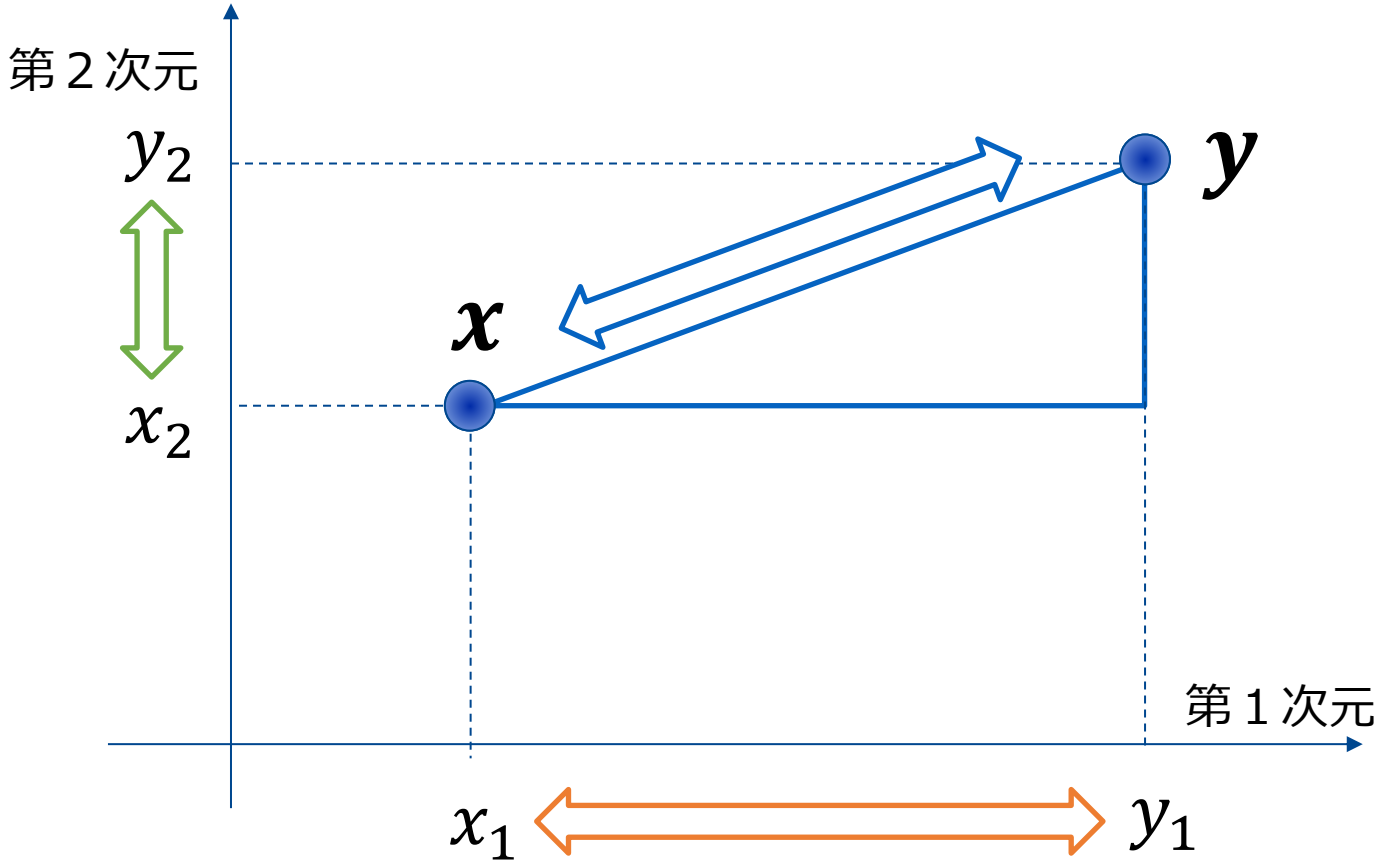
最も代表的な距離：ユークリッド距離 (2/2)

- ご存じ「三平方の定理」（ピタゴラスの定理）

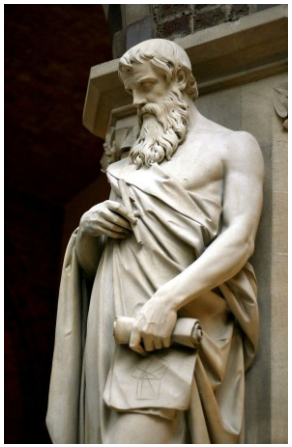


ユークリッド距離をもう少しちゃんと式で書くと

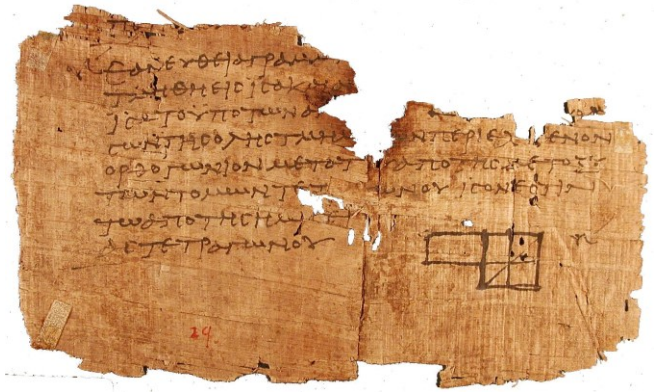
- $\boldsymbol{x}$と$\boldsymbol{y}$のユークリッド距離の二乗 $=(x_1-y_1)^2+(x_2-y_2)^2$



参考：ユークリッド＝幾何学の父



@エジプト
BC330～275年頃？

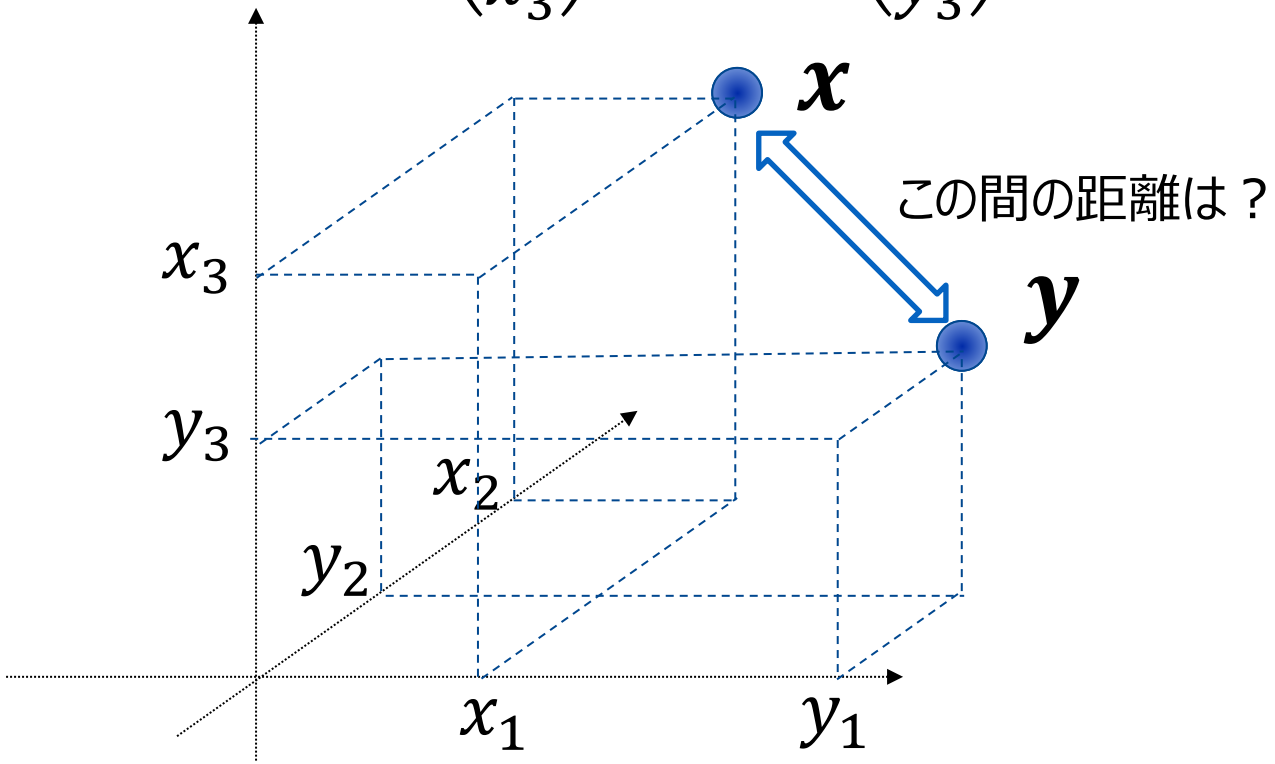


ユークリッド原論

- 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）

3 次元データでもユークリッド距離は測れる(1/2)

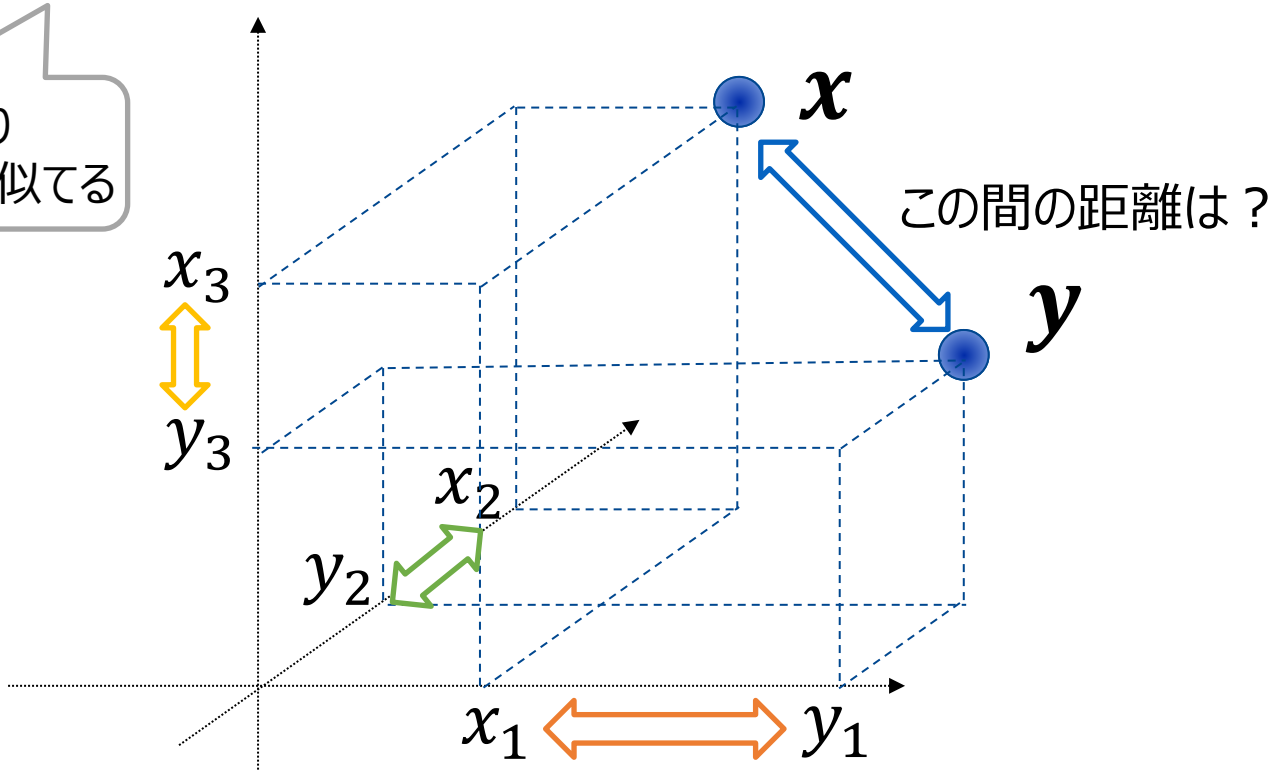
$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \quad \boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$



3 次元データでもユークリッド距離は測れる(2/2)

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

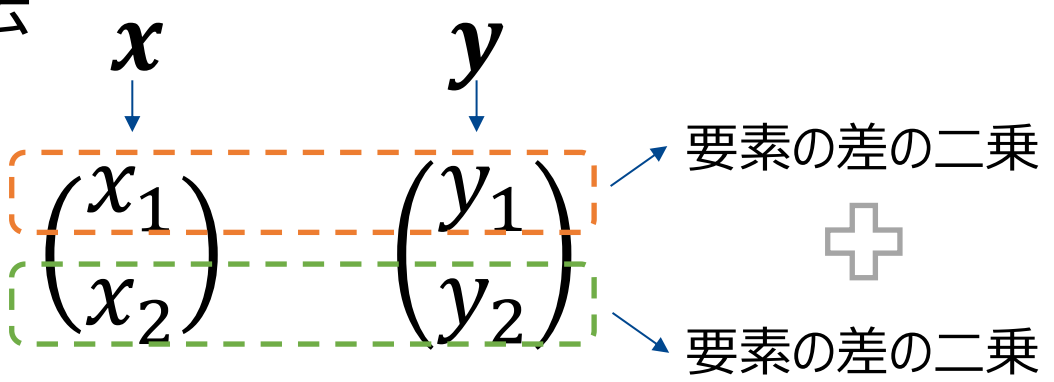
なんかやっぱり
ピタゴラスの定理に似てる



2次元と3次元での，ユークリッド距離の計算法（あれ？ 似てるな…）

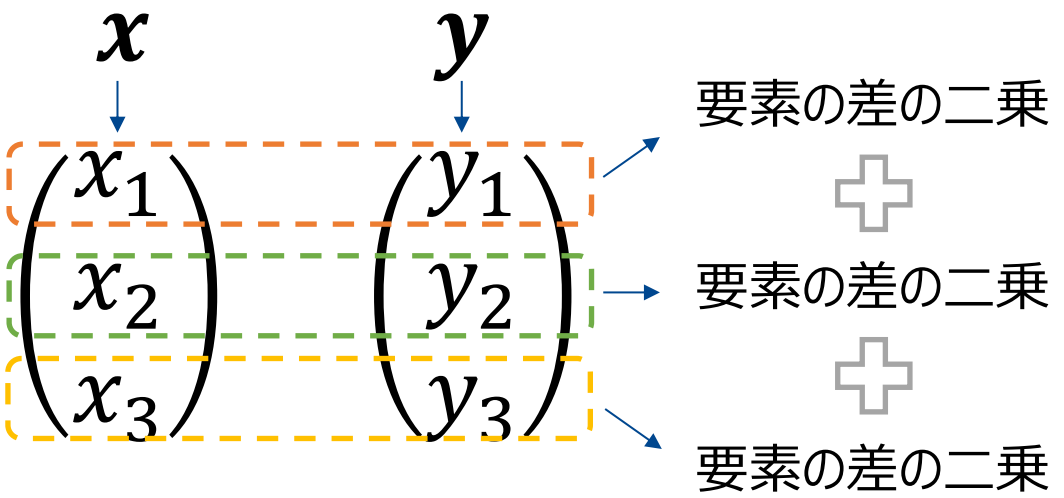
- 2次元の場合の計算法

$\boldsymbol{x}$と$\boldsymbol{y}$の距離の二乗 ＝



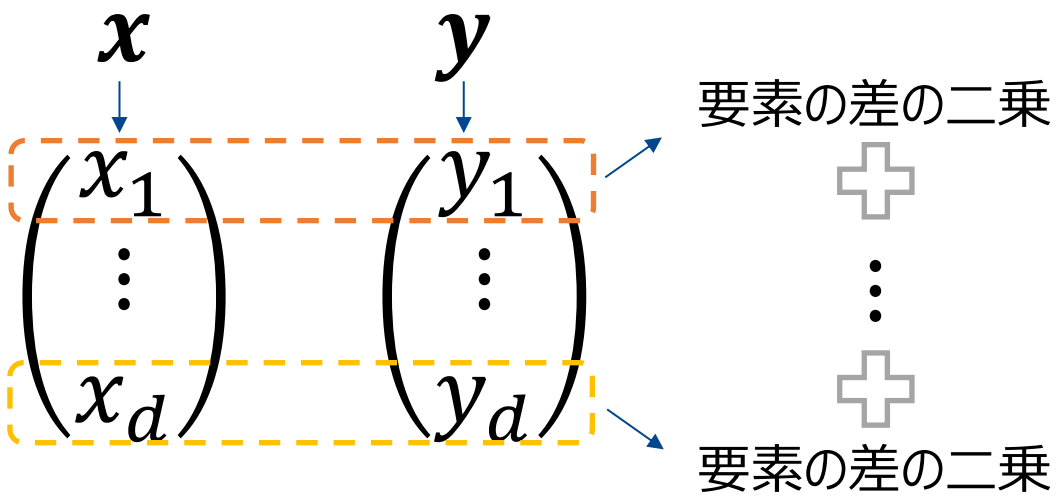
- 3次元の場合

$\boldsymbol{x}$と$\boldsymbol{y}$の距離の二乗 ＝



2次元でも3次元でも同じような計算法なら，$d\ (>3)$次元でも似たようなものでは？→ yes!

$\boldsymbol{x}$と$\boldsymbol{y}$の距離の二乗 $=$



というわけで，何次元ベクトルでも距離は計算可能

参考：いちいちd個の要素ごとの和を書くのがめんどくさい場合は，簡略表現

「要素ごとの差の二乗の合計」という意味．
結果はベクトルではなく，数値

ベクトル$\boldsymbol{x}$と$\boldsymbol{y}$のユークリッド距離の二乗 $= (\boldsymbol{x} - \boldsymbol{y})^2$

ベクトル$\boldsymbol{x}$と$\boldsymbol{y}$のユークリッド距離$= \sqrt{(\boldsymbol{x} - \boldsymbol{y})^2} = \|\boldsymbol{x} - \boldsymbol{y}\|$

なんだこの二重絶対値は？
（次スライド）

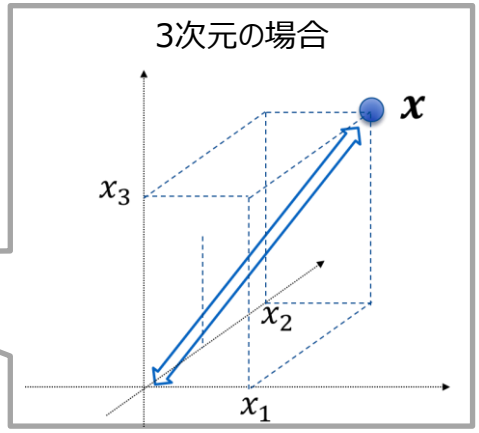
参考：なんだこの二重絶対値‖·‖は？

- $\|\boldsymbol{x}\|$はベクトル$\boldsymbol{x}$の長さを表すんです
  - ベクトル$\boldsymbol{x}$の「ノルム」とも言います！

- ベクトル$\boldsymbol{x}$の長さは
  (実はノルムにも種類があるんですが，そんなことまずは気にせずに考えれば)

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

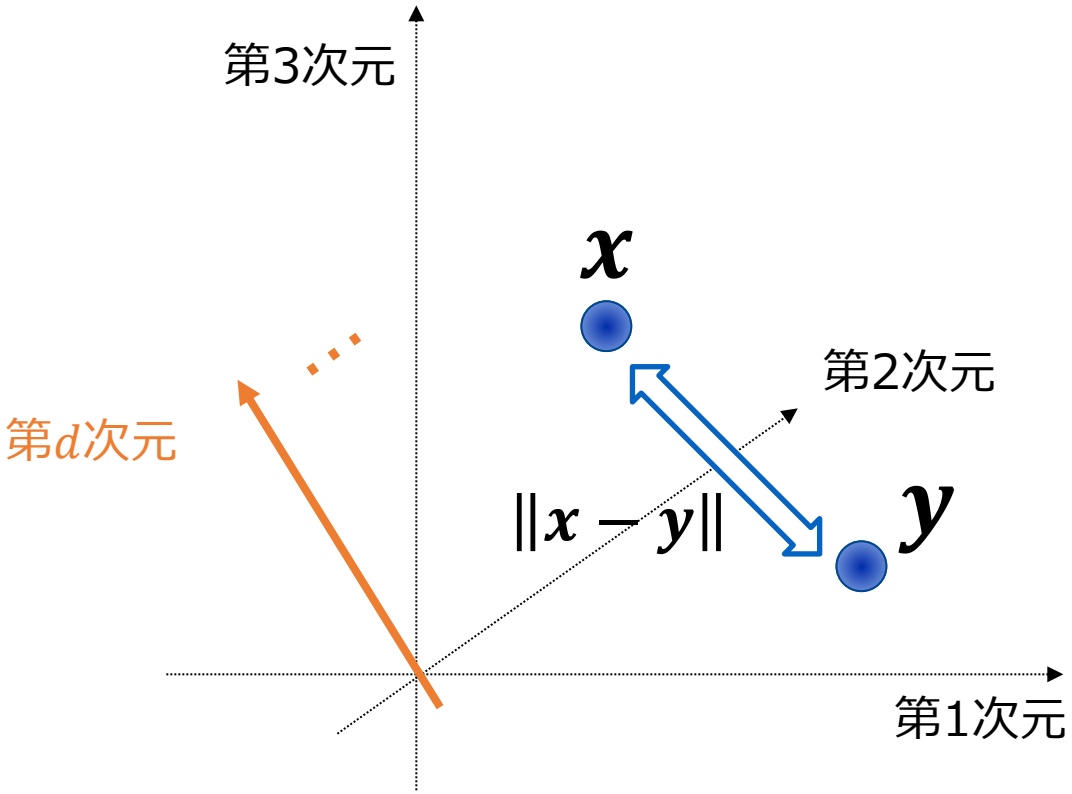
  となります



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

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

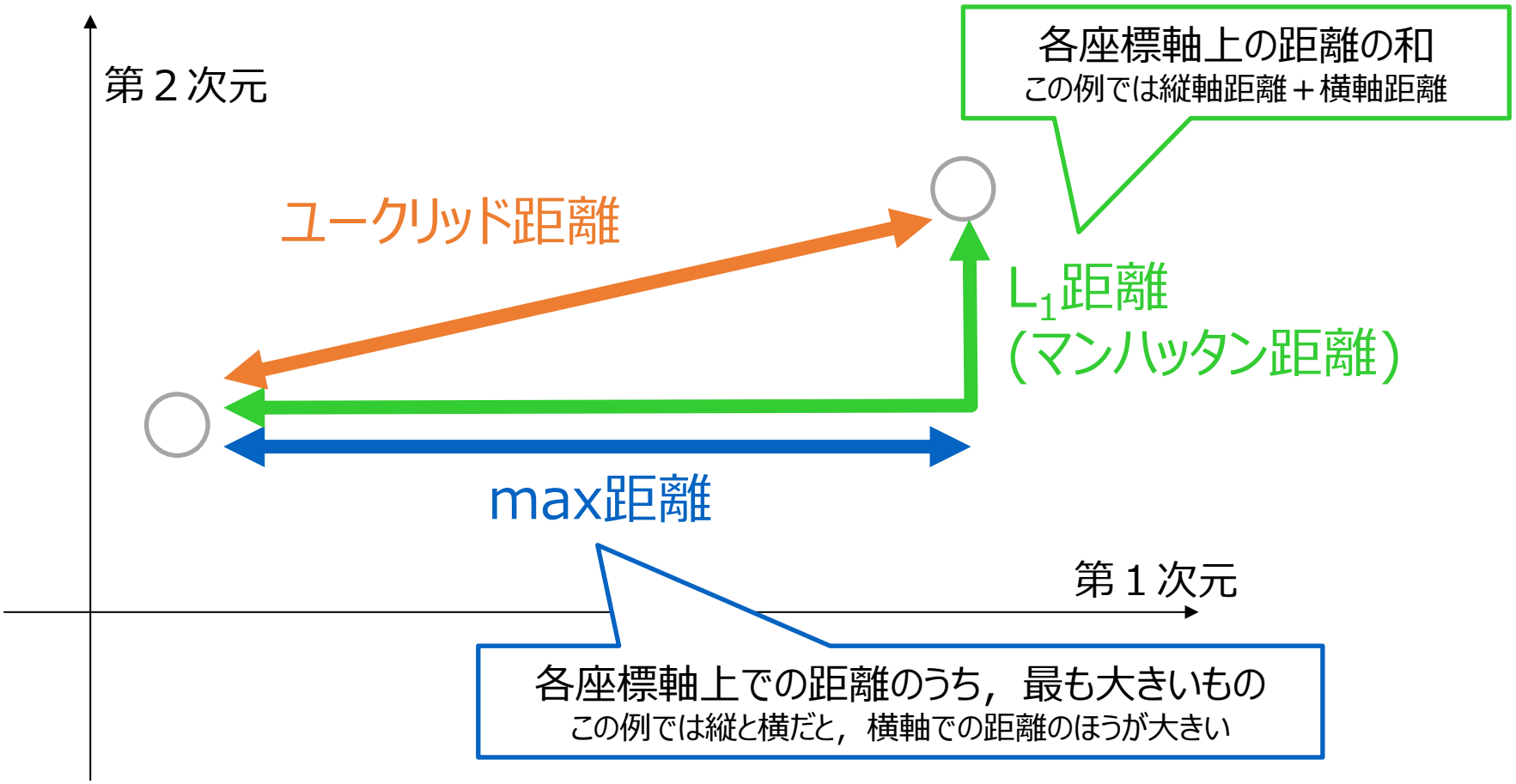
参考：図示するとやっぱりこんな感じ



様々な距離

ユークリッド距離以外にも色々ある

ユークリッド距離以外の様々な距離

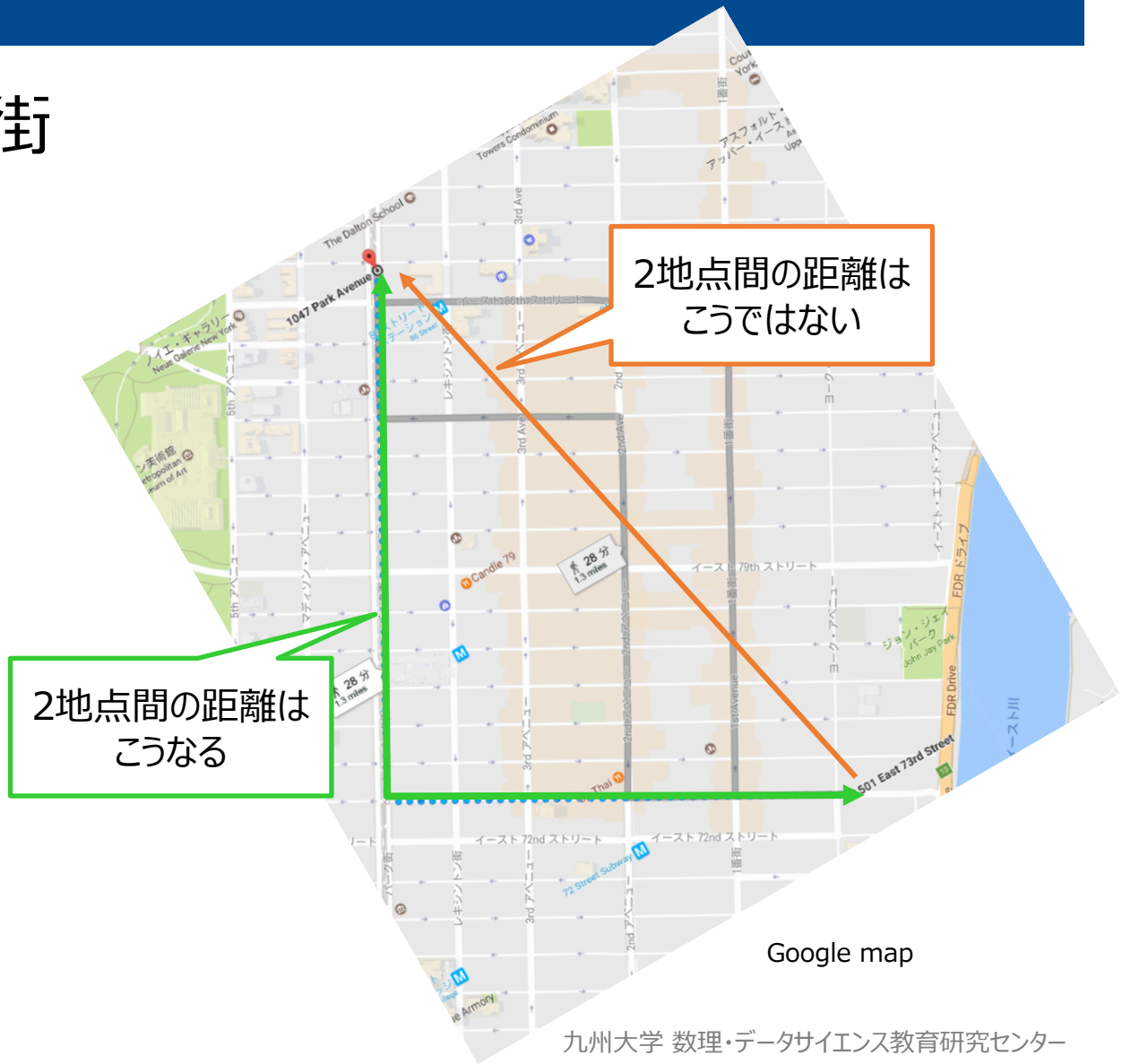


マンハッタン？

- 斜めには行けない街
  - 平安京距離
  - 平城京距離
  - 札幌距離

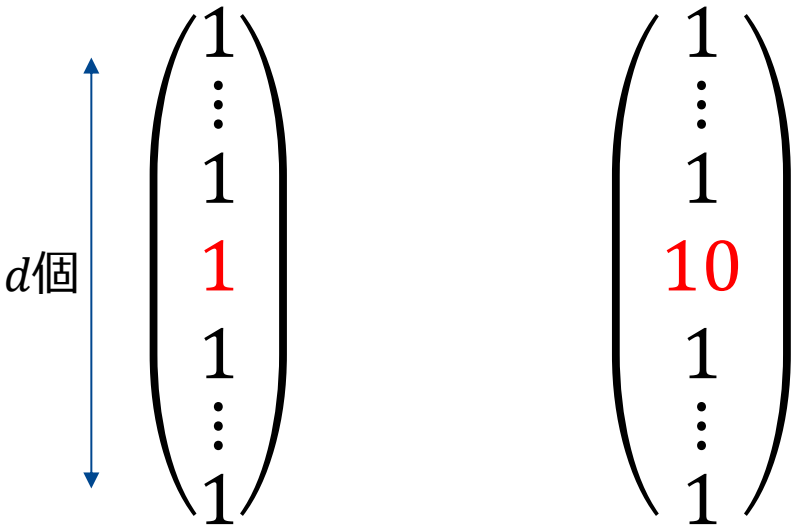
  でもいいかもね

- 「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



- 「ほとんどが一緒でも 1 要素でも大きく違ったら，それは結構違うのだ」としたいなら，max距離

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_∞距離 (上の式において$p = \infty$)

参考：L_∞がなぜmax?

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

d個の絶対値のうち，
一番大きいものが支配的

例($d = 3$)

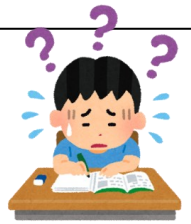
	2乗	10乗	1000乗
$\|x_1 - y_1\| =$	$10.1 \to 102.01$	$\to 1.1 \times 10^{10}$	$\to 2.1 \times 10^{1004}$
$\|x_2 - y_2\| =$	$10.2 \to 104.03$	$\to 1.2 \times 10^{10}$	$\to 3.9 \times 10^{1008}$
$\|x_3 - y_3\| =$	$10.3 \to 106.09$	$\to 1.3 \times 10^{10}$	$\to 6.8 \times 10^{1012}$

あまり差はないが…

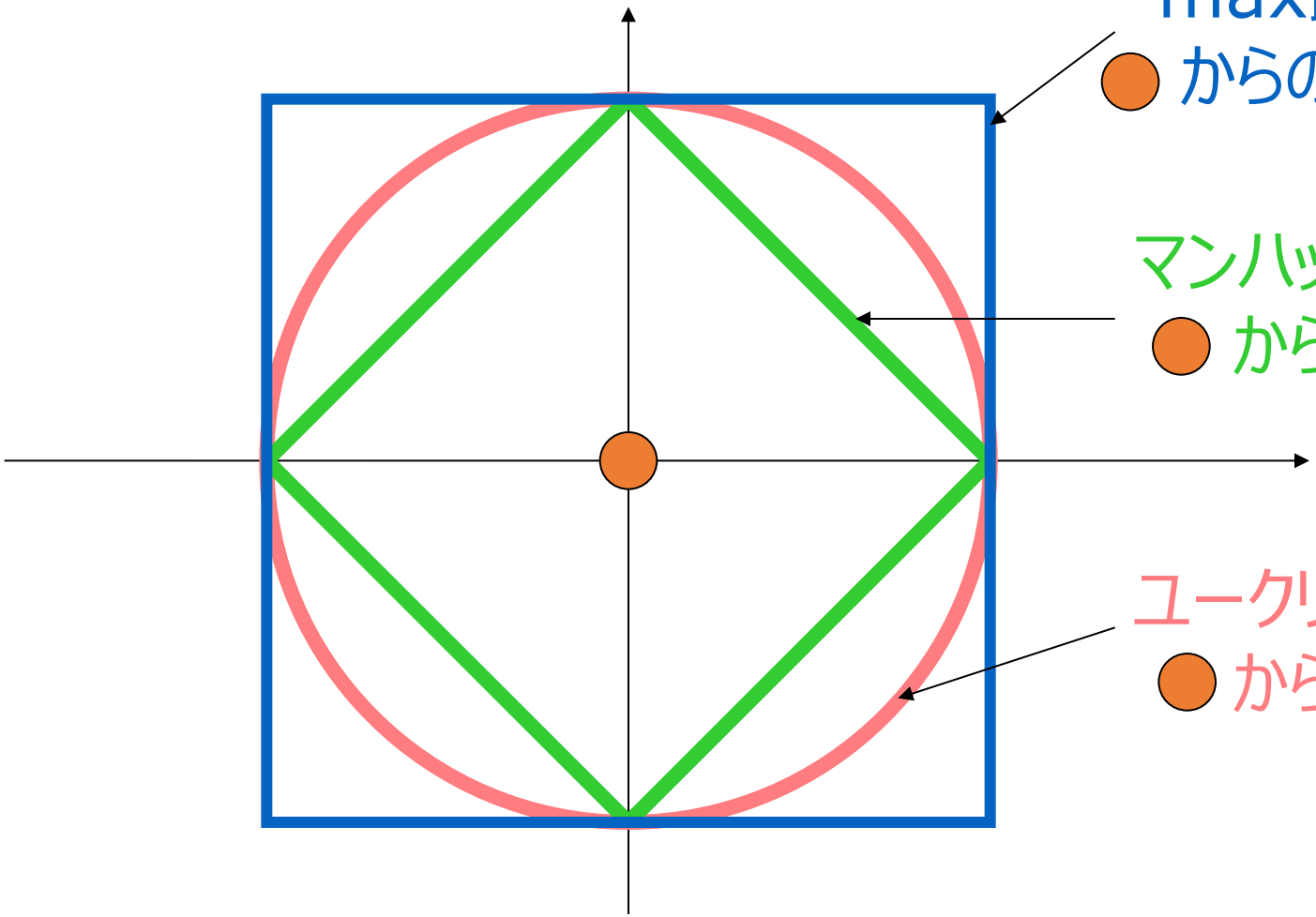
非常に大きな差となった！

最大以外のものは無視できる

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



等距離面



max距離で
● からの距離が1

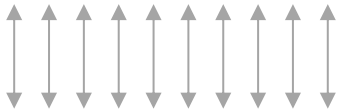
マンハッタン距離で
● からの距離が1

ユークリッド距離で
● からの距離が1

ベクトルでなくても，距離は測れる (1/2)
長さの同じ2系列間の距離

- 系列とは
  - 例えば文字列 “Synchronize”は長さ11の系列

- “Synchronize”と“Simchronise”と距離は？
  - ハミング距離 (Hamming distance)だとシンプルに測れる！
  - 要するに違うものの個数
  - “Synchronize”



  - “Simchronise” → 距離3

- 便利！でも長さが違う系列はどうする？

ベクトルでなくても，距離は測れる (2/2)
長さの異なる2系列間の距離

- 編集距離を使う！
  - 置換，挿入，削除の最小回数
  - ＝最初の文字列を修正してもう一つの文字列にするときの，最少打鍵数
    - キー一つである文字を別の文字に置換できる「謎」のキーを考えるので，注意

- 例： “This” ⇔ “These”
  - 置換1回(i⇔e) ＋ 挿入1回(e) → 回数 **2**
  - 削除1回(s) ＋ 置換(i⇔e)1回 ＋ 挿入2回(se) → 回数 **4**
  - 削除2回(is) ＋ 挿入3回(ese) → 回数 **5**
  - 削除4回(This) ＋ 挿入5回(These) → 回数 **9**
  - ….

置換技をうまく使ってる

全消しして
全打ち直し…

手順によって
必要な操作
回数が変わる

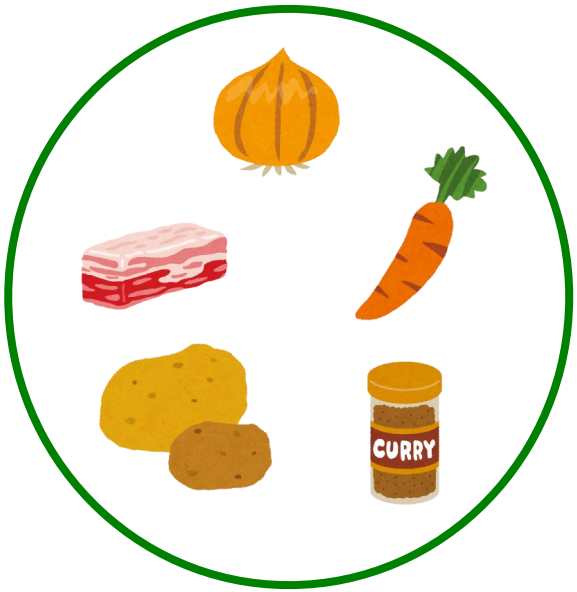
最小回数は2だから，距離2

Jaccard係数 (類似度)

- 「(数学の)集合」の類似度
  - 集合は何かの集まりを表し，入ってる/入ってないだけが重要
  - どのくらい共通しているかを測っている

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{\text{共通部分}}{\text{全要素}} = \frac{4}{6}$$

A =



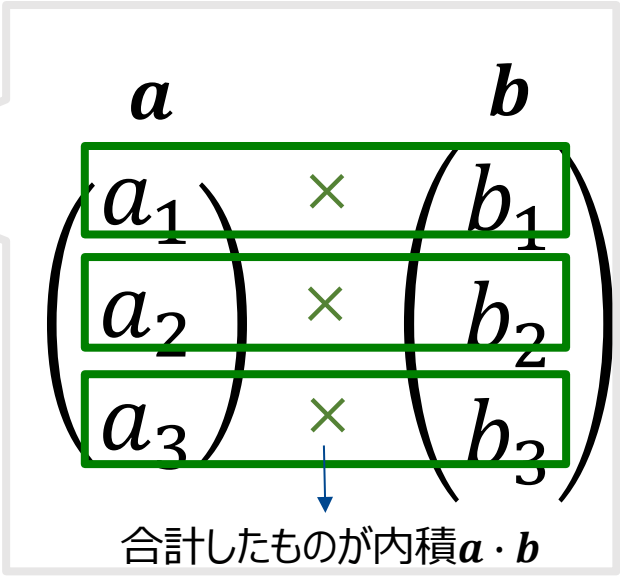
B =

参考：コサイン類似度

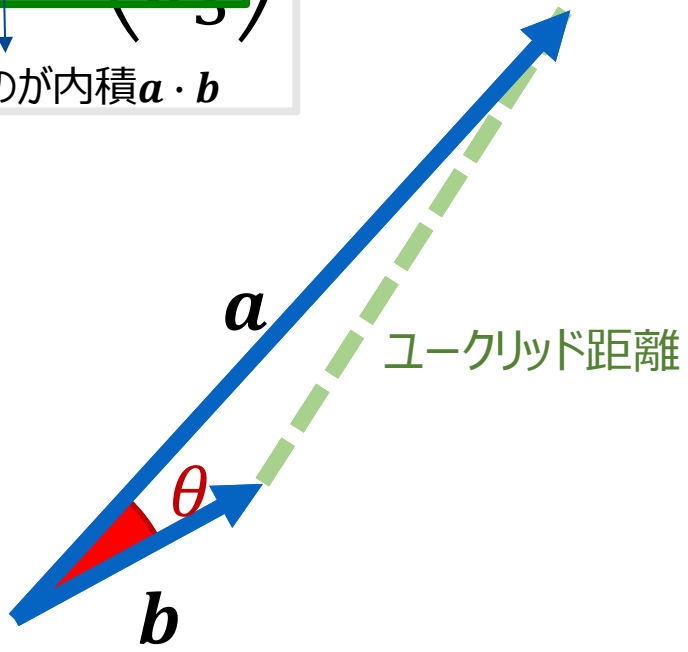
理系は高校の内積の式 $\boldsymbol{a}\cdot\boldsymbol{b}=|\boldsymbol{a}||\boldsymbol{b}|\cos\theta$ を思い出そう

- 方向性の類似度を測る方法

$$\cos\theta=\frac{\boldsymbol{a}\cdot\boldsymbol{b}}{|\boldsymbol{a}||\boldsymbol{b}|}$$



- $\cos\theta$は-1から+1の範囲で変化
  - -1 → $\boldsymbol{a},\boldsymbol{b}$は反対向き → $\boldsymbol{a},\boldsymbol{b}$は似てない
  - 0は直交
  - +1→ $\boldsymbol{a},\boldsymbol{b}$は同じ向き → $\boldsymbol{a},\boldsymbol{b}$は似てる

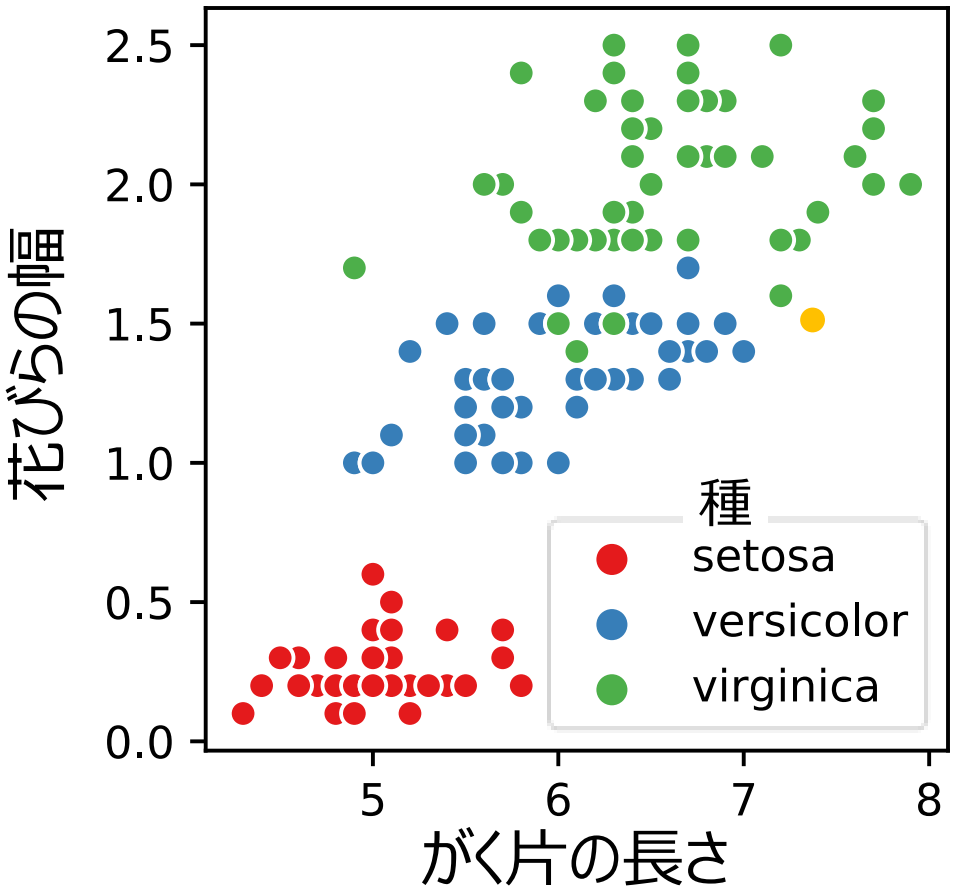


距離や類似度を利用したデータ分析①

識別（認識）

距離・類似度を使えば識別できる！
先ほどのアヤメの例

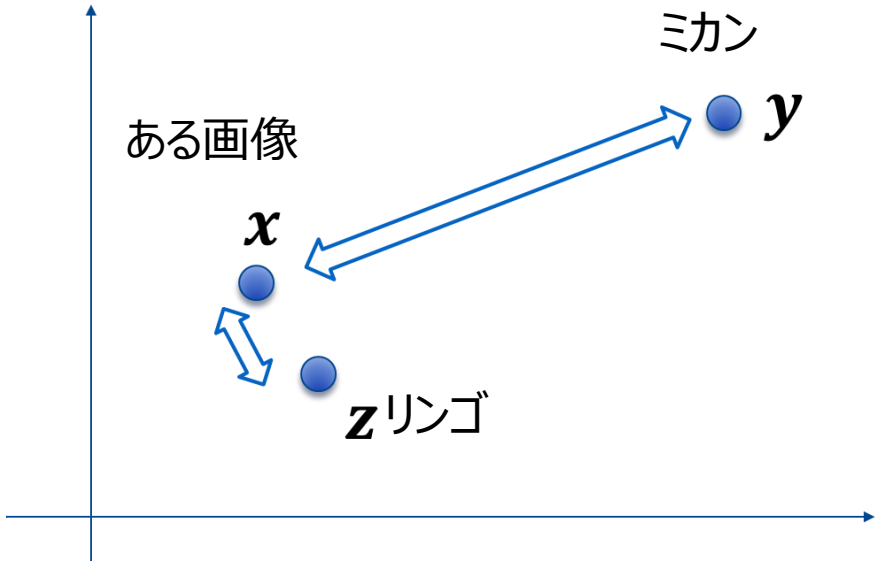
- 黄色の点は（最も近くのデータがvirginicaだから）virginicaと識別できる！



「黄色の点」の
最も近くにあるのは
「緑の点」だから
Virginicaでは？

距離・類似度で，画像認識！

- 登録されている画像データ中で，画像$\boldsymbol{x}$に最も似ているものは「リンゴ」の画像$\boldsymbol{z}$だった
  → 「画像 $\boldsymbol{x}$ はリンゴ」と識別



画像の距離をどうやって測る？
「画像をベクトルとみて，普通にユークリッド距離」が簡単

どちらも1000x1000画素の画像

100万次元ベクトル $\boldsymbol{x}$

100万次元ベクトル $\boldsymbol{y}$

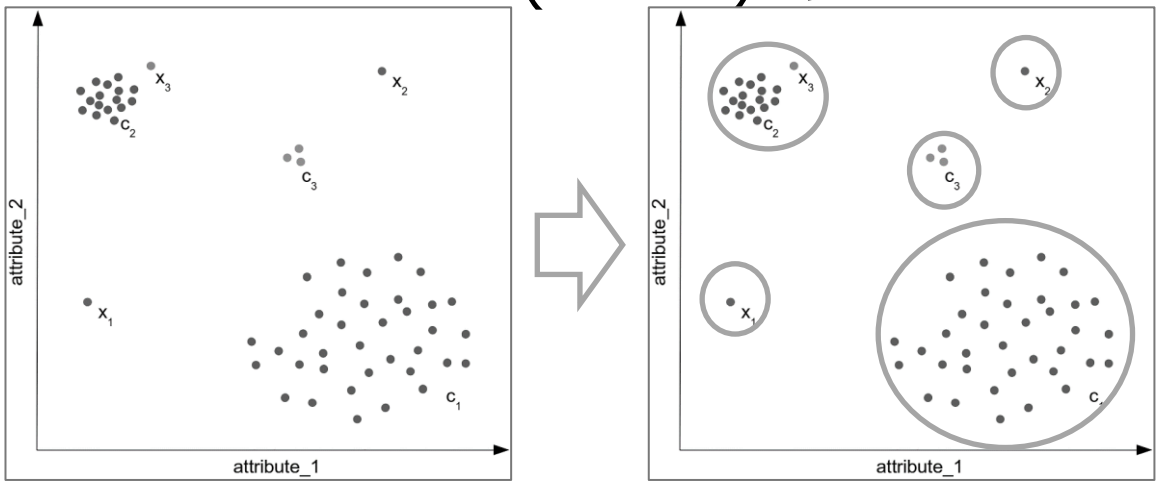
画像間距離

$$\|\boldsymbol{x}-\boldsymbol{y}\|$$

距離や類似度を利用したデータ分析②
クラスタリング

クラスタリング

- 近いデータをまとめてグループ(クラスタ)を見つけるデータ処理

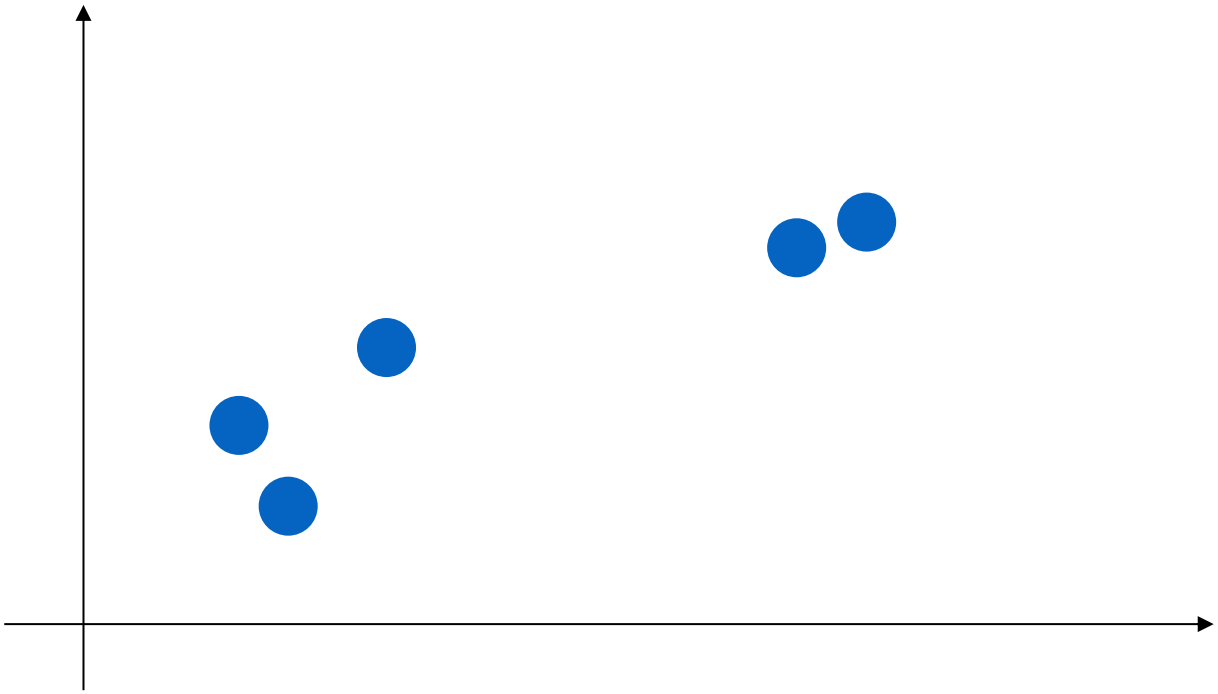


[Goldstein, Uchida, PLoSONE, 2016]

- 色々使える！
  - 例えばSNSで仲の良いグループを見つける，趣味の似た人達を見つける
  - 楽曲をまとめてジャンルを見つける
  - 遺伝子的に近い種のグループを見つける
  - 様々なニュース記事が扱っている共通の話題を見つける

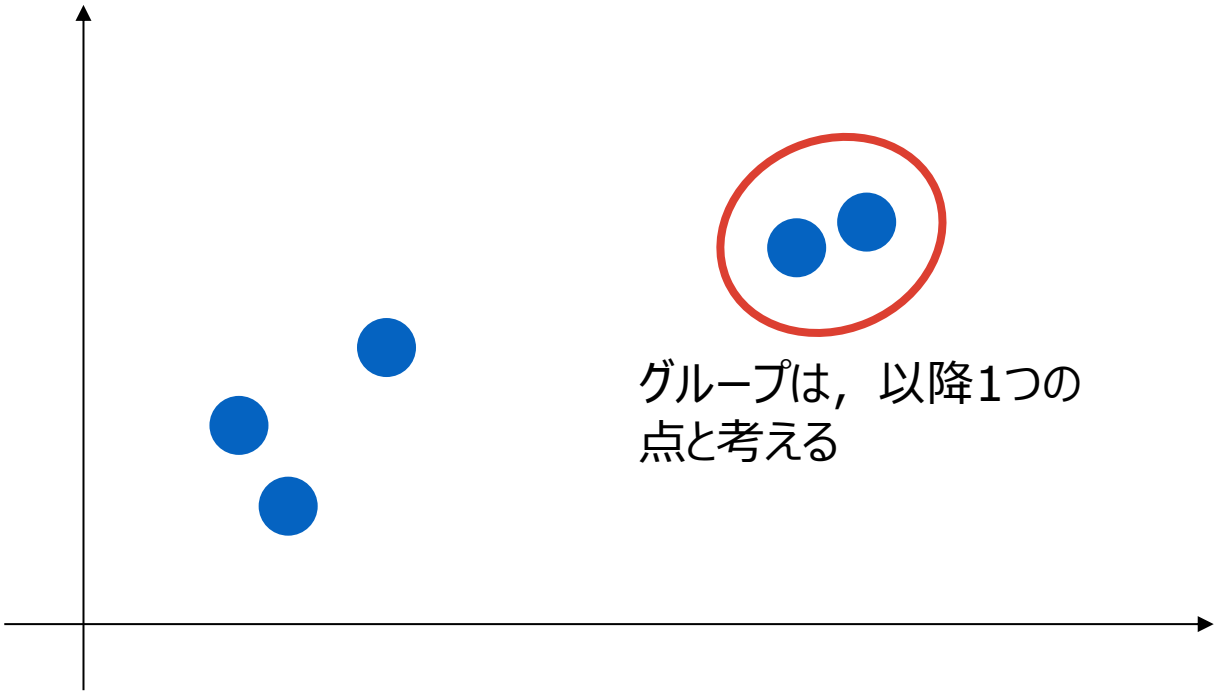
クラスタリングの例

- データをベクトルで表したところからスタート



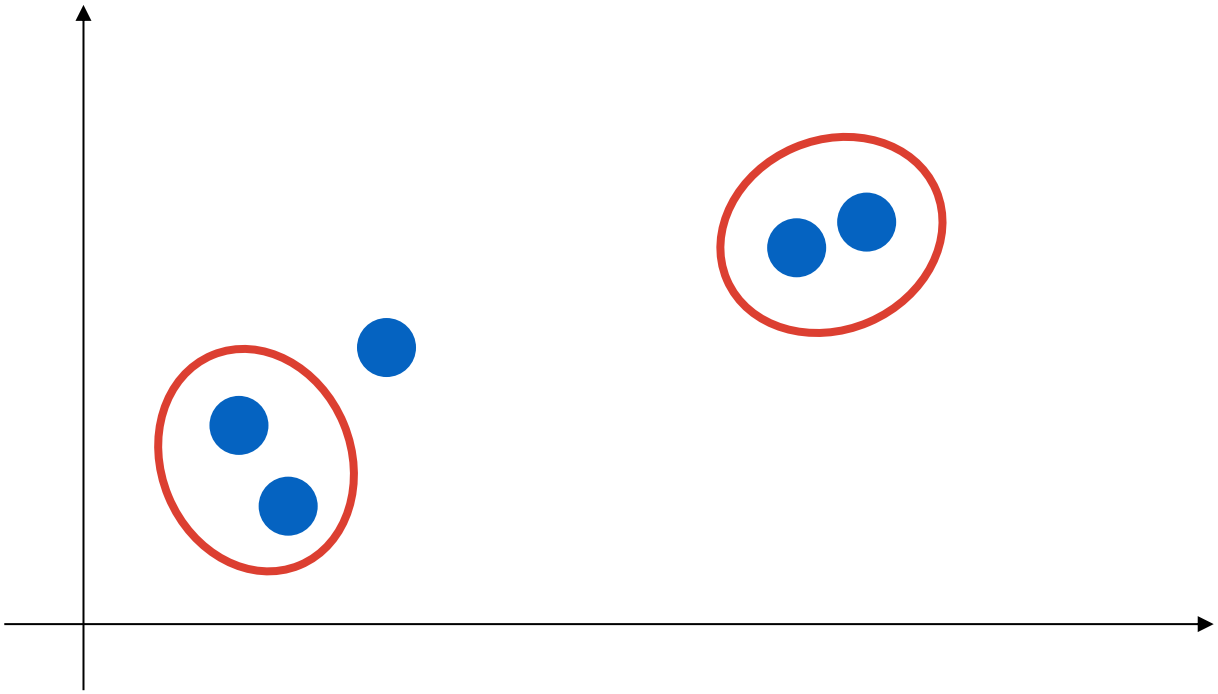
クラスタリングの例

- 全データの中で一番近い（＝距離が最も小さい）2つをグループにまとめる



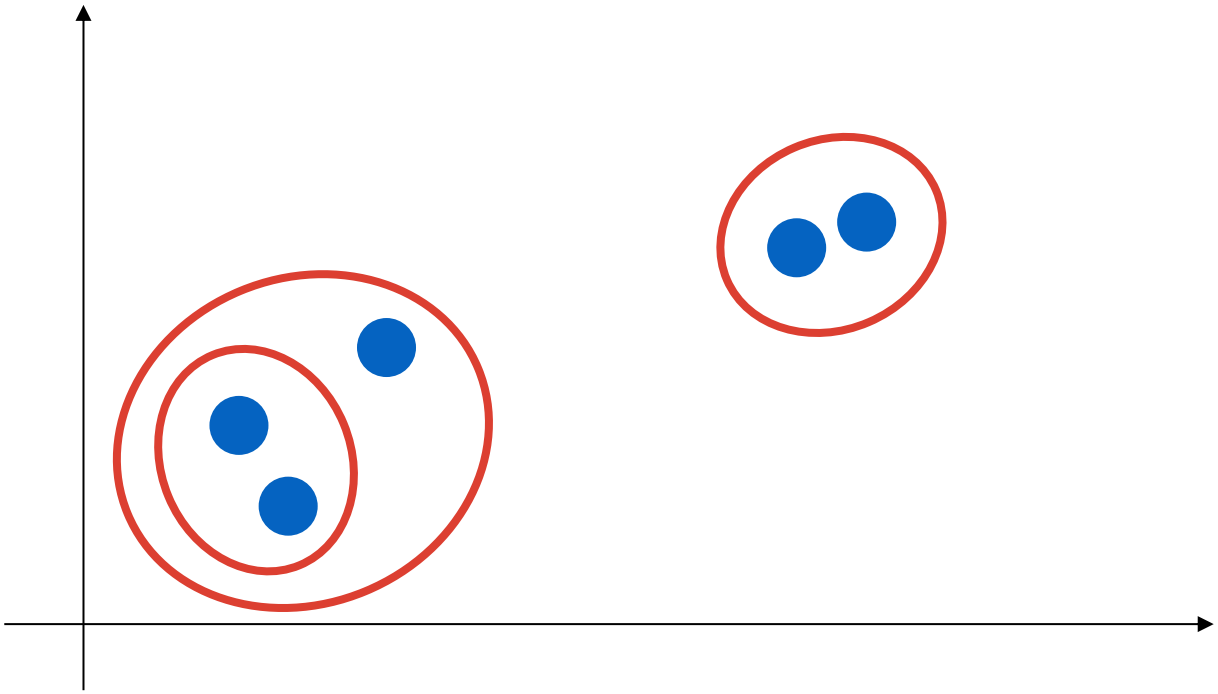
クラスタリングの例

- 次に最も近い2つをまとめる



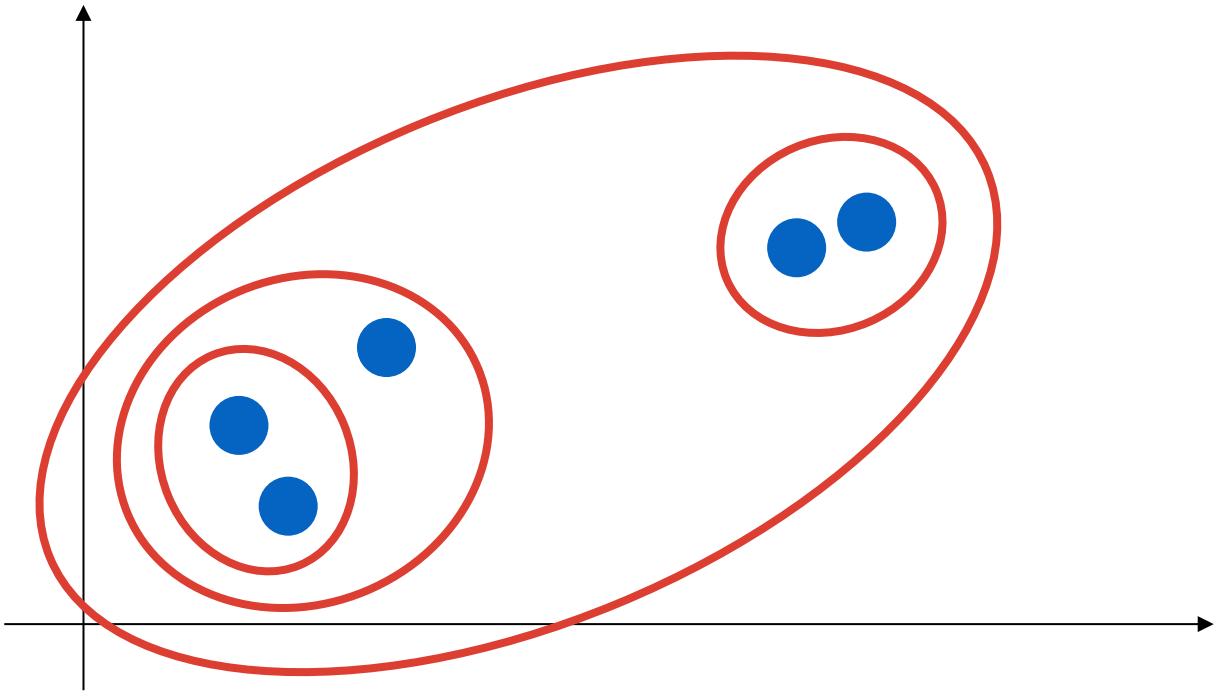
クラスタリングの例

- 今度はグループと点が1つのグループにまとめられる



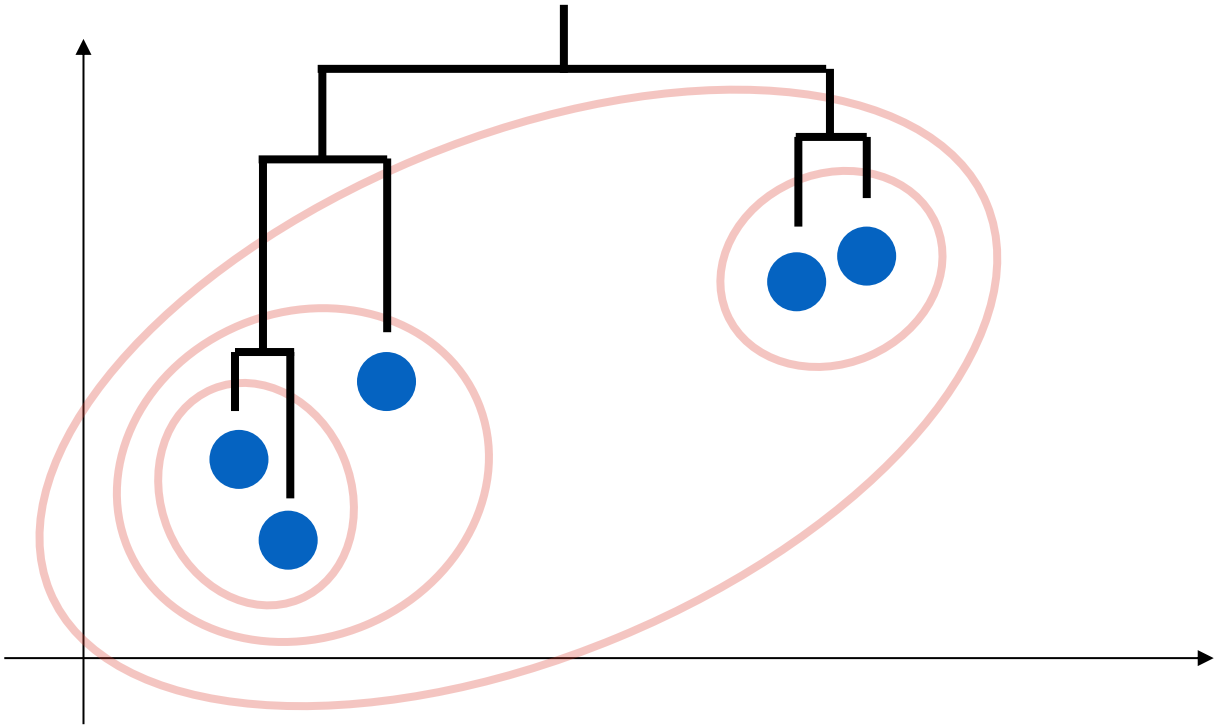
クラスタリングの例

- 全ての点が1つになるまで繰り返す



クラスタリングの例

- さらに系統樹を作ることもできる



他の有名なクラスタリング法であるk-means法→付録

まとめ

- ベクトル
  - データの代表的な表現方法の1つ
  - 何がどのくらい強い/あるを数学的に表現
  - ベクトルでの表現方法は対象によって変わってくる
- 距離と類似度
  - データの近さを測る方法
  - 対象や用途により様々な方法があり使い分ける
  - 分類（認識）やクラスタリングなどにも利用できる

付録

k-means法

Mean = 平均．だからk個の平均を求める方法．
古典的だが，いまでもクラスタリングの代表的方法（便利！）

k-means法（0）初期代表ベクトル

わかりやすい比喩：
代表＝アイドル，
データ＝民衆

k-means法（1）データの分割

わかりやすい比喩：
民衆は自分に一番近い
アイドルのファンになる

代表ベクトルの
ファンクラブ
（支持者集合）

注：用いる距離によって分割は変わる

k-means法（2）代表ベクトル更新

わかりやすい比喩：
けなげなアイドルが
ファンクラブの真ん中に移動

この中のデータの
平均に移動

k-means法（1）データ分割

わかりやすい比喩：
アイドルの移動により，
ファンクラブ構造が変ってしまう

k-means法（2）代表ベクトル更新

わかりやすい比喩：
けなげなアイドルは
新しいファンクラブの
真ん中に再び移動

以後，代表ベクトルが
動かなくなるまで反復

異常検出

距離さえあれば，異常なデータを見つけることができる！

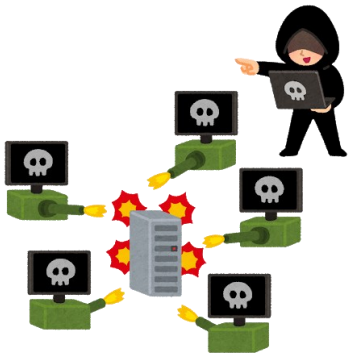
異常検出（異常検知）とは？

- 今与えられたデータが「一般的に期待していたデータ」とは異なるものであることを見出す手法

- なぜ異常は起こるか？
  - 機器やセンサの故障，身体の病気やケガ
  - うっかりや見落とし，事故や失敗など，人為的ミス
  - 侵入や破壊，悪用など，意図的な悪意のある行為
  - 甚大な自然災害など，想定外もしくは稀な現象の発生
  - Etc…

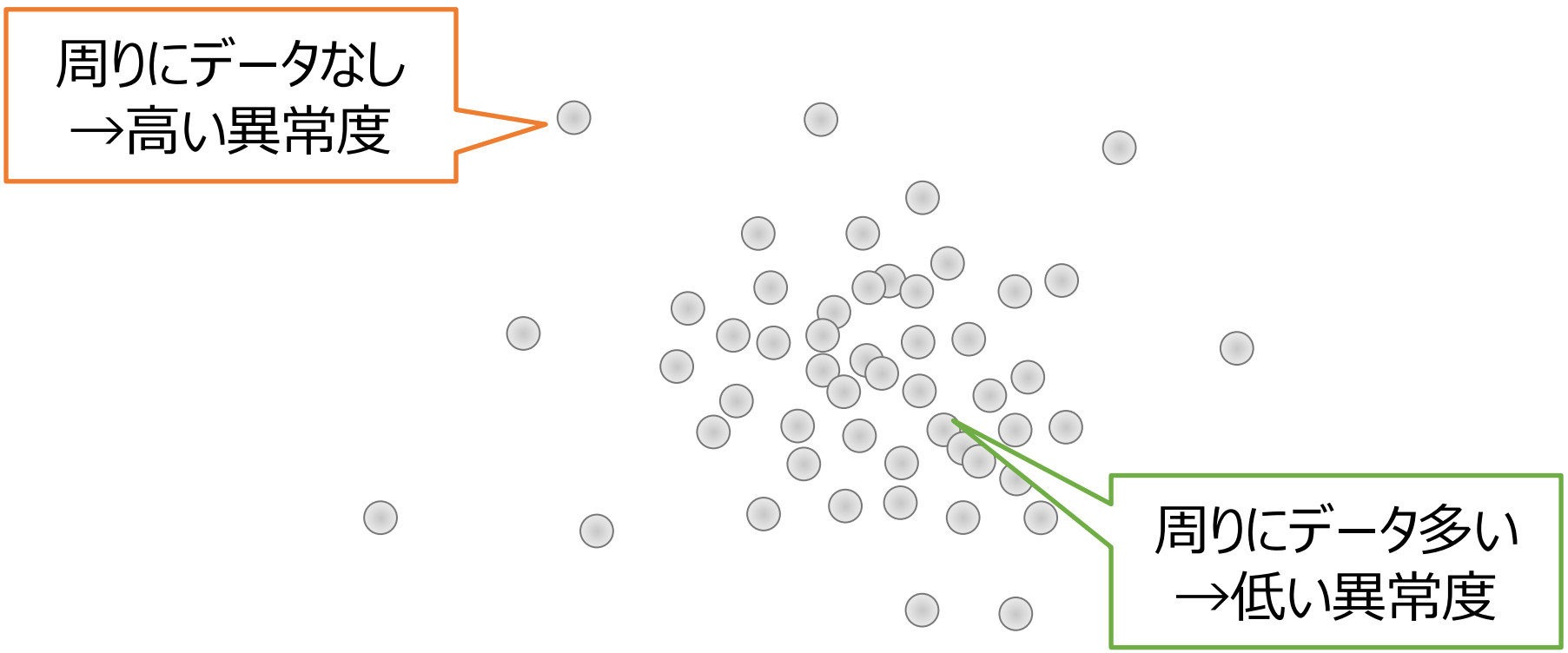
異常検出の応用例

- 人々を対象とした異常検出
  - エレベータ内のカメラで人々の動きを認識し，さらに普通でない動きを判断
  - 独居者の異常検出：日常と異なる行動パターンを通知し通知
  - 病院内での患者の異常検出：特に気づきにくい早期の異常検出
- 食品や生産物の異常検出
  - 表面の傷の検出，異物の検出
- 機械・建造物・コンピュータシステムの異常検出
  - サイバー攻撃の検出
  - 機械の故障の検出
  - 橋や道の異常検出



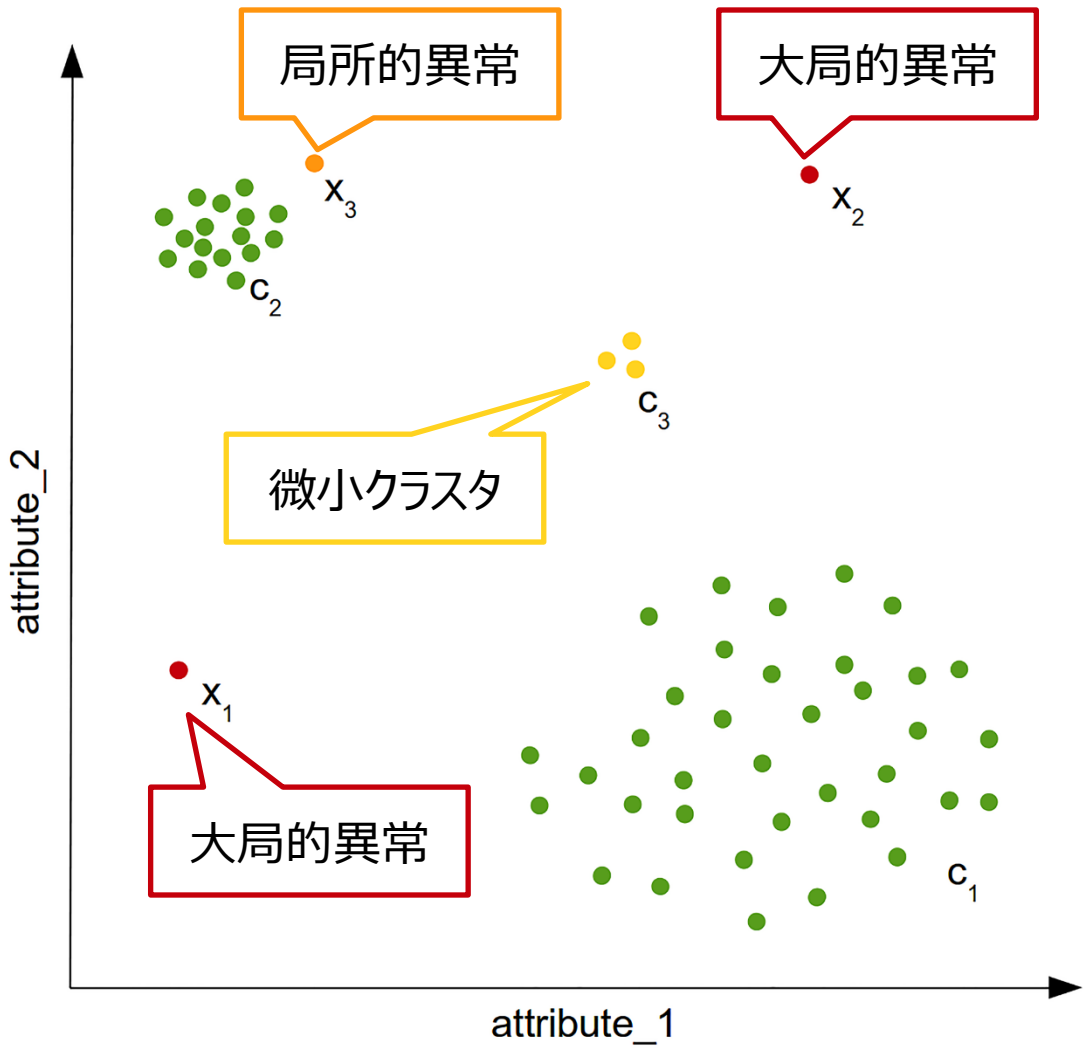
距離による異常検出の基本的な考え方

- 「注目しているデータが，他のデータから離れている(距離が遠い)」→異常度が高い



異常の種類(1/3)

- 大局的異常
  - まわりに「近い」データがない
    「似た」データがない
    ＝どう考えても異常
  - x_1, x_2

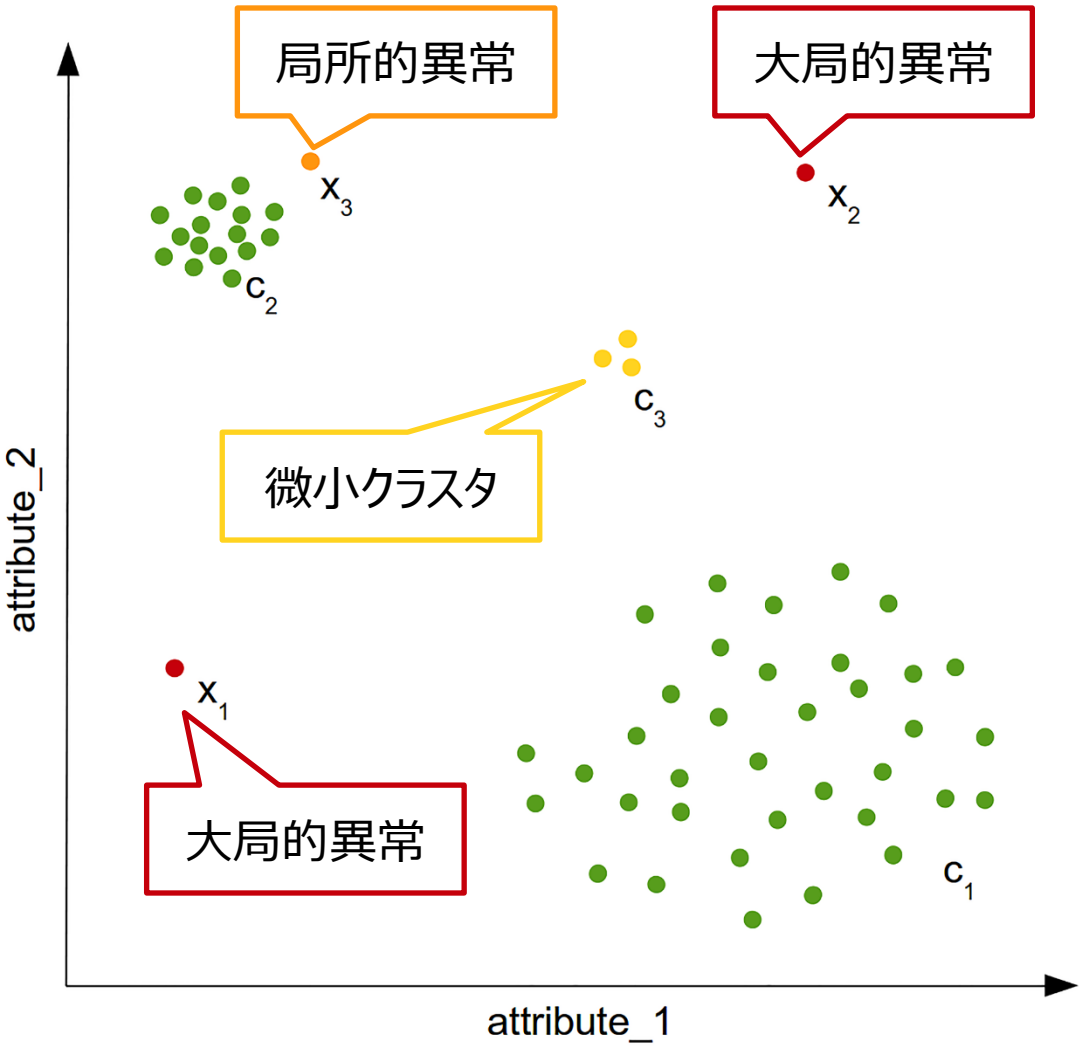


[Goldstein, Uchida, PLoSONE, 2016]

異常の種類(2/3)

- 局所的異常
  - そのデータ周辺の平均的な近さで考えると，「近くない」
  - x_3は局所的異常
    - 付近にあるc_2グループ基準でみると異常
    - ただしc_1グループ基準だとそこまで遠くはない

わかりやすい比喩：
その家がある都会では
「町はずれの一軒家」だが，
田舎視点だと十分ご近所

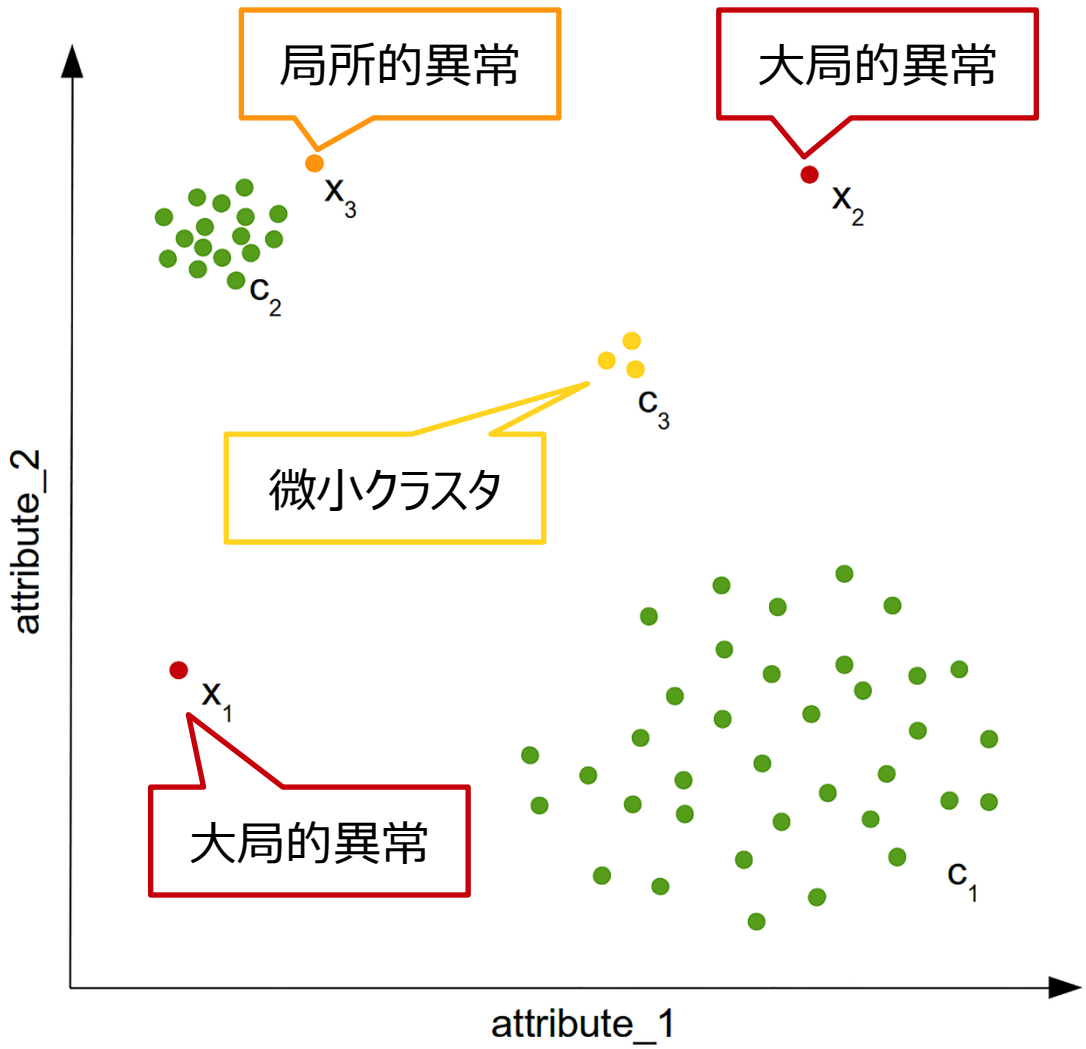


[Goldstein, Uchida, PLoSONE, 2016]

異常の種類(3/3)

- 微小クラスタ
  - 周りに近いデータはある ものの，その数は限定的
  - c_3 の3つのデータが相当

わかりやすい比喩：
要は「オタク」集団(?).
近くに同志は少数いるが，
世間全体からは浮いている



[Goldstein, Uchida, PLoSONE, 2016]

k近傍法による異常検出(1/2)
原理

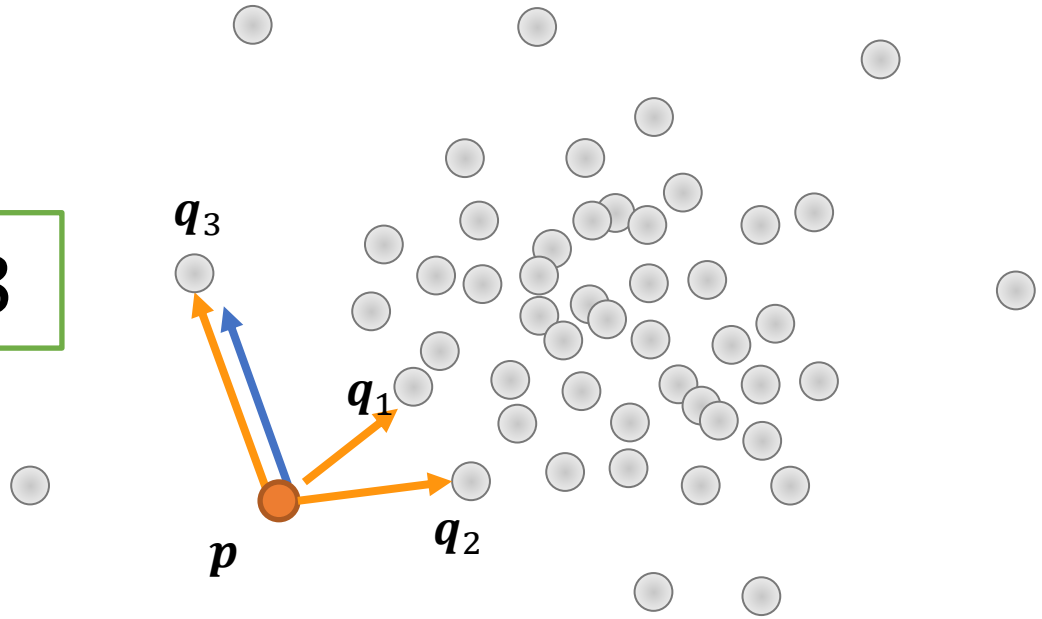
k個の近いデータ

距離登場

- あるデータについて，それに最も近いk個のデータですらあまり近くない→異常

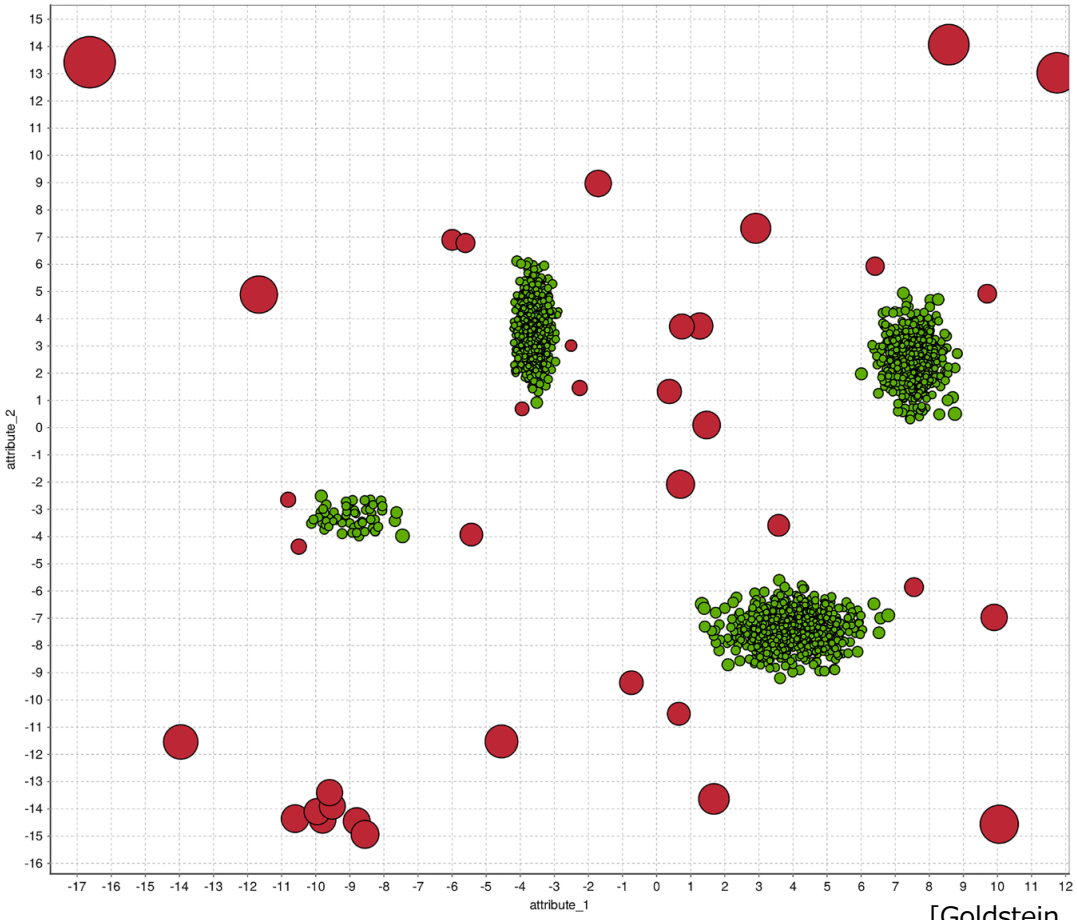
- 二つの考え方
  - 単一タイプ：k番目に近いデータへの距離 $\|\boldsymbol{p}-\boldsymbol{q}_k\|$ or
  - 合計タイプ：$\|\boldsymbol{p}-\boldsymbol{q}_1\|+\|\boldsymbol{p}-\boldsymbol{q}_2\|+\cdots+\|\boldsymbol{p}-\boldsymbol{q}_k\|$

$k=3$



k近傍法による異常検出(2/2)
異常度計算結果例($k = 10$, 合計)

- 半径が大きな点(データ)ほど異常度が高いと計算されている



[Goldstein, Uchida, PLoSONE, 2016]