

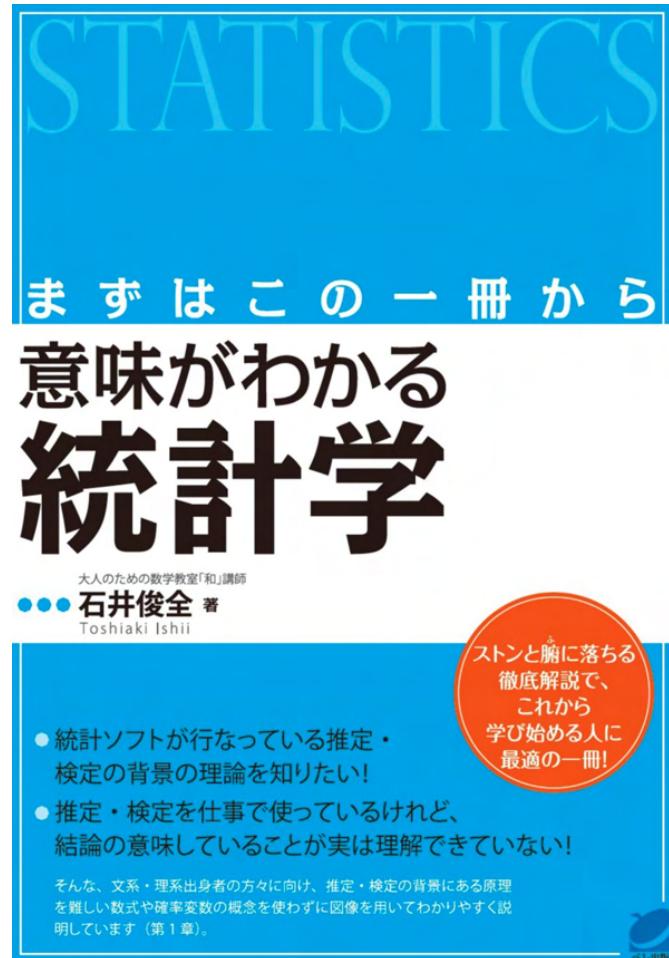
K490: データサイエンス論

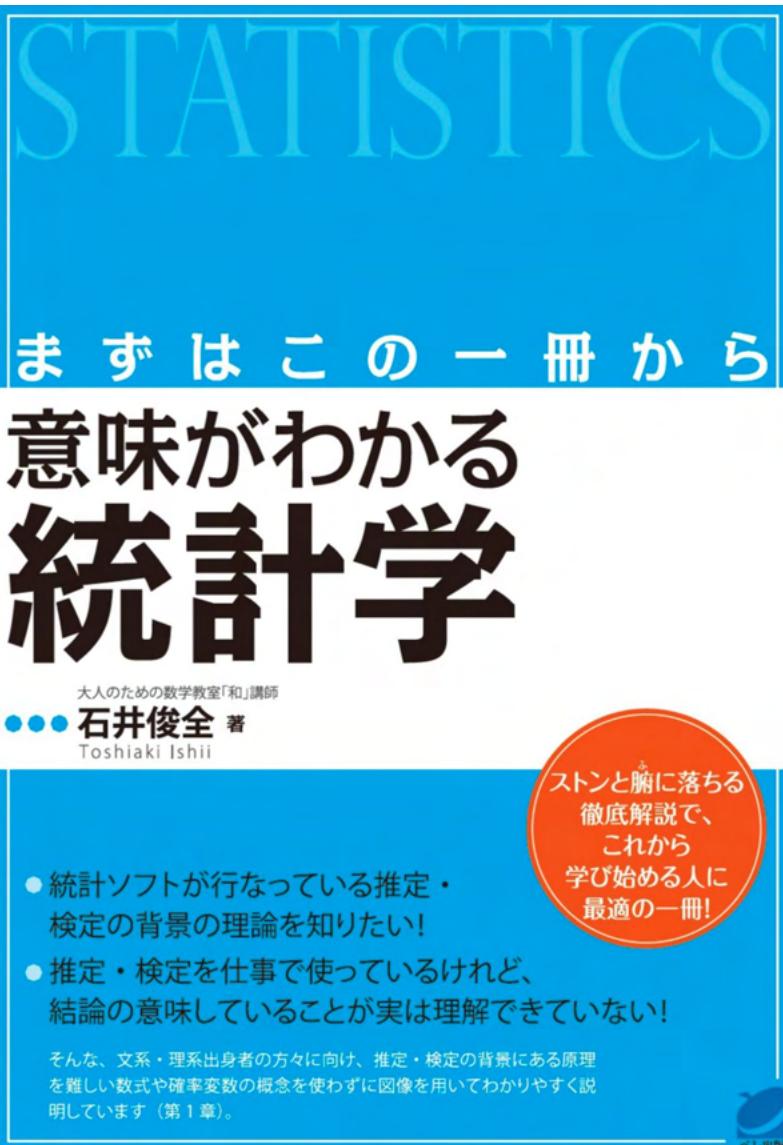
Lecturer: Hieu-Chi Dam, Takashi Isogai

講義予定

| | | |
|-----------------------------|--------------|------|
| 1. データサイエンスの紹介 | (2022年1月30日) | (ダム) |
| 2. データマイニングによる知識発見プロセス | (2022年1月30日) | (ダム) |
| 3. 確証的データ解析と探索的データ解析 | (2022年1月31日) | (ダム) |
| 4. 基礎的なデータ解析手法（1）：单变量解析 | (2022年1月31日) | (ダム) |
| 5. 基礎的なデータ解析手法（2）：多变量解析 | (2022年2月01日) | (ダム) |
| 6. 予測的データ解析手法（1）：決定木 | (2022年2月01日) | (ダム) |
| 7. 予測的データ解析手法（2）：ベイジアン分類 | (2022年2月02日) | (磯貝) |
| 8. 予測的データ解析手法（3）：サポートベクトル分類 | (2022年2月02日) | (磯貝) |
| 9. 記述的データ解析手法（1）：クラスタリング | (2022年2月03日) | (磯貝) |
| 10. 記述的データ解析手法（2）：特徴選択と次元削減 | (2022年2月03日) | (磯貝) |
| 11. 記述的データ解析手法（3）：グラフの解析 | (2022年2月04日) | (磯貝) |
| 12. データ科学と倫理問題 | (2022年2月04日) | (磯貝) |
| 13. 学生発表（1） | (2022年2月04日) | (ダム) |
| 14. 学生発表（2） | (2022年2月04日) | (ダム) |

まず読んでほしい入門書





| |
|--|
| <p>【まずはこの一冊から 意味がわかる統計学】</p> <p>もくじ</p> <p>はじめに 3 もくじ 8 この本の特徴 13</p> <p>第1章</p> <p>1 相対度数分布グラフ 19 1-1 資料をグラフに整理しよう 19 —度数分布表、ヒストグラム コラム データの代表値 24</p> <p>2 平均、分散・標準偏差 26 2-1 資料を特徴付ける3大指標 26 —平均、分散・標準偏差 コラム 分散と平均を結ぶ公式 35</p> <p>2-2 平均、分散・標準偏差をヒストグラムで実感しよう 38 —平均、分散・標準偏差の意味するところ</p> <p>2-3 資料の数値に手を加えると資料の平均、分散は? 44 —平行移動、定数倍のときの平均、分散</p> <p>6 検定の考え方 110 6-1 部分から全体の特徴を判定しよう 110 —検定の考え方 コラム 背理法と検定 123</p> <p>第2章</p> <p>1 確率変数 128 1-1 確率的にいろいろな数値をとるものを表すための記号 128 —確率変数 X 1-2 確率分布でも相対度数分布グラフが描ける! 132 —確率変数 X の相対度数分布グラフ</p> <p>1-3 連続する数値に対しても確率変数が定められる 136 —連続型確率変数</p> <p>1-4 確率変数 X について平均、分散を求めてみよう 145 —確率変数 X の平均、分散</p> <p>1-5 連続型確率変数でも平均、分散はある! 154 —連続型確率変数の平均、分散</p> <p>1-6 $X + a$、bX、$X + Y$の平均、分散は? 162 —確率変数 X の $E(X)$、$V(X)$についての公式</p> <p>【まずはこの一冊から 意味がわかる統計学】</p> <p>もくじ</p> <p>はじめに 3 もくじ 8 この本の特徴 13</p> <p>第3章</p> <p>3 サンプル X の相対度数分布グラフ 51 3-1 資料からとり出してグラフを描く 51 —サンプル X の相対度数分布グラフ コラム 復元抽出と非復元抽出 59</p> <p>3-2 資料の数値に手を加えると平均、分散は? 63 —$X + a$、bX の平均、分散</p> <p>4 正規分布 70 4-1 統計解析のなかで一番重要な度数分布 70 —正規分布 コラム 偏差値 79</p> <p>4-2 正規分布が持っているすばらしい特徴 82 —再生性、中心極限定理</p> <p>5 推定の考え方 88 5-1 部分を見て、全体の様子を知る方法 88 —推定の考え方</p> <p>5-2 ピンポイントでズバリ当てましょう 91 —点推定</p> <p>5-3 幅を持たせて予想しよう 95 —区間推定</p> <p>2 二項分布 176 2-1 組合せの個数を表す記号 C 176 —コンビネーションと道筋</p> <p>2-2 5回投げたコインのうち、3回が表である確率 187 —二項分布の確率</p> <p>2-3 二項分布の平均、分散は簡単に計算できる! 191 —二項分布の平均と分散</p> <p>2-4 二項分布の極限が正規分布だ! 195 —ラプラスの定理</p> <p>3 推定の応用 203 3-1 母平均、母分散、母比率を推定する 203 —いろいろな推定 コラム 標本分散と不偏分散 222</p> <p>4 検定の応用 226 4-1 母平均、母分散を検定しよう 226 —いろいろな検定</p> <p>4-2 2つの母集団の母分散・母平均は等しいか 237 —2つの母集団に関する検定</p> <p>4-3 母比率を検定しよう 252 —適合度・独立性の検定</p> |
|--|

K490: データサイエンス論

Lecture 1: データサイエンスの紹介

Lecturer: Hieu-Chi Dam, Takashi Isogai

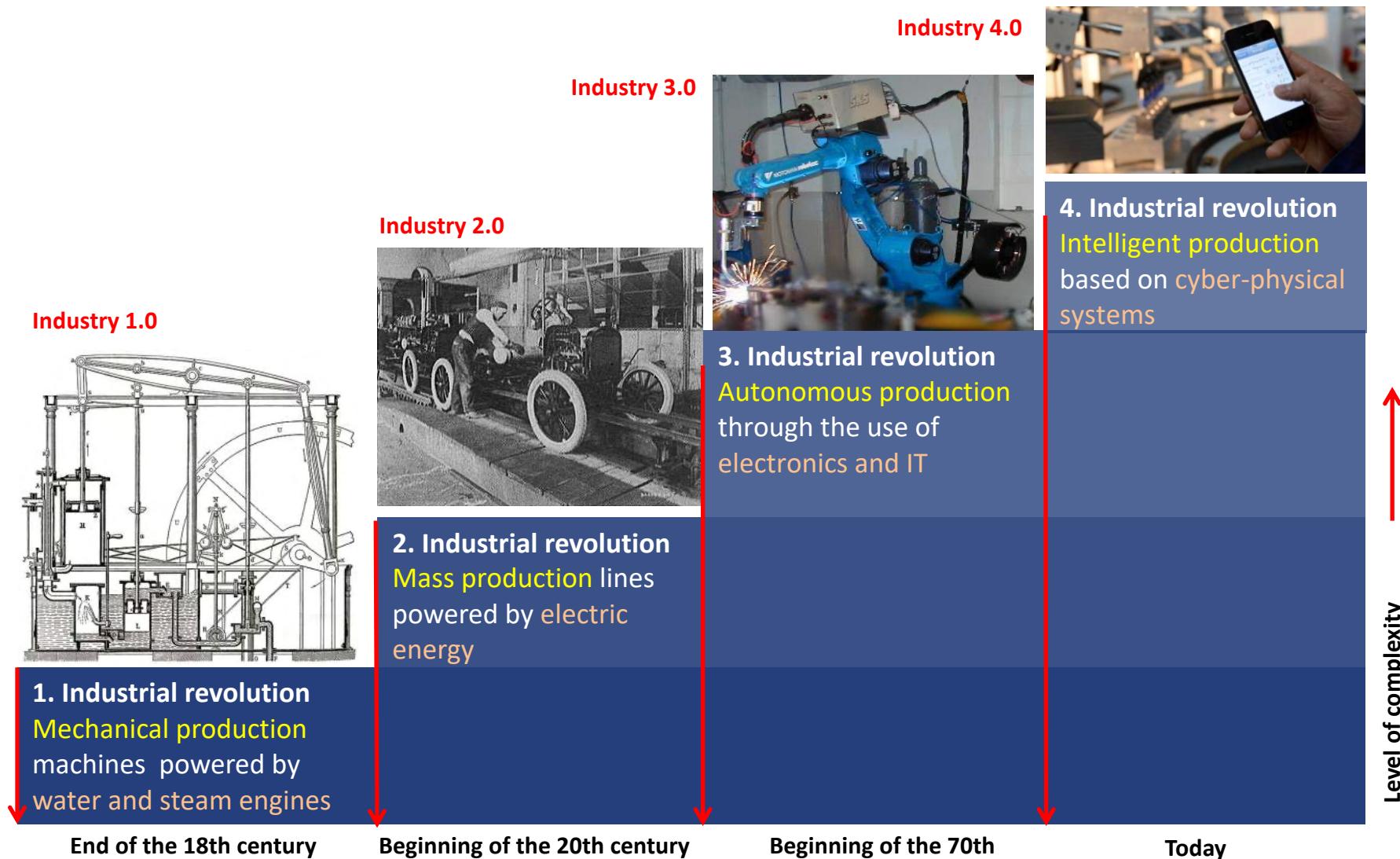
Fourth Industrial Revolution?

Characteristics of an industrial revolution

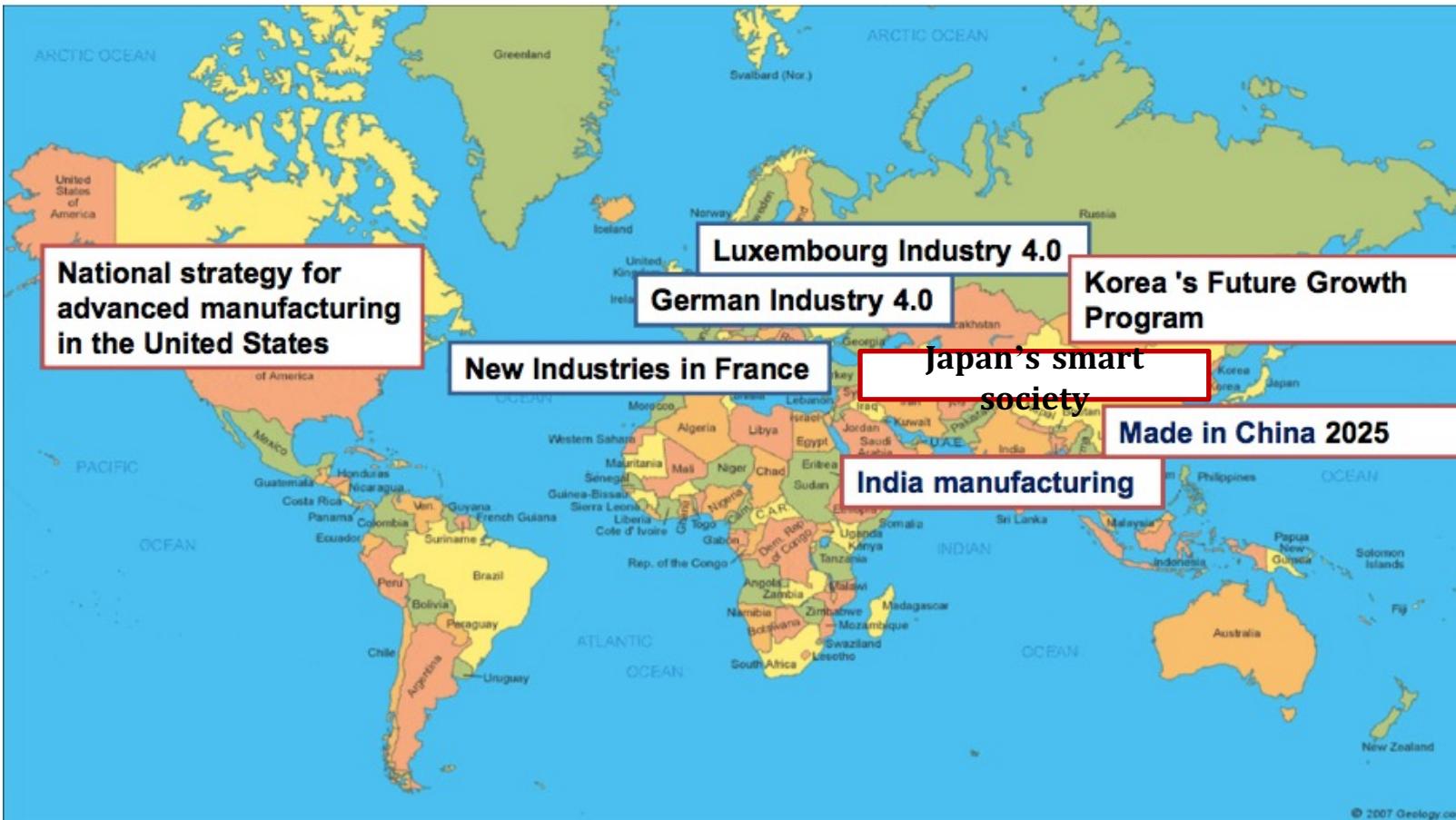
- Science and technology breakthroughs
- Power essential changes in production

Intelligent production based on information technology, biotechnology, nanotechnology with background from **breakthroughs in digital technologies in cyber-physical systems.**

The fourth industrial revolution



National strategies



Klaus Schwab (WEF), The Fourth Industrial Revolution

Alistair Nolan (OECD), Enabling the Next Production Revolution: Implications for Policy, Hanoi, 12.2016

Digitalization and cyber-physical systems

- ‘Digital version’ of objects:
Representing objects by ‘0’ and
‘1’ on computers (digitalization)
 - Ex: cars, electronic medical
records...
 - **Cyber-physical systems:**
Connection between objects by
their digital versions.



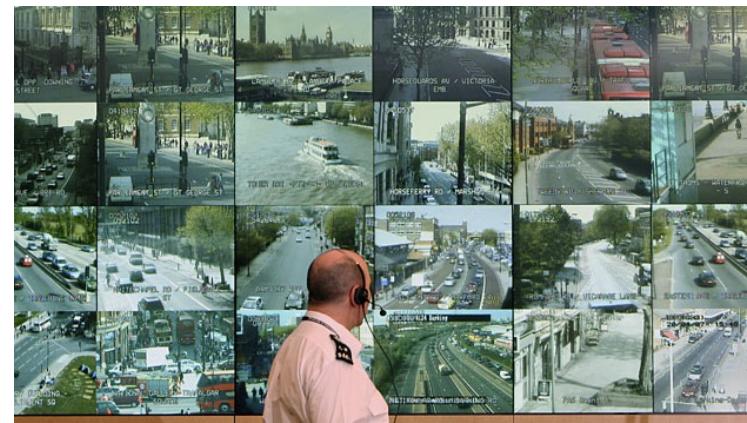
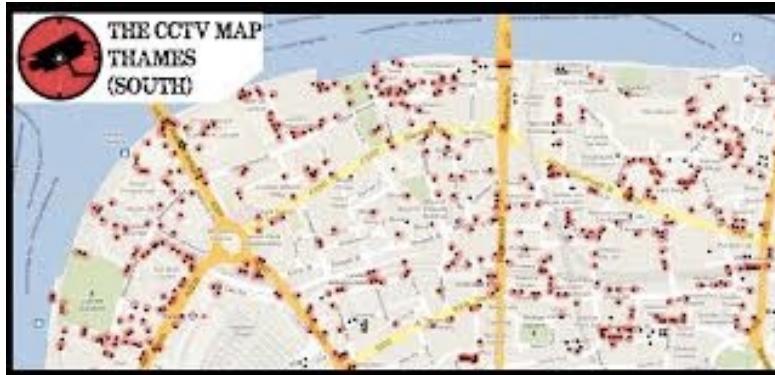
Productions in physical world of objects



Computation and control in cyber space

London CCTV (Closed circuit TV)

- 500 million of British pound (video surveillance)
- Providing 95% of information about crimes



Big data

Big data refers to data sets that are **too large** and **too complex** to manage and analyze with traditional IT techniques.

Variety:

Complexity of data in many different structures, ranging from relational, to logs, to raw text

Velocity:

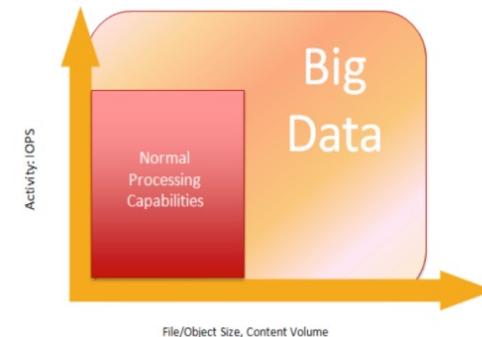
Streaming data and large volume data movement

Volume:

Scale from Terabytes to Petabytes (10^{15} bytes) to Zetabytes (10^{18} bytes)

Veracity:

Accuracy and precision, truthfulness of the data.



Data Scientist: The Sexiest Job of the 21st Century

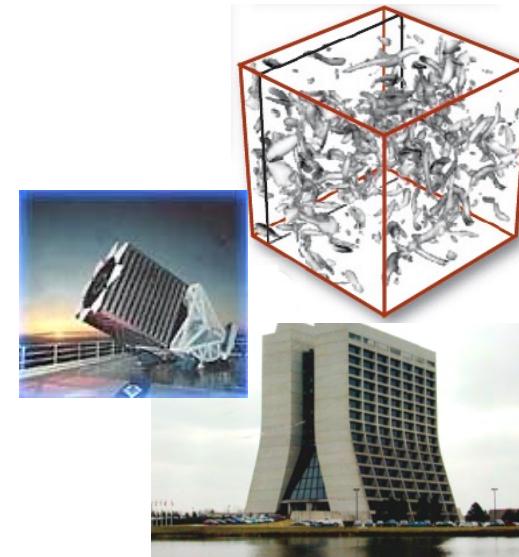
(Harvard Business Review, October 2012)

Science paradigm shifts

- Thousand years ago: science was **empirical**
Describing natural phenomena
- Last few hundred years: **Theoretical branch**
Using models, generalizations
- Last few decades: **Computational** branch
Simulating complex phenomena
- Today: **Data exploration** (e-Science)
Unify theory, experiment, and simulation
 - Data captured by instruments or generated by simulator
 - Processed by software
 - Information/knowledge stored in computer
 - Scientist analyzes databases/files using data management and statistics.

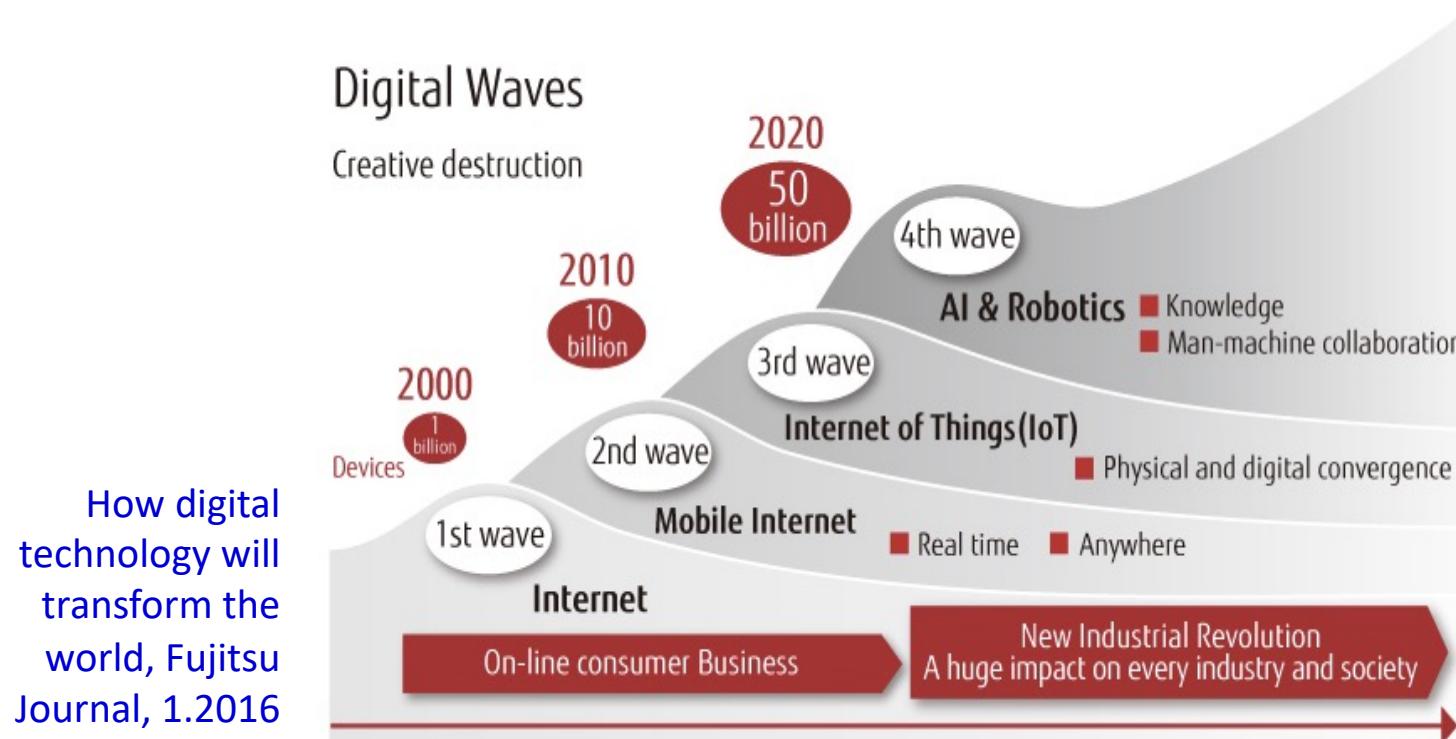


$$\left(\frac{\dot{a}}{a}\right)^2 = \frac{4\pi G p}{3} - K \frac{c^2}{a^2}$$

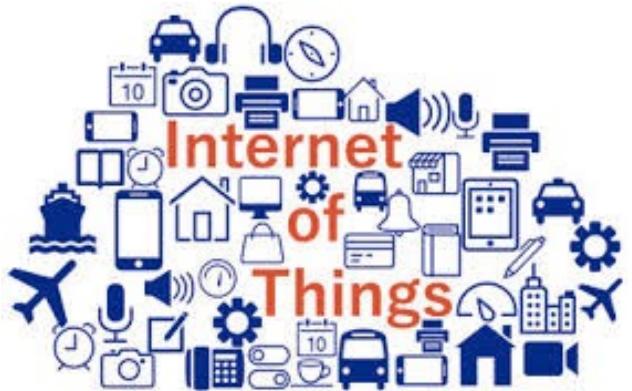


Digital technology

- Digitalization (camera, television, printing...)
- Processing digitalized data



Breakthroughs of digital technology



Some definitions of data science

- There is not yet a definition agreed by all.
- Some examples:

NIST

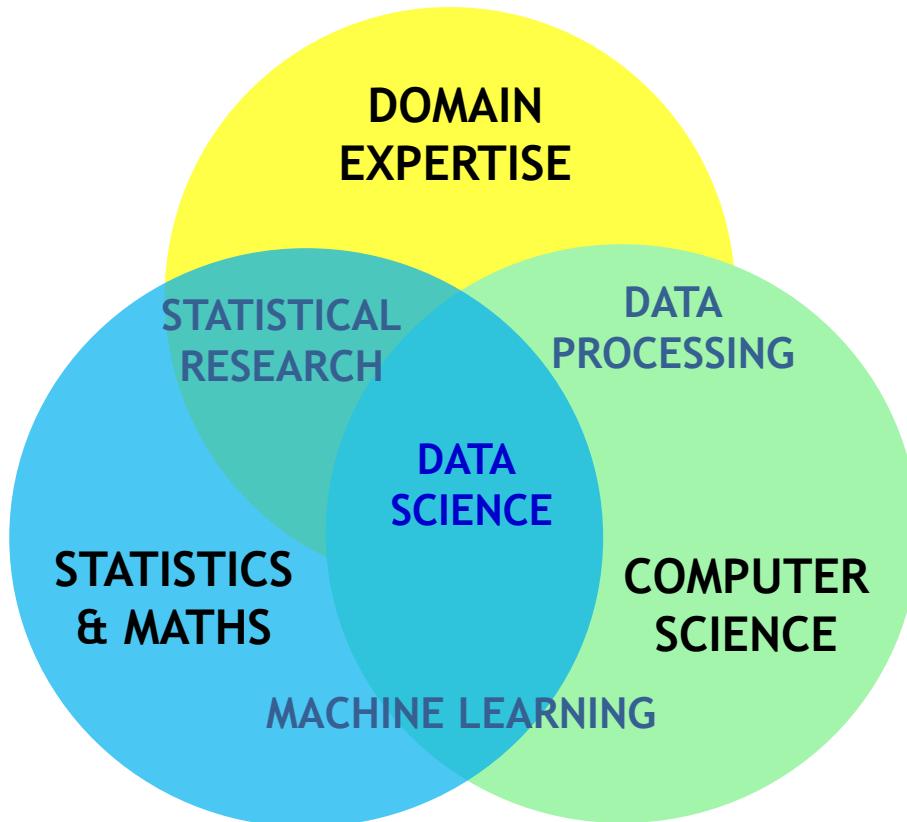
(National Institute
of Standards and
Technology)

**Data science is extraction of
actionable knowledge directly from
data through a process of discovery,
hypothesis, and hypothesis testing.**

Microsoft

**Data science is about using data to
make decisions that drive actions.**

Data science



**Data Scientist: The Sexiest
Job of the 21st Century**
**(Harvard Business Review, October
2012)**

What do we want from the data? そのデータから何を得たいのか？

- Want to exploit the data, to extract new and useful information/knowledge in the data, such as
データを探索し、例えば以下のような有用な新情報/知識を得たい



Which customers are thinking of leaving?



Which transactions are fraudulent?



Which new product has the greatest chance of success?



How can I extract insight from all of my information?

- In short, we want to “understand” and “use” the data, e.g., using data to make right decisions.

How to understand and use data?

1. Understand how people collect and manage data?

どのように人々がデータを集め、整理するのかを理解する

2. Understand the nature of data (data types, data quality)?

データの特徴（データ種類、データの特性）を理解する

3. Know different tasks when analyzing a dataset?

分析する時のタスクを知る

4. Know methods of analyzing data for given tasks?

与えられたタスクに適切なデータ分析手法について知っている

5. Know using software to analyze the data?

分析ソフトウェアについて知っている

6. Know to explain the result of data analysis?

分析から得られた結果について説明できる

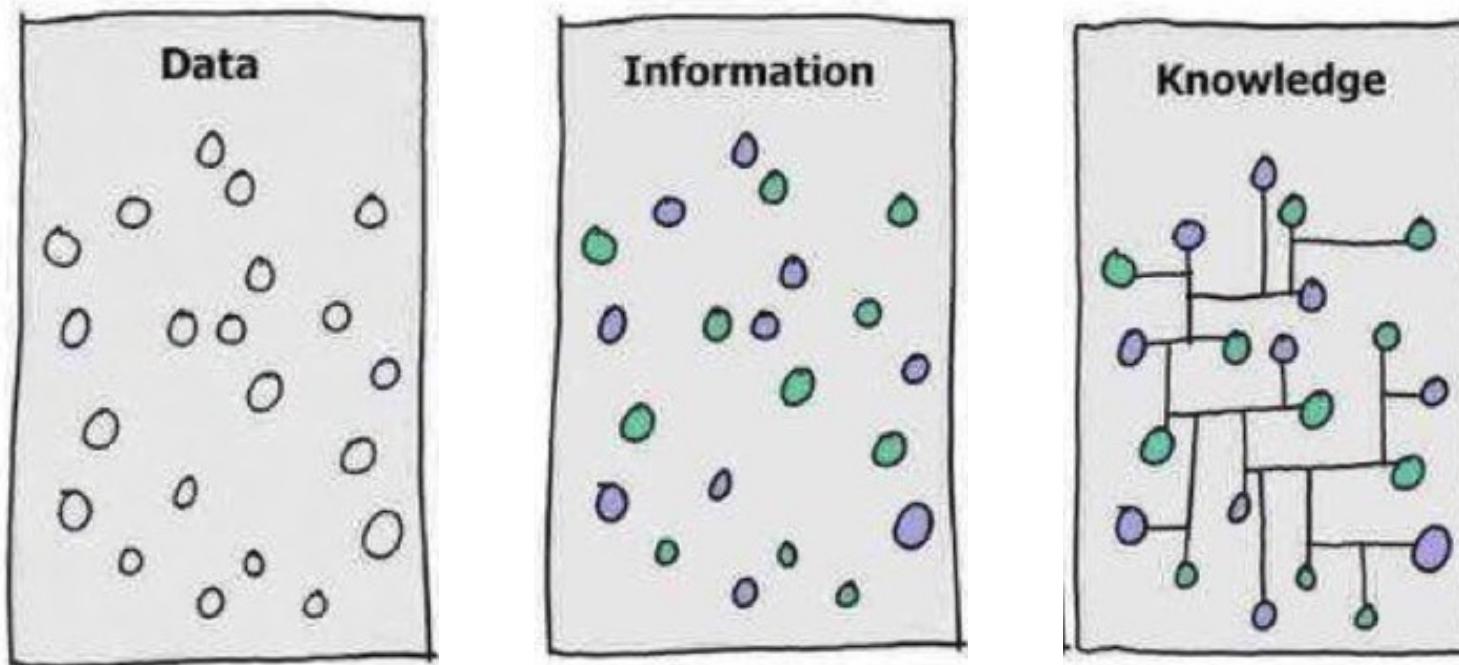


Buzzwords

- Data analytics is the study of the **generalizable extraction of knowledge from data**, used interchangeably with **data science**. Recently, the term data science is more often used.
(データ解析学はデータから知識の獲得を一般化された学問である)
- Data analytics need not be always for **big data**, however, the fact that data is scaling up makes **big data analytics** an important aspect of data analytics.
(データ解析学は必ずしもビッグデータのためではないが、ビッグデータ解析はデータ解析の重要な一側面である)
- A practitioner of data science is called a **data scientist** who solves complex data problems through employing deep expertise in some discipline.



Data, information, knowledge



From Julien Blin

Data, information, knowledge

データ・情報・知識

- **Data** is often seen as a string of bits, or numbers and symbols, or “objects” which we **collect daily** (by observation, measurements, collection, etc.).
データとは観察・測定などで得られる、単語や数字などの集まり。
- **Information** is data stripped of redundancy and reduced to the minimum necessary to **characterize the data**. 情報とはデータから冗長を取り除き特徴づけたもの。
- **Knowledge** is integrated information, including facts and their relations, which have been **perceived, discovered, or learned** as our “mental pictures” (“justified true belief”).
知識とは原因や関係性などを統合した意味づけのある情報で発見や学習などを行います。
- Knowledge can be considered data at a high level of abstraction and generalization. 知識はデータの高レベルでの抽象化と一般化と考えられます。

How does people collect data?

- Observing, measuring, or collecting the **values of features** (features, attributes, properties, variables) of the **objects** under consideration.
検証する対象の意義のある特徴（特徴・属性・特性・変数）の観察・計測・収集
- Two ways of collecting data 2通りのデータ収集方法

Randomly sampling

ランダム収集

Conventional statistics, methods were created when small or medium-sized data sets were common.

小・中規模のデータサイズが一般的であった頃に、従来の統計学手法は開発されました

Collecting all available data

可能な限り全データの収集

Many innovative multivariate techniques being developed to solve large-scale data problems.

大規模データの問題を解くために、多くの革新的な多変量技術が発達しました

Statistics

- **Statistics** provides mathematical methods and techniques to analyze, generalize and decide from the data.
- Main (traditional) **content**
 - **Descriptive statistics:** Probability distribution...
 - **Inferential statistics:** Estimation and hypothesis testing
- Experimental and observational **data**
 - Statistical data usually collected to answer predetermined questions (experiment design, survey design)
 - Mostly numerical data, few symbolic data.
- Methods were developed before having computer and for **small datasets**, for analyzing a single random variable.

Multivariate analysis

- Simultaneously analyze the relationship of multiple random variables
- **Exploratory data analysis** (EDA) produces hypotheses from data vs. Testing hypothesis by data in **Confirmatory data analysis** (CDA)
 - Factor analysis, PCA, Linear discriminant analysis
 - Regression analysis
 - Cluster analysis
- What we can see from conventional methods?
 - Poor results on large and complex data
 - Traditional methods are suitable for analyzing small datasets.
 - Price of storage and data processing decrease quickly.

Multivariate analysis

- Analytical methods were created for datasets with small or middle size and when computers were still weak.
- Multivariate data analysis is quickly changing due to computational techniques that are fast and effective. Various methods were newly developed for dealing with large scale problems (Pagerank of Google works with matrices of billion dimensions).



November 20126: Cray XK7 Titan computer,
17,590 TFlops, 560640 processors.



November 2016: [Sunway TaihuLight](#)
93,014 TFlops, 10,649,600 cores

Machine learning and data mining

研究領域とデータマイニング

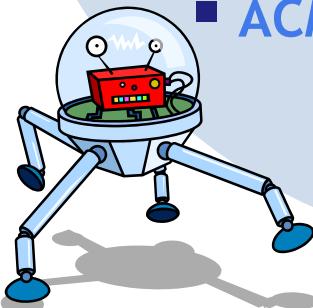


■ Machine learning 機械学習

To build computer systems that learn as well as human does (science of learning from data).

人間のように学習するコンピュータシステムを構築する。

- IJCAI since 1969 (IJCAI-16), AAAI since 1973, ECAI since 1988.
- ICML since 1982 (33th ICML in 2016), ECML since 1989.
- ECML/PKDD since 2001.
- **ACML** starts Nov. 2009.



■ Data mining データマイニング

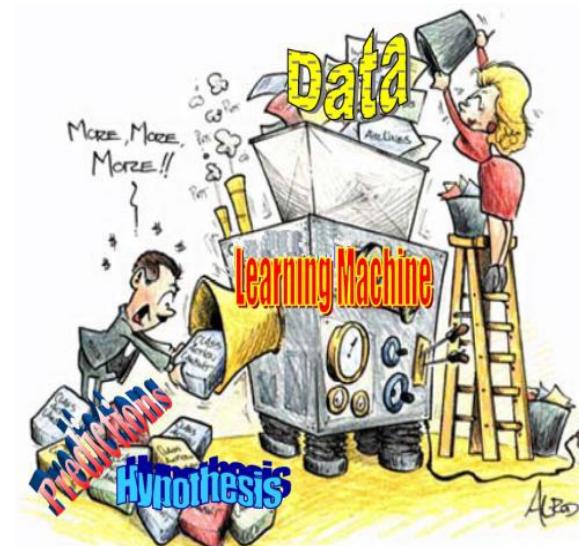
- To find new and useful knowledge from large datasets (data engineering).
大きなデータベースから新しく有用な知識を見つける.
- ACM SIGKDD since 1995, PKDD and **PAKDD** since 1997
IEEE ICDM and SIAM DM since 2000, etc.

Note: Difference between statistics, machine learning, data mining?

Machine learning

- The goal of machine learning is to build computer systems that can adapt and learn from their experience (Tom Dietterich, 1999).
- *Given*
 - $\{x_i\}$, x_i is description of an object in some space, $i = 1, 2, \dots, n$.
 - y_i is some property of x_i viewed as its label, $y_i \in \{C_1, C_2, \dots, C_K\}$ or $y_i \in \mathbb{R}$
 - $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$
- *Find*

Function $p(y|x)$ for label data
and $p(x)$ for unlabeled data



(Source: Eric Xing lecture notes)

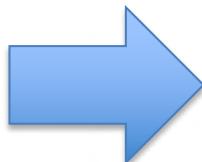
Branch of Artificial Intelligence: Reasoning, language understanding, learning

DEDUCTION [Given $f(x)$ and x_i , deduce $f(x_i)$] vs. *INDUCTION* [Given $\{x_i\}$, infer $f(x)$]

Statistics vs. Machine learning

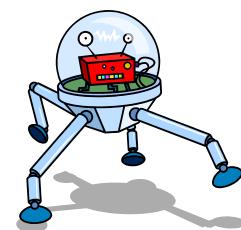
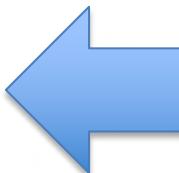
Statistics

- Emphasize on formal statistical inference (estimation, hypothesis testing).
- Based on models for problems with small size and mostly with numerical data
- Statistics is an established science, conservatively changing culture and adapt to computational power.
- Trend to move to machine learning.



Machine learning

- Emphasize on prediction problems in high dimensionality and with symbolic data.
 - In early days, the construction and use heuristics algorithms.
 - Tend to base more on statistics, build statistical models underlying the algorithms.



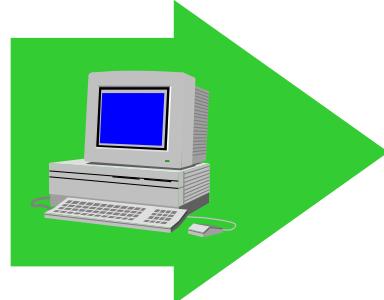
Knowledge Discovery and Data Mining

知識発見とデータマイニング

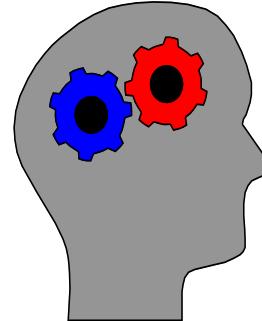
the **automatic extraction** of non-obvious, hidden **knowledge** from
large volumes of data 大量データに潜在する未発見の知識の自動抽出



データセット全体の把握・
コンピュータメモリへの展開が
困難な**大規模データベース**

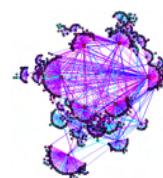
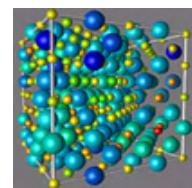
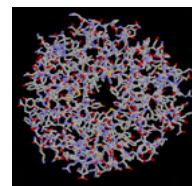


どのデータマイニング
アルゴリズムを適用するか？

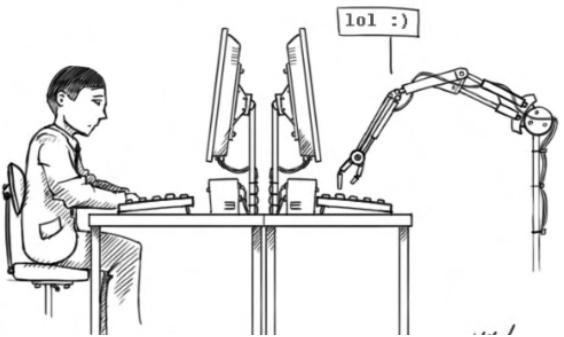


どんな知識か？
どう表現するか？

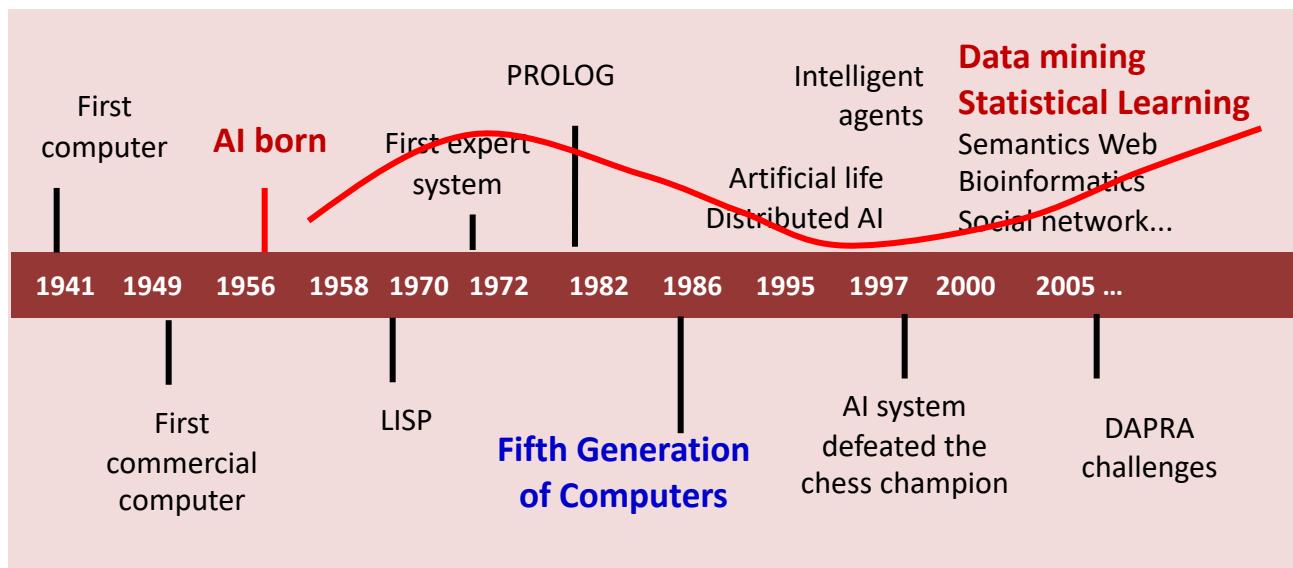
Focus on complex data



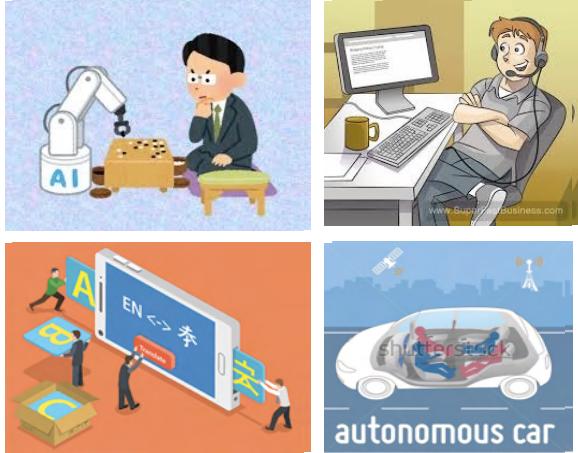
Artificial Intelligence



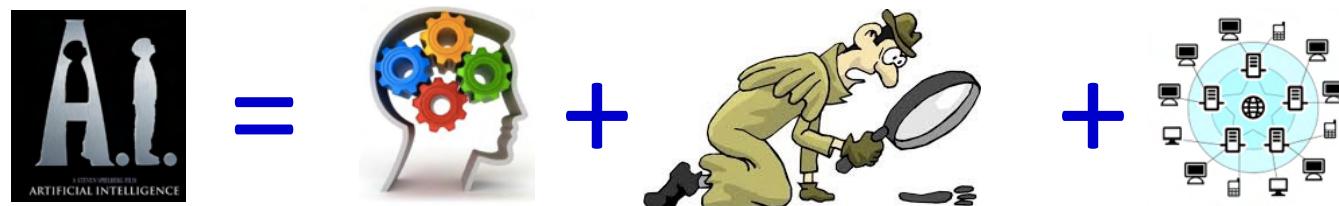
- Produce computers with some human intelligence (reasoning, language understanding, learning).
- Turing test is a way to answer the question ‘can computer think?’



Artificial Intelligence



- Produce computers with some human intelligence (reasoning, language understanding, learning).
- AlphaGo, speech recognition, cancer prediction, autonomous vehicle...
- Most recent success of AI is based on machine learning.



Main conferences: IJCAI, AAAI, ECAI, PRICAI

AI and machine learning

“Many developers of AI systems now recognize that, for many applications, it can be far easier to train a system by showing it examples of desired input-output behavior than to program it manually by anticipating the desired response for all possible inputs”.

(M.I. Jordan & T. Mitchell, *Science*, 7.2015)

Principles of data science?

Principle = a basic idea or rule that explains or controls how something happens or works (Cambridge Dict.)



1. Data type and structure
2. Process
3. Methods
4. Model selection

Data types and structure vs. methods

Data types and structures

- Flat data tables
- Relational databases
- Temporal & spatial data
- Transactional databases
- Multimedia data
- Genome databases
- Materials science data
- Textual data
- Web data
- etc.



Mining tasks and methods

- **Classification/Prediction**
 - Decision trees
 - Bayesian classification
 - Neural networks
 - Rule induction
 - Support vector machines
 - Hidden Markov Model
 - etc.
- **Description**
 - Association analysis
 - Clustering
 - Summarization
 - etc.



Data types and structure vs. methods

Data types and structures

- Flat data tables
- Relational databases
- Temporal & spatial data
- Transactional databases
- Multimedia data
- Genome databases
- Materials science data
- Textual data
- Web data
- etc.

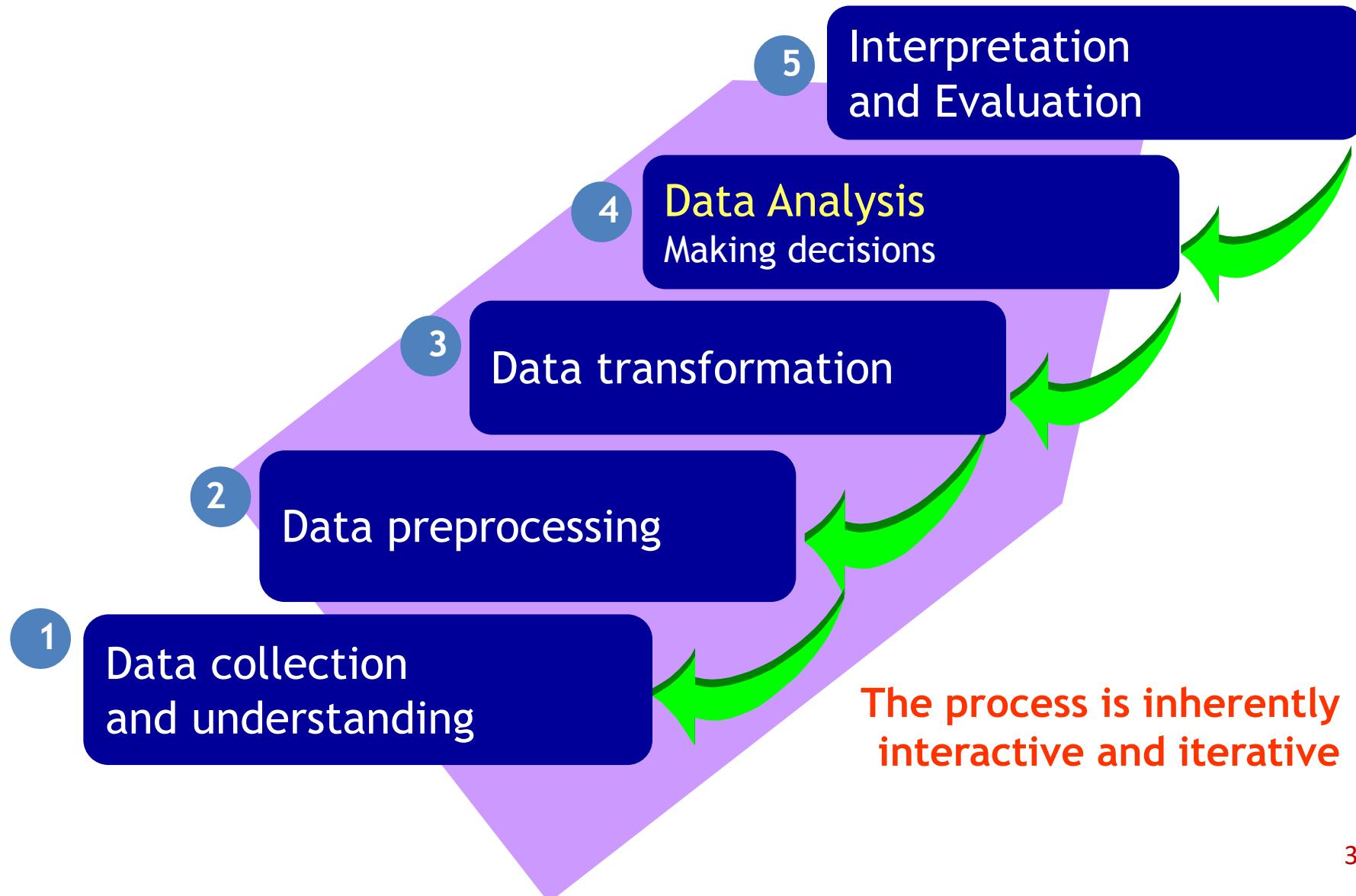


Mining tasks and methods

- **Classification/Prediction**
 - Decision trees
 - Bayesian classification
 - Neural networks
 - Rule induction
 - Support vector machines
 - Hidden Markov Model
 - etc.
- **Description**
 - Association analysis
 - Clustering
 - Summarization
 - etc.



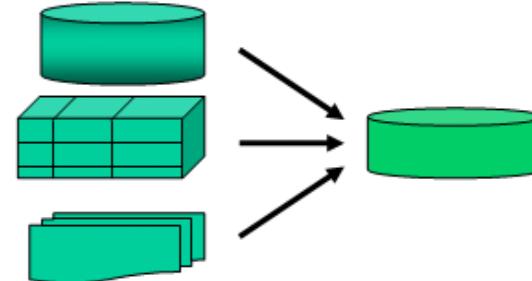
The data analytics process



Major tasks in data preprocessing



1 Data cleaning



-2, 32, 100, 59, 48 → -0.02, 0.32, 1.00, 0.59, 0.48

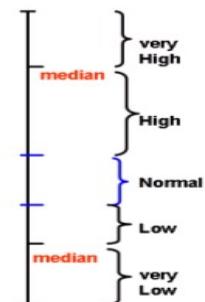
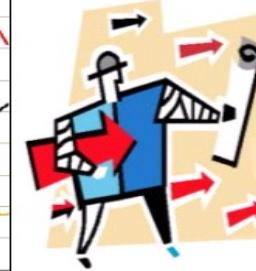
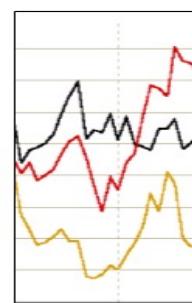
2 Data integration and transformation

| | A1 | A2 | A3 | ... | A126 |
|-------|----|----|----|-----|------|
| T1 | | | | ... | |
| T2 | | | | ... | |
| T3 | | | | ... | |
| ... | | | | ... | |
| T2000 | | | | ... | |



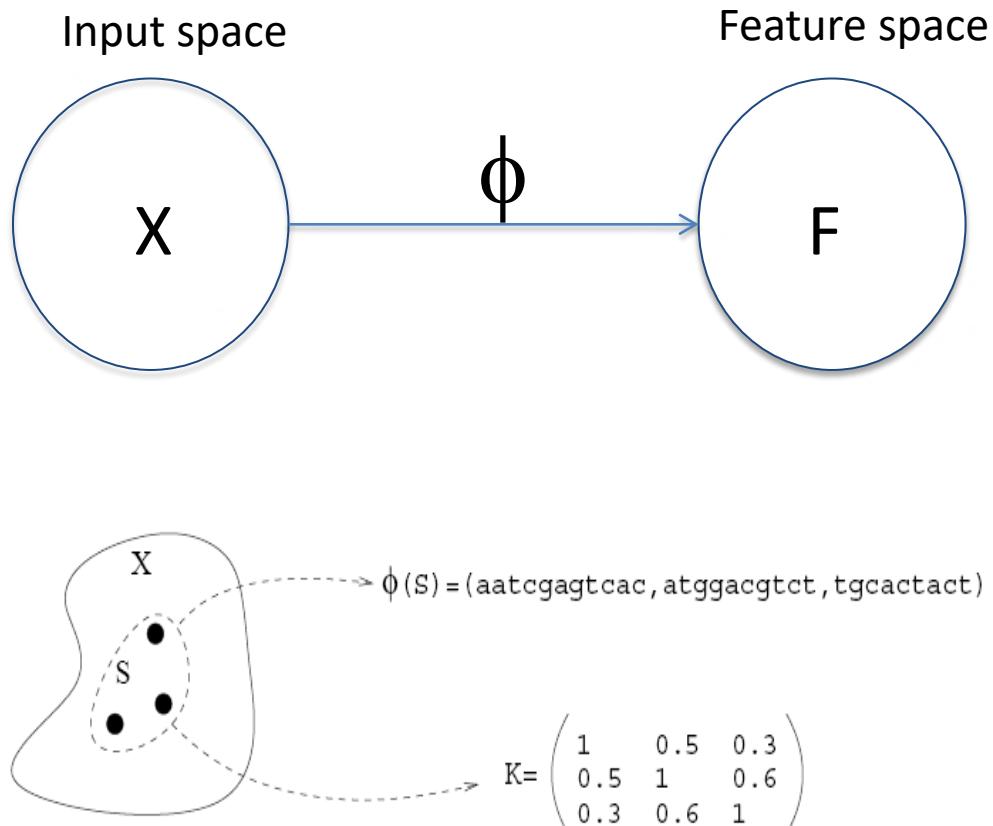
| | A1 | A3 | ... | A115 |
|-------|----|----|-----|------|
| T1 | | | ... | |
| T4 | | | ... | |
| ... | | | ... | |
| T1456 | | | ... | |

3 Data reduction
(instances and dimensions)



4 Data discretization

Data transformation



$\phi: X \rightarrow F$ where
the problem can
be solved in F

X is the set of all
oligonucleotides,
S consists of three
oligonucleotides, and
S is represented in F
as a matrix of pairwise
similarity between its
elements.

Data transformation

Example: Latent semantic indexing

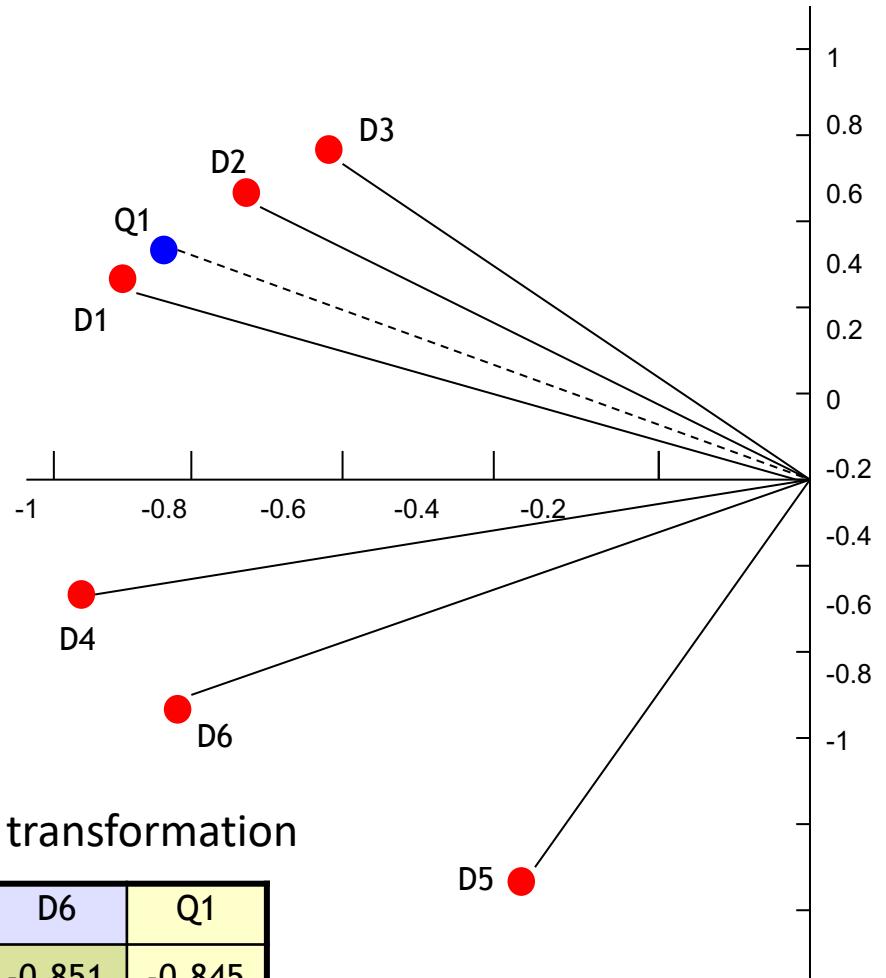
- LSI (Deerwester, 1990) clusters documents in the reduced-dimension semantic space according to word co-occurrence patterns.
- Query Q1 shares common words with D4 and D6 but Q1 is more closed to D3 in meaning.

| | D1 | D2 | D3 | D4 | D5 | D6 | Q1 |
|---------|----|----|----|----|----|----|----|
| rock | 2 | 1 | 0 | 2 | 0 | 1 | 1 |
| granite | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| marble | 1 | 2 | 0 | 0 | 0 | 0 | 1 |
| music | 0 | 0 | 0 | 1 | 2 | 0 | 0 |
| song | 0 | 0 | 0 | 1 | 0 | 2 | 0 |
| band | 0 | 0 | 0 | 0 | 1 | 0 | 0 |



Explaining the meaning after transformation

| | D1 | D2 | D3 | D4 | D5 | D6 | Q1 |
|--------|--------|--------|--------|--------|--------|--------|--------|
| Dim. 1 | -0.888 | -0.759 | -0.615 | -0.961 | -0.388 | -0.851 | -0.845 |
| Dim. 2 | 0.460 | 0.652 | 0.789 | -0.276 | -0.922 | -0.525 | 0.534 |



Machine learning: View by nature of methods

| Tribes | Origins | Master Algorithms |
|----------------|----------------------|-------------------------|
| Symbolists | Logic, philosophy | Inverse deduction |
| Evolutionaries | Evolutionary biology | Genetic programming |
| Connectionists | Neuroscience | Backpropagation |
| Bayesians | Statistics | Probabilistic inference |
| Analogizers | Psychology | Kernel machines |

The five tribes of machine learning, Pedro Domingos

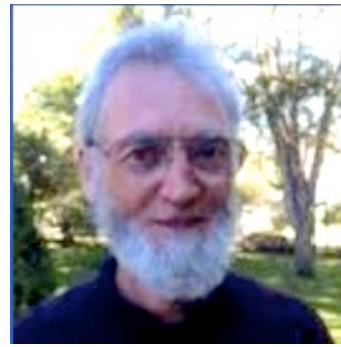
Symbolists



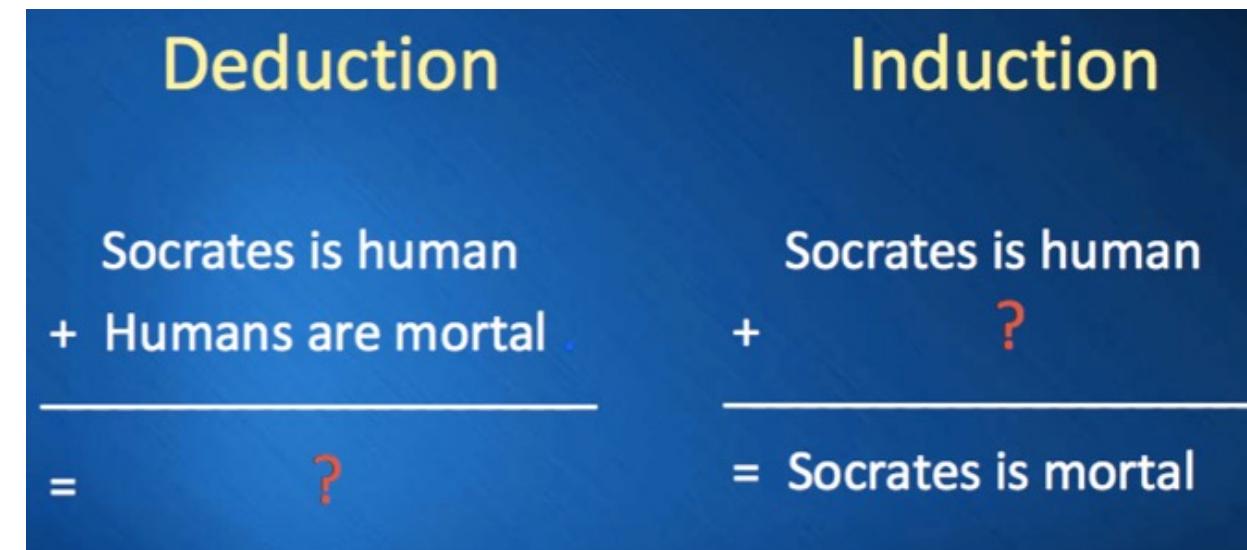
Tom Mitchell



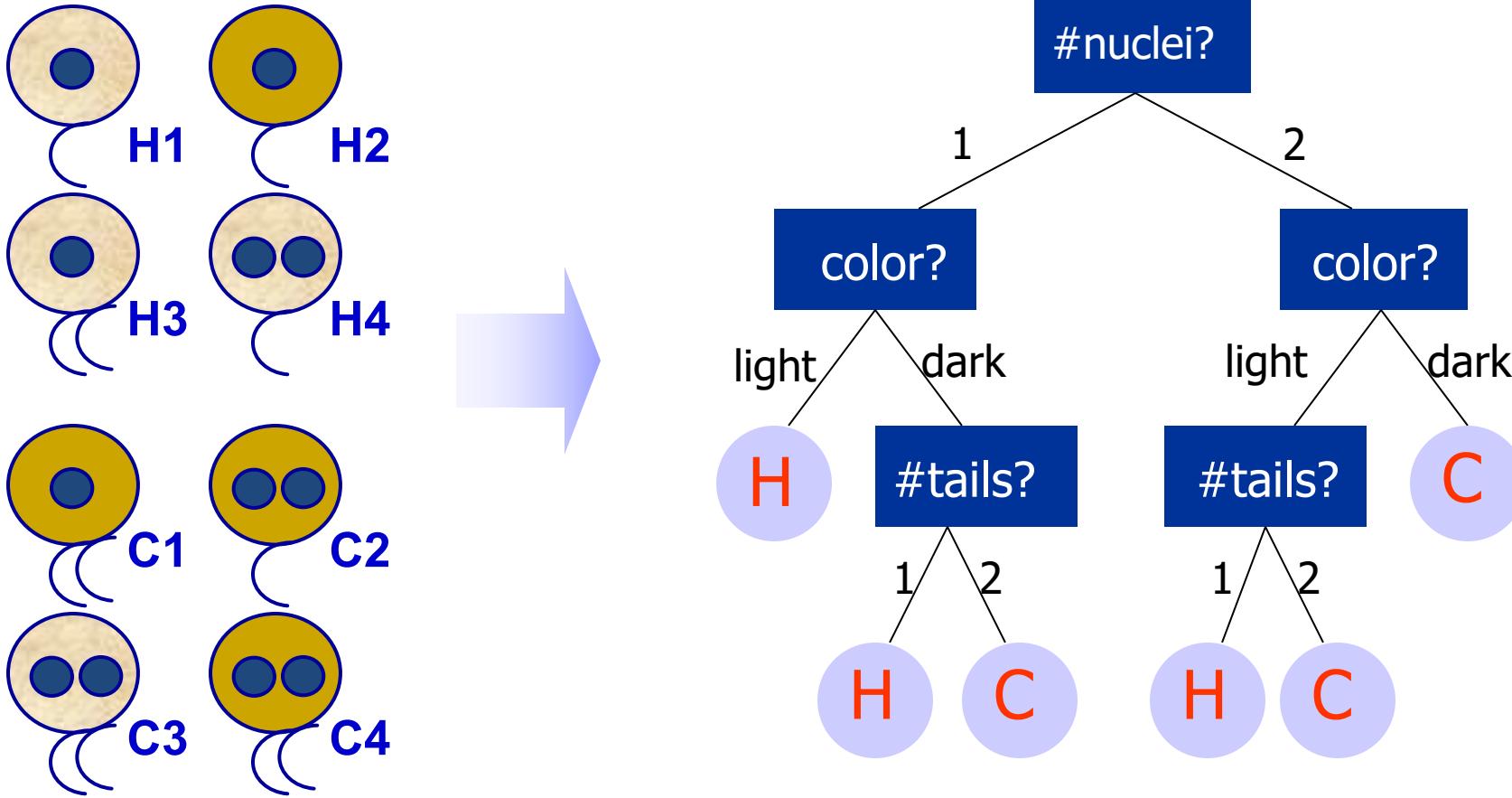
Steve Muggleton



Ross Quinlan



Classification with decision trees



Evolutionaries



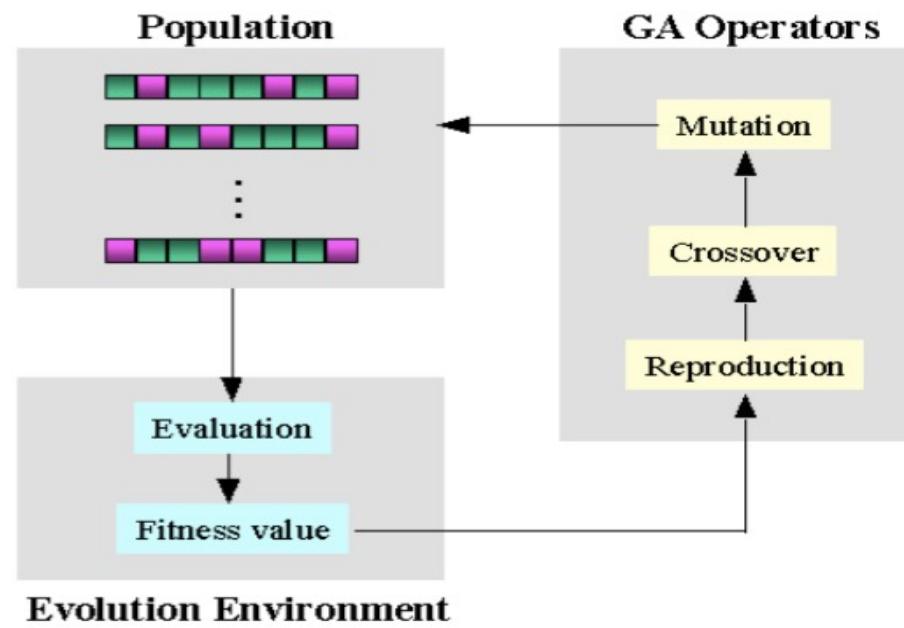
John Koza



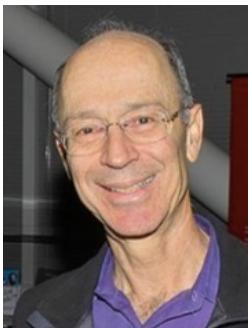
John Holland



Hod Lipson



Analoziger



Peter Hart



Vladimir Vapnik



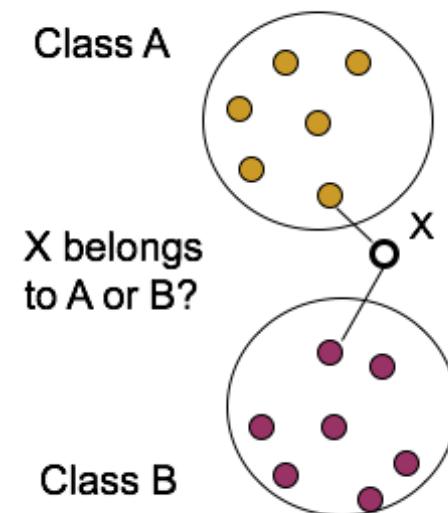
Douglas Hofstadter

■ Instance-based classification

- Using most similar individual instances known in the past to classify a new instance

■ Typical approaches

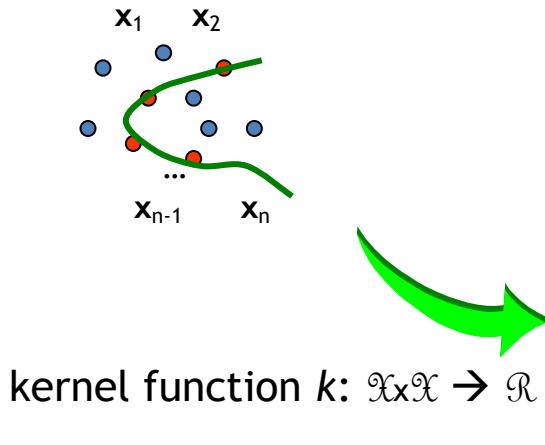
- **k-nearest neighbor approach**
 - Instances represented as points in a Euclidean space



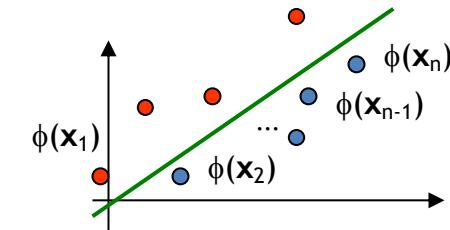
Kernel methods

The basic ideas

Input space X



Feature space F



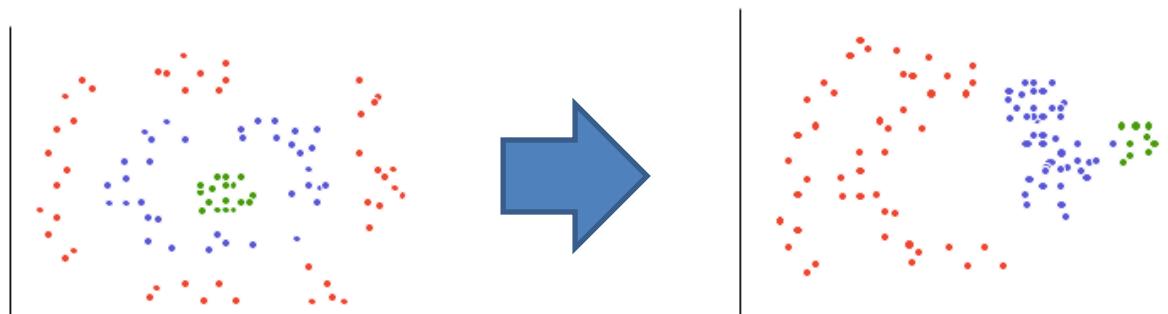
inverse map ϕ^{-1}

$\phi(\mathbf{x})$

$$k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$$

Kernel matrix $K_{n \times n}$

kernel-based algorithm on K
(computation done on kernel matrix)



Connectionists



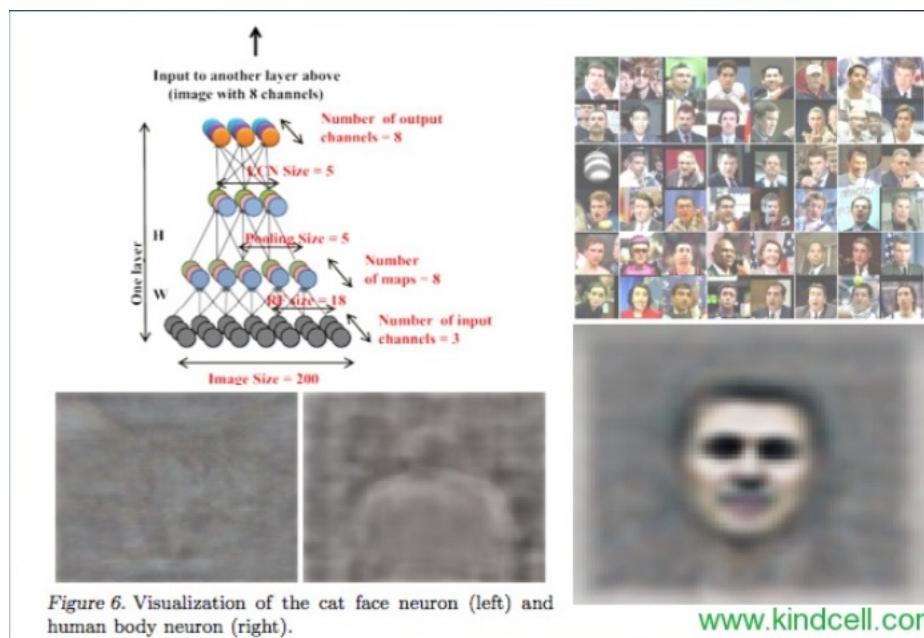
Yann LeCun



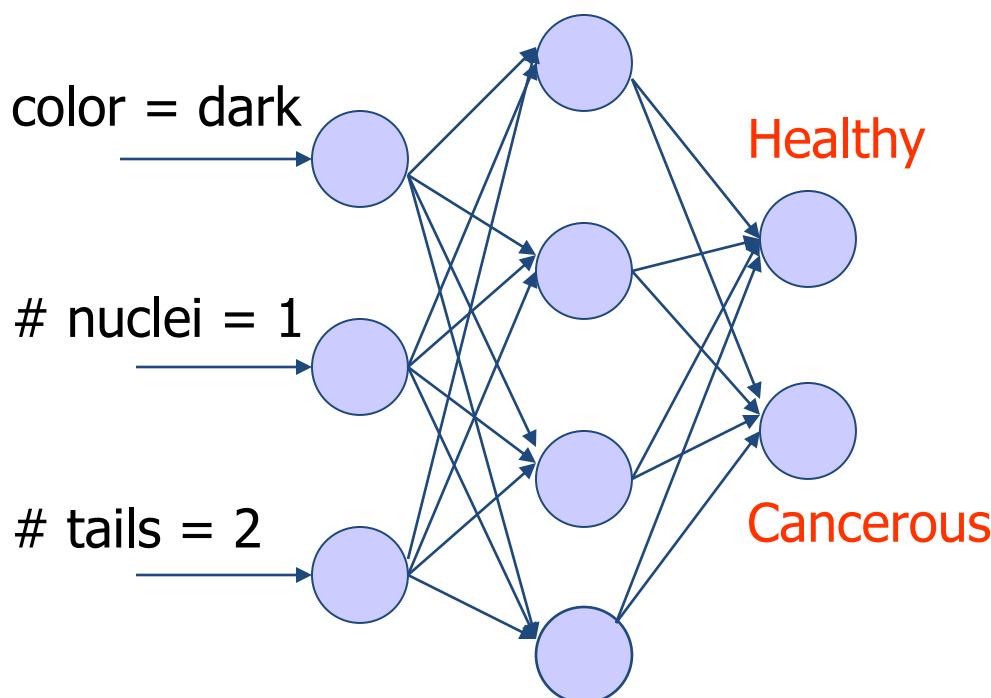
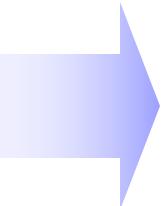
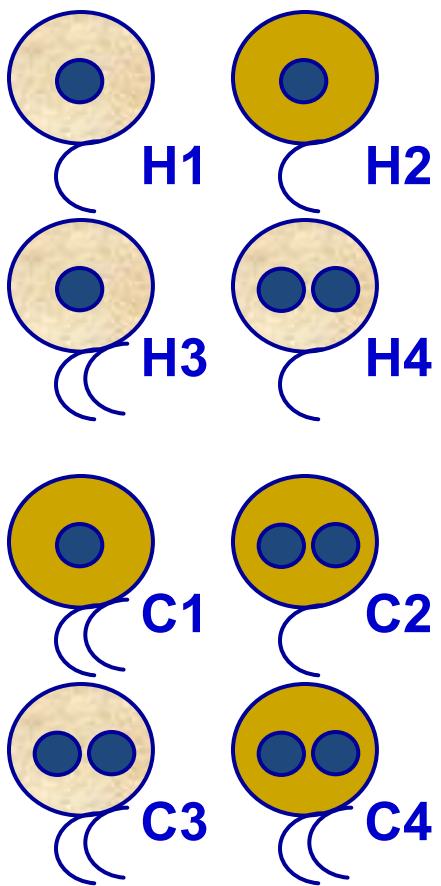
Geoff Hinton



Yoshua Bengio

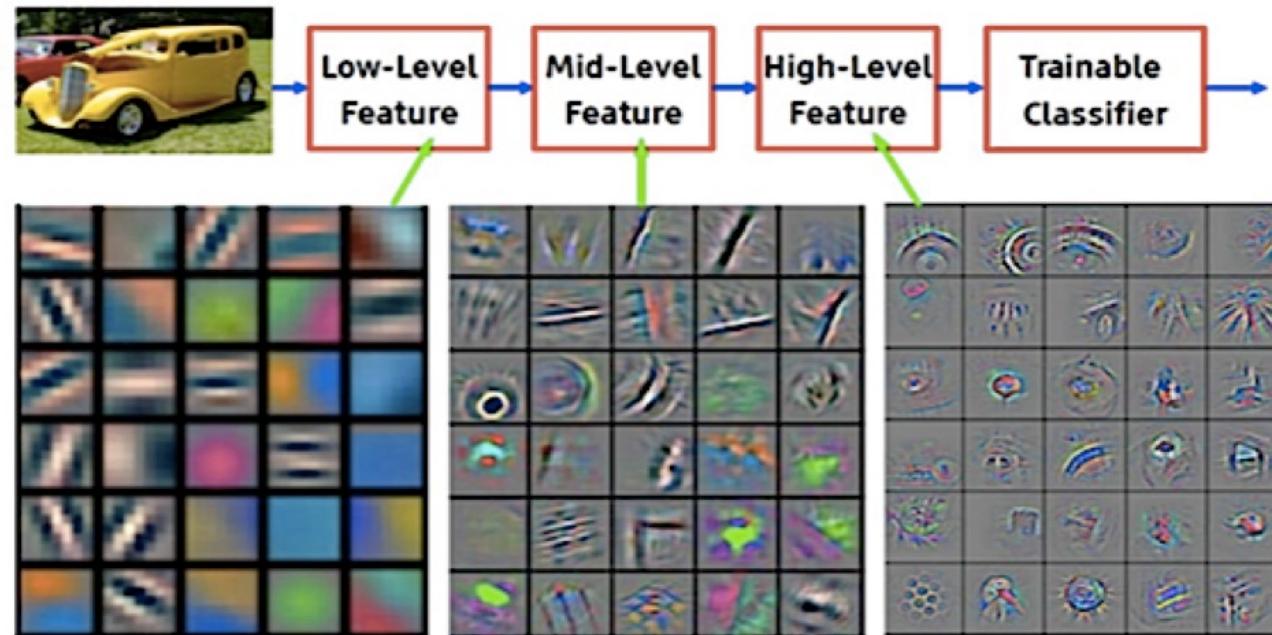


Classification with neural networks



Deep learning

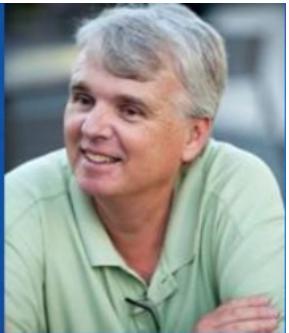
“Deep Learning: machine learning algorithms based on learning **multiple levels** of representation and abstraction” Joshua Bengio



Feature visualization of CNN trained on ImageNet

[Zeiler and Fergus 2013]

Bayesians in machine learning



David Heckerman



Judea Pearl



Michael Jordan

The diagram illustrates the four components of Bayes' Theorem:

- Likelihood**: How probable is the evidence given that our hypothesis is true?
- Prior**: How probable was our hypothesis before observing the evidence?
- Posterior**: How probable is our hypothesis given the observed evidence?
(Not directly computable)
- Marginal**: How probable is the new evidence under all possible hypotheses?
$$P(e) = \sum P(e | H_i) P(H_i)$$

$$P(H | e) = \frac{P(e | H) P(H)}{P(e)}$$

Which algorithms do best at which tasks?

| Algorithm | Pros | Cons | Good at |
|-------------------------|--|--|--|
| Linear regression | - Very fast (runs in constant time) - Easy to understand the model - Less prone to overfitting | - Unable to model complex relationships - Unable to capture nonlinear relationships without first transforming the inputs | - The first look at a dataset - Numerical data with lots of features |
| Decision trees | - Fast - Robust to noise and missing values - Accurate | - Complex trees are hard to interpret - Duplication within the same sub-tree is possible | - Star classification - Medical diagnosis - Credit risk analysis |
| Neural networks | - Extremely powerful - Can model even very complex relationships - No need to understand the underlying data - Almost works by "magic" | - Prone to overfitting - Long training time - Requires significant computing power for large datasets - Model is essentially unreadable | - Images - Video - "Human-intelligence" type tasks like driving or flying - Robotics |
| Support Vector Machines | - Can model complex, nonlinear relationships - Robust to noise (because they maximize margins) | - Need to select a good kernel function - Model parameters are difficult to interpret - Sometimes numerical stability problems - Requires significant memory and processing power | - Classifying proteins - Text classification - Image classification - Handwriting recognition |
| K-Nearest Neighbors | - Simple - Powerful - No training involved ("lazy") - Naturally handles multiclass classification and regression | - Expensive and slow to predict new instances - Must define a meaningful distance function - Performs poorly on high-dimensionality datasets | - Low-dimensional datasets - Computer security: intrusion detection - Fault detection in semi-conductor manufacturing - Video content retrieval - Gene expression - Protein-protein interaction |

<http://www.lauradhamilton.com/machine-learning-algorithm-cheat-sheet>

Model selection

- **Problem:** Choosing *the most appropriate* model(s) given a dataset and the task.

- Relating to selecting
 - Models that can be appropriated
 - Parameters of those models

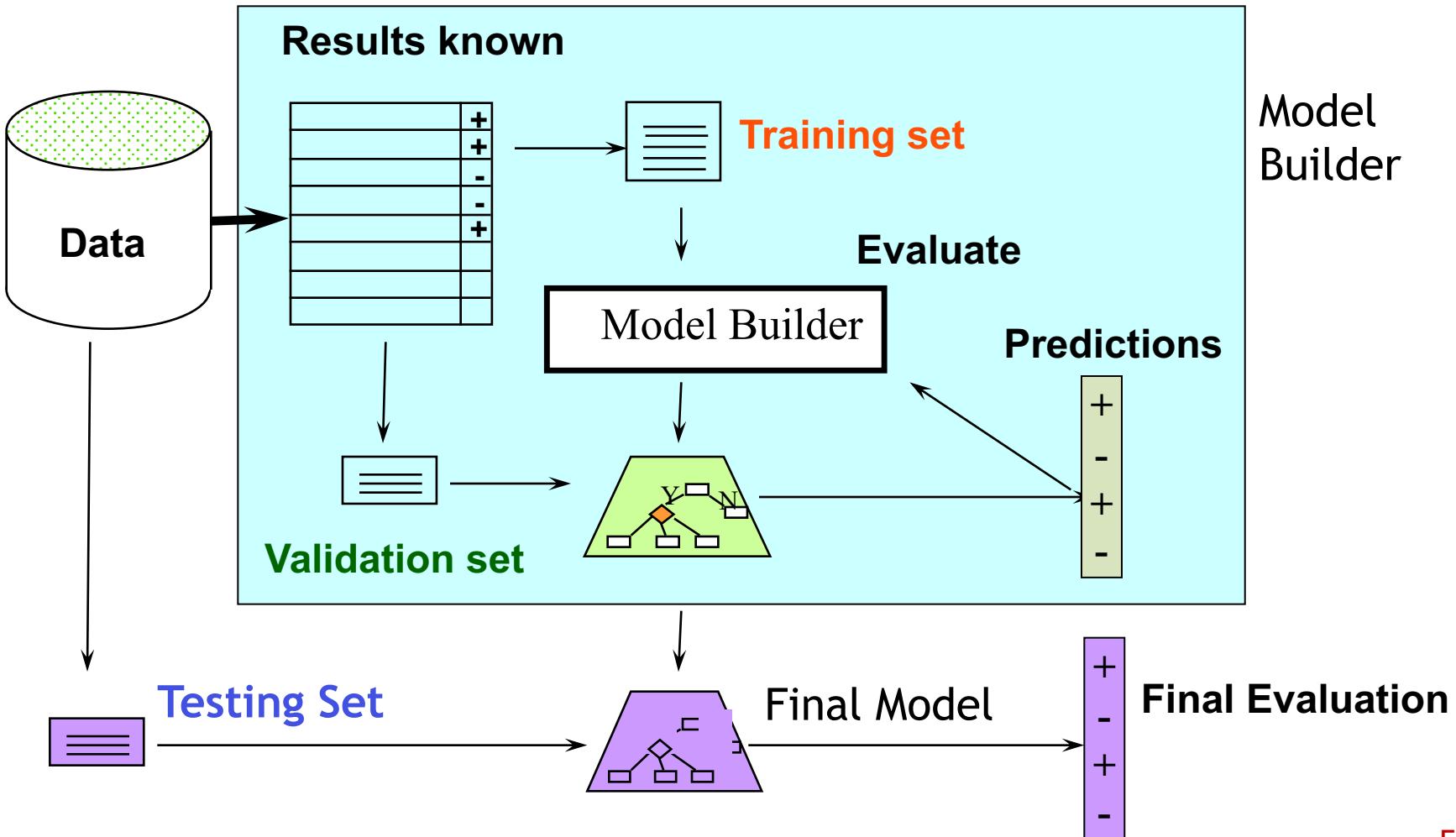
- Examples of model selection problems

- Is it a linear or non-linear regression I should choose?
- Which neural net architecture gives the best generalization error?
- How many neighbors should I take in consideration in k-NN?
- Should I use a linear model, a decision tree, a neural net, a local learning algorithms?
- Which of the 50 features are relevant for this problem?



(1919-2013)

Classification: Train, Validation, Test



K490: データサイエンス論

Lecture 2: Data Mining process

Lecturer: Hieu-Chi Dam, Takashi Isogai

講義予定

| | | |
|-----------------------------|--------------|------|
| 1. データサイエンスの紹介 | (2022年1月30日) | (ダム) |
| 2. データマイニングによる知識発見プロセス | (2022年1月30日) | (ダム) |
| 3. 確証的データ解析と探索的データ解析 | (2022年1月31日) | (ダム) |
| 4. 基礎的なデータ解析手法（1）：单変量解析 | (2022年1月31日) | (ダム) |
| 5. 基礎的なデータ解析手法（2）：多変量解析 | (2022年2月01日) | (ダム) |
| 6. 予測的データ解析手法（1）：決定木 | (2022年2月01日) | (ダム) |
| 7. 予測的データ解析手法（2）：ベイジアン分類 | (2022年2月02日) | (磯貝) |
| 8. 予測的データ解析手法（3）：サポートベクトル分類 | (2022年2月02日) | (磯貝) |
| 9. 記述的データ解析手法（1）：クラスタリング | (2022年2月03日) | (磯貝) |
| 10. 記述的データ解析手法（2）：特徴選択と次元削減 | (2022年2月03日) | (磯貝) |
| 11. 記述的データ解析手法（3）：グラフの解析 | (2022年2月04日) | (磯貝) |
| 12. データ科学と倫理問題 | (2022年2月04日) | (磯貝) |
| 13. 学生発表（1） | (2022年2月04日) | (ダム) |
| 14. 学生発表（2） | (2022年2月04日) | (ダム) |

Outline

1. Why Preprocess the Data?
2. Data Cleaning
3. Data Integration
4. Data Reduction
5. Data Transformation

Why preprocess the data?

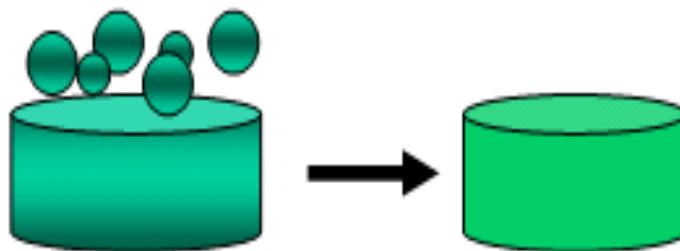
Common properties of large real-world databases:

- **Incomplete**: lacking attribute values or certain of interest
- **Noisy**: containing errors or outliers
- **Inconsistent**: containing discrepancies in codes or names

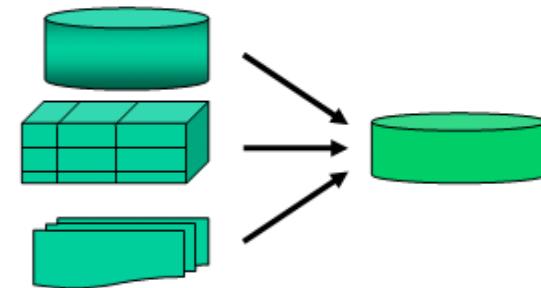
Veracity problem!

No quality data, no quality analysis results!

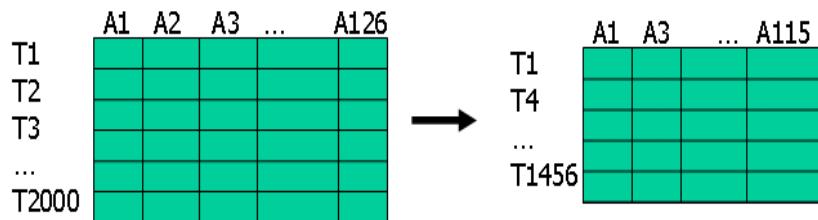
Major tasks in data preprocessing



1 Data cleaning



2 Data integration



3 Data reduction
(instances and dimensions)



4 Data transformation

Major tasks in data preprocessing

- **Data cleaning**
 - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- **Data integration**
 - Integration of multiple databases, data cubes, or files
- **Data transformation**
 - Normalization and aggregation
- **Data reduction**
 - Obtains reduced representation in volume but produces the same or similar analytical results
- **Data discretization**
 - Part of data reduction but with particular importance, especially for numerical data

Outline

1. Why Preprocess the Data?
2. **Data Cleaning**
3. Data Integration
4. Data Reduction
5. Data Transformation

Data cleaning tasks

- Fill in missing values
- Identify outliers and smooth out noisy data
- Correct inconsistent data

Missing data

- Data is not always available
 - e.g., many tuples have no recorded value for several attributes, such as customer income in sales data
- Missing data may be due to
 - equipment malfunction
 - inconsistent with other recorded data and thus deleted
 - data not entered due to misunderstanding
 - certain data may not be considered important at the time of entry
 - not register history or changes of the data
- Missing data may need to be inferred.

Missing values in databases

- Missing values may hide a true answer underlying in the data
- Many data mining programs cannot be applied with data that includes missing values

| | | | | | | |
|---------|-----|----|----|-----------|---------|-----|
| norm | yes | 23 | 12 | none | absent | dna |
| norm | no | 43 | 14 | fungicide | present | dna |
| lt-norm | yes | 14 | 33 | fungicide | absent | dna |
| gt-norm | ? | 35 | 6 | ? | ? | ? |
| norm | yes | 23 | 11 | none | absent | dna |
| gt-norm | ? | ? | 8 | ? | absent | dna |
| norm | no | ? | ? | none | present | dna |
| norm | yes | 35 | 36 | none | absent | dna |
| norm | yes | 31 | 12 | fungicide | present | dna |
| norm | ? | ? | 13 | ? | absent | dna |
| norm | ? | 23 | 14 | ? | ? | ? |
| norm | ? | ? | ? | ? | absent | dna |
| gt-norm | ? | 35 | ? | ? | ? | ? |
| norm | no | 26 | ? | none | absent | dna |

Class attribute: norm, lt-norm, gt-norm

Other six attributes all have missing values

Missing values in databases

Methods

1. Ignore the tuples
2. Fill in the missing value manually
(tedious + infeasible?)
3. Use a global constant to fill in the missing value
4. Use the attribute mean to fill the missing values
5. Use the attribute mean (or mode for categorical attribute) for all samples belonging to the same class as the given tuple.
6. Use the most probable value to fill the missing value
7. Others

| | Methods: | 2 | 4 | 5 | 3 | 6 | 6 |
|------|----------|-------|------|-----------|-----------|------------|-----|
| norm | yes | 23 | 12 | none | absent | dna | |
| | no | 43 | 14 | fungicide | present | dna | |
| | lt-norm | 14 | 33 | fungicide | absent | dna | |
| | gt-norm | ? yes | 35 | 6 | ? none | ? unknown? | dna |
| | norm | yes | 23 | 11 | none | absent | dna |
| | gt-norm | ? no | ? 29 | 8 | ? none | absent | dna |
| | norm | no | ? 29 | ? 13 | none | present | dna |
| | norm | yes | 35 | 36 | none | absent | dna |
| | norm | yes | 31 | 12 | fungicide | present | dna |
| | norm | ? yes | ? 29 | 13 | ? none | absent | dna |
| | norm | ? no | 23 | 14 | ? none | ? unknown? | dna |
| | norm | ? no | ? 29 | ? 13 | ? none | absent | dna |
| | gt-norm | ? yes | 35 | ? 7 | ? none | ? unknown? | dna |
| | norm | no | 26 | ? 13 | none | absent | dna |

Noisy data

- Noise: random error or variance in a measured variable
- Incorrect attribute values may due to
 - faulty data collection instruments
 - data entry problems
 - data transmission problems
 - technology limitation
 - inconsistency in naming convention
- Other data problems which requires data cleaning
 - duplicate records
 - incomplete data
 - inconsistent data

How to handle noisy data?

- **Binning method**
 - first sort data and partition into (equi-depth) bins
 - then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
- **Clustering**
 - detect and remove outliers
- **Combined computer and human inspection**
 - detect suspicious values and check by human
- **Regression**
 - smooth by fitting the data into regression functions

How to handle noisy data?

Binning: to smooth a sorted data value by consulting its “neighborhood”, that is, the value around it (local smoothing)

- **Smoothing by bin means:** each value in a bin is replaced by the mean value of the bin
- **Smoothing by bin medians:** each bin value is replaced by the bin median
- **Smoothing by bin boundaries:** the minimum and maximum values in each bin are identified as bin boundaries

How to handle noisy data?

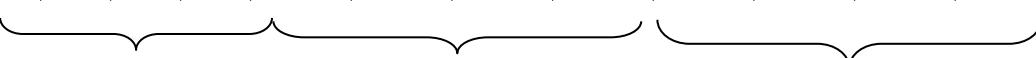
- The original data

9, 21, 24, 21, 4, 26, 28, 34, 29, 8, 15, 25

- Sort data in the increasing order, and partition into (equidepth) bins:

4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34


- Smoothing by bin means

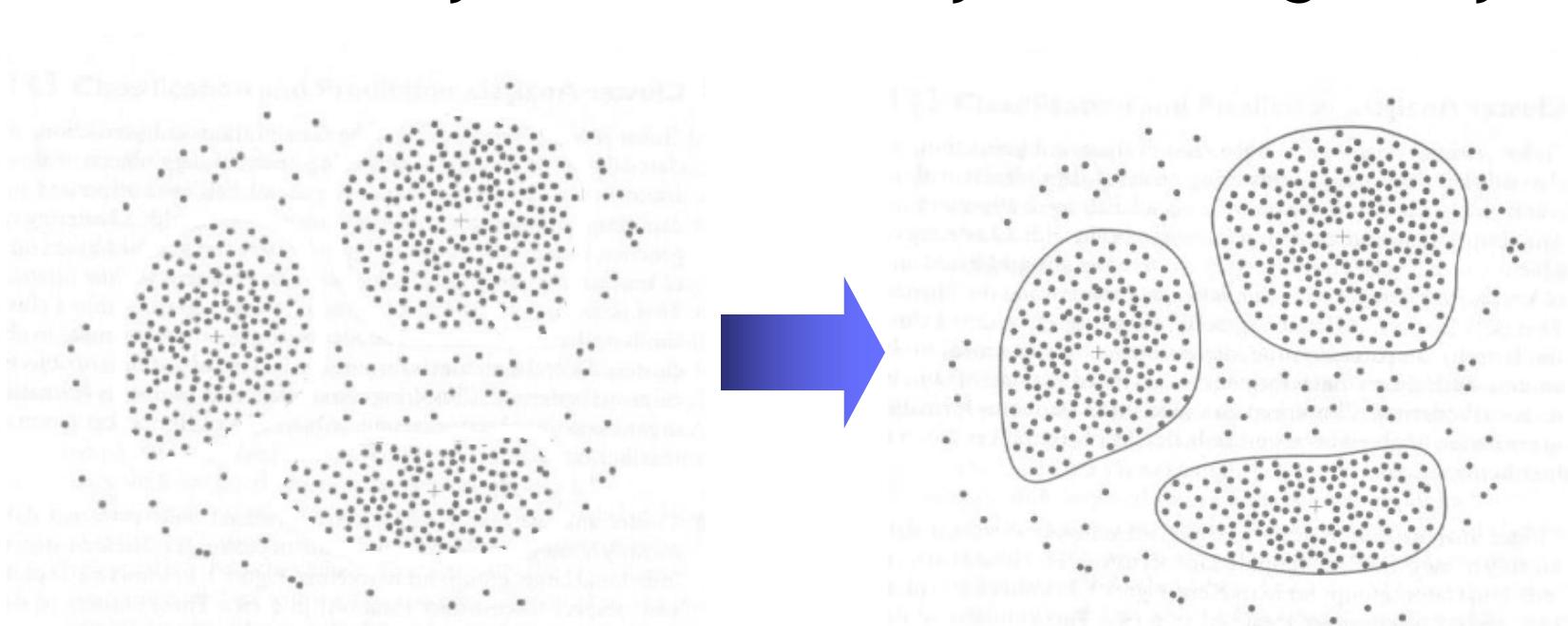
9, 9, 9, 9, 22, 22, 22, 22, 29, 29, 29, 29


- Smoothing by bin boundaries (replaced by the closest boundary)

4, 4, 4, 15, 21, 21, 25, 25, 26, 26, 26, 34


How to handle noisy data?

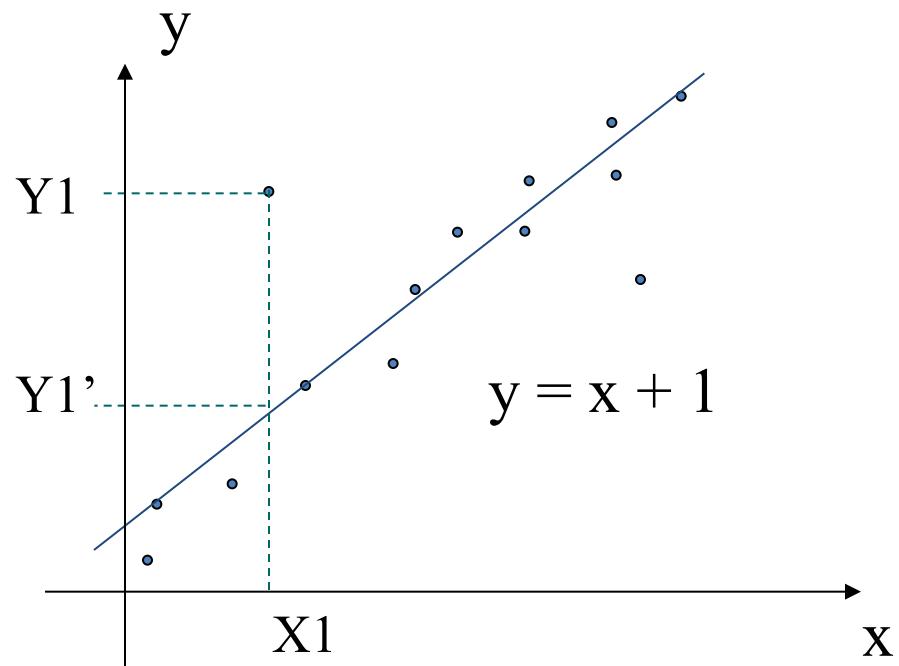
- Outliers may be detected by clustering analysis



Values that fall outside of the set of clusters may be considered outliers

How to handle noisy data?

- **Combined computer and human inspection:** Output patterns with surprise content to a list. A human can identify the actual garbage ones.
- **Regression:** by fitting the data to a function, such as with regression
 - Linear regression
 - Multiple linear regression: more than two variables and the data are fit to a multidimensional surface



Outline

1. Why Preprocess the Data?
2. Data Cleaning
3. Data Integration
4. Data Reduction
5. Data Transformation

Data integration

- **Data integration** combines data from multiple sources (multiple DBs, data cubes, flat files) into a coherent data store.
- **Schema integration** (entity identification problem): How can equivalent entities from multiple data sources be matched up?
- **Redundancy**: An attribute may be redundant if it can be “derived” from another table.

Data integration

- **Redundancy:** can be detected by correlation analysis (correlation coefficient), e.g., how strongly one attribute implies another attribute.

$$r_{A,B} = \frac{\sum (A - \bar{A})(B - \bar{B})}{(n-1)\sigma_A\sigma_B}$$

- Detection and resolution of data value conflicts

Outline

1. Why Preprocess the Data?
2. Data Cleaning
3. Data Integration
4. Data Reduction
5. Data Transformation

Strategies for data reduction

- Data cube aggregation
- Dimension reduction
- Data compression
- Numerosity reduction
- Discretization and concept hierarchy generation

Data compression: Attribute selection

Attribute subset selection (also called “feature selection”)

- Stepwise forward selection
- Stepwise backward elimination
- Combination of forward and backward elimination
- Many other methods

Forward selection

Initial attribute set:
 $\{A_1, A_2, A_3, A_4, A_5, A_6\}$

Initial reduced set:
{}
→ $\{A_1\}$
→ $\{A_1, A_4\}$
→ Reduced attribute set:
 $\{A_1, A_4, A_6\}$

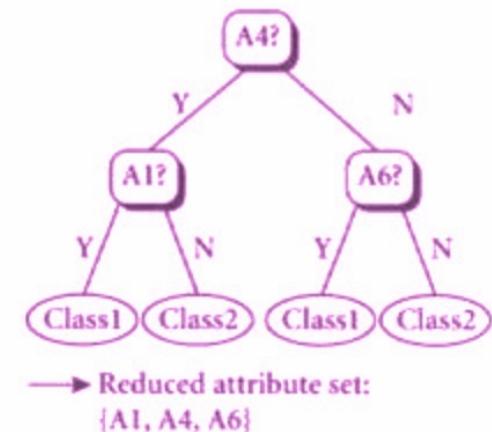
Backward elimination

Initial attribute set:
 $\{A_1, A_2, A_3, A_4, A_5, A_6\}$

→ $\{A_1, A_3, A_4, A_5, A_6\}$
→ $\{A_1, A_4, A_5, A_6\}$
→ Reduced attribute set:
 $\{A_1, A_4, A_6\}$

Decision tree induction

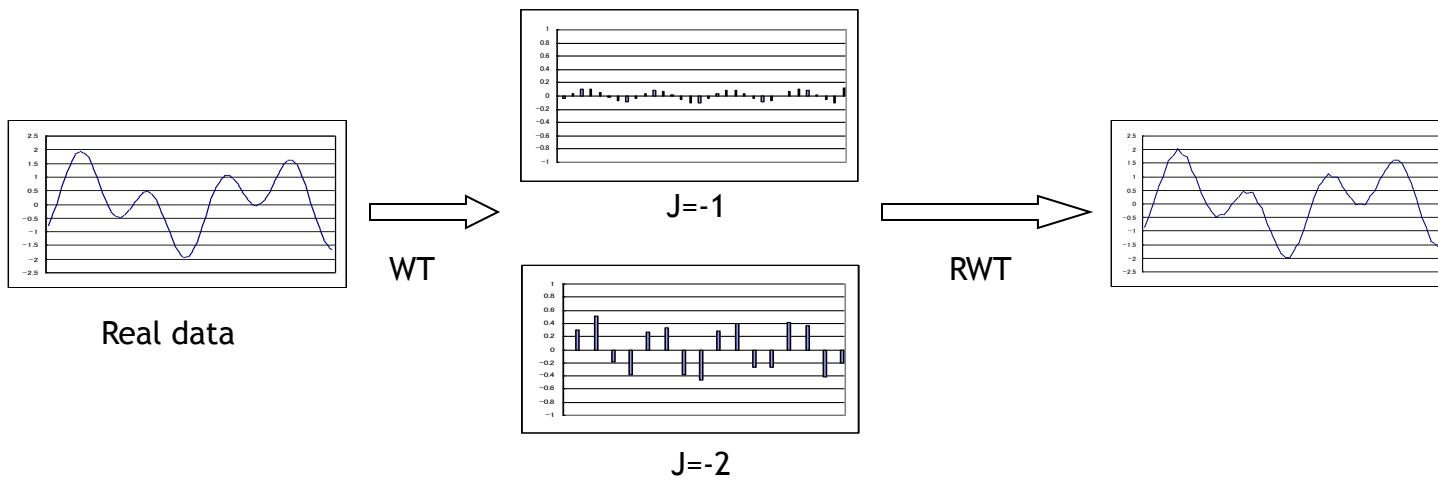
Initial attribute set:
 $\{A_1, A_2, A_3, A_4, A_5, A_6\}$



→ Reduced attribute set:
 $\{A_1, A_4, A_6\}$

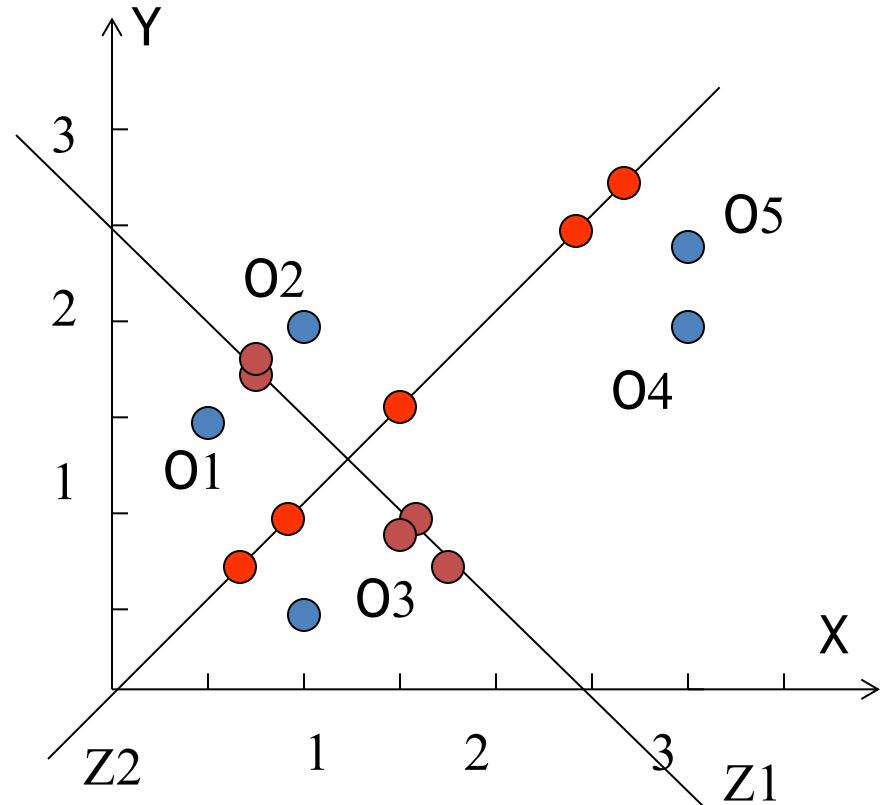
Data compression: Wavelet transforms

- **Discrete wavelet transformation (DWT):** a linear signal processing technique that, when applied to a data vector D , transforms it to a numerically different vector D' of wavelet coefficients.
- Store only a small fraction of the strongest of the wavelet coefficients



Data compression: PCA

- **Principal Components Analysis:** transform data points from k -dimensions into c -dimensions ($c \leq k$) with minimum loss of information
- PCA searches for c -dimensional orthogonal vectors that can best be used to represent data. The original data are thus projected onto a much smaller space of c dimensions (c principal components)
- Only used for numerical data

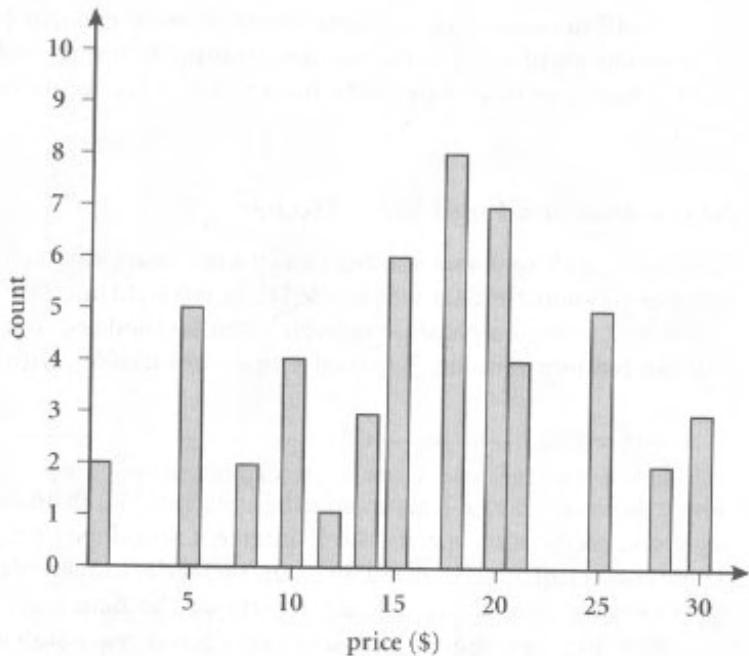


Question: Reduction to one dimension?
Z1 and Z2, which is better?

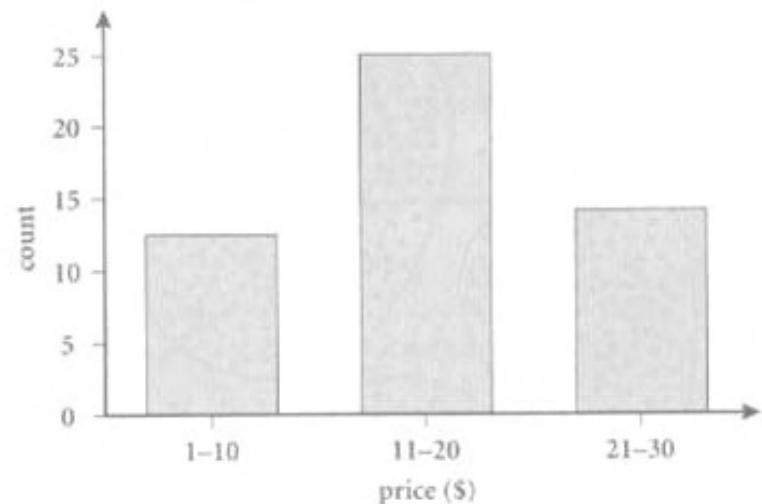
Numerosity reduction

- Can we reduce the data volume by choosing alternative, 'smaller' forms of data representation?
- **Parameter methods:** a model is used to estimate the data, so that typically only the data parameters need be stored, instead of the actual data
 - Regression and Log-Linear Models: $y = \alpha x + \beta$
- **Non-parameter methods:** for storing reduced representations of the data include
 - Histograms
 - Clustering
 - Sampling

Numerosity reduction: histogram



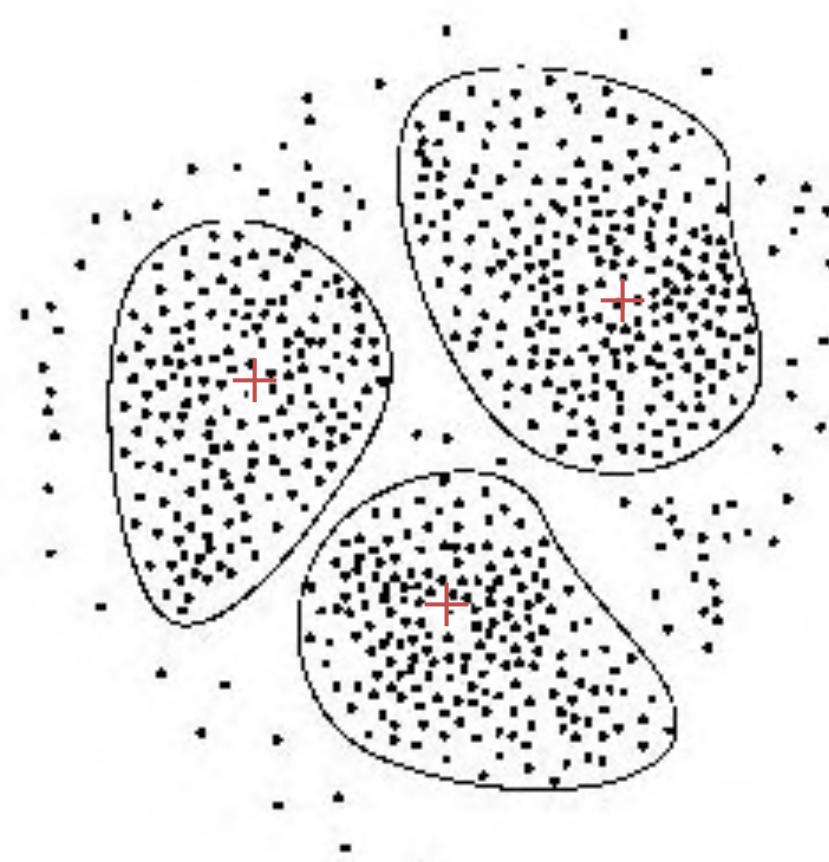
Singleton buckets: Each bucket represents one price-value/frequency pair



An equiwidth histogram, where values are aggregated so that each bucket has a uniform width of \$10

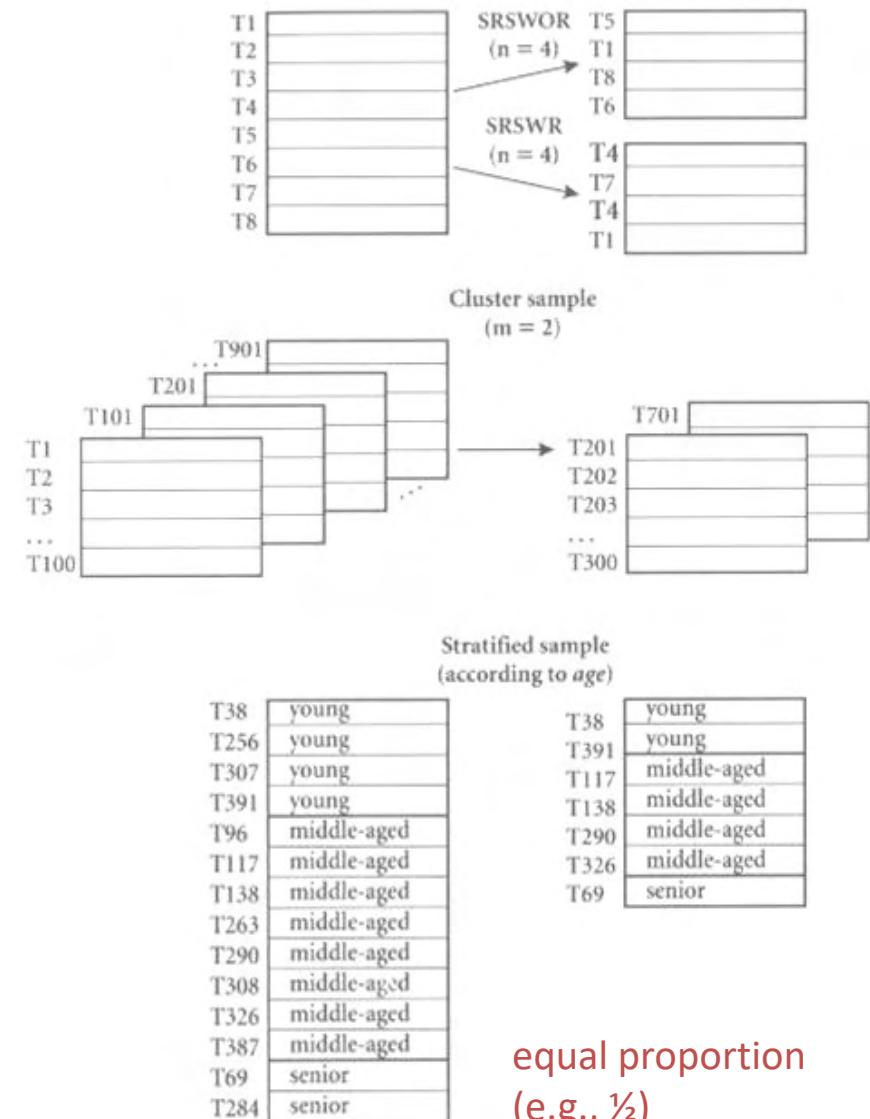
Numerosity reduction: Clustering

A 2-D plot of customer data with respect to customer locations in a city, showing three data clusters. Each cluster “center” is marked with a “+”



Numerosity reduction: Sampling

- Simple random sample **without replacement** of size n (SRSWOR)
- Simple random sample **with replacement** of size n (SRSWR)
- Cluster sample
- Stratified sample



Outline

1. Why Preprocess the Data?
2. Data Cleaning
3. Data Integration
4. Data Reduction
5. Data Transformation

Data transformation

- **Smoothing:** to remove noise from data
- **Aggregation:** summary or aggregation are applied to the data
- **Generalization:** low-level or “primitive” data are replaced by higher-level concepts using concept hierarchy
- **Normalization:** attribute data are scaled to fall within a small specified range, says 0.0 to 1.0
- **Attribute construction:** new attributes are constructed and added from the given set of attributes to help the mining process: from continuous to discrete (discretization) and from discrete to continuous (word embedding).

Min-max and z-score normalization

- **min-max normalization:** Suppose \min_A and \max_A are minimum and maximum values of attribute. We map a value v of A to v' in the range $[\text{newmin}_A, \text{newmax}_A]$ by
- **Example:** Suppose \min_A and \max_A are \$12,000 and \$98,000. We want to map minimum and maximum values of attribute. We want to map income to the range [0.0, 1.0]. So, \$73,600 is transformed to
- **z-score normalization:** The values for an attribute A are normalized based on the mean and standard deviation of A
- **Example:** If the mean and standard deviation are \$54,000 and \$16,000, the \$73,600 is transformed to

$$v' = \frac{v - \min_A}{\max_A - \min_A} (\text{newmax}_A - \text{newmin}_A) + \text{newmin}_A$$

$$\frac{73,600 - 12,000}{98,000 - 12,000} (1.0 - 0.0) + 0 = 0.716$$

$$v' = \frac{v - \bar{A}}{\sigma_A}$$

$$\frac{73,600 - 54,000}{16,000} = 1.225$$

Discretization

- **Three types of attributes:**
 - Nominal (categorical): red, yellow, blue, green
 - Ordinal: small, middle, large, extreme large
 - Continuous: real numbers
- **Discretization:** divide the range of a continuous attribute into intervals
 - Some classification algorithms only accept categorical attributes.
 - Reduce data size by discretization
 - Prepare for further analysis

Discretization

- Binning
- Histogram analysis
- Cluster analysis
- Entropy-based discretization
- Segmentation by Natural Partitioning

Entropy-based discretization

- Given a set of samples S , if S is partitioned into two intervals S_1 and S_2 using boundary T , the entropy after partitioning is

$$E(S, T) = \frac{|S_1|}{|S|} Ent(S_1) + \frac{|S_2|}{|S|} Ent(S_2)$$

- The boundary that minimizes the entropy function over all possible boundaries is selected as a binary discretization.
- The process is recursively applied to partitions obtained until some stopping criterion is met, e.g.,

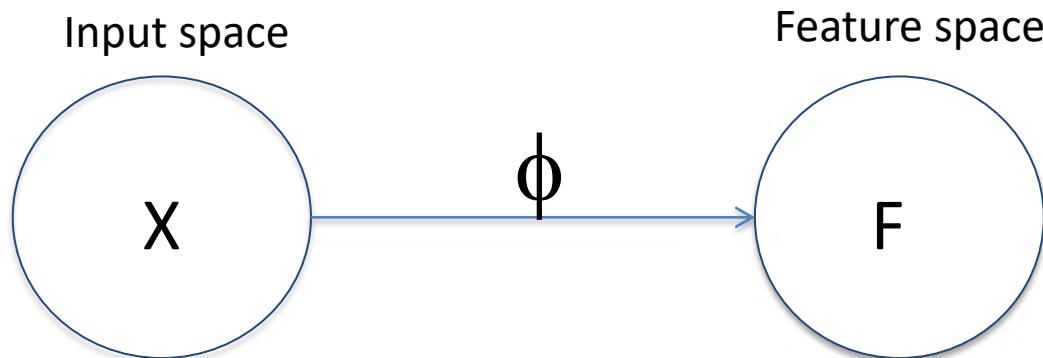
$$Ent(S) - E(T, S) > \delta$$

- Experiments show that it may reduce data size and improve classification accuracy

What is word embedding?

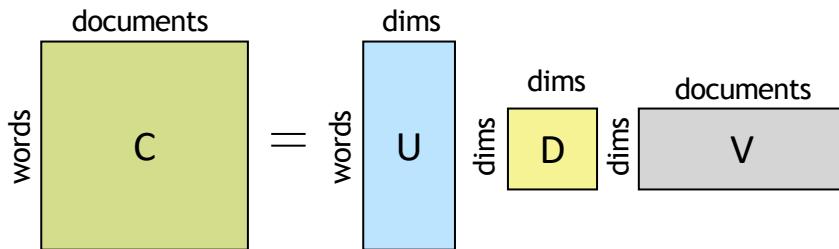
- **Word embedding:** Mapping a word (or phrase) from its original high dimensional input space to a lower-dimensional numerical vector space.
- **Word2vec** is a group of related models that are used to produce word embeddings.
 - These models are shallow, two-layer neural networks that are trained to reconstruct linguistic contexts of words.
 - Word2vec takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space.
- **Word vectors** are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the space.

Some more complex data transformation

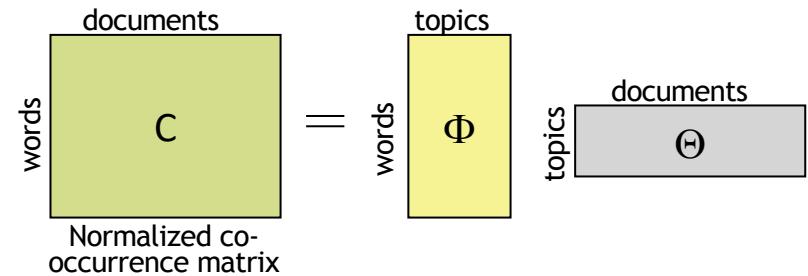


$\phi: X \rightarrow F$
where the
problem can
be solved in F

Latent semantic indexing



Topic models



講義予定

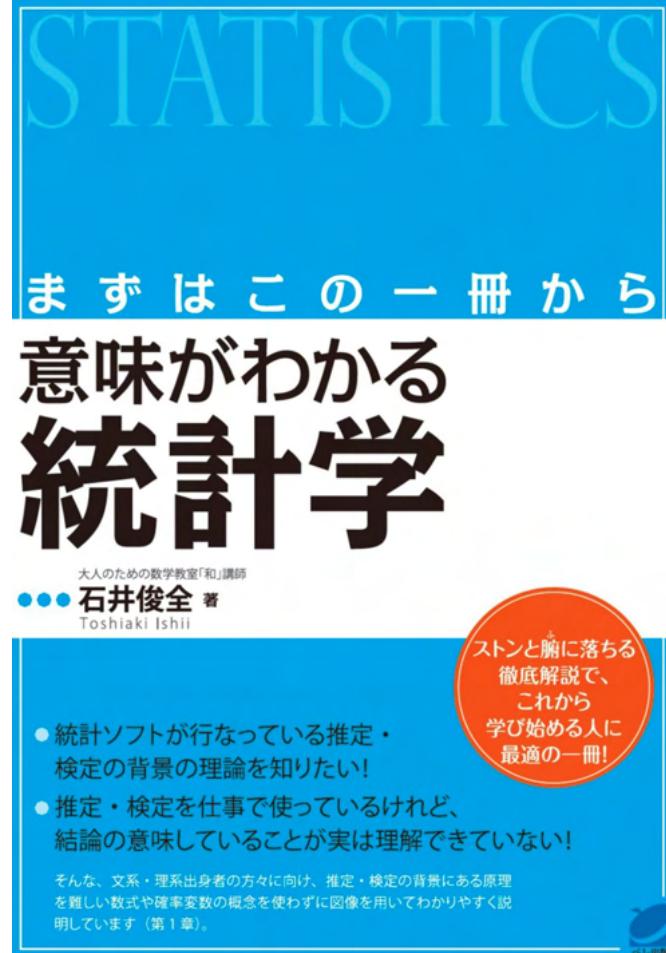
| | | |
|-----------------------------|--------------|------|
| 1. データサイエンスの紹介 | (2022年1月30日) | (ダム) |
| 2. データマイニングによる知識発見プロセス | (2022年1月30日) | (ダム) |
| 3. 確証的データ解析と探索的データ解析 | (2022年1月31日) | (ダム) |
| 4. 基礎的なデータ解析手法（1）：单变量解析 | (2022年1月31日) | (ダム) |
| 5. 基礎的なデータ解析手法（2）：多变量解析 | (2022年2月01日) | (ダム) |
| 6. 予測的データ解析手法（1）：決定木 | (2022年2月01日) | (ダム) |
| 7. 予測的データ解析手法（2）：ベイジアン分類 | (2022年2月02日) | (磯貝) |
| 8. 予測的データ解析手法（3）：サポートベクトル分類 | (2022年2月02日) | (磯貝) |
| 9. 記述的データ解析手法（1）：クラスタリング | (2022年2月03日) | (磯貝) |
| 10. 記述的データ解析手法（2）：特徴選択と次元削減 | (2022年2月03日) | (磯貝) |
| 11. 記述的データ解析手法（3）：グラフの解析 | (2022年2月04日) | (磯貝) |
| 12. データ科学と倫理問題 | (2022年2月04日) | (磯貝) |
| 13. 学生発表（1） | (2022年2月04日) | (ダム) |
| 14. 学生発表（2） | (2022年2月04日) | (ダム) |

K490: データサイエンス論

Lecture 3: 確証的データ解析と探索データ解析

Lecturer: Hieu-Chi Dam, Takashi Isogai

参考書の紹介



What is statistics? 統計学とは?

Statistics provides principles and methodology
for designing the process of:

統計学は下記プロセス設計のための原理や方法論を提供

- Data Collection データ収集
- Summarizing and Interpreting the data
データ要約と解釈
- Drawing Conclusions or Generalities
結論・概括

What is statistics? 統計学とは?



| Sport | 1990 | 1981 | 1972 | 1960 | 1948 |
|-------------|------|------|------|------|------|
| Football | 35% | 38% | 36% | 21% | 17% |
| Baseball | 16% | 16% | 21% | 34% | 39% |
| Baseketball | 15% | 9% | 8% | 9% | 10% |
| Others | 33% | 37% | 35% | 36% | 34% |

Data Collection
データ収集

Summarization – Interpretation
要約 – 解釈

Population and sample 母集団と標本

- The **population** is the *complete* collection of persons, objects... whose characteristics are of interest.
母集団とは、求める情報をもつ単位の完全な集合.
- A **sample** from a population is the set of objects whose data are actually collected in the course of an investigation.
母集団から抽出された標本とは、調査中に実際に収集したデータを備えた単位の集合.
- Good sample should be *randomly* collected (**random sample**, 無作為標本).



Topic: Which motorbikes are preferred by different groups of people for daily transportation?



Is this a good sample?

Sampling distribution

- **Random sampling** from a population refers to independent selections where each observation has the same distribution as the population.
母集団からの無作為抽出とはそれぞれの観察が母集団と同じ分布であるよう独立選択すること。
- When random sampling from a population, a statistic is a random variable. The probability distribution of a statistic is called its **sampling distribution**.
母集団から無作為抽出するときの統計量は確率変数。 統計量の確率分布をその標本分布と呼ぶ。

目次

- ① 統計学の基礎
- ② 回帰解析

統計学の基礎： 数学的推論

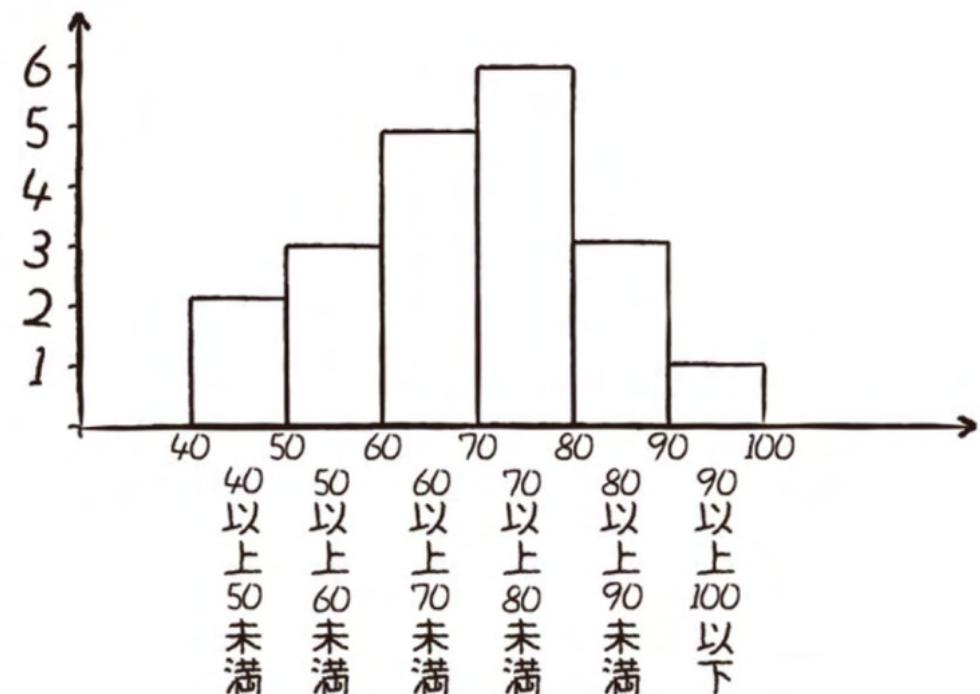
生データ

2人 3人 5人
43、47、52、52、54、61、67、67、68、69、
6人 3人 1人
70、71、71、73、76、78、82、84、84、91

整理された
データ

| 階級 | 階級値 | 度数 |
|--------------|-----|----|
| 40 以上 50 未満 | 45 | 2 |
| 50 以上 60 未満 | 55 | 3 |
| 60 以上 70 未満 | 65 | 5 |
| 70 以上 80 未満 | 75 | 6 |
| 80 以上 90 未満 | 85 | 3 |
| 90 以上 100 以下 | 95 | 1 |

可視化されたデータ



統計学の基礎： 数学的推論

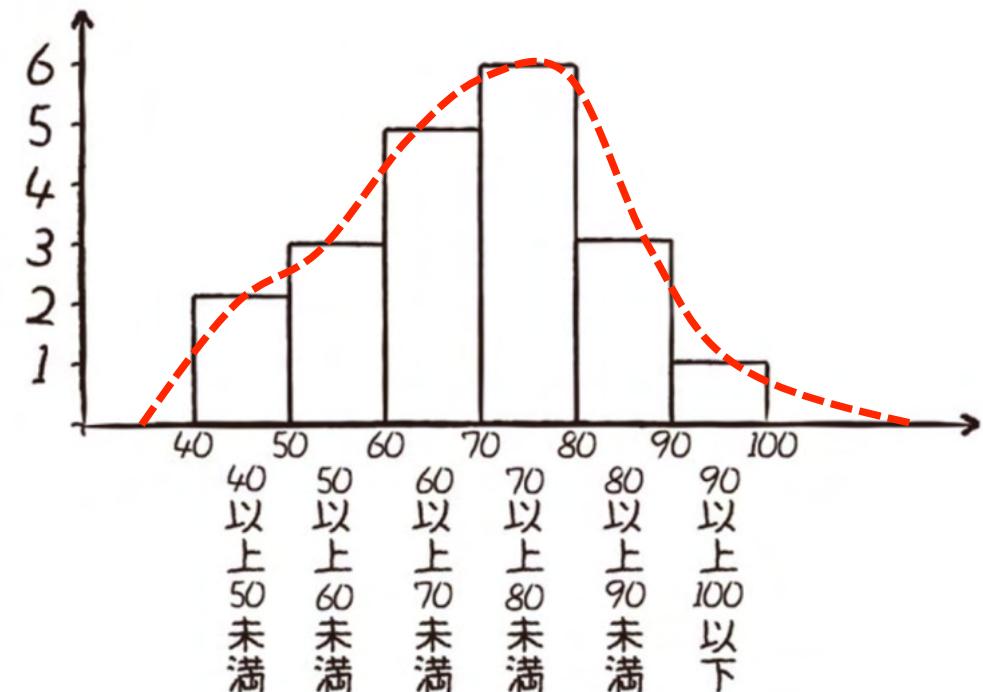
生データ

2人 3人 5人
43、47、52、52、54、61、67、67、68、69、
6人 3人 1人
70、71、71、73、76、78、82、84、84、91

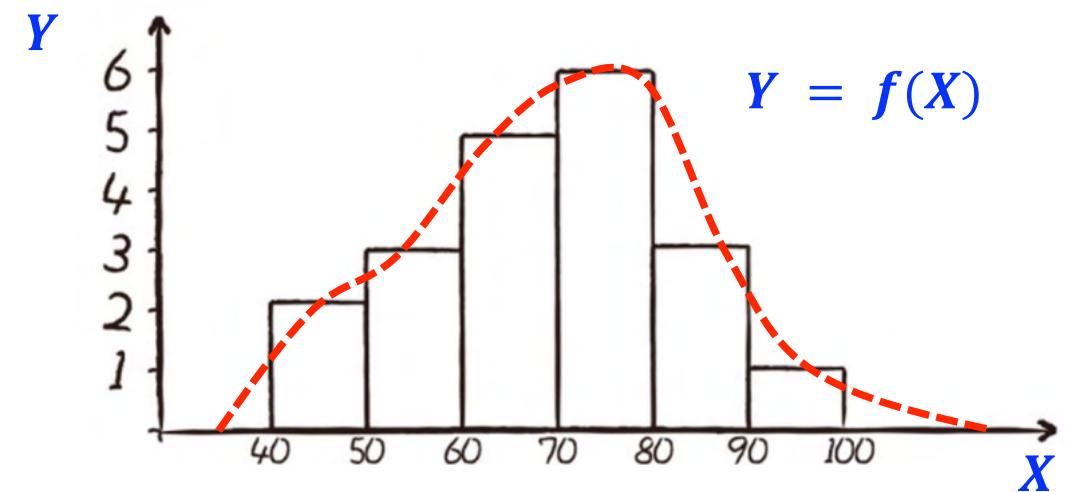
整理された
データ

| 階級 | 階級値 | 度数 |
|--------------|-----|----|
| 40 以上 50 未満 | 45 | 2 |
| 50 以上 60 未満 | 55 | 3 |
| 60 以上 70 未満 | 65 | 5 |
| 70 以上 80 未満 | 75 | 6 |
| 80 以上 90 未満 | 85 | 3 |
| 90 以上 100 以下 | 95 | 1 |

可視化されたデータ



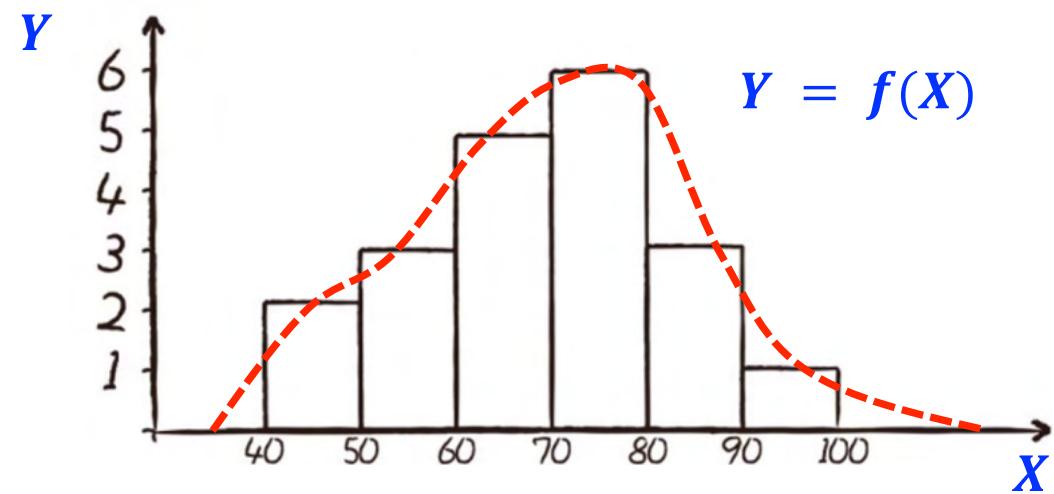
統計学の基礎： 数学的推論



統計学の基礎： 数学的推論

数学的推論

$$P(X > 60) = \int_{60}^{\infty} f(X) dX$$



Random variables 確率変数

- An **experiment** is the process of observing a phenomenon that has variation in its outcomes. 実験とは、その結果にはばらつきがある現象を観察する過程である。
- The experiment's outcomes can be numeric (1, 2, ..., 6) or non-numeric (ten or house). For computation, we qualify experiment's outcomes by assigning each of them a numerical value related to a characteristic of interest.
実験の結果の数値的特徴を考える。実験結果を量で表す際には、注目している特性に関連した数値をそれぞれに割当てる。
- A **random variable** X is a **function** that associates a numerical value with each outcome of an experiment.
確率変数 X は、実験結果のそれぞれに数値を結び付ける関数である。
- “Random” means before the experiment we do not know the outcome of an experiment or its associated value of X .
「無作為(Random)」とは、実験前には実験結果もしくはそのときの X の値が不明であることを意味する。

A random variable $X: \Omega \rightarrow \mathbb{R}$ is a measurable function from the set of possible outcomes to \mathbb{R} .

Random variables 確率変数

- A random variable is **discrete** if it has either a finite number of values or infinitely many values that can be arranged in a sequence.
確率変数のとりうる値が有限個もしくは無限であっても数列規則に従う場合の確率変数は離散型である。
 - Example: Number of cars in JAIST parking during 1 day.
JAIST駐車場の一日の車の数.
- A **continuous** random variable is a random variable that represents some measurement on a continuous scale and therefore capable of assuming all values in an interval.
連續型確率変数は、連續尺度上の計測値や、ある区間内のどんな値でもとりうる確率変数のこと。
 - Example: Rainfall after each rain during the raining season.
雨季における降雨ごとの降雨量.

統計学の基礎：平均と分散

平均、分散・標準偏差の公式

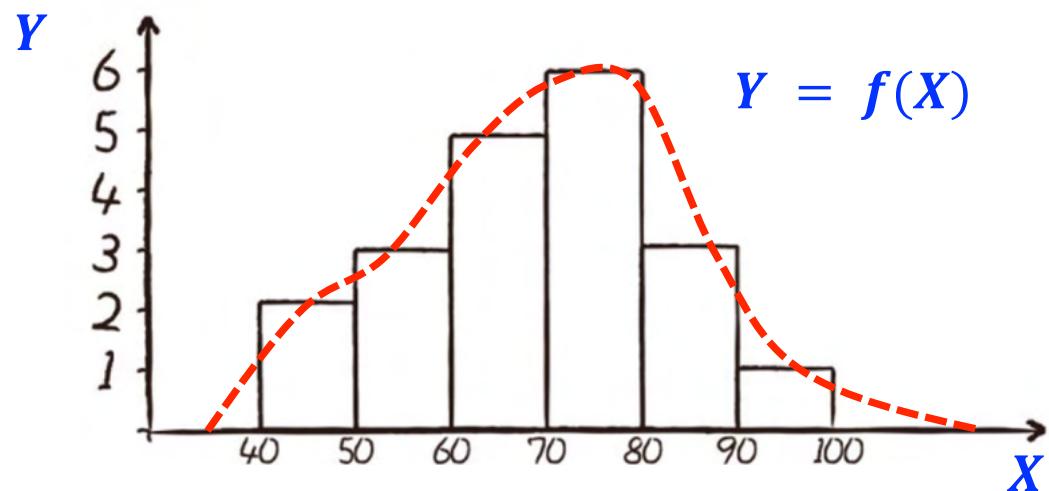
資料の個数が N 個、資料の変量が x_1, x_2, \dots, x_N のとき、平均、分散・標準偏差の計算の仕方をまとめておきます。

資料の平均を \bar{x} 、分散を s^2 、標準偏差を s であります。

$$\bar{x} = \frac{x_1 + x_2 + x_3 + \dots + x_N}{N}$$

$$s^2 = \frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + (x_3 - \bar{x})^2 + \dots + (x_N - \bar{x})^2}{N}$$

$$s = \sqrt{\frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + (x_3 - \bar{x})^2 + \dots + (x_N - \bar{x})^2}{N}}$$



統計学の基礎：平均と分散

平均、分散・標準偏差の公式

資料の個数が N 個、資料の変量が x_1, x_2, \dots, x_N のとき、平均、分散・標準偏差の計算の仕方をまとめておきます。

資料の平均を \bar{x} 、分散を s^2 、標準偏差を s であります。

$$\bar{x} = \frac{x_1 + x_2 + x_3 + \dots + x_N}{N}$$

$$s^2 = \frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + (x_3 - \bar{x})^2 + \dots + (x_N - \bar{x})^2}{N}$$

$$s = \sqrt{\frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + (x_3 - \bar{x})^2 + \dots + (x_N - \bar{x})^2}{N}}$$

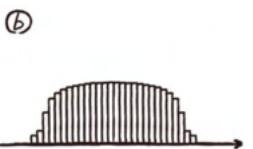
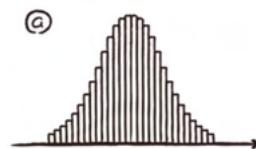
問題

(1)～(3) のそれぞれの 2 つのヒストグラムで標準偏差の大きいのは a 、 b どちらですか。ただし、2 つのヒストグラムはヨコ軸の目盛りが等しいものとします。

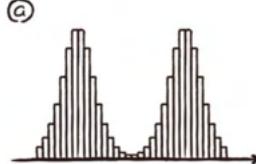
(1)



(2)



(3)



統計学の基礎：平均と分散

平均、分散・標準偏差の公式

資料の個数が N 個、資料の変量が x_1, x_2, \dots, x_N のとき、平均、分散・標準偏差の計算の仕方をまとめておきます。

資料の平均を \bar{x} 、分散を s^2 、標準偏差を s であります。

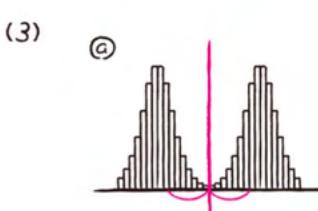
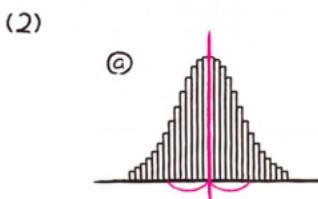
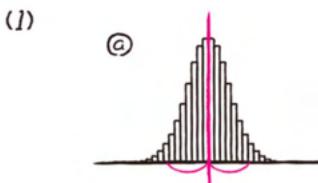
$$\bar{x} = \frac{x_1 + x_2 + x_3 + \dots + x_N}{N}$$

$$s^2 = \frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + (x_3 - \bar{x})^2 + \dots + (x_N - \bar{x})^2}{N}$$

$$s = \sqrt{\frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + (x_3 - \bar{x})^2 + \dots + (x_N - \bar{x})^2}{N}}$$

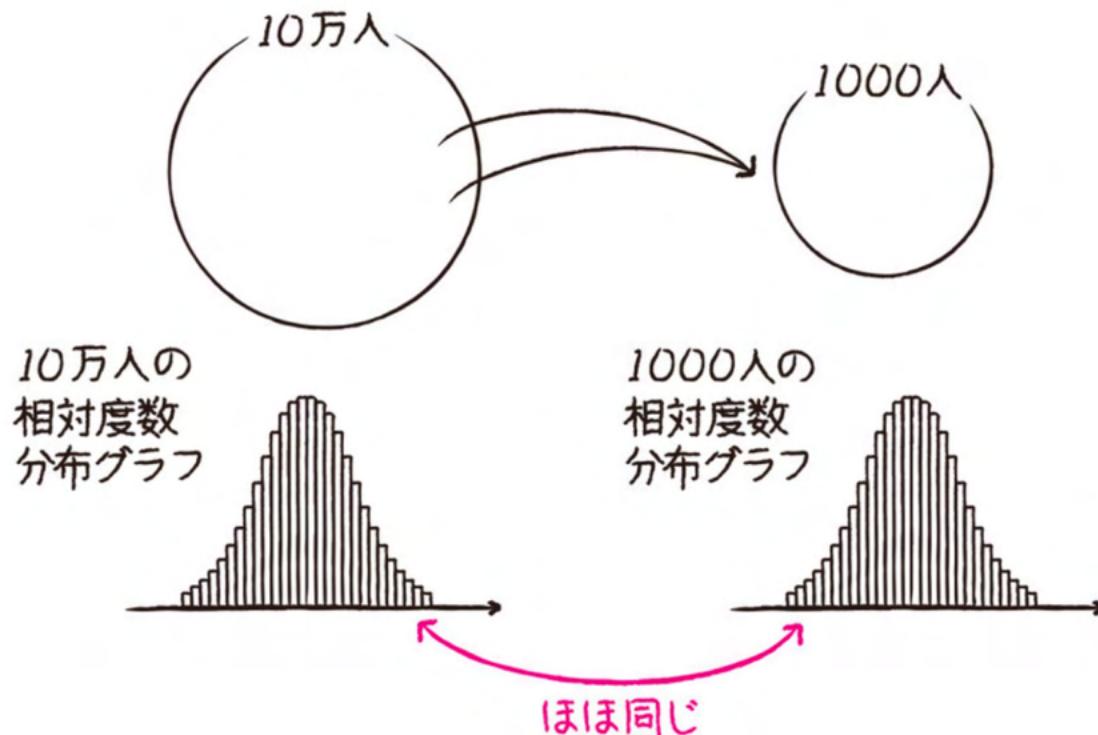
問題

(1)～(3) のそれぞれの 2 つのヒストグラムで標準偏差の大きいのは a 、 b どちらですか。ただし、2 つのヒストグラムはヨコ軸の目盛りが等しいものとします。

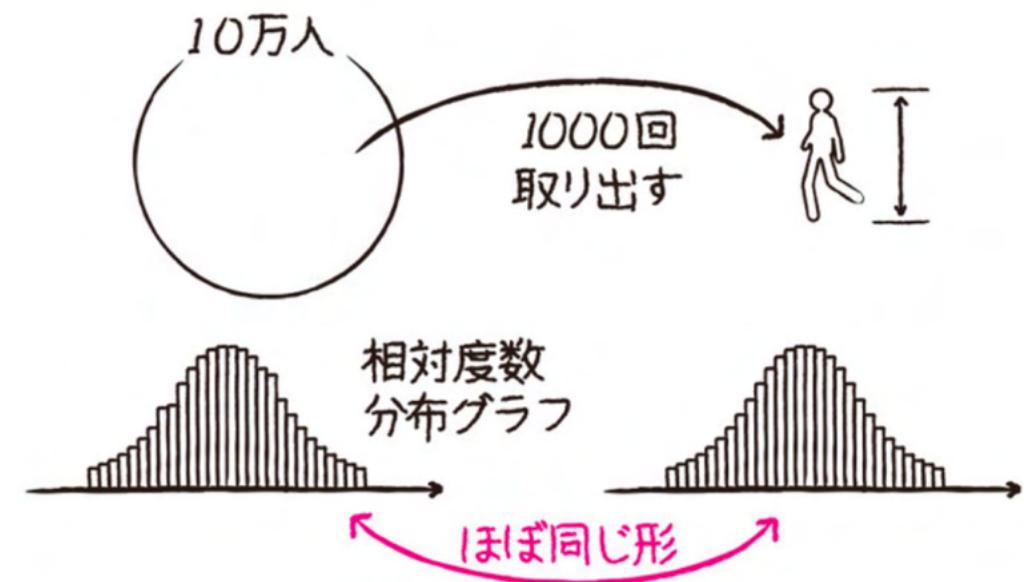


統計学の基礎：データサンプリング

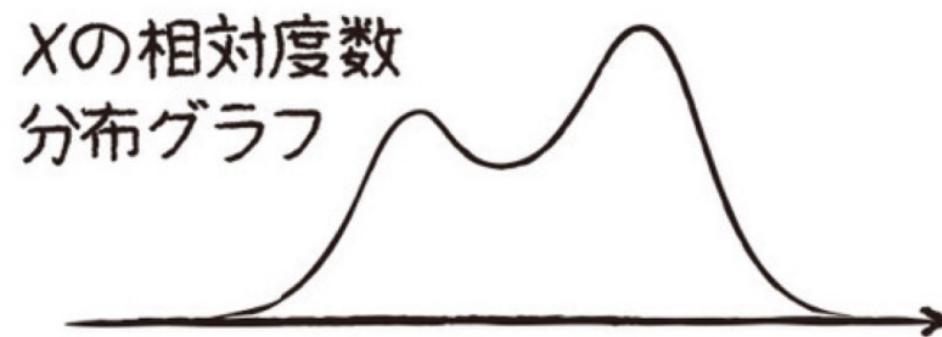
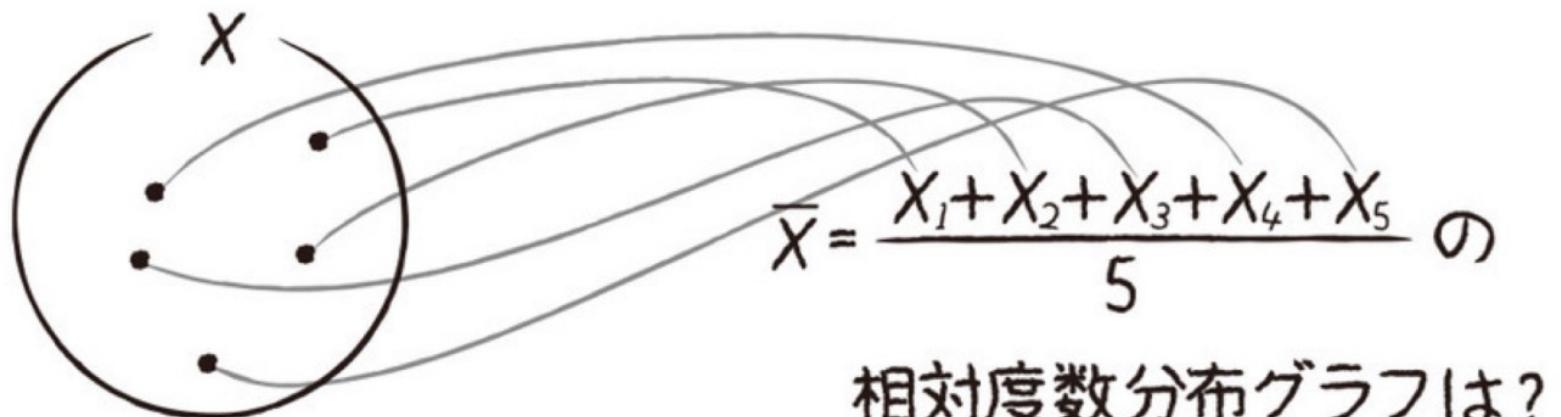
非復元抽出



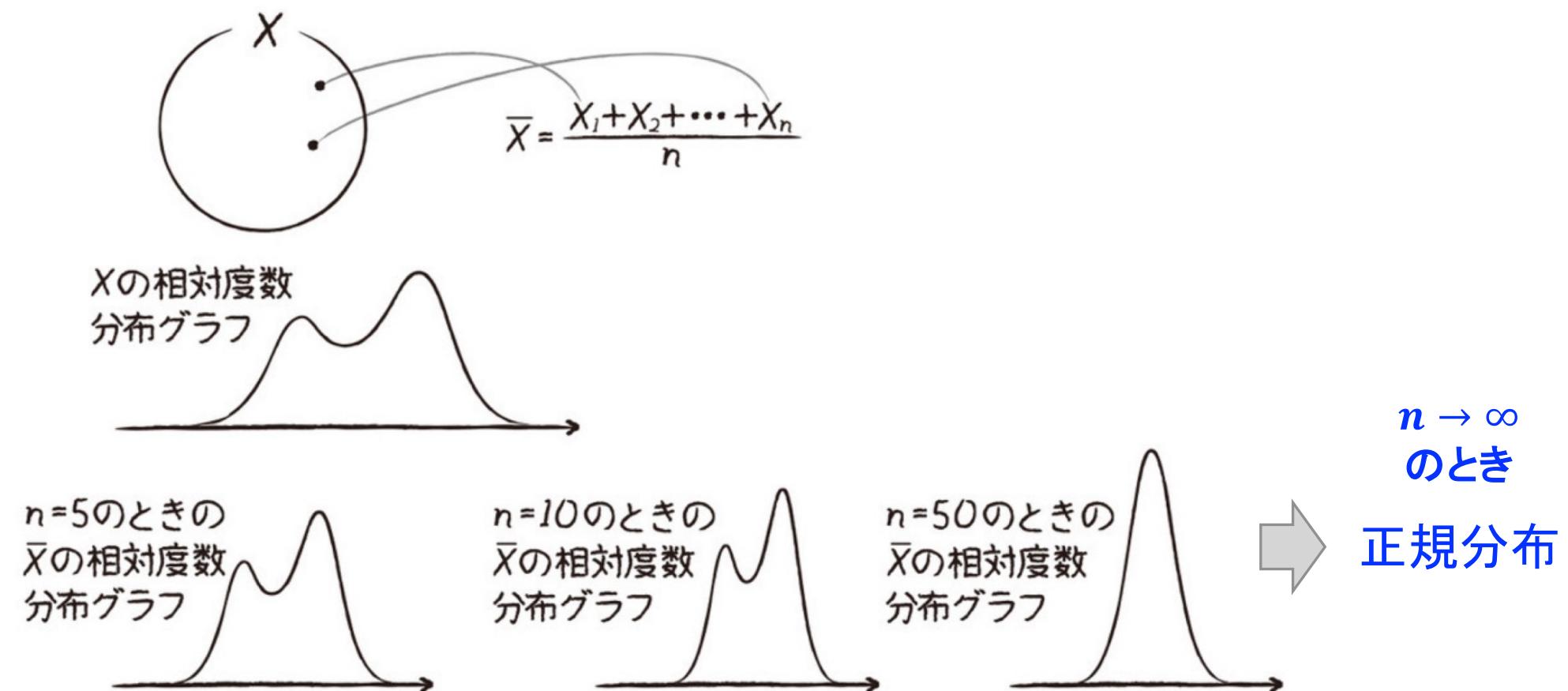
復元抽出



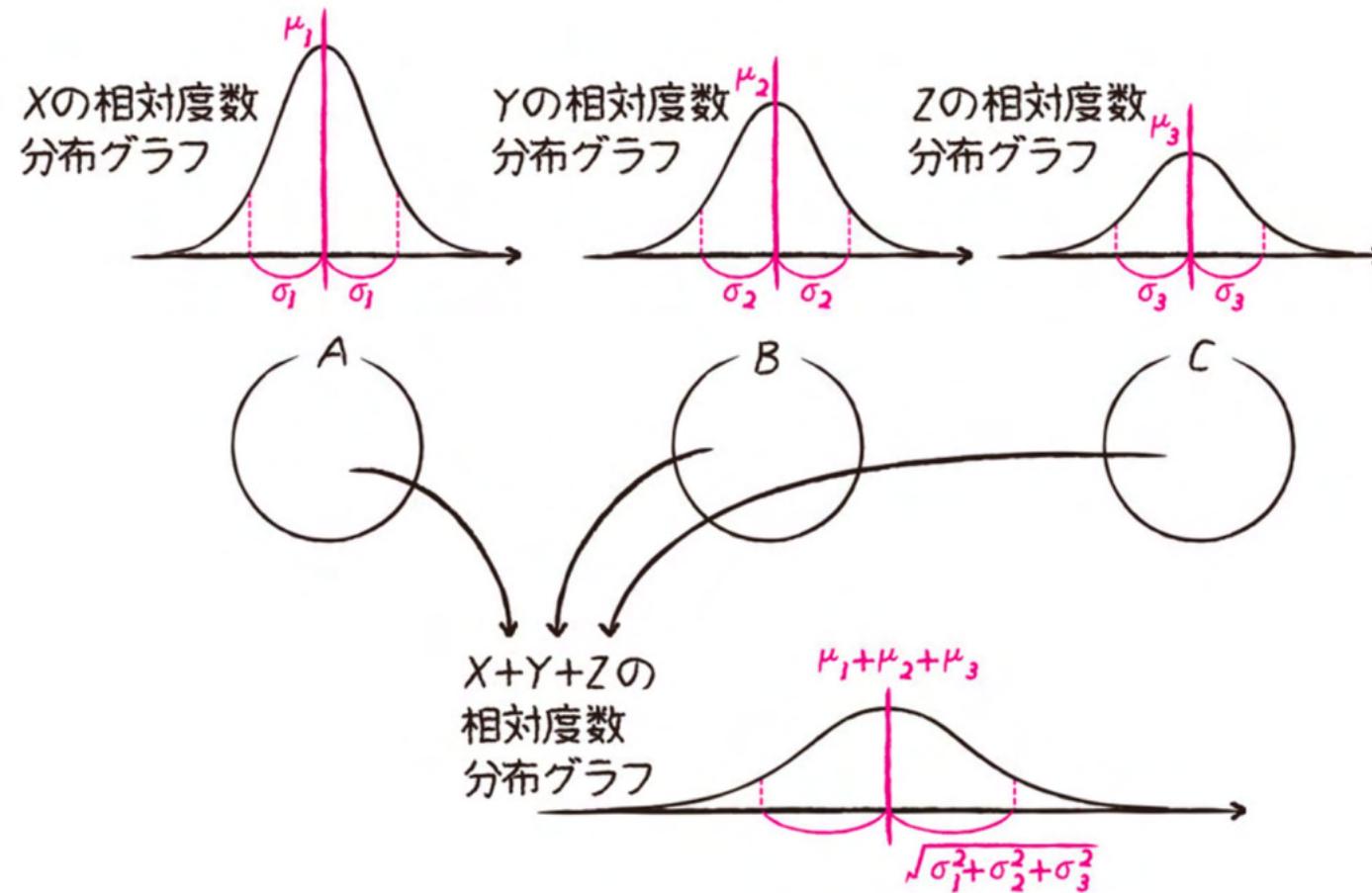
統計学の基礎： 中心極限定理



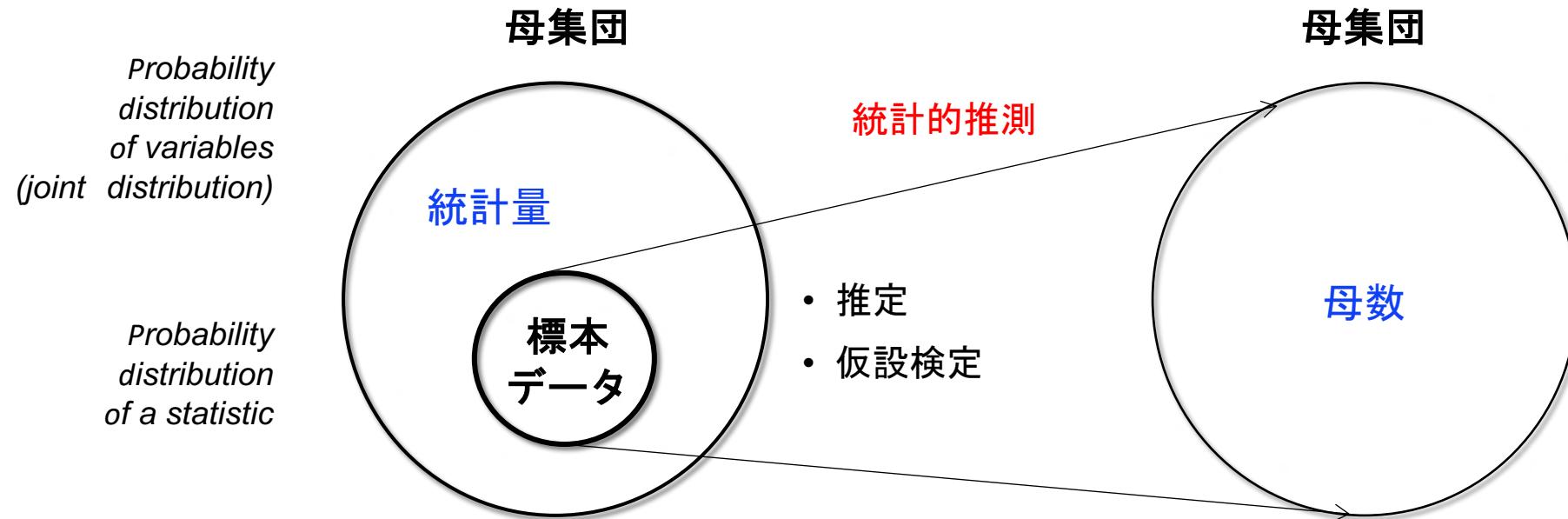
統計学の基礎： 中心極限定理



統計学の基礎：正規分布



統計学の基礎： 統計的推測



統計学的推測とは、標本データを解析して母数に関する結論を導くこと。
(母集団の数値的特徴を母数と呼ぶ)

統計学の基礎： 推定

不偏推定量による推定

最尤法による推定

統計学の基礎： 推定

平均、分散・標準偏差の公式

資料の個数が N 個、資料の変量が x_1, x_2, \dots, x_N のとき、平均、分散・標準偏差の計算の仕方をまとめておきます。

資料の平均を \bar{x} 、分散を s^2 、標準偏差を s でおきます。

$$\bar{x} = \frac{x_1 + x_2 + x_3 + \dots + x_N}{N}$$

$$s^2 = \frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + (x_3 - \bar{x})^2 + \dots + (x_N - \bar{x})^2}{N}$$

$$s = \sqrt{\frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + (x_3 - \bar{x})^2 + \dots + (x_N - \bar{x})^2}{N}}$$

不偏推定量による推定

統計学の基礎： 推定

不偏推定量による推定

平均、分散・標準偏差の公式

資料の個数が N 個、資料の変量が x_1, x_2, \dots, x_N のとき、平均、分散・標準偏差の計算の仕方をまとめておきます。

資料の平均を \bar{x} 、分散を s^2 、標準偏差を s でおきます。

$$\bar{x} = \frac{x_1 + x_2 + x_3 + \dots + x_N}{N}$$

$$s^2 = \frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + (x_3 - \bar{x})^2 + \dots + (x_N - \bar{x})^2}{N}$$

$$s = \sqrt{\frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + (x_3 - \bar{x})^2 + \dots + (x_N - \bar{x})^2}{N}}$$

統計学の基礎： 推定

不偏推定量による推定

推定に用いる統計量

① σ が既知のとき、 μ を区間推定するには、

「統計量 $T = \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{n}}}$ は、標準正規分布 $N(0, 1^2)$ に従う」

② σ が未知のとき、 μ を区間推定するには、

「統計量 $T = \frac{\bar{X} - \mu}{\frac{U}{\sqrt{n}}}$ は、自由度 $n - 1$ の t 分布に従う」

ここで、②の U は、例の不偏分散 U^2 の U です。書き下せば、

$$U = \sqrt{\frac{(X_1 - \bar{X})^2 + (X_2 - \bar{X})^2 + \cdots + (X_n - \bar{X})^2}{n - 1}}$$

となります。

③ μ が既知のとき、 σ を区間推定するには、

「統計量 $T = \frac{(X_1 - \mu)^2 + (X_2 - \mu)^2 + \cdots + (X_n - \mu)^2}{\sigma^2}$ は、

自由度 n の χ^2 (カイ2乗)分布に従う」

④ μ が未知のとき、 σ を区間推定するには、

「統計量 $T = \frac{(X_1 - \bar{X})^2 + (X_2 - \bar{X})^2 + \cdots + (X_n - \bar{X})^2}{\sigma^2}$ は、

自由度 $n - 1$ の χ^2 (カイ2乗)分布に従う」

統計学の基礎： 推定

与えられたデータは**現実に出現**している



考えられる数理モデルの集合を検討する

最尤法による推定



与えられた**データを再現する数理モデル**は
比較的高い確率が割り振られている

統計学の基礎：コイン投げ



100回: 70表 & 30裏

$$p(\text{表}) = ?$$

統計学の基礎：コイン投げ



100回: 70表 & 30裏

$$p(\text{表}) = \frac{70}{100} = 0.7$$

統計学の基礎：コイン投げ



なぜ?

本当ですか? どれぐらい信頼できるか?
この計算はデータ解析なのか?
このデータ解析が何を基にしているか?
背景にはどのような仮説があるか?

100回: 70表 & 30裏

$$p(\text{表}) = \frac{70}{100} = 0.7$$

統計学の基礎：コイン投げ



仮定

$$p(\text{表}) = \theta$$

尤度

$$\mathcal{L}(\theta) = {}_{100}C_{70}\theta^{70}(1 - \theta)^{30}$$

考え方：

与えられたデータを再現する数理モデルは
比較的高い確率が割り振られている

100回: 70表 & 30裏

統計学の基礎：コイン投げ



100回: 70表 & 30裏

仮定

$$p(\text{表}) = \theta$$

尤度

$$\mathcal{L}(\theta) = {}_{100}C_{70}\theta^{70}(1 - \theta)^{30}$$

$$\theta \triangleq \arg \max_{\theta \in [0,1]} \log p(\mathcal{D}|\theta)$$

$$\triangleq \arg \max_{\theta \in [0,1]} \log \mathcal{L}(\theta)$$

$$\triangleq \arg \max_{\theta \in [0,1]} \log {}_{100}C_{70}\theta^{70}(1 - \theta)^{30}$$

統計学の基礎：コイン投げ



100回: 70表 & 30裏

仮定

$$p(\text{表}) = \theta$$

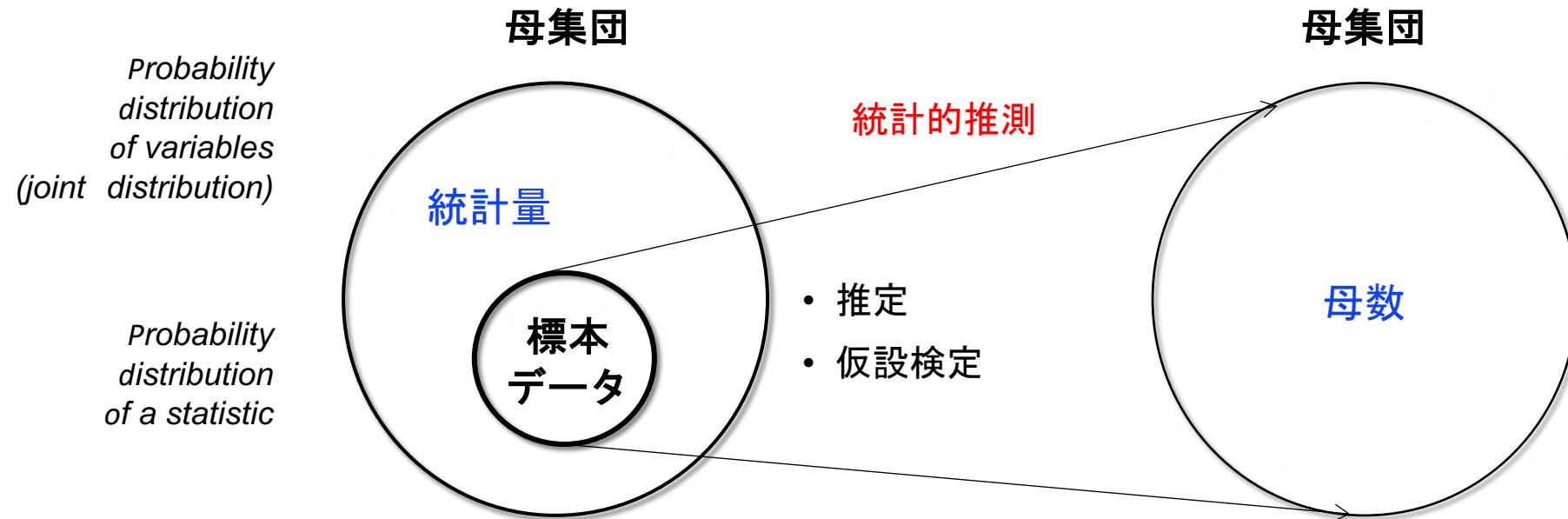
尤度

$$\mathcal{L}(\theta) = {}_{100}C_{70}\theta^{70}(1-\theta)^{30}$$

$$\frac{d}{d\theta} \log \mathcal{L}(\theta) = \frac{70}{\theta} - \frac{30}{1-\theta} = 0$$

$$\theta = \frac{70}{100} = 0.7$$

統計学の基礎： 統計的推測



統計学的推測とは、標本データを解析して母数に関する結論を導くこと。
(母集団の数値的特徴を母数と呼ぶ)

Statistical inference 統計的推測

Statistical inference deals with drawing conclusions about population parameters from an analysis of the sample data.

統計学的推測とは、標本データを解析して母数に関する結論を導くこと。

Two most important types of inferences: もっとも重要な2種類の推測

1. Estimation of parameter(s) 母数の推定

- ***Point estimation*** 点推定: Point estimation involves the use of sample data to calculate a single value (statistic) which is to serve as a “best guess” or “best estimate” of an unknown population parameter.
- ***Interval estimation*** 区間推定: interval estimation is the use of sample data to calculate an interval of possible (or probable) values of an unknown population parameter.

2. Testing of statistical hypotheses 統計的仮説検定

Point estimation of a population mean

母平均値の点推定

- Random data sample: X_1, X_2, \dots, X_n .

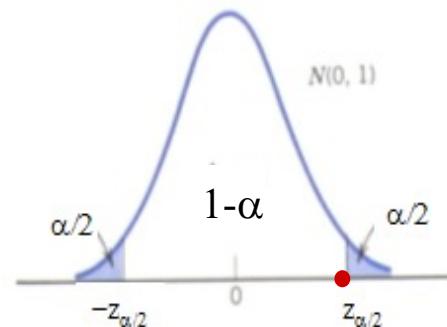
$$\text{Estimated sample mean } SE(\bar{X}) = \frac{S}{\sqrt{n}}, S = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n-1}}$$

For large n , the $100(1 - \alpha)\%$ error margin is $z_{\alpha/2} \frac{\sigma}{\sqrt{n}}$

$z_{\alpha/2}$ = the upper $\alpha/2$ point of the standard normal distribution. That is, the area to the right of $z_{\alpha/2}$ is $\alpha/2$, and the area between $-z_{\alpha/2}$ and $z_{\alpha/2}$ is $1 - \alpha$.

$Z_{\alpha/2}$ は標準正規分布において全領域に対する比が上から $\alpha/2$ となる地点。つまり $Z_{\alpha/2}$ の右側は全体の $\alpha/2$ を占めるため、 $-Z_{\alpha/2}$ と $Z_{\alpha/2}$ の間の領域は全領域の $1-\alpha$ である。

| Some values of $z_{\alpha/2}$ よく使う確率とz値 | | | | | |
|---|------|------|-------|------|------|
| $1 - \alpha$ | 0.80 | 0.85 | 0.90 | 0.95 | 0.99 |
| $z_{\alpha/2}$ | 1.28 | 1.44 | 1.645 | 1.96 | 2.58 |



Confidence interval for a parameter 母数の信頼区間

An interval (L, U) is a $100(1 - \alpha)\%$ confidence interval for a parameter

if

$$P[L < \text{parameter} < U] = 1 - \alpha$$

and the endpoints L and U are computable from the sample.

ある母数について区間 (L, U) が $100(1-\alpha)\%$ の信頼区間であるのは

$P[L < \text{母数} < U] = 1 - \alpha$ であり、区間の下限 L および上限 U が標本から求められる場合

- When the parameter is μ we have 母数が母平均 m であるとき、その信頼区間は

$$\left(L = \bar{X} - Z_{\alpha/2} \frac{s}{\sqrt{n}}, \bar{X} + Z_{\alpha/2} \frac{s}{\sqrt{n}} \right)$$

統計学の基礎：仮説検定

状況：街（日本）で向こうを歩いている身長2mの人を見えました。

疑問：歩いてきた人は外国人ではないか？

統計学の基礎：仮説検定

状況：街（日本）で向こうを歩いている身長2mの人を見ました。

疑問：歩いてきた人は外国人ではないか？

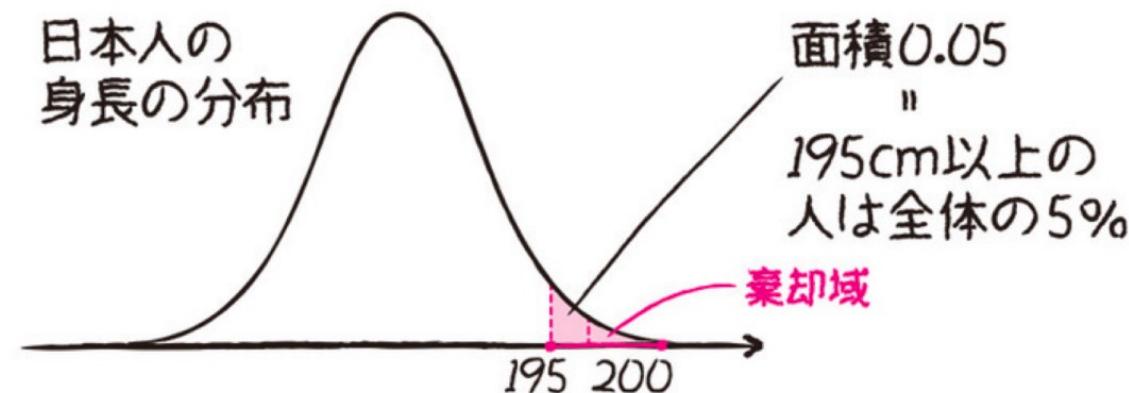
理由：日本人だとすれば、大きすぎる。常識から考えて、あり得ないと思う。

統計学の基礎：仮説検定

状況：街（日本）で向こうを歩いている身長2mの人を見ました。

疑問：歩いてきた人は外国人ではないか？

理由：日本人だとすれば、大きすぎる。常識から考えて、あり得ないと思う。

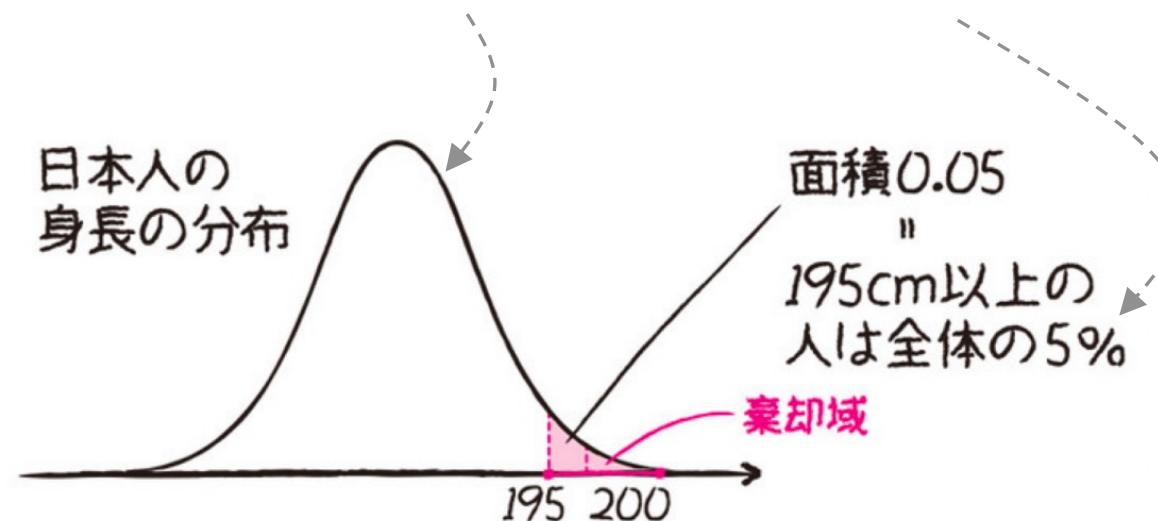


統計学の基礎：仮説検定

状況：街（日本）で向こうを歩いている身長2mの人を見ました。

疑問：歩いてきた人は外国人ではないか？

理由：日本人だとすれば、大きすぎる。常識から考えて、あり得ないと思う。



統計学の基礎：仮説検定

「むこうを歩いている人が $\leftarrow H_0$ を仮定 帰無仮説

日本人であるとすれば、

大きすぎるなあ。

ありえないよ。

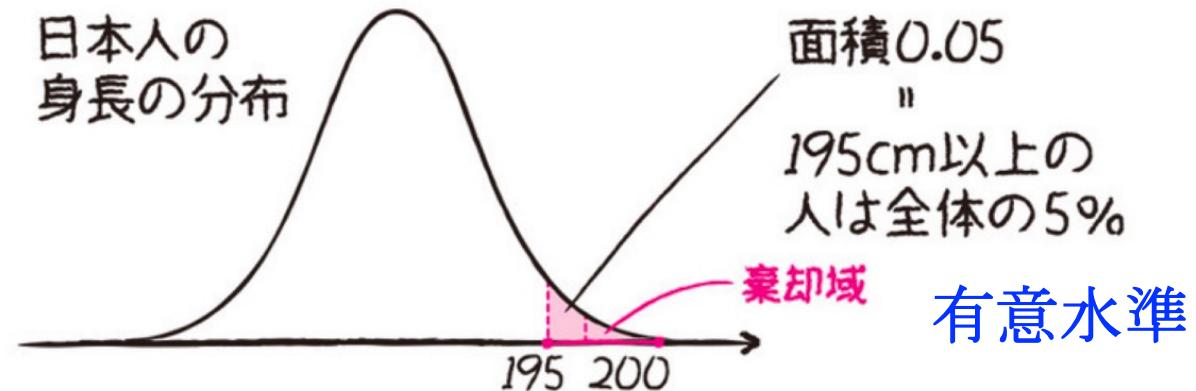
日本人じゃないな。

外国人じゃないの」

\leftarrow 起こる確率が 5% 以下の
できごとが起きた

$\leftarrow H_0$ 棄却

$\leftarrow H_1$ 採択 対立仮説



統計学の基礎：仮説検定

「むこうを歩いている人が $\leftarrow H_0$ を仮定 帰無仮説

日本人であるとすれば、

大きすぎるなあ。

ありえないよ。

日本人じゃないな。

外国人じゃないの」

\leftarrow 起こる確率が 5% 以下の
できごとが起きた

$\leftarrow H_0$ 番却

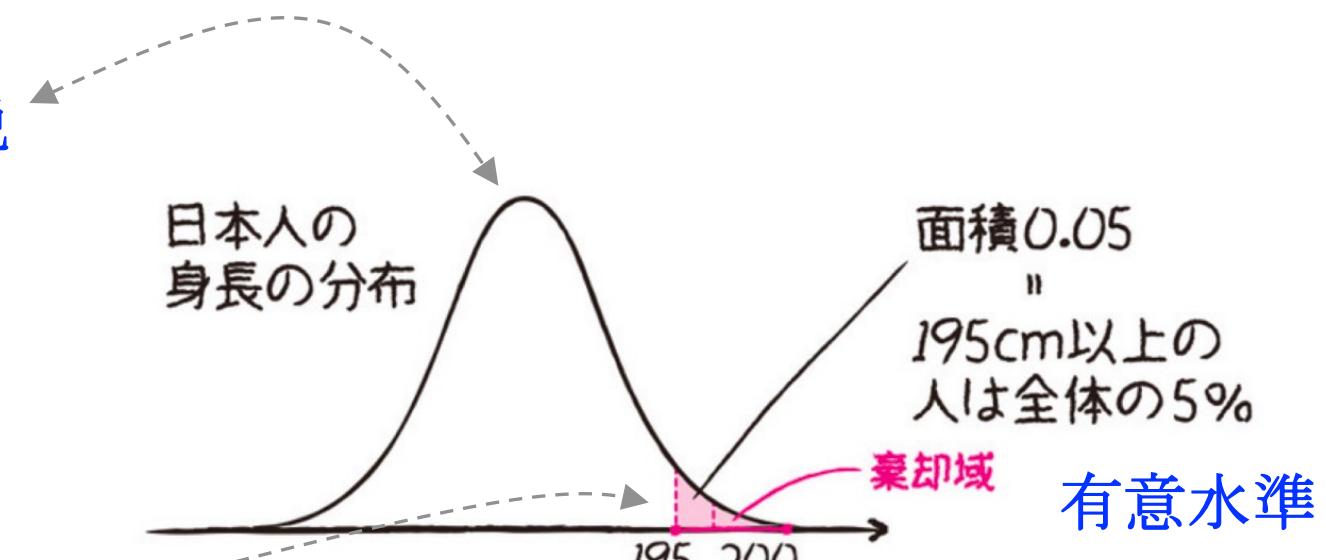
$\leftarrow H_1$ 採択 対立仮説

日本人の
身長の分布

面積 0.05
" 195cm 以上の
人は全体の 5%

有意水準

| 判断結果 | 帰無仮説 H_0 を採択 | 帰無仮説 H_0 を棄却 |
|----------------|----------------|----------------|
| 事実 | | |
| 帰無仮説 H_0 が本当 | 正しい判断 | 第 1 種の誤り |
| 帰無仮説 H_0 が嘘 | 第 2 種の誤り | 正しい判断 |



統計学の基礎：仮説検定と背理法

状況：街（日本）で向こうを歩いている身長2mの人を見ました。

疑問：歩いてきた人は外国人ではないか？

理由：日本人だとすれば、大きすぎる。常識から考えて、あり得ないと思う。

統計学の基礎：仮説検定と背理法

状況：街（日本）で向こうを歩いている身長2mの人を見ました。

疑問：歩いてきた人は外国人ではないか？

理由：日本人だとすれば、大きすぎる。常識から考えて、あり得ないと思う。

背理法：

- ① 日本人だと仮定する。
- ② 日本人なら身長が“それほど”背が高くない。
- ③ 見えた人の身長が2m
- ④ ③が②と矛盾しているので①が正しくない
(矛盾しない場合は①が正しくないことはまだ言えない)

統計学の基礎：仮説検定と背理法

状況：街（日本）で向こうを歩いている身長2mの人を見ました。

疑問：歩いてきた人は外国人ではないか？

理由：日本人だとすれば、大きすぎる。常識から考えて、あり得ないと思う。

背理法：

- ① 日本人だと仮定する。
- ② 日本人なら身長が“それほど”背が高くない。
- ③ 見えた人の身長が2m
- ④ ③が②と矛盾しているので①が正しくない
(矛盾しない場合は①が正しくないことはまだ言えない)

対立仮説 H_1 ← 証明したいこと

帰無仮説 H_0 を仮定する ← 反対のことを仮定する

このもとで確率を計算

ありえそうもないことが起きている

対立仮説 H_1 を採択 ← 証明完了

} 矛盾を導く

統計学の基礎：仮説検定と背理法

状況：街（日本）で向こうを歩いている身長2mの人を見ました。

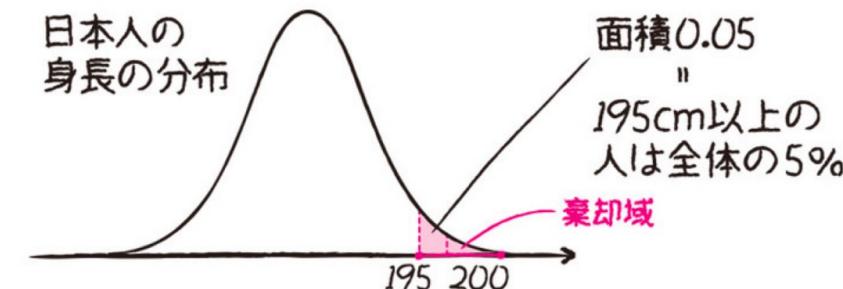
疑問：歩いてきた人は外国人ではないか？

理由：日本人だとすれば、大きすぎる。常識から考えて、あり得ないと思う。

背理法：

- ① 日本人だと仮定する。
- ② 日本人なら身長が“それほど”背が高くない。
- ③ 見えた人の身長が2m
- ④ ③が②と矛盾しているので①が正しくない
(矛盾しない場合は①が正しくないことはまだ言えない)

危険率付きの背理法



The steps for testing hypotheses

仮説検定の手順

1. Formulate the null hypothesis H_0 and alternative hypothesis H_1 .
帰無仮説 H_0 と対立仮説 H_1 を設定する
2. Test criterion: State the test statistic and the form of the rejection region.
検定基準: 検定統計量と棄却域の形式を定義
3. With a specified α , determine the rejection region. α を指定し棄却域を決定
4. Calculate the test statistic from the data.
データから検定統計量を算出
5. Draw a conclusion: State whether or not H_0 is rejected at the specified α and interpret the conclusion in the context of the problem. Also, it is a good statistical practice to calculate the P -value and strengthen the conclusion.
結論付け: 指定した α について H_0 を棄却するかどうかを述べ、問題の文脈で結論を解釈する。また、統計学的実践としては、P値を計算し結論を補強すると尚よい。

Example: weight loss diet 減量用食餌療法

- A new diet program states that the participants are expected to lose over 22 pounds in 5 weeks. From the data of the 5-week weight losses of 56 participants, the sample mean and the std. deviation are found 23.5 and 10.2 pounds.

新減量プログラムの参加者は5週間で22ポンド超減量できると主張。56人の5週間の減量データの標本平均は23.5ポンド、標準偏差が10.2ポンド。

- Is the statement substantiated on the basis of these findings? Test with level of significance 0.05. Calculate the P -value and interpret the result.

この結果から、減量プログラムの主張は実証されるか？有意水準0.05で仮説検定し、P値を求め結果を解釈せよ。

- SOLUTION: We have

$$n = 56; \mu_0 = 22; \bar{x} = 23.5; S = 10.2; \alpha = 0.05$$

- Hypothesis $H_0 : \mu = 22$ versus $H_1 : \mu > 22$

Example (continued)

2

Test statistic $Z = \frac{\bar{X} - \mu_0}{S/\sqrt{n}} = \frac{\bar{X} - 22}{S/\sqrt{56}}$

H_1 is right - sided \Rightarrow 対立仮説は右側検定なので棄却域は
Rejection region for H_0 is $R : Z \geq c$.

3

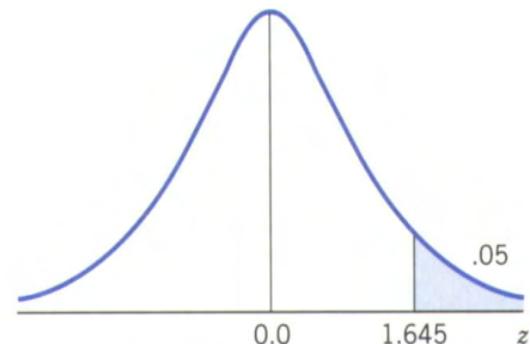
$$z_\alpha = z_{0.05} = 1.645 \Rightarrow R : Z \geq 1.645$$

4

$$z = \frac{23.5 - 22}{10/\sqrt{56}} = 1.1 \notin \text{Region } R : Z \geq 1.645$$

5

We do not reject H_0 , with $\alpha = 0.05$ the stated claim that $\mu > 22$ is not substantiated. P - value = $P[Z \geq 1.10] = 0.1357$.
0.1357 is the smallest α at which H_0 could be rejected. As it is not negligible, the data do not provide a strong basis for rejection of H_0 .



平均減量値が22ポンドを超えるという主張 H_0 を有意水準0.05では棄却しない。
 H_0 を棄却できる α の最小値は0.1357で有意水準を越えるため無視できない。
よって、このデータは棄却の根拠にはならない。

演繹法, 帰納法, およびアブダクション

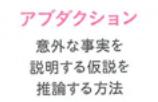
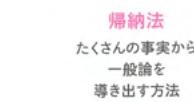


大前提が正しければ結論が正しい(保証される)

妥当性の検証
が必要

不確かさの定
量評価が必要

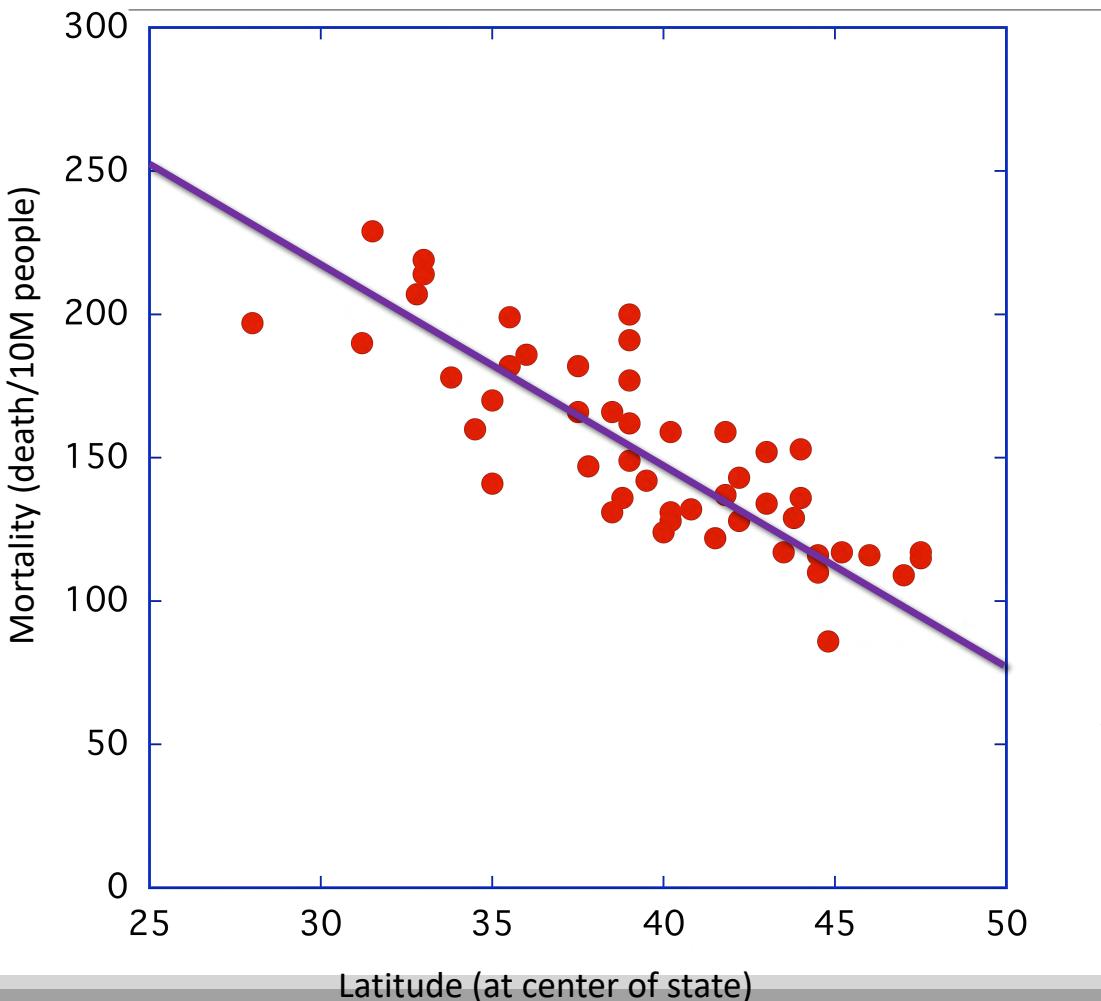
諸前提が正しいであっても必ずしも結論が正しいと限らない(保証されない)



目次

- ① 統計学の基礎
- ② 回帰解析

線形回帰



Hypothesis

$$y = \beta_0 + \beta_1 x$$



Evaluation

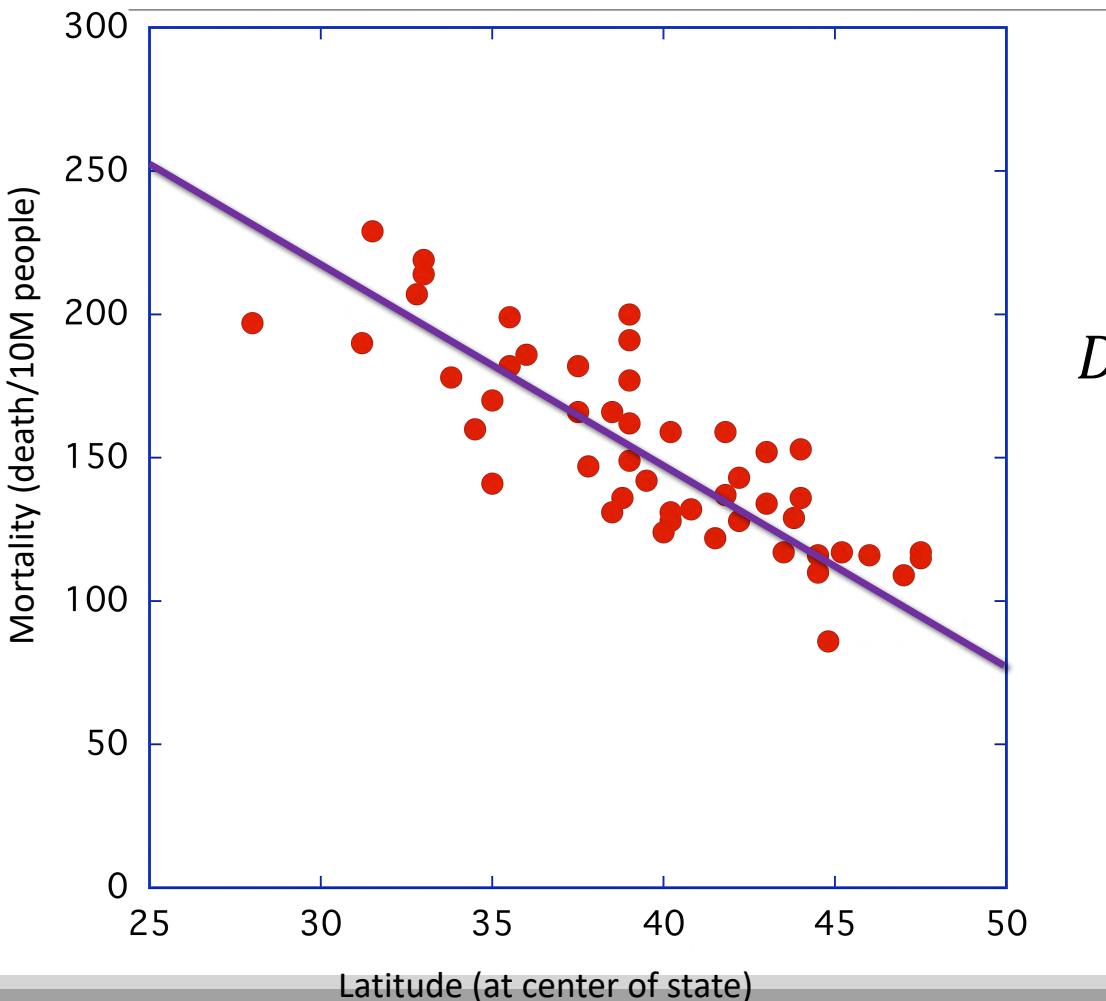
$$D = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2$$

Original hypothesis

Smaller $D \equiv$ better linear model?

| | A | B | C | D | E |
|----|----------------|----------|-----------|-------|-----------|
| 1 | State | Latitude | Longitude | Ocean | Mortality |
| 2 | Alabama | 33 | 87 | 1 | 219 |
| 3 | Arizona | 34.5 | 112 | 0 | 160 |
| 4 | Arkansas | 35 | 92.5 | 0 | 170 |
| 5 | California | 37.5 | 119.5 | 1 | 182 |
| 6 | Colorado | 39 | 105.5 | 0 | 149 |
| 7 | Connecticut | 41.8 | 72.8 | 1 | 159 |
| 8 | Delaware | 39 | 75.5 | 1 | 200 |
| 9 | Wash.D.C. | 39 | 77 | 0 | 177 |
| 10 | Florida | 28 | 82 | 1 | 197 |
| 11 | Georgia | 33 | 83.5 | 1 | 214 |
| 12 | Idaho | 44.5 | 114 | 0 | 116 |
| 13 | Illinois | 40 | 89.5 | 0 | 124 |
| 14 | Indiana | 40.2 | 86.2 | 0 | 128 |
| 15 | Iowa | 42.2 | 93.8 | 0 | 128 |
| 16 | Kansas | 38.5 | 98.5 | 0 | 166 |
| 17 | Kentucky | 37.8 | 85 | 0 | 147 |
| 18 | Louisiana | 31.2 | 91.8 | 1 | 190 |
| 19 | Maine | 45.2 | 69 | 1 | 117 |
| 20 | Maryland | 39 | 76.5 | 1 | 162 |
| 21 | Massachusetts | 42.2 | 71.8 | 1 | 143 |
| 22 | Michigan | 43.5 | 84.5 | 0 | 117 |
| 23 | Minnesota | 46 | 94.5 | 0 | 116 |
| 24 | Mississippi | 32.8 | 90 | 1 | 207 |
| 25 | Missouri | 38.5 | 92 | 0 | 131 |
| 26 | Montana | 47 | 110.5 | 0 | 109 |
| 27 | Nebraska | 41.5 | 99.5 | 0 | 122 |
| 28 | Nevada | 39 | 117 | 0 | 191 |
| 29 | New Hampshire | 43.8 | 71.5 | 1 | 129 |
| 30 | New Jersey | 40.2 | 74.5 | 1 | 159 |
| 31 | New Mexico | 35 | 106 | 0 | 141 |
| 32 | New York | 43 | 75.5 | 1 | 152 |
| 33 | North Carolina | 35.5 | 79.5 | 1 | 199 |
| 34 | North Dakota | 47.5 | 100.5 | 0 | 115 |
| 35 | Ohio | 40.2 | 82.8 | 0 | 131 |
| 36 | Oklahoma | 35.5 | 97.2 | 0 | 182 |
| 37 | Oregon | 44 | 120.5 | 1 | 136 |
| 38 | Pennsylvania | 40.8 | 77.8 | 0 | 132 |
| 39 | Rhode Island | 41.8 | 71.5 | 1 | 137 |
| 40 | South Carolina | 33.8 | 81 | 1 | 178 |
| 41 | South Dakota | 44.8 | 100 | 0 | 86 |
| 42 | Tennessee | 36 | 86.2 | 0 | 186 |
| 43 | Texas | 31.5 | 98 | 1 | 229 |
| 44 | Utah | 39.5 | 111.5 | 0 | 142 |
| 45 | Vermont | 44 | 72.5 | 1 | 153 |
| 46 | Virginia | 37.5 | 78.5 | 1 | 166 |
| 47 | Washington | 47.5 | 121 | 1 | 117 |
| 48 | West Virginia | 38.8 | 80.8 | 0 | 136 |
| 49 | Wisconsin | 44.5 | 90.2 | 0 | 110 |
| 50 | Wyoming | 43 | 107.5 | 0 | 134 |

線形回帰



$$y = \beta_0 + \beta_1 x$$

$$D = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2$$

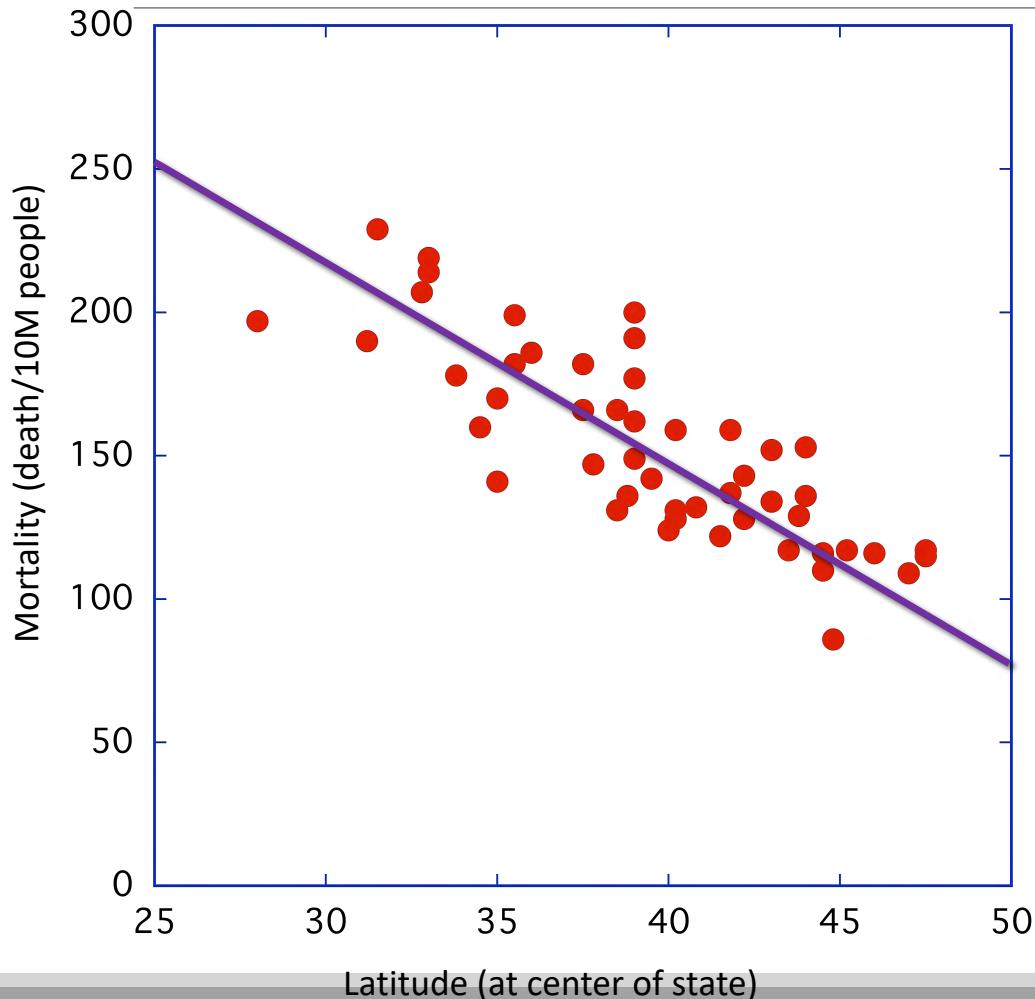
$$D = D(\beta_0, \beta_1)$$

$$\frac{\partial D}{\partial \beta_0} = 0 \text{ } \& \text{ } \frac{\partial D}{\partial \beta_1} = 0$$

Smaller $D \equiv$ better linear model?

| | A | B | C | D | E |
|----|----------------|----------|-----------|-------|-----------|
| 1 | State | Latitude | Longitude | Ocean | Mortality |
| 2 | Alabama | 33 | 87 | 1 | 219 |
| 3 | Arizona | 34.5 | 112 | 0 | 160 |
| 4 | Arkansas | 35 | 92.5 | 0 | 170 |
| 5 | California | 37.5 | 119.5 | 1 | 182 |
| 6 | Colorado | 39 | 105.5 | 0 | 149 |
| 7 | Connecticut | 41.8 | 72.8 | 1 | 159 |
| 8 | Delaware | 39 | 75.5 | 1 | 200 |
| 9 | Wash.D.C. | 39 | 77 | 0 | 177 |
| 10 | Florida | 28 | 82 | 1 | 197 |
| 11 | Georgia | 33 | 83.5 | 1 | 214 |
| 12 | Idaho | 44.5 | 114 | 0 | 116 |
| 13 | Illinois | 40 | 89.5 | 0 | 124 |
| 14 | Indiana | 40.2 | 86.2 | 0 | 128 |
| 15 | Iowa | 42.2 | 93.8 | 0 | 128 |
| 16 | Kansas | 38.5 | 98.5 | 0 | 166 |
| 17 | Kentucky | 37.8 | 85 | 0 | 147 |
| 18 | Louisiana | 31.2 | 91.8 | 1 | 190 |
| 19 | Maine | 45.2 | 69 | 1 | 117 |
| 20 | Maryland | 39 | 76.5 | 1 | 162 |
| 21 | Massachusetts | 42.2 | 71.8 | 1 | 143 |
| 22 | Michigan | 43.5 | 84.5 | 0 | 117 |
| 23 | Minnesota | 46 | 94.5 | 0 | 116 |
| 24 | Mississippi | 32.8 | 90 | 1 | 207 |
| 25 | Missouri | 38.5 | 92 | 0 | 131 |
| 26 | Montana | 47 | 110.5 | 0 | 109 |
| 27 | Nebraska | 41.5 | 99.5 | 0 | 122 |
| 28 | Nevada | 39 | 117 | 0 | 191 |
| 29 | New Hampshire | 43.8 | 71.5 | 1 | 129 |
| 30 | New Jersey | 40.2 | 74.5 | 1 | 159 |
| 31 | New Mexico | 35 | 106 | 0 | 141 |
| 32 | New York | 43 | 75.5 | 1 | 152 |
| 33 | North Carolina | 35.5 | 79.5 | 1 | 199 |
| 34 | North Dakota | 47.5 | 100.5 | 0 | 115 |
| 35 | Ohio | 40.2 | 82.8 | 0 | 131 |
| 36 | Oklahoma | 35.5 | 97.2 | 0 | 182 |
| 37 | Oregon | 44 | 120.5 | 1 | 136 |
| 38 | Pennsylvania | 40.8 | 77.8 | 0 | 132 |
| 39 | Rhode Island | 41.8 | 71.5 | 1 | 137 |
| 40 | South Carolina | 33.8 | 81 | 1 | 178 |
| 41 | South Dakota | 44.8 | 100 | 0 | 86 |
| 42 | Tennessee | 36 | 86.2 | 0 | 186 |
| 43 | Texas | 31.5 | 98 | 1 | 229 |
| 44 | Utah | 39.5 | 111.5 | 0 | 142 |
| 45 | Vermont | 44 | 72.5 | 1 | 153 |
| 46 | Virginia | 37.5 | 78.5 | 1 | 166 |
| 47 | Washington | 47.5 | 121 | 1 | 117 |
| 48 | West Virginia | 38.8 | 80.8 | 0 | 136 |
| 49 | Wisconsin | 44.5 | 90.2 | 0 | 110 |
| 50 | Wyoming | 43 | 107.5 | 0 | 134 |

線形回帰



$$y = \beta_0 + \beta_1 x$$

$$D = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2$$

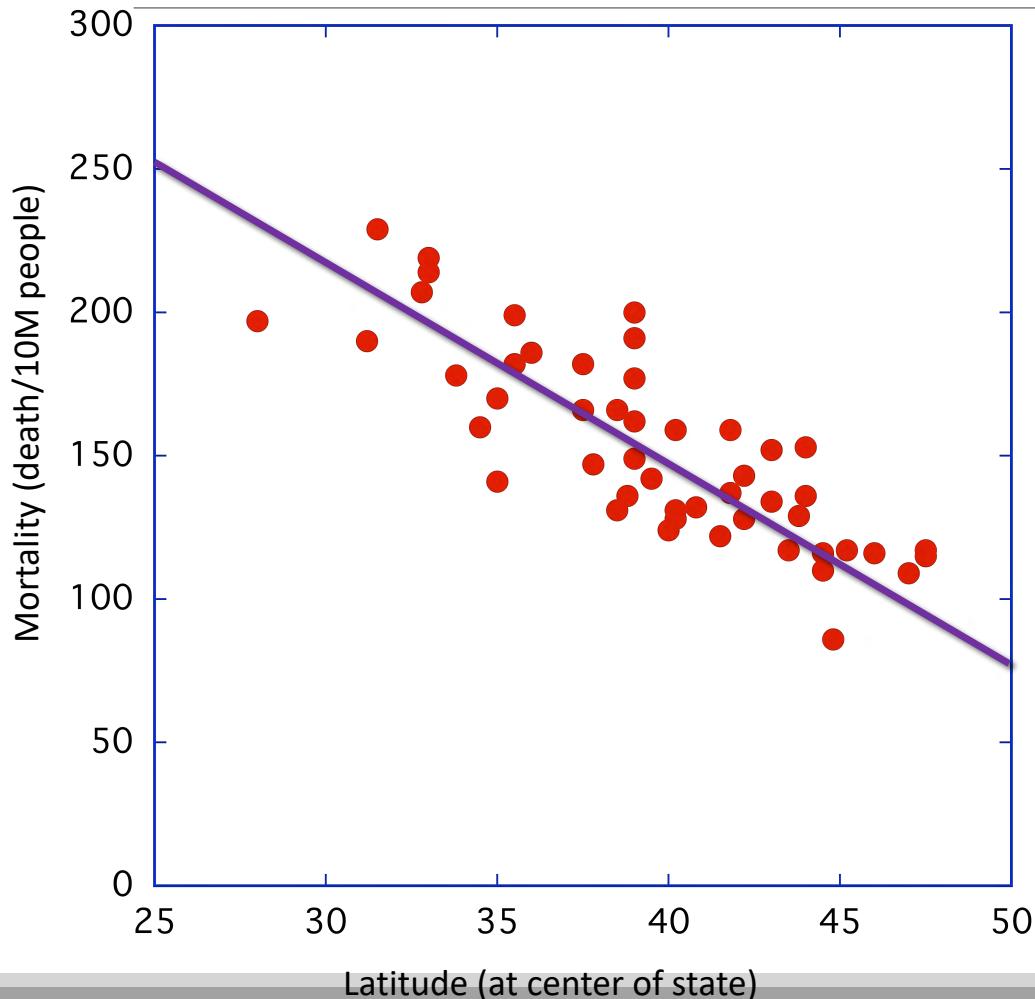
$$D = \sum_{i=1}^n (y_i^2 + \beta_0^2 + \beta_1^2 x_i^2 - 2y_i\beta_0 - 2y_i x_i \beta_1 + 2\beta_0 \beta_1 x_i)$$

$$\frac{\partial D}{\partial \beta_0} = 0 \quad \& \quad \frac{\partial D}{\partial \beta_1} = 0$$

$$\frac{\partial D}{\partial \beta_0} = -2 \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i) = 0$$

$$\frac{\partial D}{\partial \beta_1} = -2 \sum_{i=1}^n x_i (y_i - \beta_0 - \beta_1 x_i) = 0$$

線形回帰



$$y = \beta_0 + \beta_1 x$$

$$D = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2$$

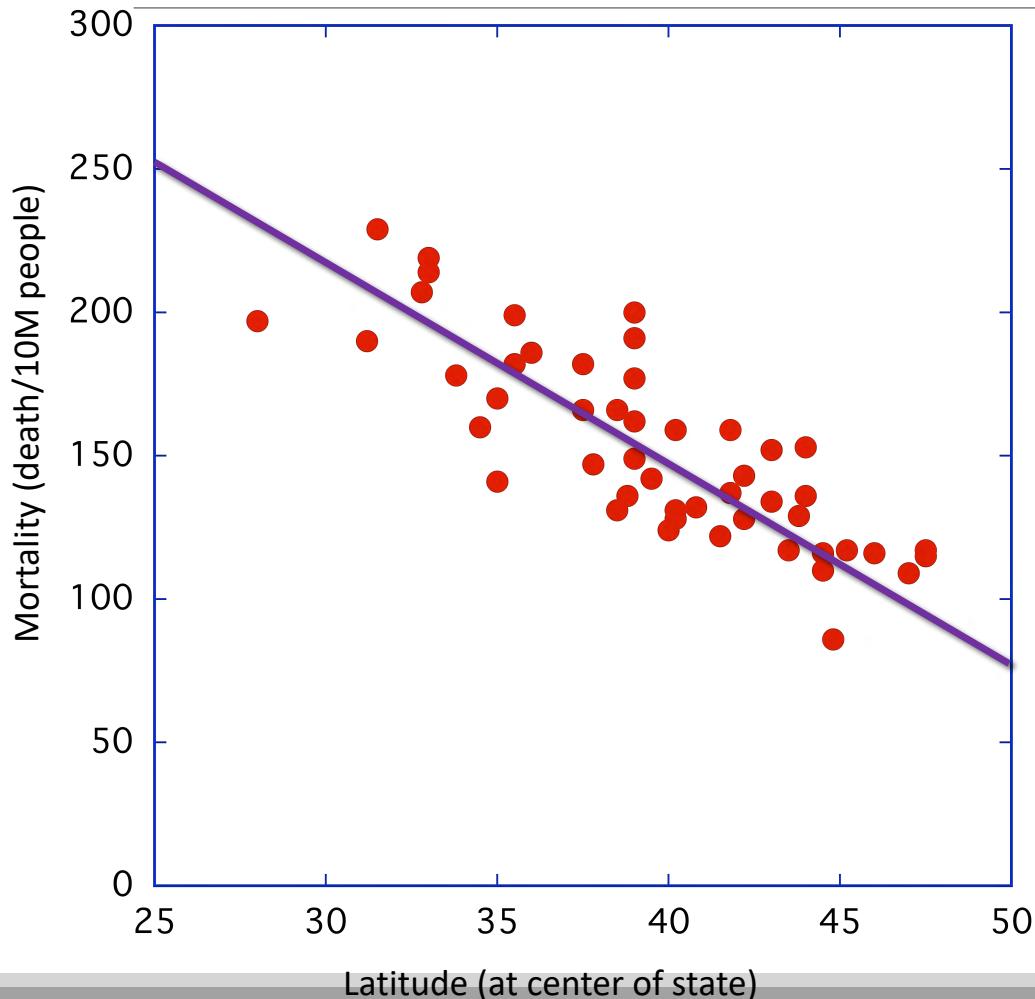
$$\frac{\partial D}{\partial \beta_0} = -2 \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i) = 0$$

$$\sum_{i=1}^n y_i = n\beta_0 + \beta_1 \sum_{i=1}^n x_i$$

$$\frac{\partial D}{\partial \beta_1} = -2 \sum_{i=1}^n x_i (y_i - \beta_0 - \beta_1 x_i) = 0$$

$$\sum_{i=1}^n y_i x_i = \beta_0 \sum_{i=1}^n x_i + \beta_1 \sum_{i=1}^n x_i^2$$

線形回帰



$$y = \beta_0 + \beta_1 x$$

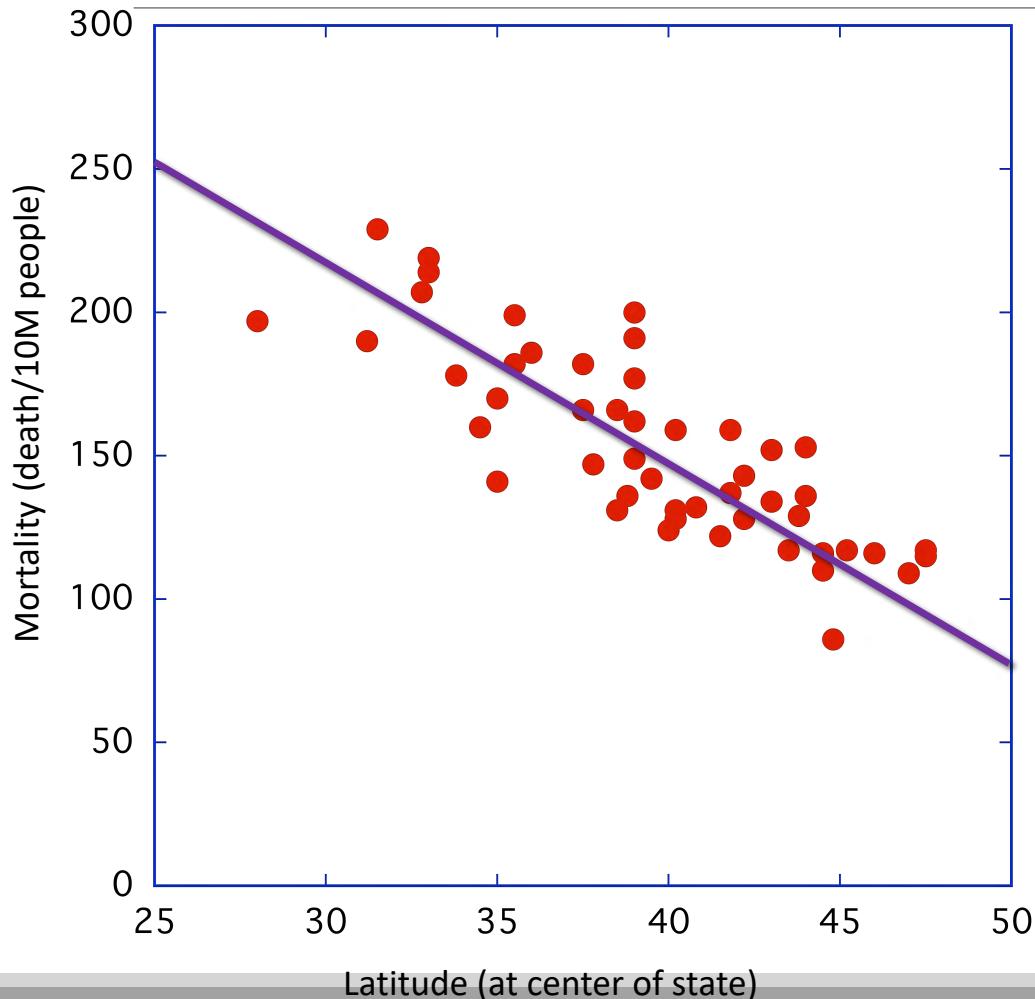
$$\sum_{i=1}^n x_i \rightarrow \sum_{i=1}^n y_i = n\beta_0 + \beta_1 \sum_{i=1}^n x_i$$

$$n \rightarrow \sum_{i=1}^n y_i x_i = \beta_0 \sum_{i=1}^n x_i + \beta_1 \sum_{i=1}^n x_i^2$$

$$\beta_1 = \frac{\sum_{i=1}^n x_i \sum_{i=1}^n y_i - n \sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i \sum_{i=1}^n x_i - n \sum_{i=1}^n x_i^2}$$

$$\beta_0 = \frac{\sum_{i=1}^n y_i - \beta_1 \sum_{i=1}^n x_i}{n}$$

線形回帰



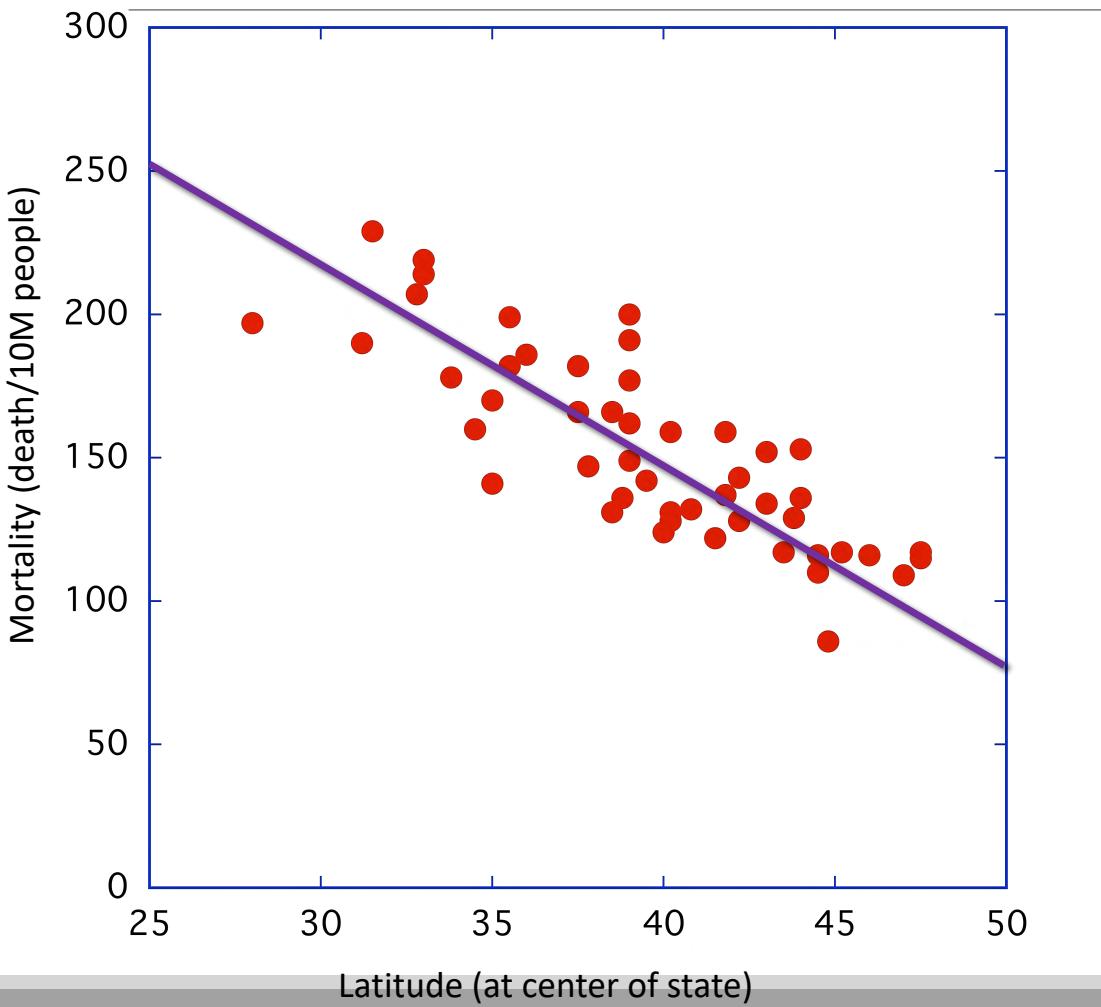
$$y = \beta_0 + \beta_1 x$$

$$D = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2$$

$$\beta_1 = \frac{\sum_{i=1}^n x_i \sum_{i=1}^n y_i - n \sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i \sum_{i=1}^n x_i - n \sum_{i=1}^n x_i^2}$$

$$\beta_0 = \frac{\sum_{i=1}^n y_i - \beta_1 \sum_{i=1}^n x_i}{n}$$

線形回帰



$$y = f(x)$$



$$\begin{aligned}\hat{y} &= \hat{f}(x) \\ &= \hat{\beta}_0 + \hat{\beta}_1 x\end{aligned}$$

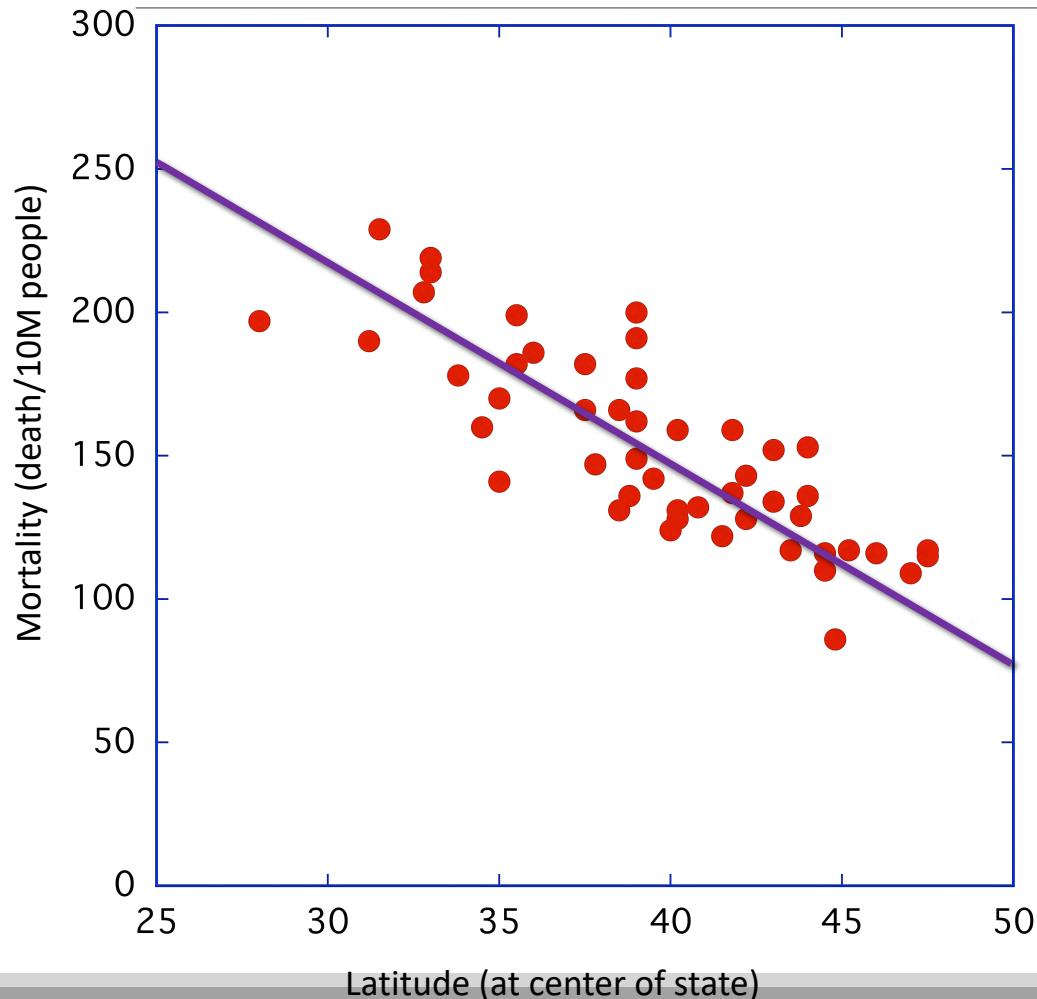
\hat{y} : prediction of y

\hat{f} : estimation of f

$\hat{\beta}_0 + \hat{\beta}_1 x$: just one form of f

| | A | B | C | D | E |
|----|----------------|----------|-----------|-------|-----------|
| 1 | State | Latitude | Longitude | Ocean | Mortality |
| 2 | Alabama | 33 | 87 | 1 | 219 |
| 3 | Arizona | 34.5 | 112 | 0 | 160 |
| 4 | Arkansas | 35 | 92.5 | 0 | 170 |
| 5 | California | 37.5 | 119.5 | 1 | 182 |
| 6 | Colorado | 39 | 105.5 | 0 | 149 |
| 7 | Connecticut | 41.8 | 72.8 | 1 | 159 |
| 8 | Delaware | 39 | 75.5 | 1 | 200 |
| 9 | Wash.D.C. | 39 | 77 | 0 | 177 |
| 10 | Florida | 28 | 82 | 1 | 197 |
| 11 | Georgia | 33 | 83.5 | 1 | 214 |
| 12 | Idaho | 44.5 | 114 | 0 | 116 |
| 13 | Illinois | 40 | 89.5 | 0 | 124 |
| 14 | Indiana | 40.2 | 86.2 | 0 | 128 |
| 15 | Iowa | 42.2 | 93.8 | 0 | 128 |
| 16 | Kansas | 38.5 | 98.5 | 0 | 166 |
| 17 | Kentucky | 37.8 | 85 | 0 | 147 |
| 18 | Louisiana | 31.2 | 91.8 | 1 | 190 |
| 19 | Maine | 45.2 | 69 | 1 | 117 |
| 20 | Maryland | 39 | 76.5 | 1 | 162 |
| 21 | Massachusetts | 42.2 | 71.8 | 1 | 143 |
| 22 | Michigan | 43.5 | 84.5 | 0 | 117 |
| 23 | Minnesota | 46 | 94.5 | 0 | 116 |
| 24 | Mississippi | 32.8 | 90 | 1 | 207 |
| 25 | Missouri | 38.5 | 92 | 0 | 131 |
| 26 | Montana | 47 | 110.5 | 0 | 109 |
| 27 | Nebraska | 41.5 | 99.5 | 0 | 122 |
| 28 | Nevada | 39 | 117 | 0 | 191 |
| 29 | New Hampshire | 43.8 | 71.5 | 1 | 129 |
| 30 | New Jersey | 40.2 | 74.5 | 1 | 159 |
| 31 | New Mexico | 35 | 106 | 0 | 141 |
| 32 | New York | 43 | 75.5 | 1 | 152 |
| 33 | North Carolina | 35.5 | 79.5 | 1 | 199 |
| 34 | North Dakota | 47.5 | 100.5 | 0 | 115 |
| 35 | Ohio | 40.2 | 82.8 | 0 | 131 |
| 36 | Oklahoma | 35.5 | 97.2 | 0 | 182 |
| 37 | Oregon | 44 | 120.5 | 1 | 136 |
| 38 | Pennsylvania | 40.8 | 77.8 | 0 | 132 |
| 39 | Rhode Island | 41.8 | 71.5 | 1 | 137 |
| 40 | South Carolina | 33.8 | 81 | 1 | 178 |
| 41 | South Dakota | 44.8 | 100 | 0 | 86 |
| 42 | Tennessee | 36 | 86.2 | 0 | 186 |
| 43 | Texas | 31.5 | 98 | 1 | 229 |
| 44 | Utah | 39.5 | 111.5 | 0 | 142 |
| 45 | Vermont | 44 | 72.5 | 1 | 153 |
| 46 | Virginia | 37.5 | 78.5 | 1 | 166 |
| 47 | Washington | 47.5 | 121 | 1 | 117 |
| 48 | West Virginia | 38.8 | 80.8 | 0 | 136 |
| 49 | Wisconsin | 44.5 | 90.2 | 0 | 110 |
| 50 | Wyoming | 43 | 107.5 | 0 | 134 |

線形回帰



Hypothesis

Estimation

Evaluation

\hat{y} : prediction of y

\hat{f} : estimation of f

$$y = f(x)$$

$$\hat{y} = \hat{f}(x)$$

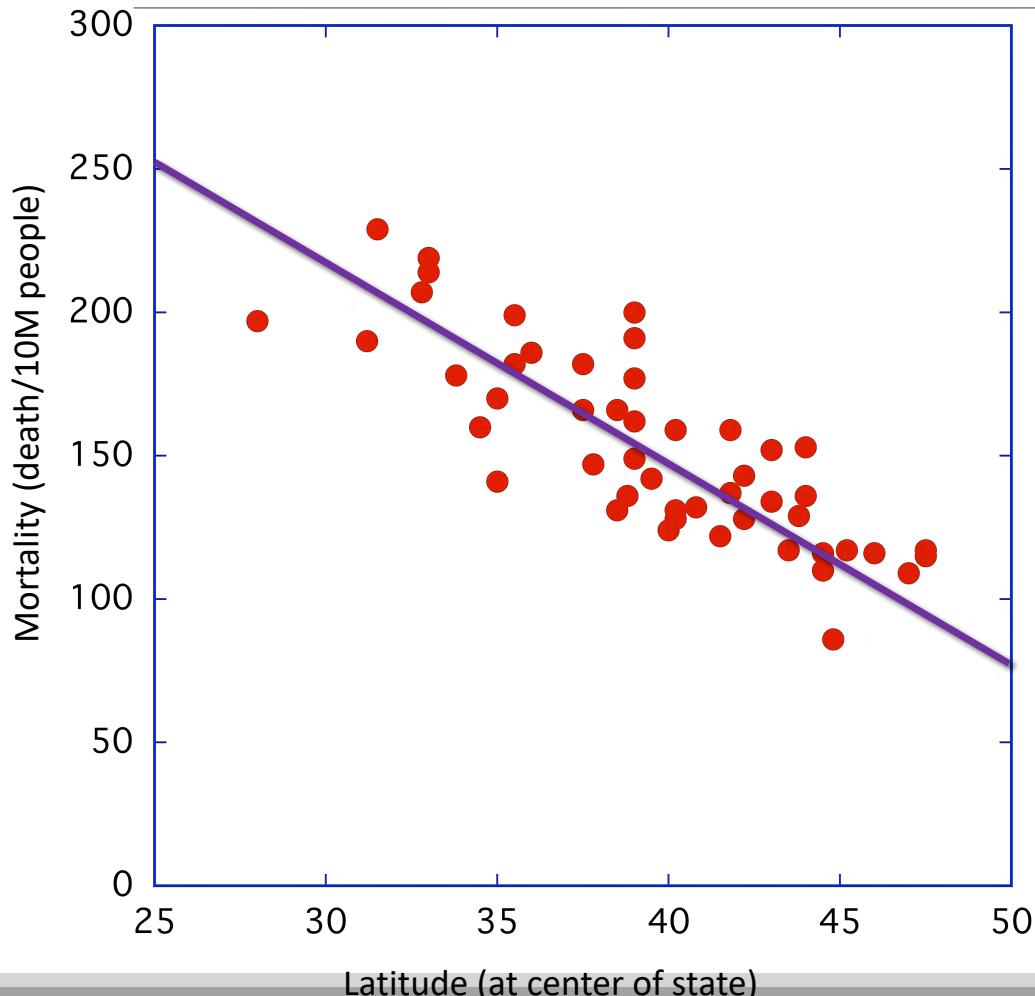
$$y_{obs} = y_{predict} + \epsilon$$

$\epsilon_{reducible}$

ϵ

$\epsilon_{irreducible}$

線形回帰



Hypothesis

$$y = f(x)$$

Estimation

$$\hat{y} = \hat{f}(x)$$

Evaluation

$$y_{obs} = y_{predict} + \epsilon$$

Statistical thinking

Understand and reduce

ϵ

$\epsilon_{reducible}$

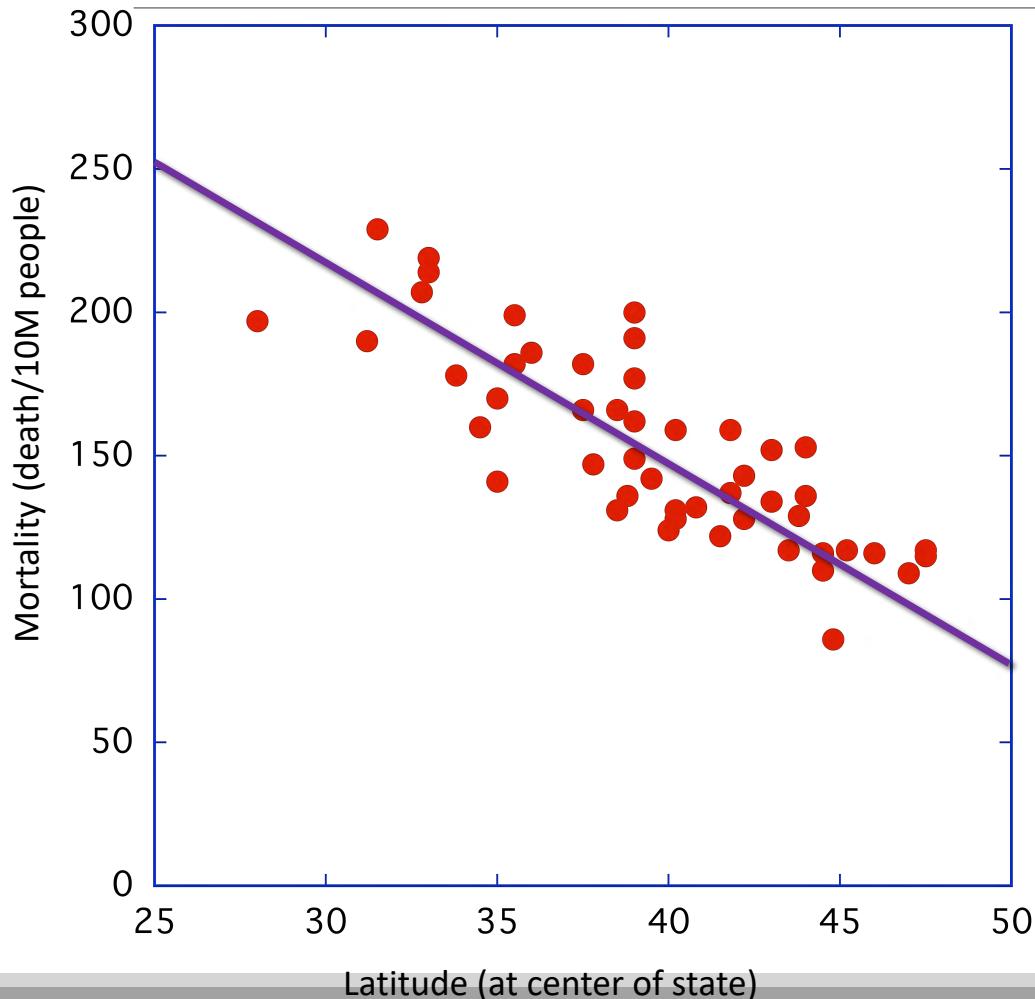
$\epsilon_{irreducible}$

Understand the limitation of the hypothesis

\hat{y} : prediction of y

\hat{f} : estimation of f

線形回帰



Hypothesis

$$y = f(x)$$

Estimation

$$\hat{y} = \hat{f}(x)$$

Evaluation

$$y_{obs} = y_{predict} + \epsilon$$

Statistical thinking for prediction

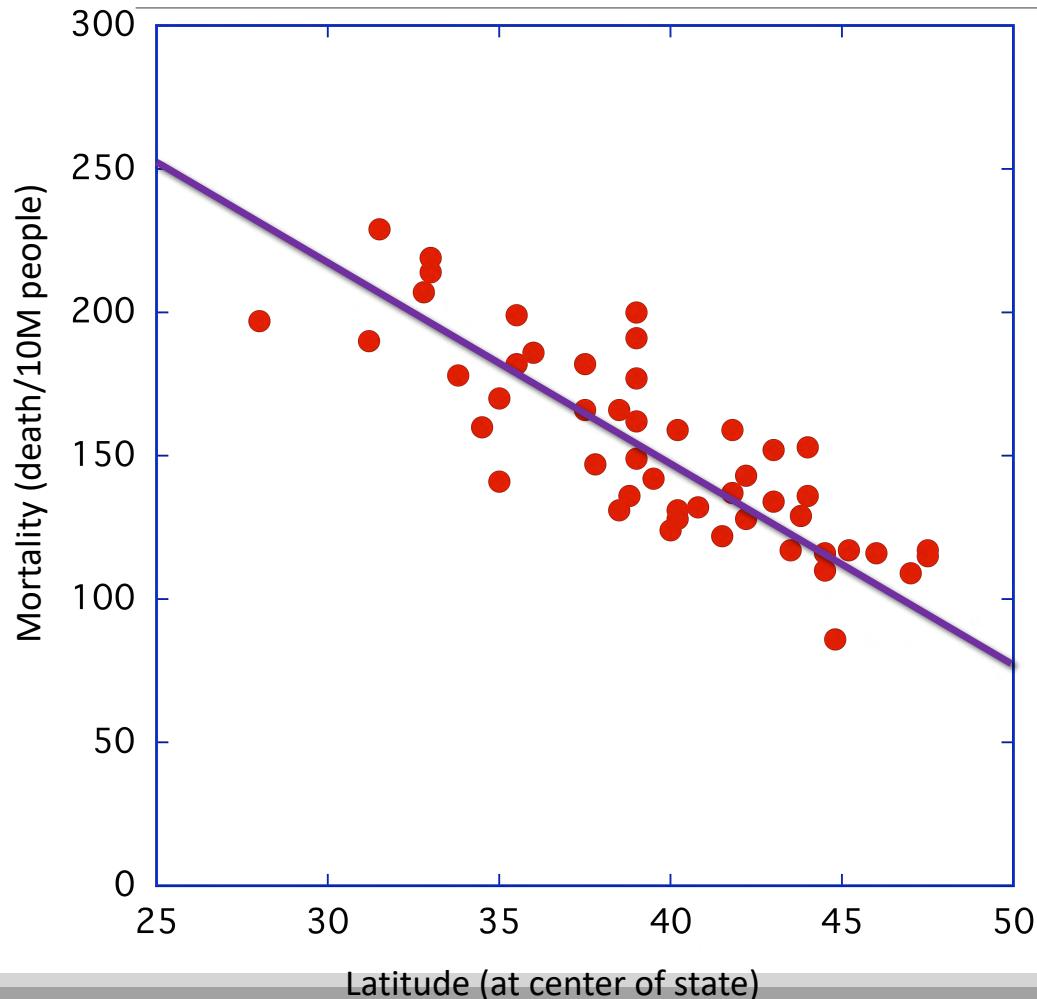
Minimization

$$E(y_{obs} - y_{predict})^2 = [f(x) - \hat{f}(x)]^2 + \text{Var } (\epsilon_{irreducible})$$

\hat{y} : prediction of y

\hat{f} : estimation of f

線形回帰



Hypothesis

$$y = f(x)$$

Estimation

$$\hat{y} = \hat{f}(x)$$

Evaluation

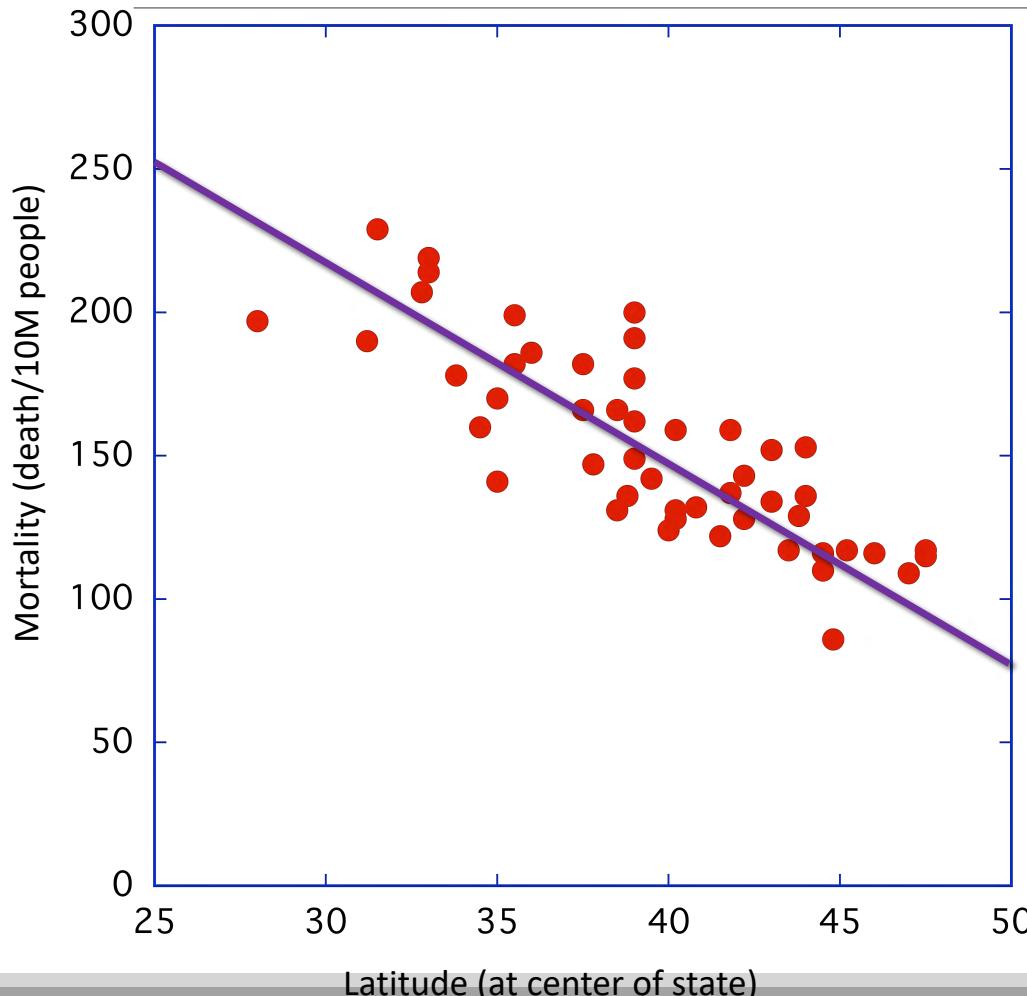
$$y_{obs} = y_{predict} + \epsilon$$

Statistical thinking for inference

Understand
the relationship
between x and y

$$E(y_{obs} - y_{predict})^2 = [f(x) - \hat{f}(x)]^2 + \text{Var } (\epsilon_{irreducible})$$

線形回帰



Hypothesis

$$y = f(x)$$

Estimation

$$\hat{y} = \hat{f}(x)$$

Evaluation

$$y_{obs} = y_{predict} + \epsilon$$

From the nature
of the relationship
between x and y

From the fixity in
the form of the
hypothetical function

$\epsilon_{irreducible}$

Understand the limitation of the hypothesis⁵⁴

\hat{y} : prediction of y

\hat{f} : estimation of f

線形回帰

$$\begin{aligned}\hat{y} &= \hat{f}(x) \\ &= \hat{\beta}_0 + \hat{\beta}_1 x\end{aligned}$$

$$E(y_{obs} - y_{predict})^2 = [f(x) - \hat{f}(x)]^2 + \text{Var}(\epsilon_{irreducible})$$

Mystery **Estimation**
 $*_0 + *_1 x$ $\hat{\beta}_0 + \hat{\beta}_1 x$



$$y(x) = \underbrace{\beta_0 + \beta_1 x}_{\text{Mystery}} + \underbrace{\mathcal{N}(0, \sigma^2)}_{\text{Normal distribution}}$$

Hypothesis

線形回帰

$$\begin{aligned}\hat{y} &= \hat{f}(x) \\ &= \hat{\beta}_0 + \hat{\beta}_1 x\end{aligned}$$

Hypothesis

$$y(x) = \beta_0 + \beta_1 x + \mathcal{N}(0, \sigma^2) \longrightarrow p(y|x, \beta_0, \beta_1, \sigma^2) = \mathcal{N}(y|\beta_0 + \beta_1 x, \sigma^2)$$

fixed noise

$$Data = \{x_i, y_i\}_{i=1 \div N}$$

Observation

Sample



Most strongly support

Assumption

Population

線形回帰

$$\begin{aligned}\hat{y} &= \hat{f}(x) \\ &= \hat{\beta}_0 + \hat{\beta}_1 x\end{aligned}$$

Model parameter
 θ

Hypothesis

$$y(x) = \beta_0 + \beta_1 x + \mathcal{N}(0, \sigma^2)$$

Maximum likelihood

$$\hat{\beta}_0, \hat{\beta}_1 \triangleq \arg \max_{\beta_0, \beta_1 \in \mathbb{R}} \log p(\mathcal{D} \mid \underbrace{\beta_0, \beta_1, \sigma^2}_{\theta})$$

$$Data = \{x_i, y_i\}_{i=1 \div N}$$

Observation

線形回帰

Observation

$$\mathcal{D}ata = \{x_i, y_i\}_{i=1 \div N}$$

Log-likelihood

$$\begin{aligned}\ell(\boldsymbol{\theta}) &\triangleq \log p(\mathcal{D} | \beta_0, \beta_1, \sigma^2) = \sum_{i=1}^N \log p(y_i | \mathbf{x}_i, \boldsymbol{\theta}) \\ &= \sum_{i=1}^N \log \left[\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(y_i - \beta_0 - \beta_1 x_i)^2} \right] \\ &= -\frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \beta_0 - \beta_1 x_i)^2 + \frac{N}{\sqrt{2\pi}\sigma}\end{aligned}$$

線形回帰

Observation

$$\mathcal{D}ata = \{x_i, y_i\}_{i=1 \div N}$$

Log-likelihood

$$\begin{aligned}\ell(\boldsymbol{\theta}) &\triangleq \log p(\mathcal{D} | \beta_0, \beta_1, \sigma^2) = \sum_{i=1}^N \log p(y_i | \mathbf{x}_i, \boldsymbol{\theta}) \\ &= \sum_{i=1}^N \log \left[\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(y_i - \beta_0 - \beta_1 x_i)^2} \right] \\ &= \boxed{-\frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \beta_0 - \beta_1 x_i)^2} + \frac{N}{\sqrt{2\pi}\sigma}\end{aligned}$$

Maximum log-likelihood
= Least square fit

Statistical thinking in data science

データ科学における統計的思考

- Statistical thinking relates processes and statistics, and is based on the following principles:
 - All work occurs in a system of interconnected processes.
 - Variation exists in all processes
 - Understanding and reducing variation are keys to success.

統計的思考は、データを理解し、管理し、導かれる結論の誤差を低減する方法に重点をおく思考プロセスとして定義できる

- Statistical thinking plays an essential role in data science.

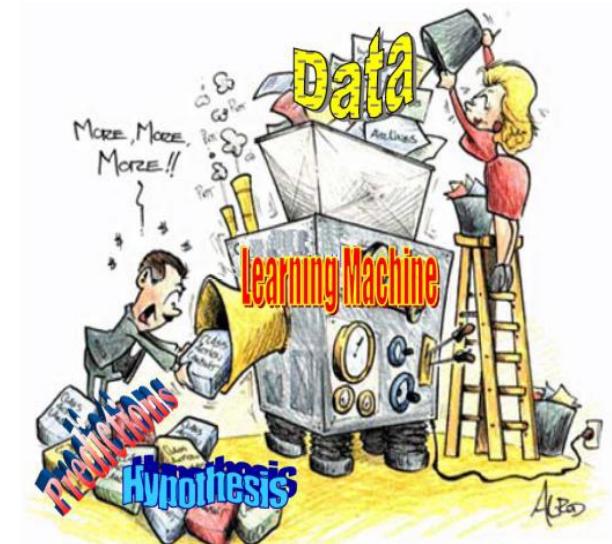
K490: データサイエンス論

Lecture 5: 多変量解析

Lecturer: Hieu-Chi Dam, Takashi Isogai

Machine learning

- The goal of machine learning is to build computer systems that can adapt and learn from their experience (Tom Dietterich, 1999).
- *Given* (input for the learning process – experience will be given to computer)
 - $\{x_i\}$, x_i is description of an object in some space, $i = 1, 2, \dots, n$.
 - (experience with supervision) y_i is some property of x_i viewed as its label, $y_i \in \{C_1, C_2, \dots, C_K\}$ or $y_i \in \mathbb{R}$
 - $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$
- *Find* (output from the learning process)
 - Method to predict y that corresponding to a given x – for label data
 - Method to describe the $\{x_i\}$ – for unlabeled data



(Source: Eric Xing lecture notes)

Branch of Artificial Intelligence: Reasoning, language understanding, learning

DEDUCTION [Given $f(x)$ and x_i , deduce $f(x_i)$] vs. *INDUCTION* [Given $\{x_i\}$, infer $f(x)$]

review articles

DOI:10.1145/2347736.2347755

Tapping into the “folk knowledge” needed to advance machine learning applications.

BY PEDRO DOMINGOS

A Few Useful Things to Know About Machine Learning

is needed to successfully develop machine learning applications is not



Many different types of machine learning exist, but for illustration purposes I will focus on the most mature and widely used one: classification. Nevertheless, the issues I will discuss apply across all of machine learning. A *classifier* is a system that inputs (typically) a vector of discrete and/or continuous *feature values* and outputs a single discrete value, the *class*. For example, a spam filter classifies email messages into “spam” or “not spam,” and its input may be a Boolean vector $\mathbf{x} = (x_1, \dots, x_j, \dots, x_d)$, where $x_j = 1$ if the j^{th} word in the dictionary appears in the email and $x_j = 0$ otherwise. A *learner* inputs a training set of examples (\mathbf{x}_i, y_i) , where $\mathbf{x}_i = (x_{i1}, \dots, x_{id})$ is an observed input and y_i is the corresponding output, and outputs a classifier. The test of the learner is whether this classifier produces the

Learning = Representation + Evaluation + Optimization

Suppose you have an application that you think machine learning might be good for. The first problem facing you is the bewildering variety of learning algorithms available. Which one to use? There are literally thousands available, and hundreds more are published each year. The key to not getting lost in this huge space is to realize that it consists of combinations of just three components. The components are:

► **Representation.** A classifier must be represented in some formal language that the computer can handle. Conversely, choosing a representation for a learner is tantamount to choosing the set of classifiers that it can possibly learn. This set is called the *hypothesis space* of the learner. If a classifier is not in the hypothesis space, it cannot be learned. A related

or *scoring function*) is needed to distinguish good classifiers from bad ones. The evaluation function used internally by the algorithm may differ from the external one that we want the classifier to optimize, for ease of optimization and due to the issues I will discuss.

► **Optimization.** Finally, we need a method to search among the classifiers in the language for the highest-scoring one. The choice of optimization technique is key to the efficiency of the learner, and also helps determine the classifier produced if the evaluation function has more than one optimum. It is common for new learners to start out using off-the-shelf optimizers, which are later replaced by custom-designed ones.

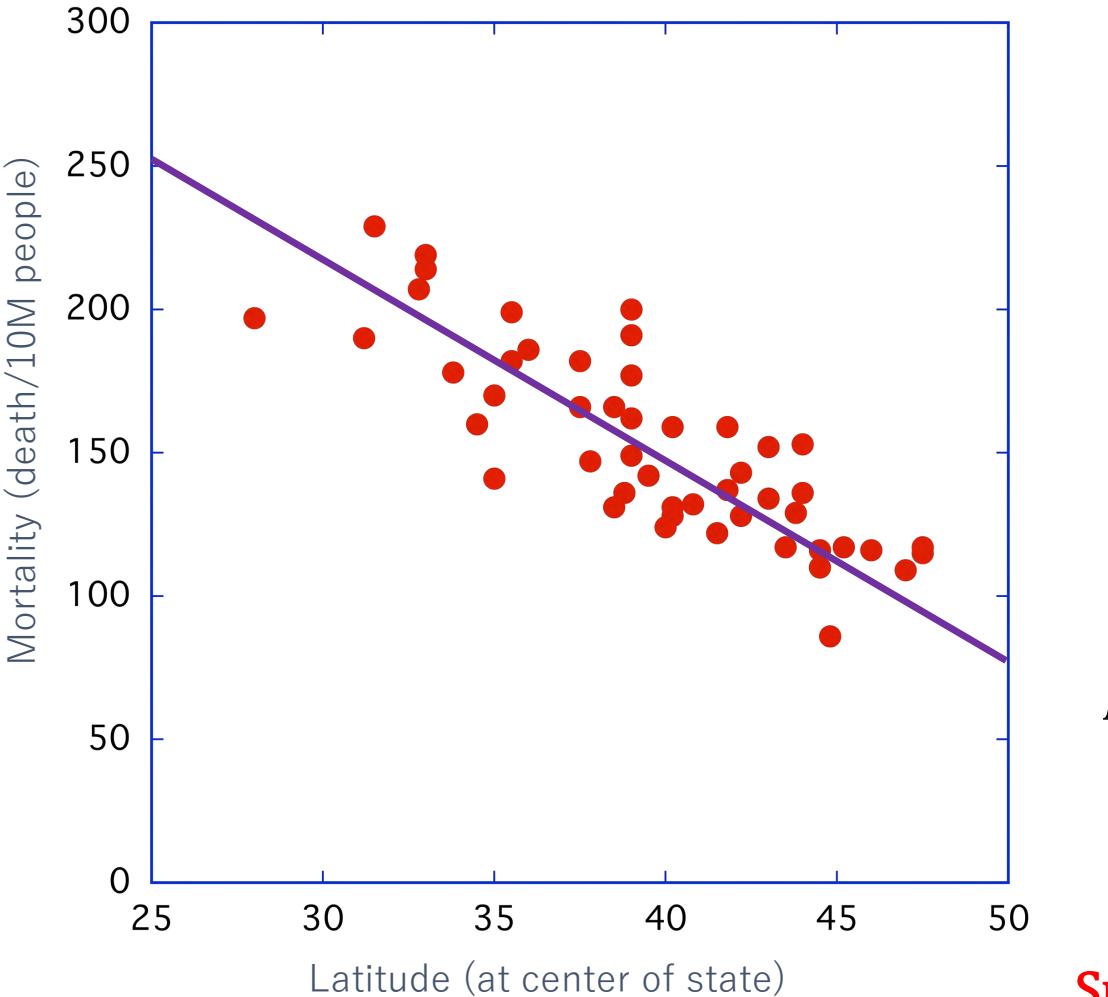
The accompanying table shows common examples of each of these three components. For example, k -

Learning = Representation + Evaluation + Optimization

Table 1. The three components of learning algorithms.

| Representation | Evaluation | Optimization |
|---------------------------|-----------------------|----------------------------|
| Instances | Accuracy/Error rate | Combinatorial optimization |
| K-nearest neighbor | Precision and recall | Greedy search |
| Support vector machines | Squared error | Beam search |
| Hyperplanes | Likelihood | Branch-and-bound |
| Naive Bayes | Posterior probability | Continuous optimization |
| Logistic regression | Information gain | Unconstrained |
| Decision trees | K-L divergence | Gradient descent |
| Sets of rules | Cost/Utility | Conjugate gradient |
| Propositional rules | Margin | Quasi-Newton methods |
| Logic programs | | Constrained |
| Neural networks | | Linear programming |
| Graphical models | | Quadratic programming |
| Bayesian networks | | |
| Conditional random fields | | |

A straight-line regression model



Hypothesis

$$y = \beta_0 + \beta_1 x$$



Evaluation

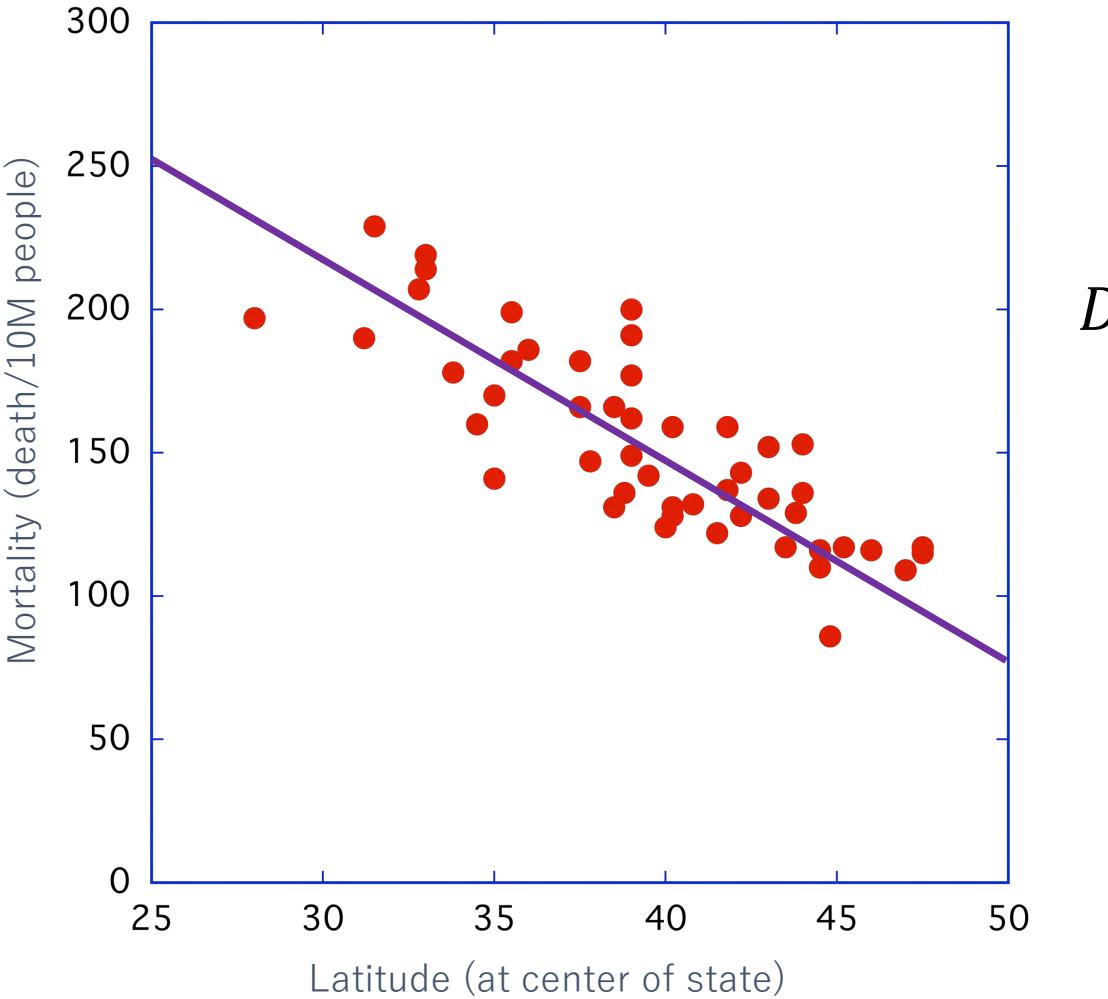
$$D = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2$$

Original hypothesis

Smaller $D \equiv$ better linear model?

| | A | B | C | D | E |
|----|----------------|----------|-----------|-------|-----------|
| 1 | State | Latitude | Longitude | Ocean | Mortality |
| 2 | Alabama | 33 | 87 | 1 | 219 |
| 3 | Arizona | 34.5 | 112 | 0 | 160 |
| 4 | Arkansas | 35 | 92.5 | 0 | 170 |
| 5 | California | 37.5 | 119.5 | 1 | 182 |
| 6 | Colorado | 39 | 105.5 | 0 | 149 |
| 7 | Connecticut | 41.8 | 72.8 | 1 | 159 |
| 8 | Delaware | 39 | 75.5 | 1 | 200 |
| 9 | Wash.D.C. | 39 | 77 | 0 | 177 |
| 10 | Florida | 28 | 82 | 1 | 197 |
| 11 | Georgia | 33 | 83.5 | 1 | 214 |
| 12 | Idaho | 44.5 | 114 | 0 | 116 |
| 13 | Illinois | 40 | 89.5 | 0 | 124 |
| 14 | Indiana | 40.2 | 86.2 | 0 | 128 |
| 15 | Iowa | 42.2 | 93.8 | 0 | 128 |
| 16 | Kansas | 38.5 | 98.5 | 0 | 166 |
| 17 | Kentucky | 37.8 | 85 | 0 | 147 |
| 18 | Louisiana | 31.2 | 91.8 | 1 | 190 |
| 19 | Maine | 45.2 | 69 | 1 | 117 |
| 20 | Maryland | 39 | 76.5 | 1 | 162 |
| 21 | Massachusetts | 42.2 | 71.8 | 1 | 143 |
| 22 | Michigan | 43.5 | 84.5 | 0 | 117 |
| 23 | Minnesota | 46 | 94.5 | 0 | 116 |
| 24 | Mississippi | 32.8 | 90 | 1 | 207 |
| 25 | Missouri | 38.5 | 92 | 0 | 131 |
| 26 | Montana | 47 | 110.5 | 0 | 109 |
| 27 | Nebraska | 41.5 | 99.5 | 0 | 122 |
| 28 | Nevada | 39 | 117 | 0 | 191 |
| 29 | New Hampshire | 43.8 | 71.5 | 1 | 129 |
| 30 | New Jersey | 40.2 | 74.5 | 1 | 159 |
| 31 | New Mexico | 35 | 106 | 0 | 141 |
| 32 | New York | 43 | 75.5 | 1 | 152 |
| 33 | North Carolina | 35.5 | 79.5 | 1 | 199 |
| 34 | North Dakota | 47.5 | 100.5 | 0 | 115 |
| 35 | Ohio | 40.2 | 82.8 | 0 | 131 |
| 36 | Oklahoma | 35.5 | 97.2 | 0 | 182 |
| 37 | Oregon | 44 | 120.5 | 1 | 136 |
| 38 | Pennsylvania | 40.8 | 77.8 | 0 | 132 |
| 39 | Rhode Island | 41.8 | 71.5 | 1 | 137 |
| 40 | South Carolina | 33.8 | 81 | 1 | 178 |
| 41 | South Dakota | 44.8 | 100 | 0 | 86 |
| 42 | Tennessee | 36 | 86.2 | 0 | 186 |
| 43 | Texas | 31.5 | 98 | 1 | 229 |
| 44 | Utah | 39.5 | 111.5 | 0 | 142 |
| 45 | Vermont | 44 | 72.5 | 1 | 153 |
| 46 | Virginia | 37.5 | 78.5 | 1 | 166 |
| 47 | Washington | 47.5 | 121 | 1 | 117 |
| 48 | West Virginia | 38.8 | 80.8 | 0 | 136 |
| 49 | Wisconsin | 44.5 | 90.2 | 0 | 110 |
| 50 | Wyoming | 43 | 107.5 | 0 | 134 |

A straight-line regression model



$$y = \beta_0 + \beta_1 x$$

$$D = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2$$

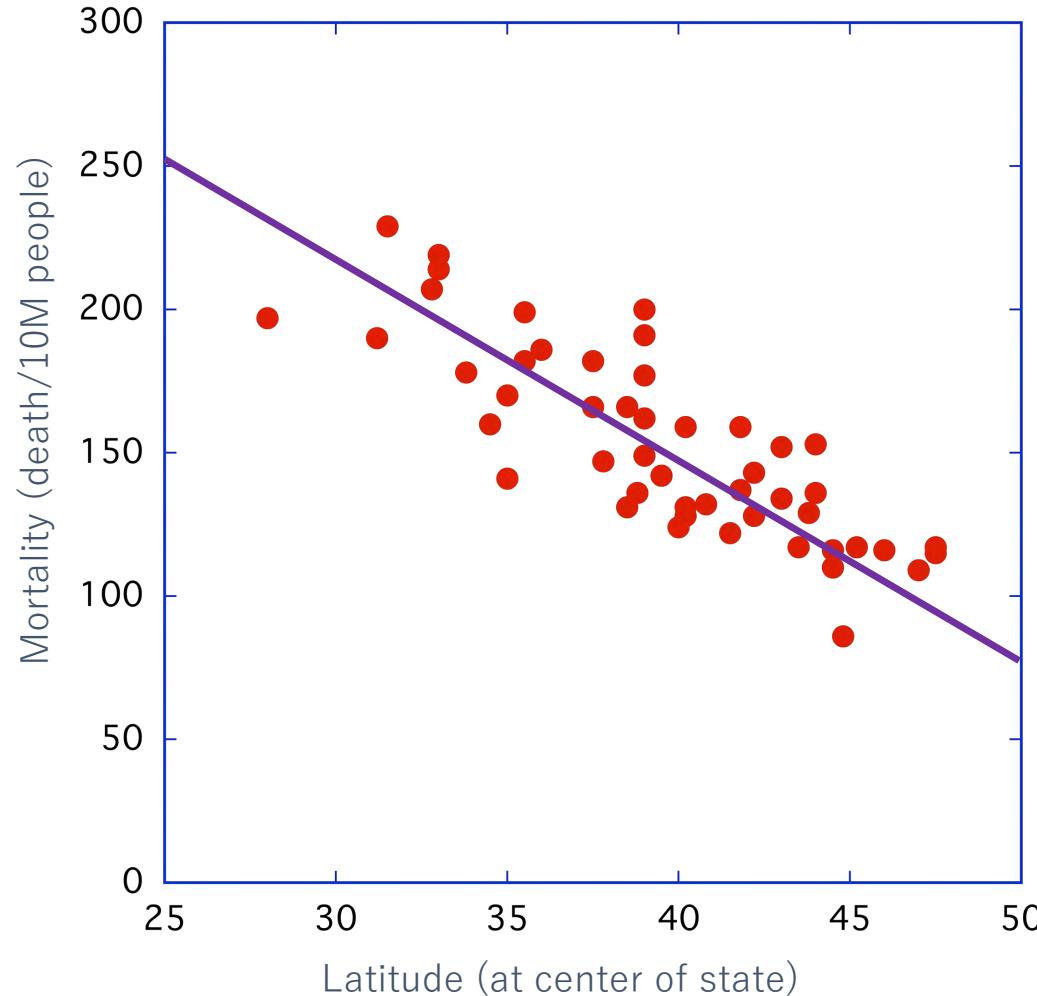
$$D = D(\beta_0, \beta_1)$$

$$\frac{\partial D}{\partial \beta_0} = 0 \text{ & } \frac{\partial D}{\partial \beta_1} = 0$$

Smaller $D \equiv$ better linear model?

| | A | B | C | D | E |
|----|----------------|----------|-----------|-------|-----------|
| 1 | State | Latitude | Longitude | Ocean | Mortality |
| 2 | Alabama | 33 | 87 | 1 | 219 |
| 3 | Arizona | 34.5 | 112 | 0 | 160 |
| 4 | Arkansas | 35 | 92.5 | 0 | 170 |
| 5 | California | 37.5 | 119.5 | 1 | 182 |
| 6 | Colorado | 39 | 105.5 | 0 | 149 |
| 7 | Connecticut | 41.8 | 72.8 | 1 | 159 |
| 8 | Delaware | 39 | 75.5 | 1 | 200 |
| 9 | Wash.D.C. | 39 | 77 | 0 | 177 |
| 10 | Florida | 28 | 82 | 1 | 197 |
| 11 | Georgia | 33 | 83.5 | 1 | 214 |
| 12 | Idaho | 44.5 | 114 | 0 | 116 |
| 13 | Illinois | 40 | 89.5 | 0 | 124 |
| 14 | Indiana | 40.2 | 86.2 | 0 | 128 |
| 15 | Iowa | 42.2 | 93.8 | 0 | 128 |
| 16 | Kansas | 38.5 | 98.5 | 0 | 166 |
| 17 | Kentucky | 37.8 | 85 | 0 | 147 |
| 18 | Louisiana | 31.2 | 91.8 | 1 | 190 |
| 19 | Maine | 45.2 | 69 | 1 | 117 |
| 20 | Maryland | 39 | 76.5 | 1 | 162 |
| 21 | Massachusetts | 42.2 | 71.8 | 1 | 143 |
| 22 | Michigan | 43.5 | 84.5 | 0 | 117 |
| 23 | Minnesota | 46 | 94.5 | 0 | 116 |
| 24 | Mississippi | 32.8 | 90 | 1 | 207 |
| 25 | Missouri | 38.5 | 92 | 0 | 131 |
| 26 | Montana | 47 | 110.5 | 0 | 109 |
| 27 | Nebraska | 41.5 | 99.5 | 0 | 122 |
| 28 | Nevada | 39 | 117 | 0 | 191 |
| 29 | New Hampshire | 43.8 | 71.5 | 1 | 129 |
| 30 | New Jersey | 40.2 | 74.5 | 1 | 159 |
| 31 | New Mexico | 35 | 106 | 0 | 141 |
| 32 | New York | 43 | 75.5 | 1 | 152 |
| 33 | North Carolina | 35.5 | 79.5 | 1 | 199 |
| 34 | North Dakota | 47.5 | 100.5 | 0 | 115 |
| 35 | Ohio | 40.2 | 82.8 | 0 | 131 |
| 36 | Oklahoma | 35.5 | 97.2 | 0 | 182 |
| 37 | Oregon | 44 | 120.5 | 1 | 136 |
| 38 | Pennsylvania | 40.8 | 77.8 | 0 | 132 |
| 39 | Rhode Island | 41.8 | 71.5 | 1 | 137 |
| 40 | South Carolina | 33.8 | 81 | 1 | 178 |
| 41 | South Dakota | 44.8 | 100 | 0 | 86 |
| 42 | Tennessee | 36 | 86.2 | 0 | 186 |
| 43 | Texas | 31.5 | 98 | 1 | 229 |
| 44 | Utah | 39.5 | 111.5 | 0 | 142 |
| 45 | Vermont | 44 | 72.5 | 1 | 153 |
| 46 | Virginia | 37.5 | 78.5 | 1 | 166 |
| 47 | Washington | 47.5 | 121 | 1 | 117 |
| 48 | West Virginia | 38.8 | 80.8 | 0 | 136 |
| 49 | Wisconsin | 44.5 | 90.2 | 0 | 110 |
| 50 | Wyoming | 43 | 107.5 | 0 | 134 |

The least squares fit



$$y = \beta_0 + \beta_1 x$$

$$D = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2$$

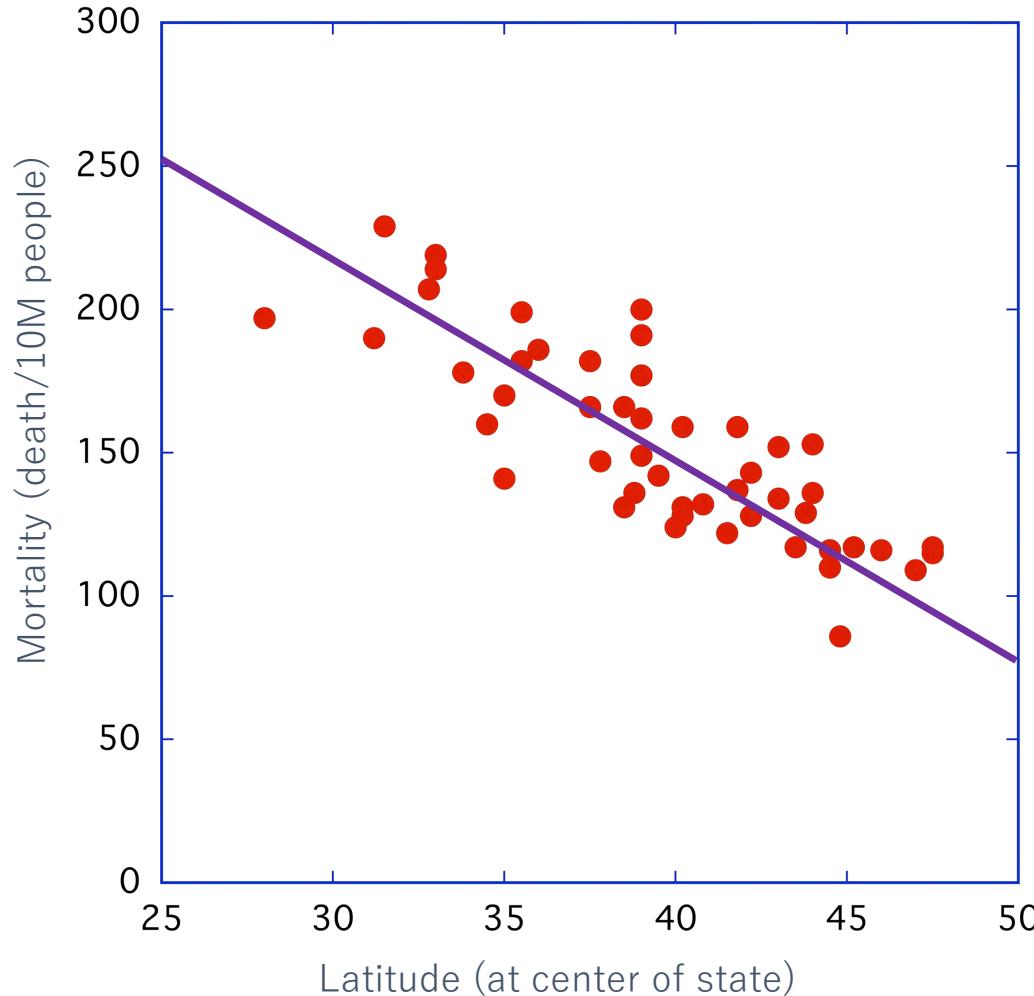
$$D = \sum_{i=1}^n (y_i^2 + \beta_0^2 + \beta_1^2 x_i^2 - 2y_i\beta_0 - 2y_i x_i \beta_1 + 2\beta_0 \beta_1 x_i)$$

$$\frac{\partial D}{\partial \beta_0} = 0 \quad \& \quad \frac{\partial D}{\partial \beta_1} = 0$$

$$\frac{\partial D}{\partial \beta_0} = -2 \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i) = 0$$

$$\frac{\partial D}{\partial \beta_1} = -2 \sum_{i=1}^n x_i (y_i - \beta_0 - \beta_1 x_i) = 0$$

The least squares fit



$$y = \beta_0 + \beta_1 x$$

$$D = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2$$

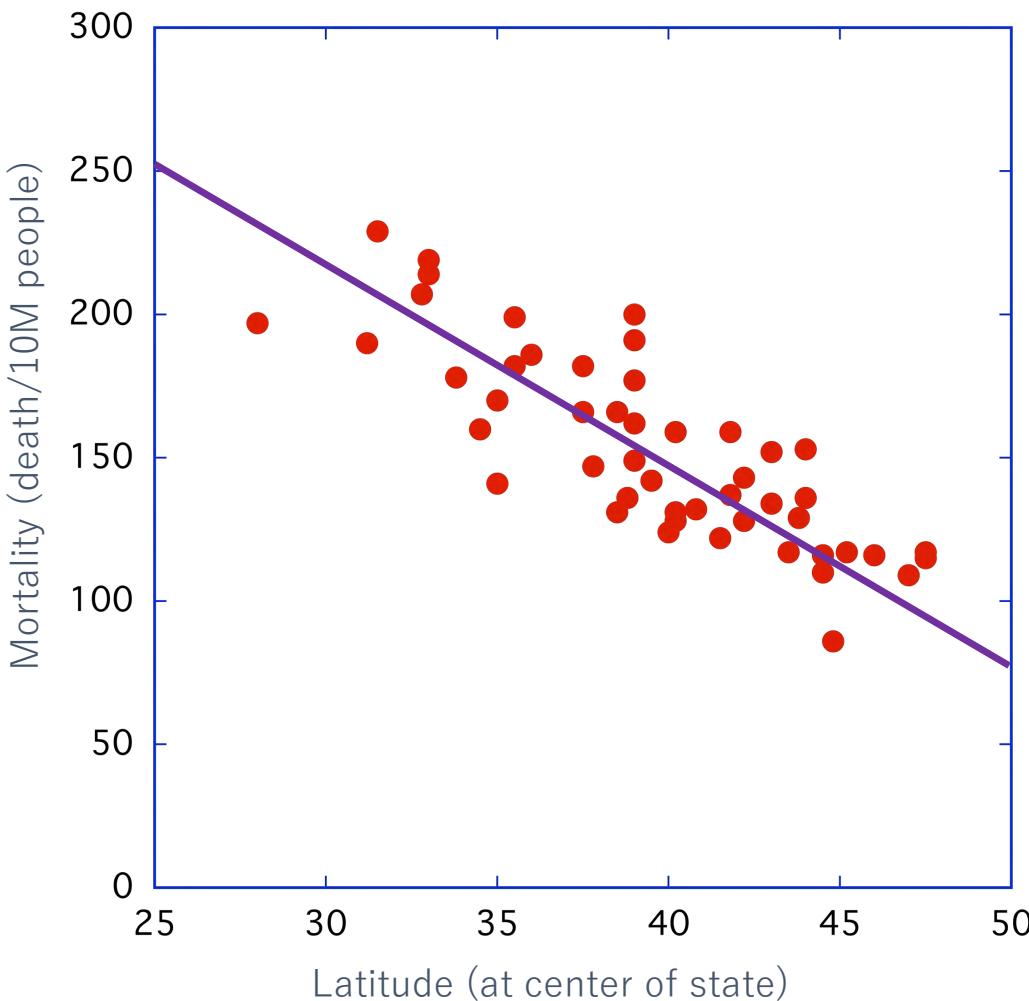
$$\frac{\partial D}{\partial \beta_0} = -2 \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i) = 0$$

$$\sum_{i=1}^n y_i = n\beta_0 + \beta_1 \sum_{i=1}^n x_i$$

$$\frac{\partial D}{\partial \beta_1} = -2 \sum_{i=1}^n x_i (y_i - \beta_0 - \beta_1 x_i) = 0$$

$$\sum_{i=1}^n y_i x_i = \beta_0 \sum_{i=1}^n x_i + \beta_1 \sum_{i=1}^n x_i^2$$

The least squares fit



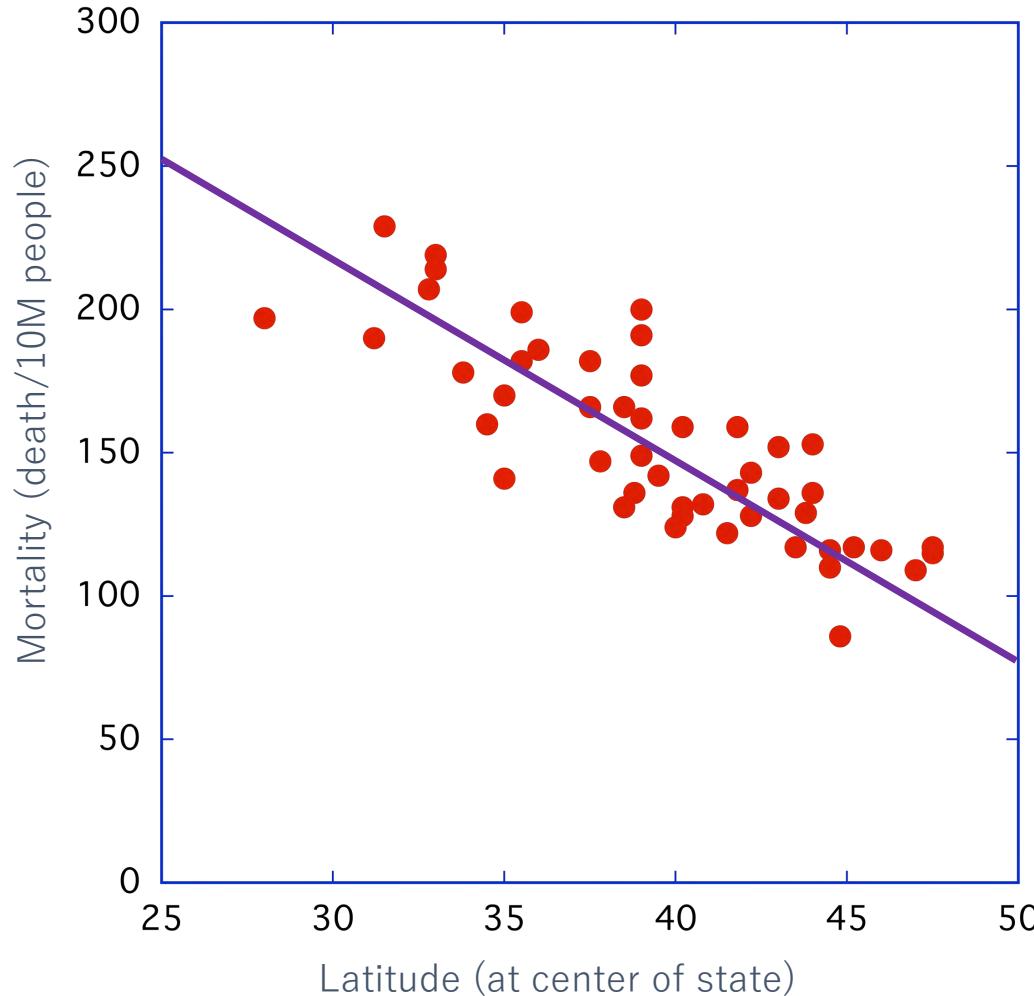
$$\sum_{i=1}^n x_i \quad \xrightarrow{\hspace{1cm}} \quad \sum_{i=1}^n y_i = n\beta_0 + \beta_1 \sum_{i=1}^n x_i$$

$$n \quad \xrightarrow{\hspace{1cm}} \quad \sum_{i=1}^n y_i x_i = \beta_0 \sum_{i=1}^n x_i + \beta_1 \sum_{i=1}^n x_i^2$$

$$\beta_1 = \frac{\sum_{i=1}^n x_i \sum_{i=1}^n y_i - n \sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i \sum_{i=1}^n x_i - n \sum_{i=1}^n x_i^2}$$

$$\beta_0 = \frac{\sum_{i=1}^n y_i - \beta_1 \sum_{i=1}^n x_i}{n}$$

Learning a simple linear regression model



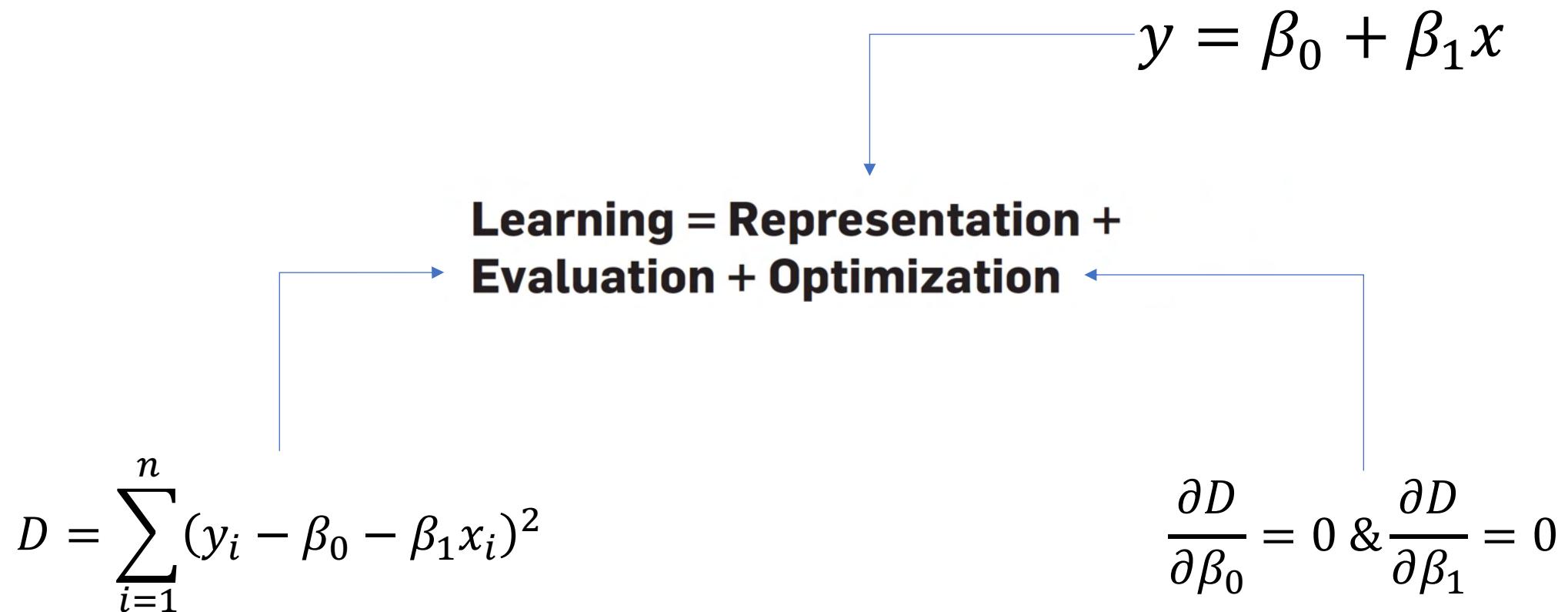
$$y = \beta_0 + \beta_1 x$$

$$D = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2$$

$$\beta_1 = \frac{\sum_{i=1}^n x_i \sum_{i=1}^n y_i - n \sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i \sum_{i=1}^n x_i - n \sum_{i=1}^n x_i^2}$$

$$\beta_0 = \frac{\sum_{i=1}^n y_i - \beta_1 \sum_{i=1}^n x_i}{n}$$

Learning a simple linear regression model



Learning a simple linear regression model

Why?

Is it an absolute truth? How sure?

Learning from observation?

What does the learning process base on?

What is the underlying hypothesis?

$$y = \beta_0 + \beta_1 x$$

**Learning = Representation +
Evaluation + Optimization**

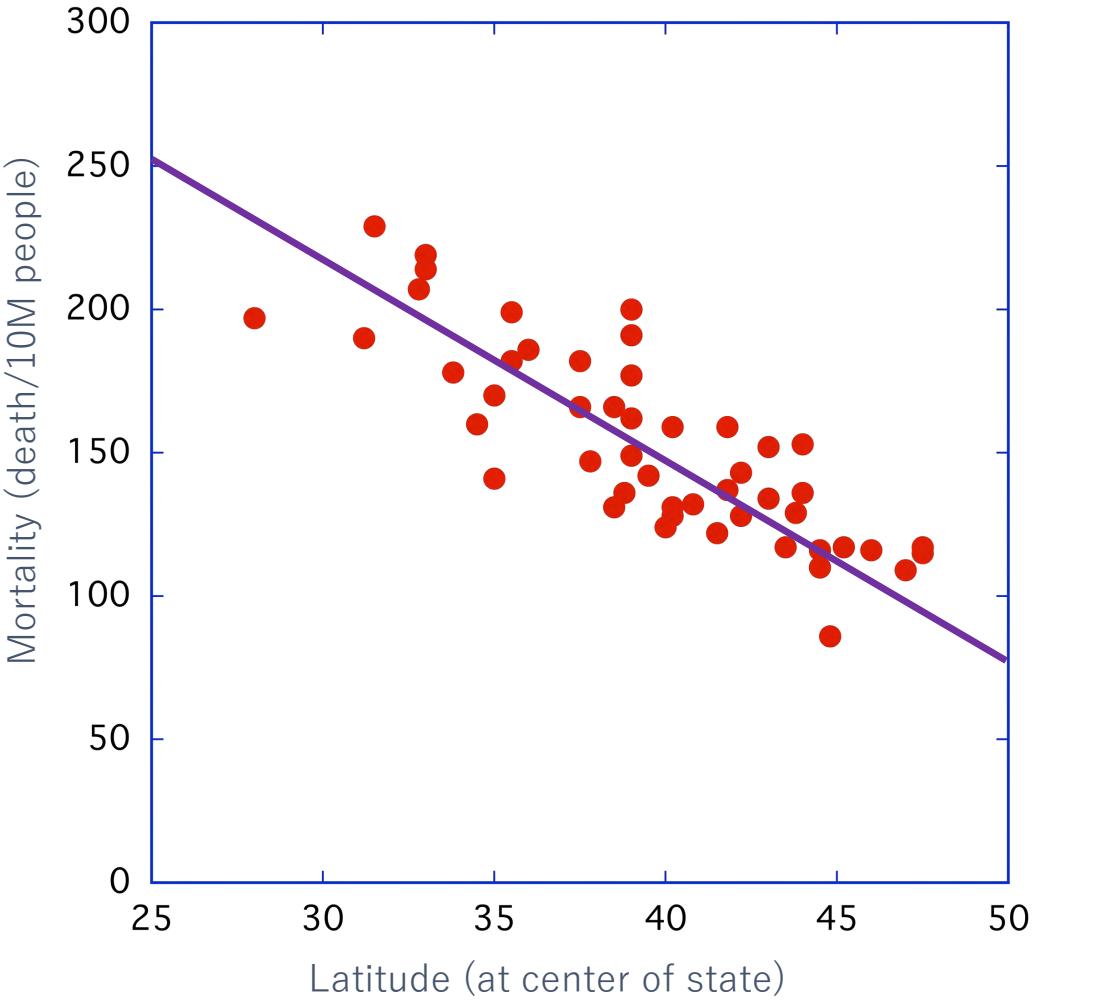
$$D = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2$$

$$\frac{\partial D}{\partial \beta_0} = 0 \text{ } \& \text{ } \frac{\partial D}{\partial \beta_1} = 0$$

Outline

1. Introduction
2. Warming-up exercises
3. Regression analysis with statistical learning
 - Learning simple linear regression models by least square fit
 - Estimation and evaluation of linear regression models by statistical learning

A straight-line regression model



$$y = f(x)$$



$$\hat{y} = \hat{f}(x) \\ = \hat{\beta}_0 + \hat{\beta}_1 x$$

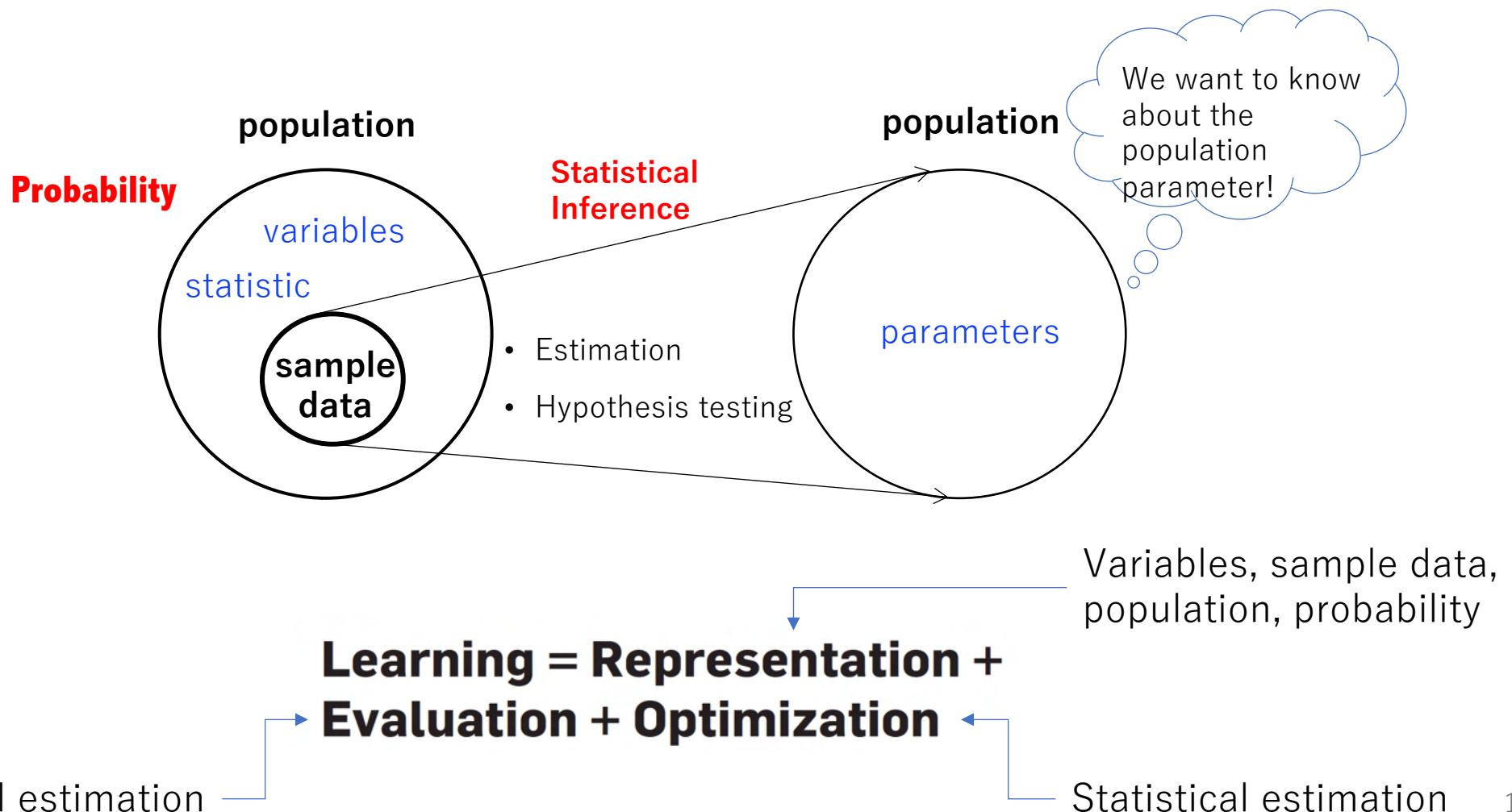
\hat{y} : prediction of y

\hat{f} : estimation of f

$\hat{\beta}_0 + \hat{\beta}_1 x$: just one form of f

| | A | B | C | D | E |
|----|----------------|----------|-----------|-------|-----------|
| 1 | State | Latitude | Longitude | Ocean | Mortality |
| 2 | Alabama | 33 | 87 | 1 | 219 |
| 3 | Arizona | 34.5 | 112 | 0 | 160 |
| 4 | Arkansas | 35 | 92.5 | 0 | 170 |
| 5 | California | 37.5 | 119.5 | 1 | 182 |
| 6 | Colorado | 39 | 105.5 | 0 | 149 |
| 7 | Connecticut | 41.8 | 72.8 | 1 | 159 |
| 8 | Delaware | 39 | 75.5 | 1 | 200 |
| 9 | Wash.D.C. | 39 | 77 | 0 | 177 |
| 10 | Florida | 28 | 82 | 1 | 197 |
| 11 | Georgia | 33 | 83.5 | 1 | 214 |
| 12 | Idaho | 44.5 | 114 | 0 | 116 |
| 13 | Illinois | 40 | 89.5 | 0 | 124 |
| 14 | Indiana | 40.2 | 86.2 | 0 | 128 |
| 15 | Iowa | 42.2 | 93.8 | 0 | 128 |
| 16 | Kansas | 38.5 | 98.5 | 0 | 166 |
| 17 | Kentucky | 37.8 | 85 | 0 | 147 |
| 18 | Louisiana | 31.2 | 91.8 | 1 | 190 |
| 19 | Maine | 45.2 | 69 | 1 | 117 |
| 20 | Maryland | 39 | 76.5 | 1 | 162 |
| 21 | Massachusetts | 42.2 | 71.8 | 1 | 143 |
| 22 | Michigan | 43.5 | 84.5 | 0 | 117 |
| 23 | Minnesota | 46 | 94.5 | 0 | 116 |
| 24 | Mississippi | 32.8 | 90 | 1 | 207 |
| 25 | Missouri | 38.5 | 92 | 0 | 131 |
| 26 | Montana | 47 | 110.5 | 0 | 109 |
| 27 | Nebraska | 41.5 | 99.5 | 0 | 122 |
| 28 | Nevada | 39 | 117 | 0 | 191 |
| 29 | New Hampshire | 43.8 | 71.5 | 1 | 129 |
| 30 | New Jersey | 40.2 | 74.5 | 1 | 159 |
| 31 | New Mexico | 35 | 106 | 0 | 141 |
| 32 | New York | 43 | 75.5 | 1 | 152 |
| 33 | North Carolina | 35.5 | 79.5 | 1 | 199 |
| 34 | North Dakota | 47.5 | 100.5 | 0 | 115 |
| 35 | Ohio | 40.2 | 82.8 | 0 | 131 |
| 36 | Oklahoma | 35.5 | 97.2 | 0 | 182 |
| 37 | Oregon | 44 | 120.5 | 1 | 136 |
| 38 | Pennsylvania | 40.8 | 77.8 | 0 | 132 |
| 39 | Rhode Island | 41.8 | 71.5 | 1 | 137 |
| 40 | South Carolina | 33.8 | 81 | 1 | 178 |
| 41 | South Dakota | 44.8 | 100 | 0 | 86 |
| 42 | Tennessee | 36 | 86.2 | 0 | 186 |
| 43 | Texas | 31.5 | 98 | 1 | 229 |
| 44 | Utah | 39.5 | 111.5 | 0 | 142 |
| 45 | Vermont | 44 | 72.5 | 1 | 153 |
| 46 | Virginia | 37.5 | 78.5 | 1 | 166 |
| 47 | Washington | 47.5 | 121 | 1 | 117 |
| 48 | West Virginia | 38.8 | 80.8 | 0 | 136 |
| 49 | Wisconsin | 44.5 | 90.2 | 0 | 110 |
| 50 | Wyoming | 43 | 107.5 | 0 | 134 |

Statistics vs. Machine learning



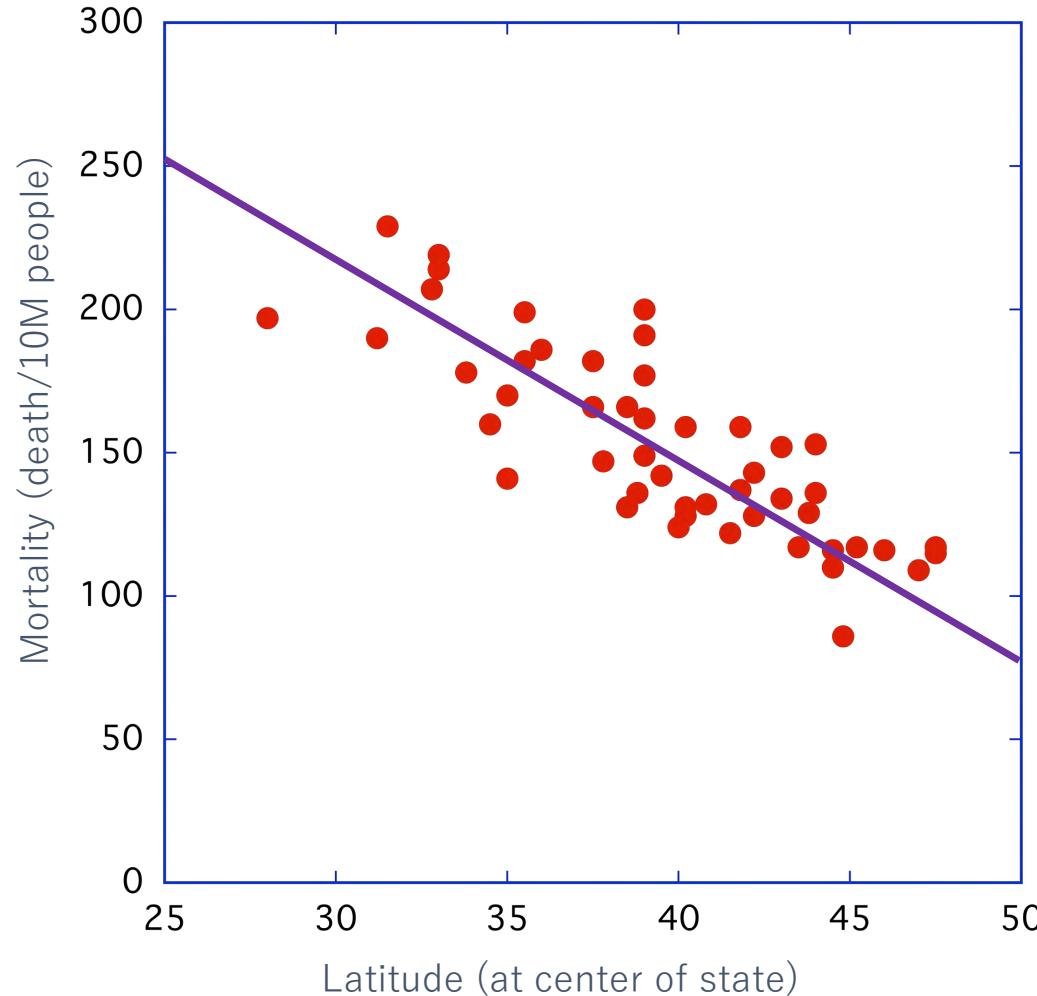
Statistical thinking in data science

1. Statistical thinking relates processes and statistics, and is based on the following principles:
 - ✓ All work occurs in a system of **interconnected** processes
 - ✓ Variation **exists** in all processes
 - ✓ **Understanding** and **reducing** variation are keys to success
2. Statistical thinking plays an essential role in data science

\hat{y} : prediction of y

\hat{f} : estimation of f

Evaluation of a regression model



Hypothesis

Estimation

Evaluation

$$y_{obs} = y_{predict} + \epsilon$$

$\epsilon_{reducible}$

$\epsilon_{irreducible}$

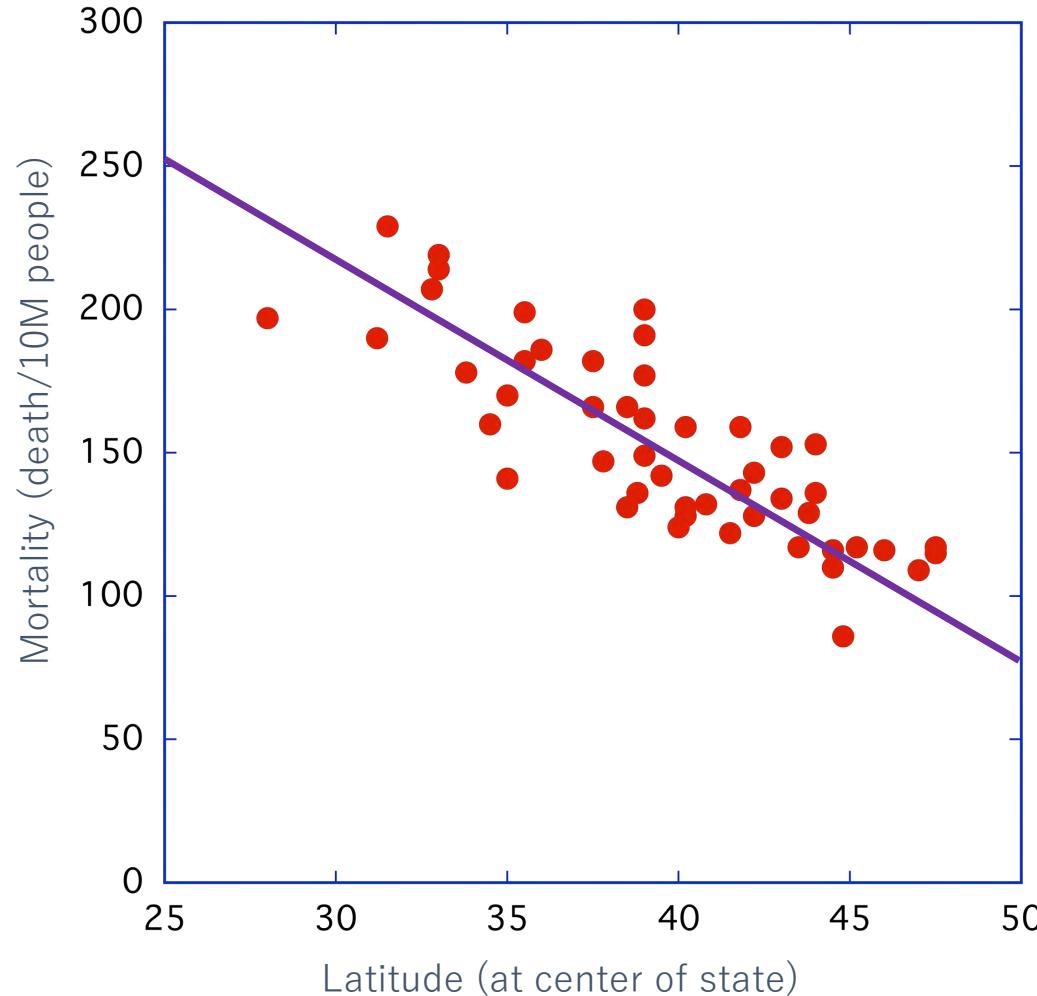
$$y = f(x)$$

$$\hat{y} = \hat{f}(x)$$

\hat{y} : prediction of y

\hat{f} : estimation of f

Evaluation of a regression model



Hypothesis

Estimation

Evaluation

Statistical thinking

$$y = f(x)$$

$$\hat{y} = \hat{f}(x)$$

$$y_{obs} = y_{predict} + \epsilon$$

Understand and reduce

$\epsilon_{reducible}$

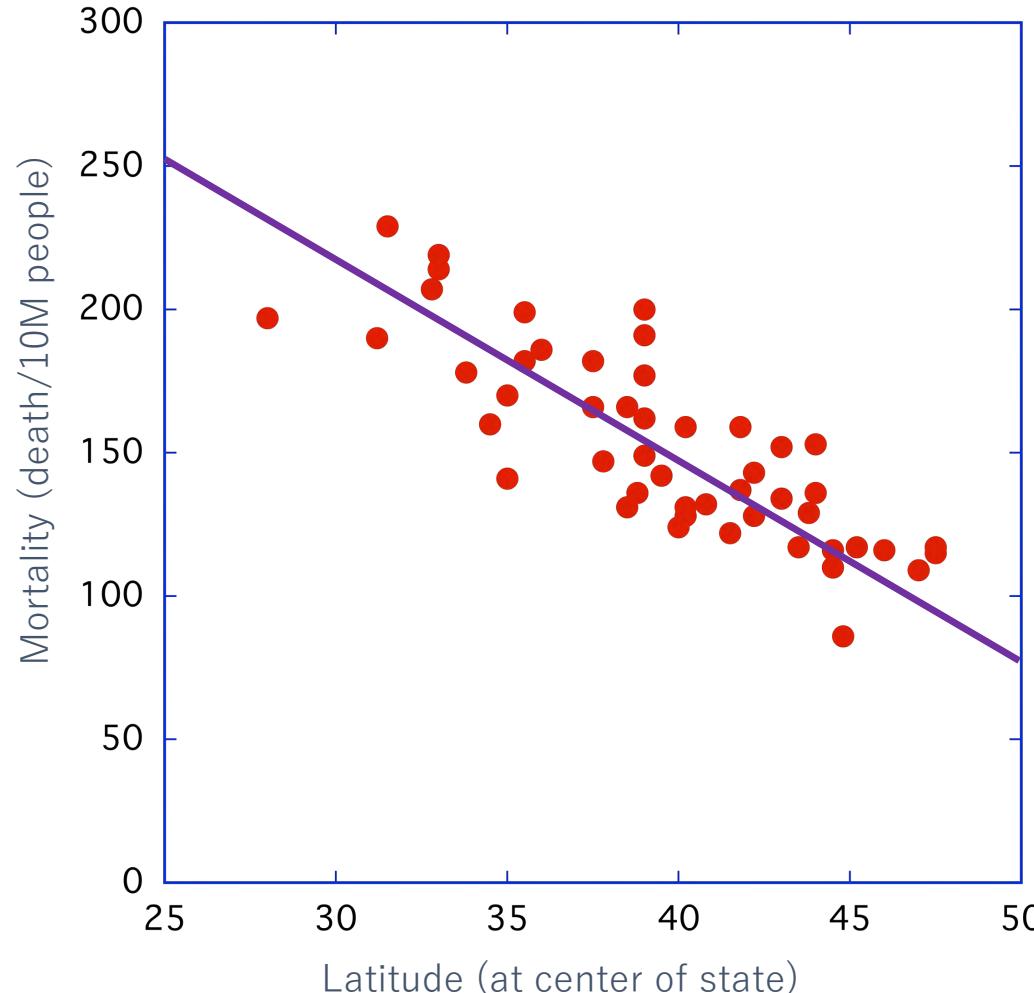
$\epsilon_{irreducible}$

Understand the limitation of the hypothesis¹⁷

\hat{y} : prediction of y

\hat{f} : estimation of f

Evaluation of a regression model



Hypothesis

$$y = f(x)$$

Estimation

$$\hat{y} = \hat{f}(x)$$

Evaluation

$$y_{obs} = y_{predict} + \epsilon$$

Statistical thinking for prediction

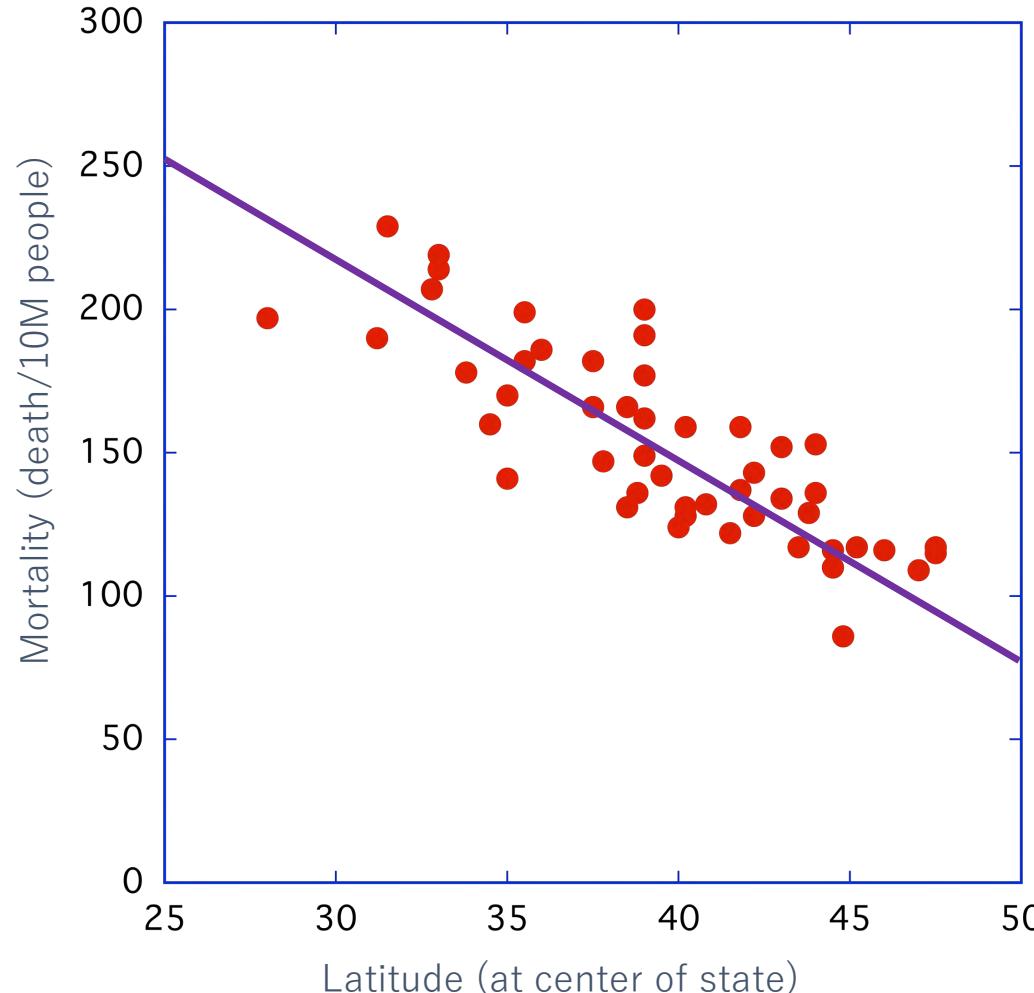
Minimization

$$E(y_{obs} - y_{predict})^2 = [f(x) - \hat{f}(x)]^2 + \text{Var } (\epsilon_{irreducible})$$

\hat{y} : prediction of y

\hat{f} : estimation of f

Evaluation of a regression model



Hypothesis

$$y = f(x)$$

Estimation

$$\hat{y} = \hat{f}(x)$$

Evaluation

$$y_{obs} = y_{predict} + \epsilon$$

Statistical thinking for inference

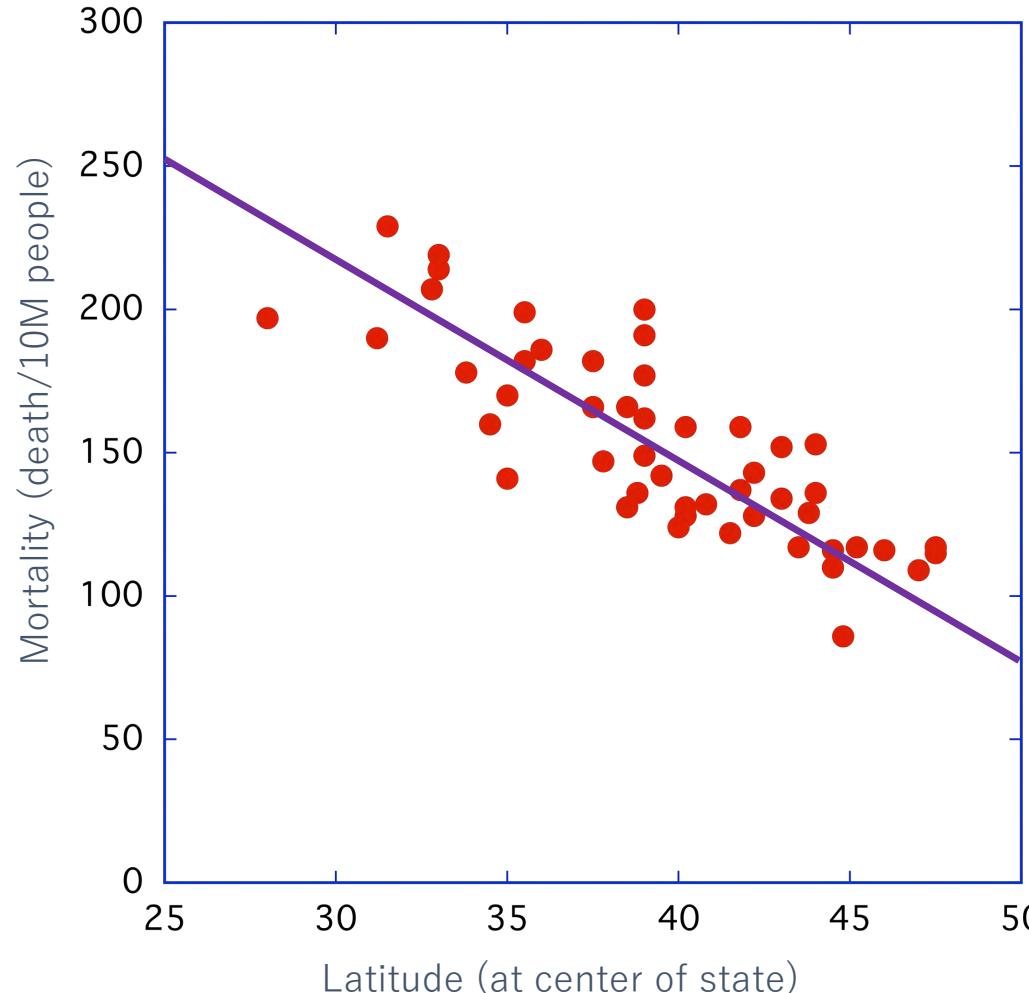
$$E(y_{obs} - y_{predict})^2 = [f(x) - \hat{f}(x)]^2 + \text{Var } (\epsilon_{irreducible})$$

Understand
the relationship
between x and y

\hat{y} : prediction of y

\hat{f} : estimation of f

Evaluation of a regression model



Hypothesis

$$y = f(x)$$

Estimation

$$\hat{y} = \hat{f}(x)$$

Evaluation

$$y_{obs} = y_{predict} + \epsilon$$

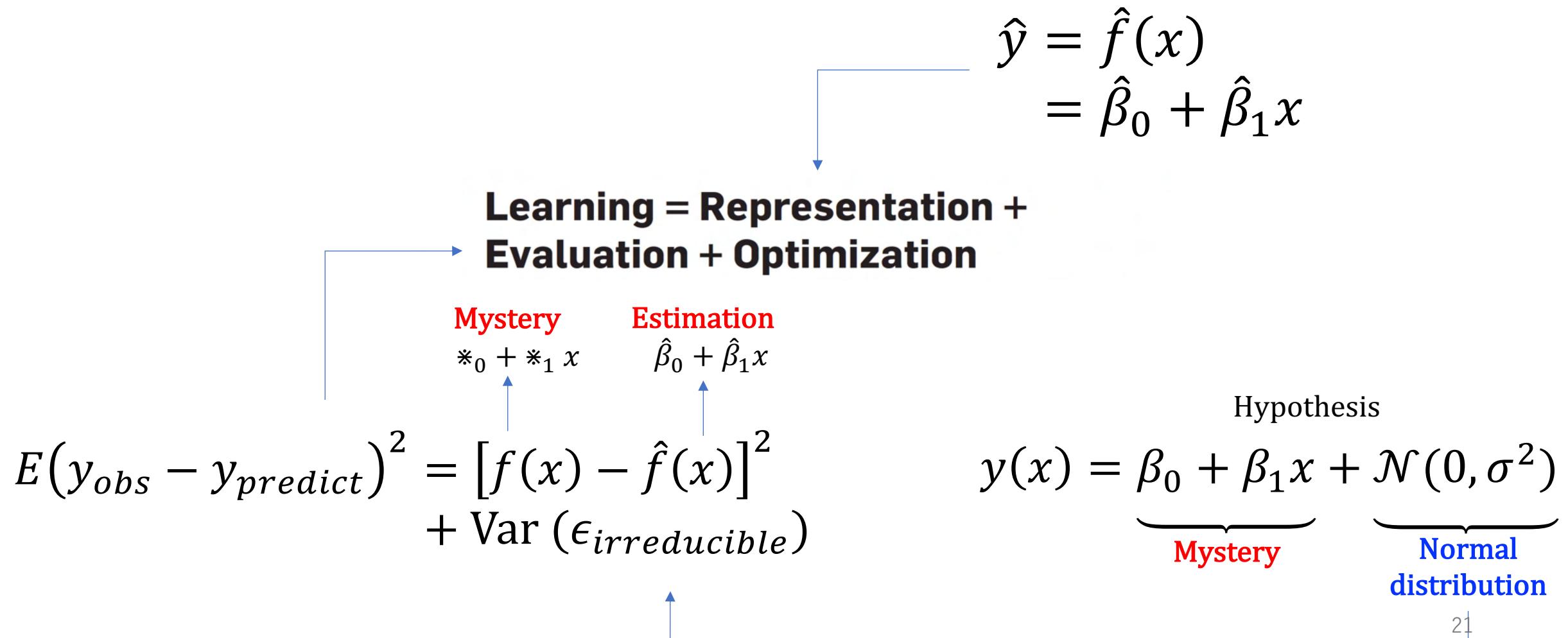
From the nature
of the relationship
between x and y

From the fixity in
the form of the
hypothetical function

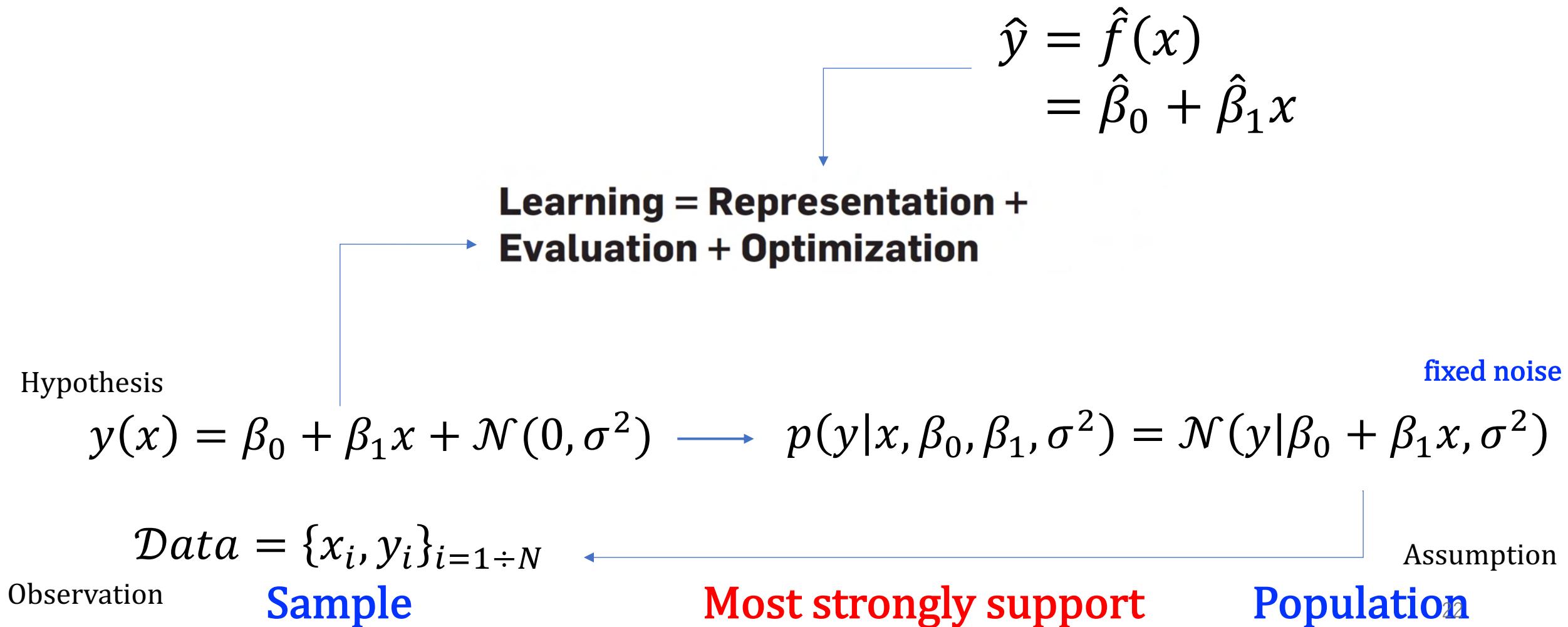
$\epsilon_{irreducible}$

Understand the limitation of the hypothesis²⁰

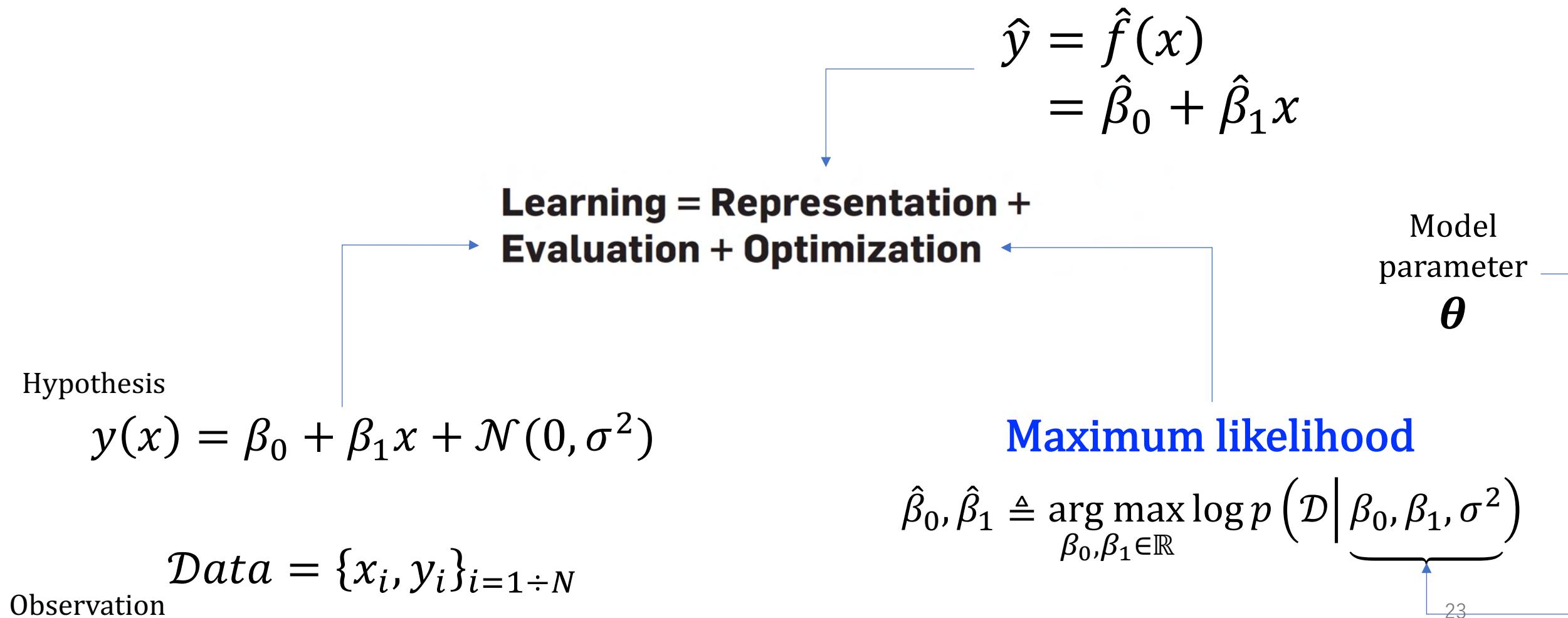
Estimation of a simple linear regression model



Estimation of a simple linear regression model



Estimation of a simple linear regression model



Estimation of a simple linear regression model

Observation

$$\mathcal{D}ata = \{x_i, y_i\}_{i=1 \div N}$$

Log-likelihood

$$\begin{aligned}\ell(\boldsymbol{\theta}) &\triangleq \log p(\mathcal{D} | \beta_0, \beta_1, \sigma^2) = \sum_{i=1}^N \log p(y_i | \mathbf{x}_i, \boldsymbol{\theta}) \\ &= \sum_{i=1}^N \log \left[\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(y_i - \beta_0 - \beta_1 x_i)^2} \right] \\ &= -\frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \beta_0 - \beta_1 x_i)^2 + \frac{N}{\sqrt{2\pi}\sigma}\end{aligned}$$

Estimation of a simple linear regression model

Observation

$$\mathcal{D}ata = \{x_i, y_i\}_{i=1 \div N}$$

Log-likelihood

$$\begin{aligned}\ell(\boldsymbol{\theta}) &\triangleq \log p(\mathcal{D} | \beta_0, \beta_1, \sigma^2) = \sum_{i=1}^N \log p(y_i | \mathbf{x}_i, \boldsymbol{\theta}) \\ &= \sum_{i=1}^N \log \left[\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(y_i - \beta_0 - \beta_1 x_i)^2} \right] \\ &= \boxed{-\frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \beta_0 - \beta_1 x_i)^2 + \frac{N}{\sqrt{2\pi}\sigma}}\end{aligned}$$

Maximum log-likelihood
= Least square fit

Statistical thinking in data science

1. Statistical thinking relates processes and statistics, and is based on the following principles:
 - ✓ All work occurs in a system of **interconnected** processes
 - ✓ Variation **exists** in all processes
 - ✓ **Understanding** and **reducing** variation are keys to success
2. Statistical thinking plays an essential role in data science

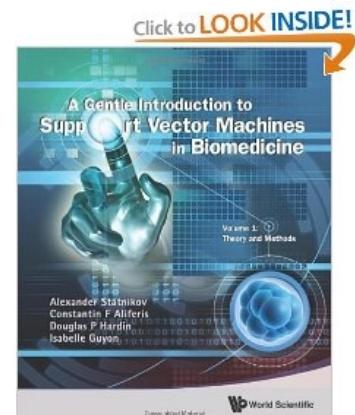
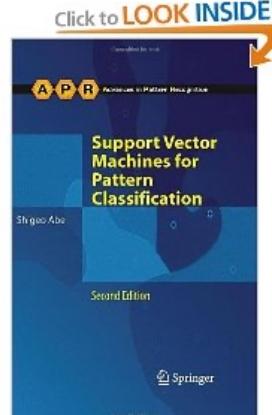
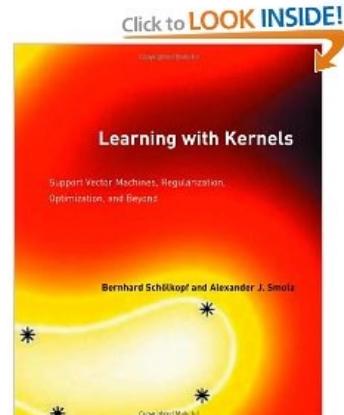
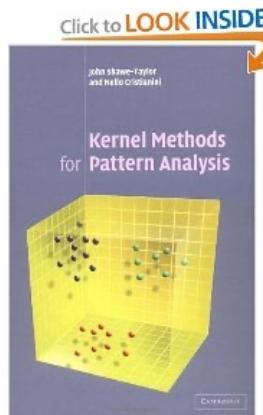
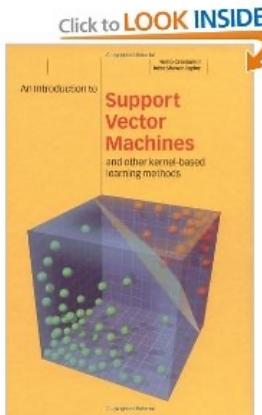
K490: データサイエンス論

Lecture 6: サポートベクトル分類

Lecturer: Hieu-Chi Dam, Takashi Isogai

Content

1. Introduction
2. Linear support vector machines
3. Nonlinear support vector machines
4. Multiclass support vector machines
5. Other issues
6. Challenges for kernel methods and SVMs

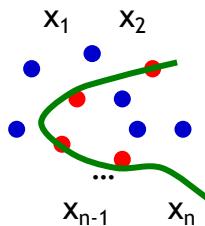


Introduction

- SVMs are of great interest to theoretical researchers and applied scientists.
- By means of the new technology of *kernel methods*, SVMs have been very successful in building highly nonlinear classifiers.
- SVMs have also been successful in dealing with situations in which there are many more variables than observations, and complexly structured data.
- Wide applications in machine learning, natural language processing, bioinformatics.

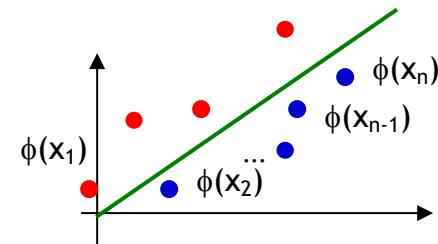
Kernel methods: key idea

Input space X



inverse map ϕ^{-1}
 $\phi(x)$

Feature space F

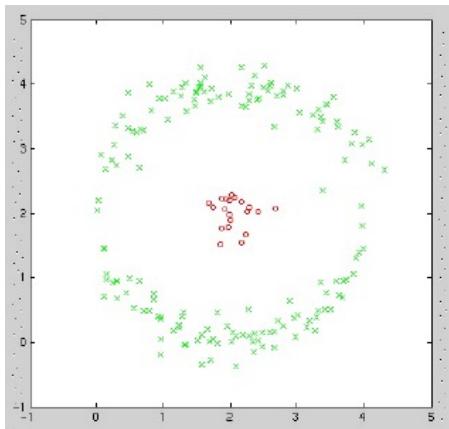


kernel function $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$

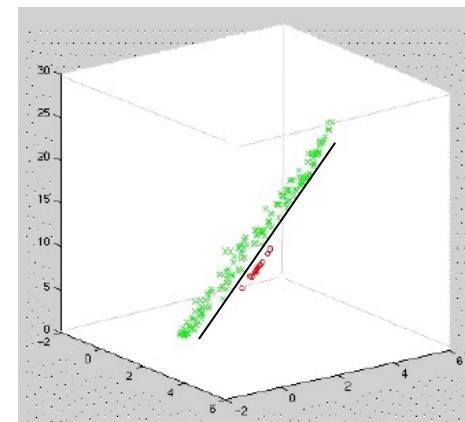
$$k(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$$

Kernel matrix $K_{n \times n}$

kernel-based algorithm on K
(computation on kernel matrix)

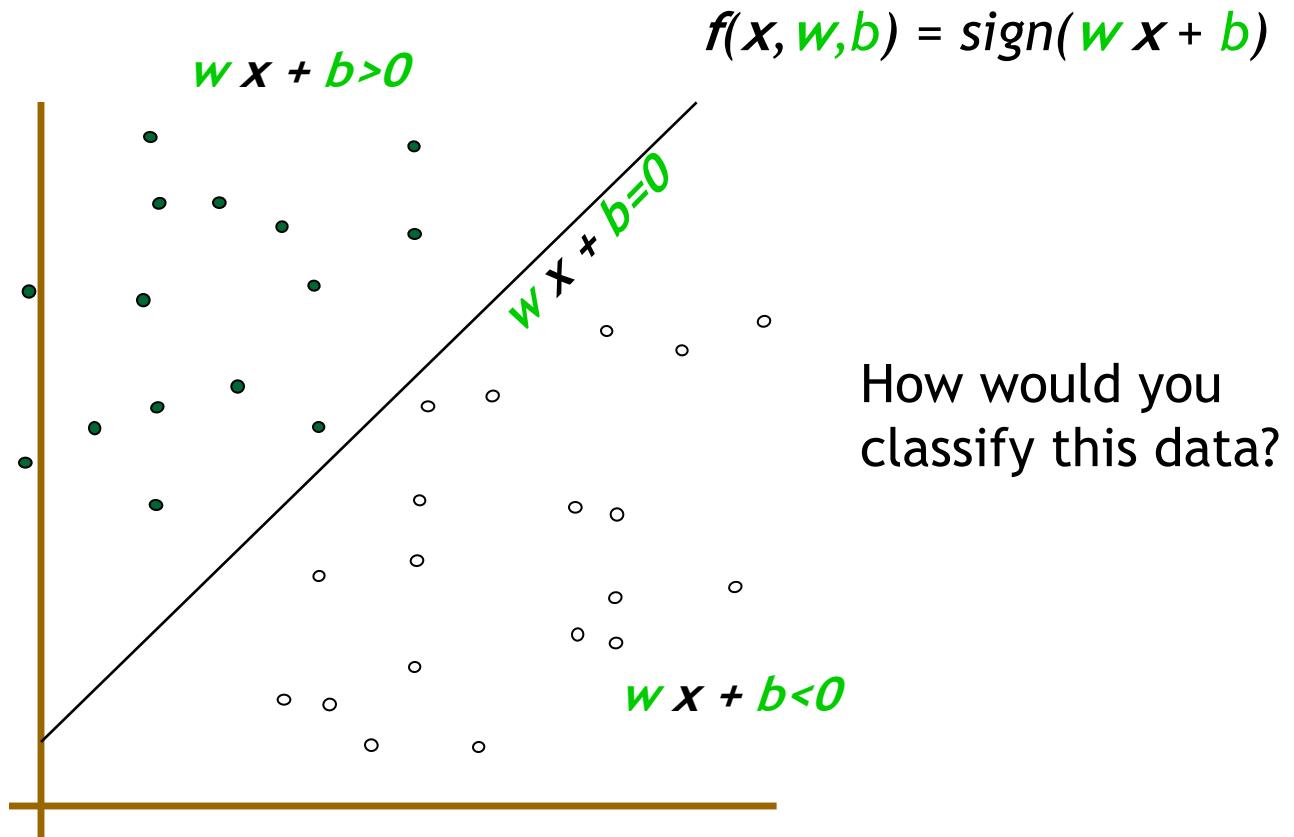


$$\begin{aligned}\phi: \mathcal{X} = \mathbb{R}^2 &\rightarrow \mathcal{H} = \mathbb{R}^3 \\ (x_1, x_2) &\mapsto (x_1, x_2, x_1^2 + x_2^2)\end{aligned}$$



Support vector machines: idea

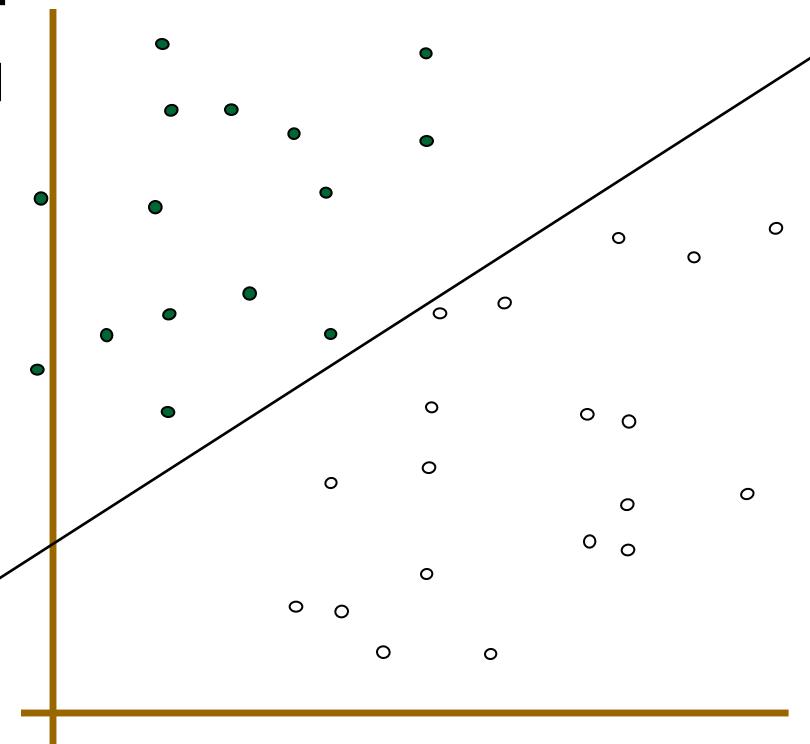
- denotes +1
- denotes -1



How would you
classify this data?

Support vector machines: idea

- denotes +1
- denotes -1

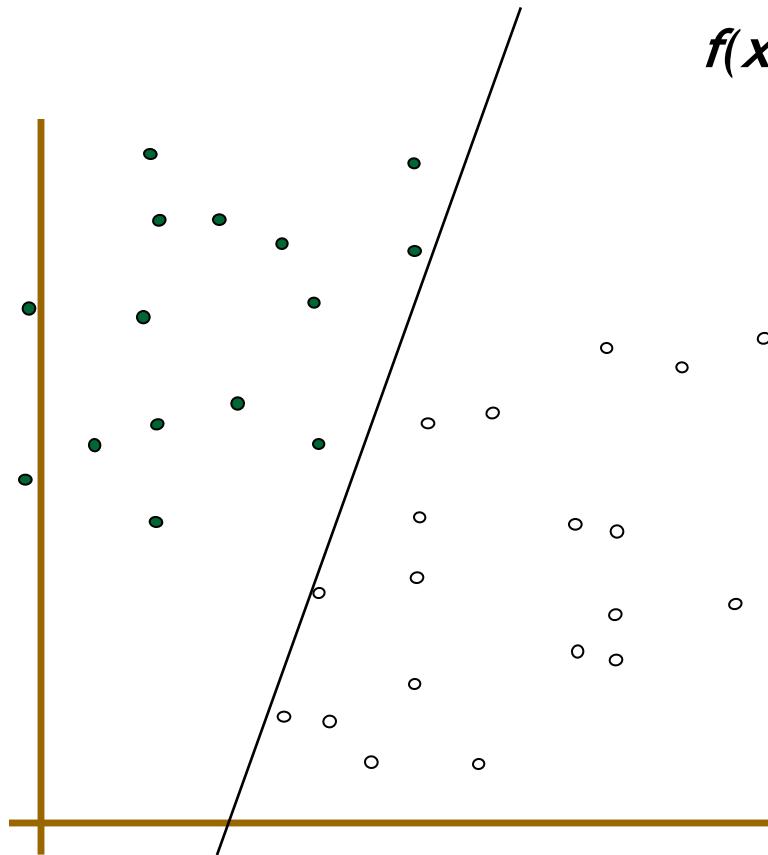


$$f(x, \mathbf{w}, b) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b)$$

How would you
classify this data?

Support vector machines: idea

- denotes +1
- denotes -1

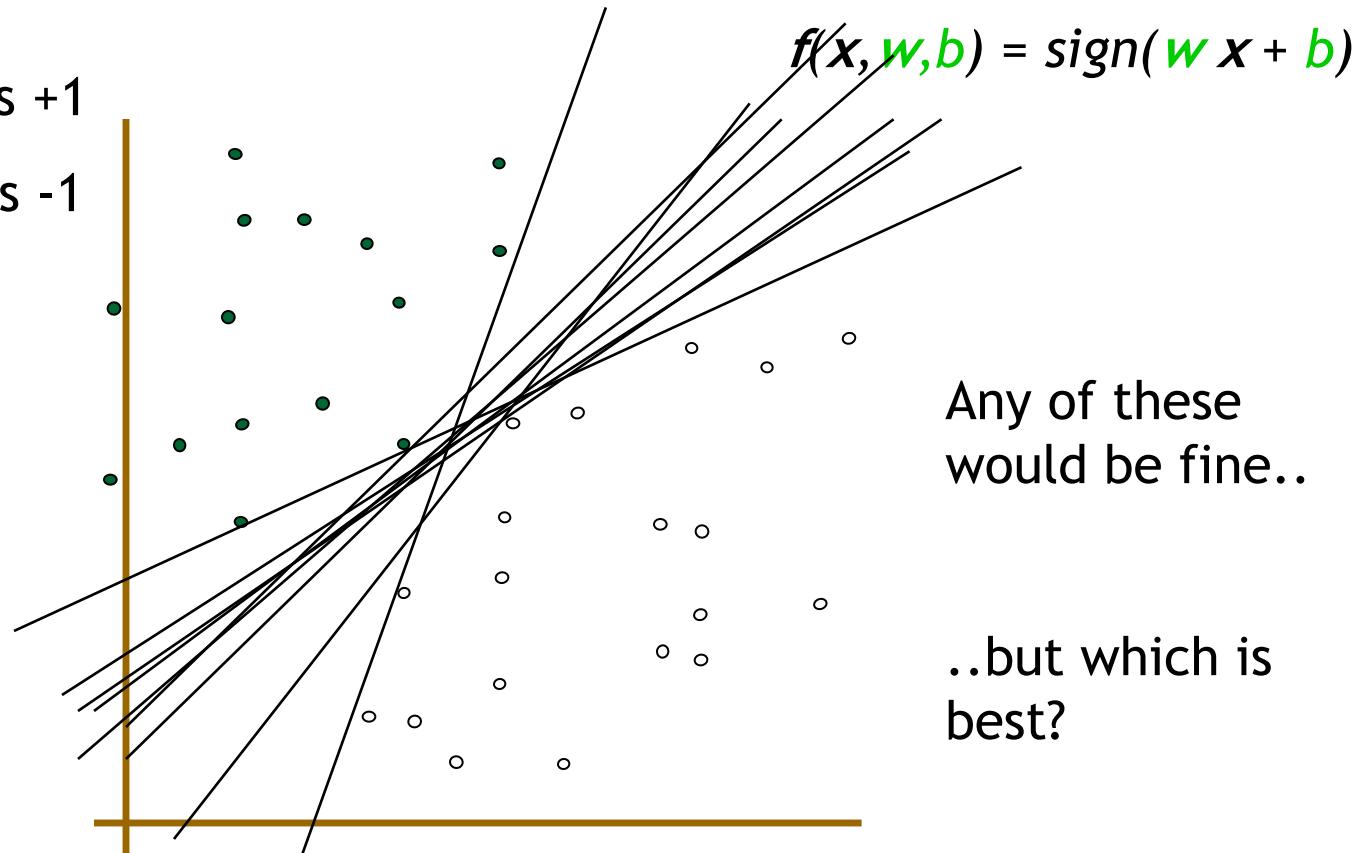


$$f(x, w, b) = \text{sign}(w x + b)$$

How would you
classify this data?

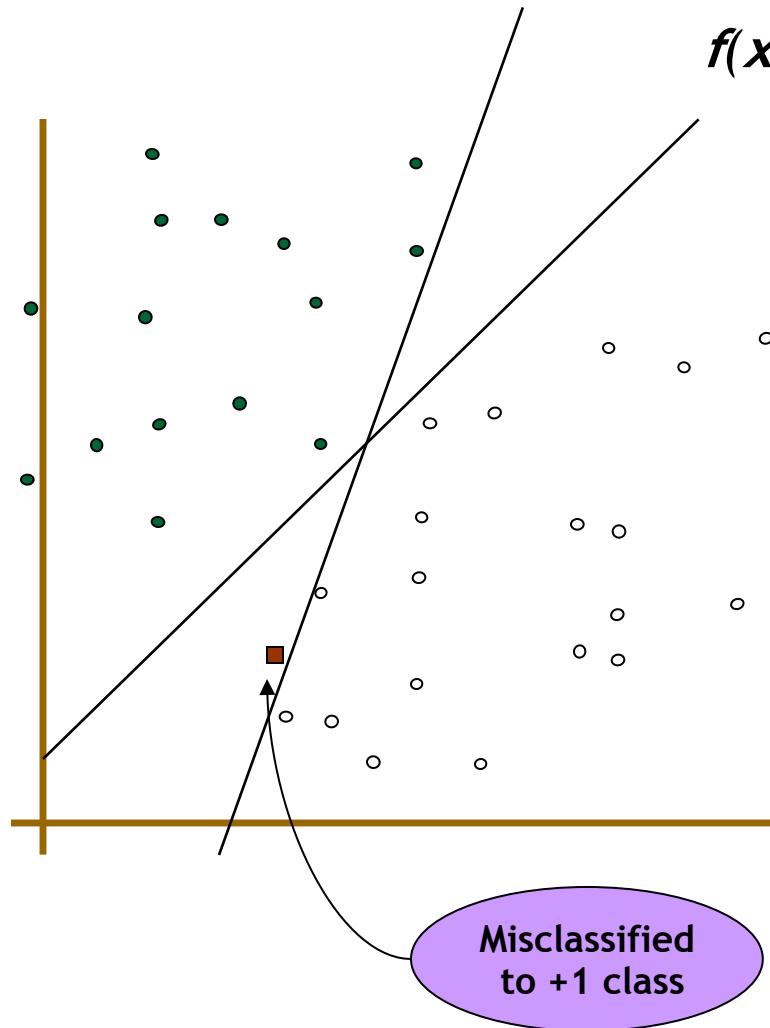
Support vector machines: idea

- denotes +1
- denotes -1



Support vector machines: idea

- denotes +1
- denotes -1

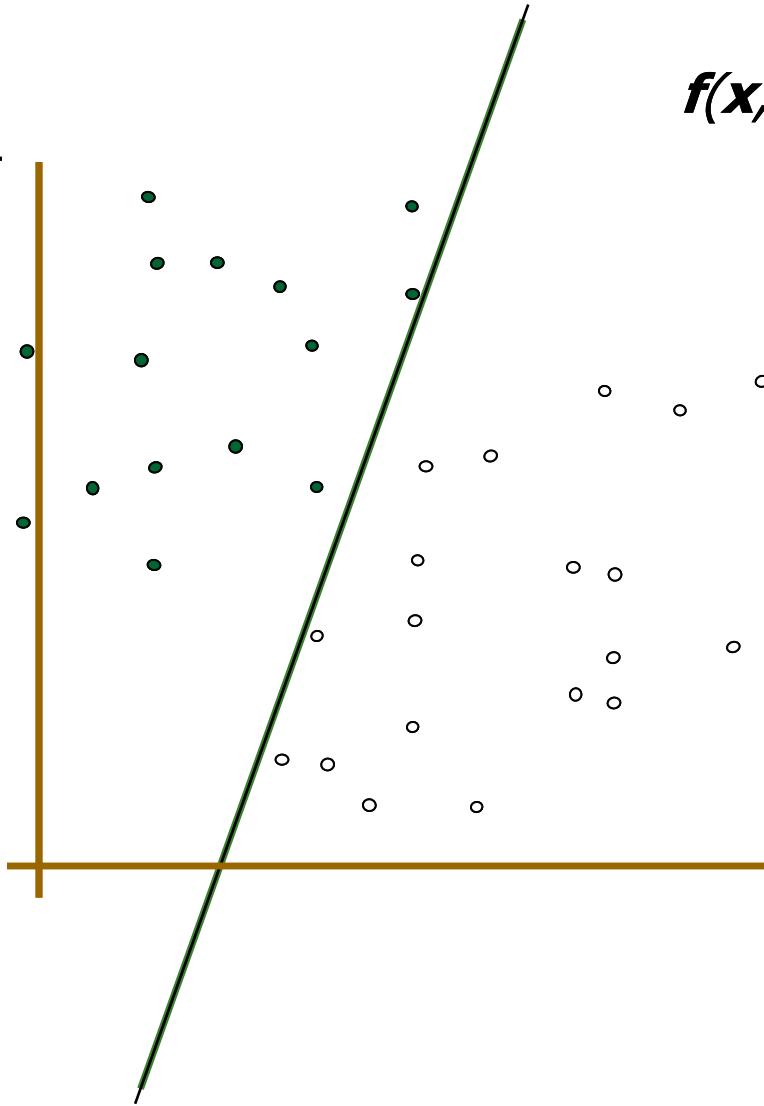


$$f(x, w, b) = \text{sign}(w x + b)$$

How would you
classify this data?

Support vector machines: idea

- denotes +1
- denotes -1



$$f(\mathbf{x}, \mathbf{w}, b) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b)$$

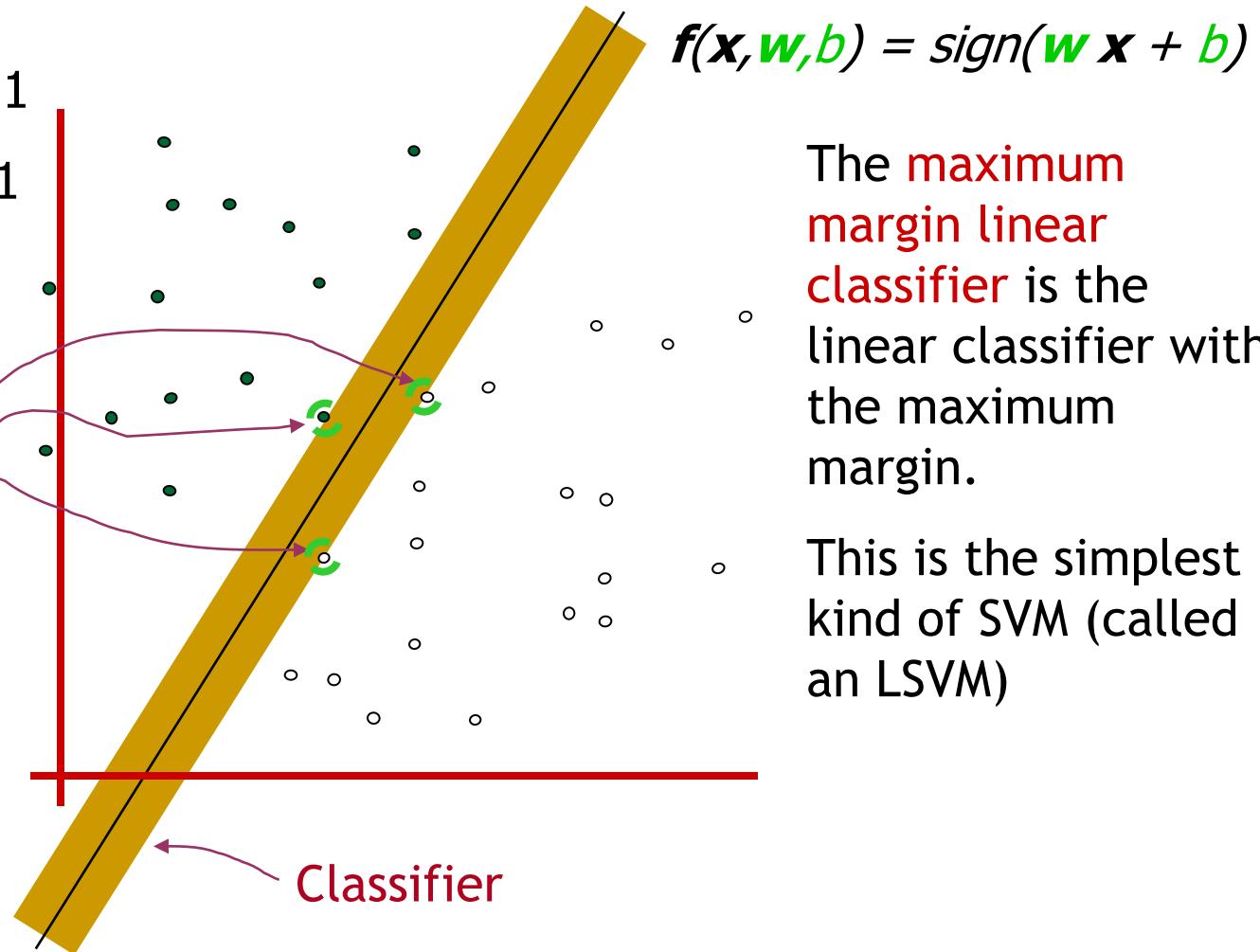
Define the **margin** of a linear classifier as the width that the boundary could be increased by before hitting a datapoint.

Maximum margin

1. Maximizing the margin is good according to intuition and PAC theory
2. Implies that only support vectors are important; other training examples are ignorable.
3. Empirically it works very very well.

- denotes +1
- denotes -1

Support vectors
are those
datapoints that
the margin
pushes up
against



Content

1. Introduction
2. Linear support vector machines
3. Nonlinear support vector machines
4. Multiclass support vector machines
5. Other issues
6. Challenges for kernel methods and SVMs

Linear support vector machines

The linearly separable case

- Learning set of data $\mathcal{L} = \{(x_i, y_i) : i = 1, 2, \dots, n\}$, $x_i \in \mathbb{R}^r$, $y_i \in \{-1, +1\}$.
- The binary classification problem is to use \mathcal{L} to construct a function $f: \mathbb{R}^r \rightarrow \mathbb{R}$ so that $C(x) = \text{sign}(f(x))$ is a classifier.
- Function f classifies each x in a test set \mathcal{T} into one of two classes, $\Pi+$ or $\Pi-$, depending upon whether $C(x)$ is $+1$ (if $f(x) \geq 0$) or -1 (if $f(x) < 0$), respectively. The goal is to have f assign all positive points in \mathcal{T} (i.e., those with $y = +1$) to $\Pi+$ and all negative points in \mathcal{T} ($y = -1$) to $\Pi-$.
- The simplest situation: positive ($y_i = +1$) and negative ($y_i = -1$) data points from the learning set \mathcal{L} can be separated by a hyperplane,

$$\{x : f(x) = \beta_0 + x^\tau \boldsymbol{\beta} = 0\} \quad (1)$$

$\boldsymbol{\beta}$ is the *weight vector* with Euclidean norm $\|\boldsymbol{\beta}\|$, and β_0 is the *bias*.

Linear support vector machines

The linearly separable case

- If no error, the hyperplane is called a *separating hyperplane*.
- Let d_- and d_+ be the shortest distance from the separating hyperplane to the nearest negative and positive data points. Then, the *margin* of the separating hyperplane is defined as $d = d_- + d_+$.
- We look for *maximal margin classifier* (optimal separating hyperplane).
- If the learning data are linearly separable, $\exists \beta_0$ and $\boldsymbol{\beta}$ such that

$$\beta_0 + \mathbf{x}_i^\tau \boldsymbol{\beta} \geq +1, \text{ if } y_i = +1 \quad (2) \quad \beta_0 + \mathbf{x}_i^\tau \boldsymbol{\beta} \leq -1, \text{ if } y_i = -1 \quad (3)$$

- If there are data vectors in \mathcal{L} such that equality holds in (1), then they lie on the hyperplane H_{+1} : $(\beta_0 - 1) + \mathbf{x}^\tau \boldsymbol{\beta} = 0$; similarly, for hyperplane H_{-1} : $(\beta_0 + 1) + \mathbf{x}^\tau \boldsymbol{\beta} = 0$. Points in \mathcal{L} that lie on either one of the hyperplanes H_{-1} or H_{+1} , are said to be *support vectors*.

Linear support vector machines

The linearly separable case

- If \mathbf{x}_{-1} lies on H_{-1} , and if \mathbf{x}_{+1} lies on H_{+1} , then

$$\beta_0 + \mathbf{x}_{-1}^\tau \boldsymbol{\beta} = -1 \text{ and } \beta_0 + \mathbf{x}_{+1}^\tau \boldsymbol{\beta} = +1$$

the difference between them is $\mathbf{x}_{+1}^\tau \boldsymbol{\beta} - \mathbf{x}_{-1}^\tau \boldsymbol{\beta} = 2$ and their sum is $\beta_0 = -\frac{1}{2}(\mathbf{x}_{+1}^\tau \boldsymbol{\beta} - \mathbf{x}_{-1}^\tau \boldsymbol{\beta})$. The perpendicular distances of the hyperplane $\beta_0 + \mathbf{x}^\tau \boldsymbol{\beta} = 0$ to \mathbf{x}_{-1} and \mathbf{x}_{+1} are

$$d_- = \frac{|\beta_0 + \mathbf{x}_{-1}^\tau \boldsymbol{\beta}|}{\|\boldsymbol{\beta}\|} = \frac{1}{\|\boldsymbol{\beta}\|}$$

$$d_+ = \frac{|\beta_0 + \mathbf{x}_{+1}^\tau \boldsymbol{\beta}|}{\|\boldsymbol{\beta}\|} = \frac{1}{\|\boldsymbol{\beta}\|}$$

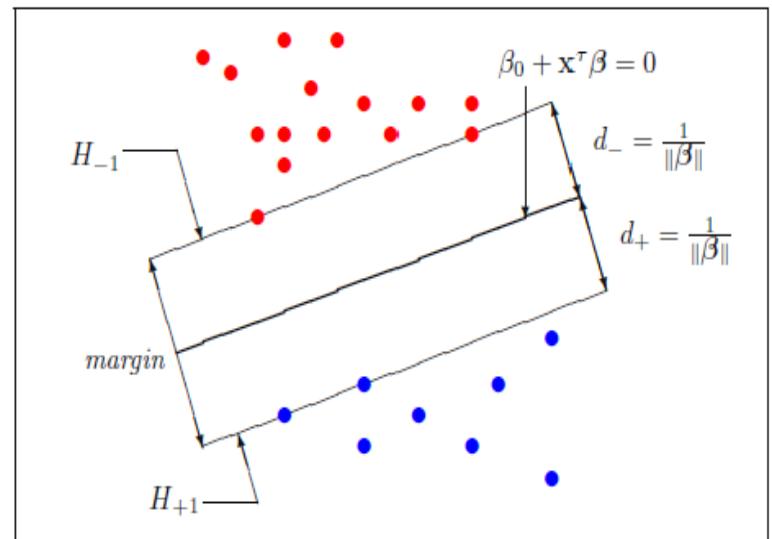


FIGURE 11.1. Support vector machines: the linearly separable case. The red points correspond to data points with $y_i = -1$, and the blue points correspond to data points with $y_i = +1$. The separating hyperplane is the line $\beta_0 + \mathbf{x}^\tau \boldsymbol{\beta} = 0$. The support vectors are those points lying on the hyperplanes H_{-1} and H_{+1} . The margin of the separating hyperplane is $d = 2 / \|\boldsymbol{\beta}\|$.

Linear support vector machines

The linearly separable case

- Combine (2) and (3) into a single set of inequalities

$$y_i(\beta_0 + \mathbf{x}_i^\tau \boldsymbol{\beta}) \geq +1, i = 1, 2, \dots, n.$$

- The quantity $y_i(\beta_0 + \mathbf{x}_i^\tau \boldsymbol{\beta})$ is called the *margin of (\mathbf{x}_i, y_i) with respect to the hyperplane (1)*, $i = 1, \dots, n$ and \mathbf{x}_i is the support vectors wrt to (1) if $y_i(\beta_0 + \mathbf{x}_i^\tau \boldsymbol{\beta}) = 1$.
- Problem: Find the hyperplane that maximizes the margin $\frac{2}{\|\boldsymbol{\beta}\|}$.
- Equivalently, find β_0 and $\boldsymbol{\beta}$ to

$$\text{minimize} \quad \frac{1}{2} \|\boldsymbol{\beta}\|^2$$

$$\text{subject to} \quad y_i(\beta_0 + \mathbf{x}_i^\tau \boldsymbol{\beta}) \geq 1, i = 1, 2, \dots, n \quad (4)$$

- Solve this *primal optimization problem* using Lagrangian multipliers.

Linear support vector machines

The linearly separable case

- Multiply the constraints, $y_i(\beta_0 + \mathbf{x}_i^\tau \boldsymbol{\beta}) - 1 \geq 0$, by positive Lagrangian multipliers and subtract each product from the objective function ...
- Dual optimization problem: Find $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)^\tau \geq 0$ (n-vector of Lagrangian coefficients) to

$$\begin{aligned} & \text{maximize } F_D(\boldsymbol{\alpha}) = \mathbf{1}_n^\tau \boldsymbol{\alpha} - \frac{1}{2} \boldsymbol{\alpha}^\tau \mathbf{H} \boldsymbol{\alpha} \\ & \text{subject to } \boldsymbol{\alpha} \geq 0, \boldsymbol{\alpha}^\tau \mathbf{y} = 0 \end{aligned} \tag{5}$$

where $\mathbf{y} = (y_1, y_2, \dots, y_n)^\tau$, $\mathbf{H} = (H_{ij})$, $H_{ij} = y_i y_j (\mathbf{x}_i^\tau \mathbf{x}_j)$, $i, j = 1, \dots, n$.

- If $\boldsymbol{\alpha}^*$ solves this problem, then $\boldsymbol{\beta}^* = \sum_{i=1}^n \alpha_i^* y_i \mathbf{x}_i \rightarrow \boldsymbol{\beta}^* = \sum_{i \in sv} \alpha_i^* y_i \mathbf{x}_i$

$$\beta_0^* = \frac{1}{|sv|} \sum_{i \in sv} \left(\frac{1 - y_i \mathbf{x}_i^\tau \boldsymbol{\beta}^*}{y_i} \right)$$

- Optimal hyperplane $f^*(\mathbf{x}) = \beta_0^* + \mathbf{x}^\tau \boldsymbol{\beta}^* = \beta_0^* + \sum_{i \in sv} \alpha_i^* y_i (\mathbf{x}^\tau \mathbf{x}_i)$

Linear support vector machines

The linearly nonseparable case

- The *nonseparable case* occurs if either the two classes are separable, but not linearly so, or that no clear separability exists between the two classes, linearly or nonlinearly (caused by, for example, noise).
- Create a more flexible formulation of the problem, which leads to a *soft-margin solution*. We introduce a nonnegative *slack variable*, ξ_i , for each observation (\mathbf{x}_i, y_i) in \mathcal{L} , $i = 1, 2, \dots, n$. Let

$$\boldsymbol{\xi} = (\xi_1, \dots, \xi_n)^\tau \geq \mathbf{0}.$$

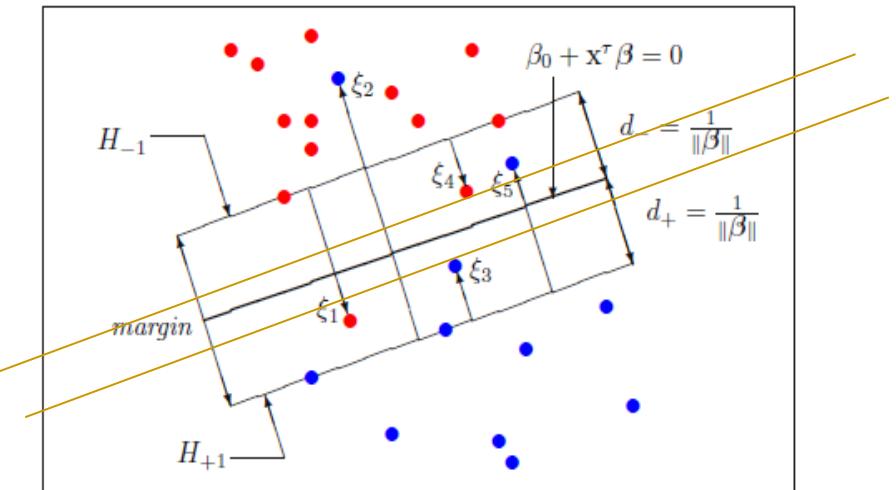


FIGURE 11.2. Support vector machines: the nonlinearly separable case. The red points correspond to data points with $y_i = -1$, and the blue points correspond to data points with $y_i = +1$. The separating hyperplane is the line $\beta_0 + \mathbf{x}^\top \beta = 0$. The support vectors are those circled points lying on the hyperplanes H_{-1} and H_{+1} . The slack variables ξ_1 and ξ_4 are associated with the red points that violate the constraint of hyperplane H_{-1} , and points marked by ξ_2 , ξ_3 , and ξ_5 are associated with the blue points that violate the constraint of hyperplane H_{+1} . Points that satisfy the constraints of the appropriate hyperplane have $\xi_i = 0$.

Linear support vector machines

The linearly nonseparable case

- The constraints in (5) become $y_i(\beta_0 + \mathbf{x}_i^\tau \boldsymbol{\beta}) + \xi_i \geq 1$ for $i = 1, 2, \dots, n$.
- Find the optimal hyperplane that controls both the margin, $\frac{2}{\|\boldsymbol{\beta}\|}$, and some computationally simple function of the slack variables, such as $g_\sigma(\xi) = \sum_{i=1}^n \xi_i^\sigma$. Consider “1-norm” ($\sigma = 1$) and “2-norm” ($\sigma = 2$).
- The 1-norm soft-margin optimization problem is to find β_0 , $\boldsymbol{\beta}$ and ξ to

$$\begin{aligned} & \text{minimize} \quad \frac{1}{2} \|\boldsymbol{\beta}\|^2 + C \sum_{i=1}^n \xi_i \\ & \text{subject to} \quad \xi_i \geq 0, y_i(\beta_0 + \mathbf{x}_i^\tau \boldsymbol{\beta}) \geq 1 - \xi_i, \quad i = 1, 2, \dots, n. \end{aligned} \tag{6}$$

where $C > 0$ is a *regularization parameter*. C takes the form of a tuning constant that controls the size of the slack variables and balances the two terms in the minimizing function.

Linear support vector machines

The linearly nonseparable case

- We can write the dual maximization problem in matrix notation as follows. Find α to

$$\begin{aligned} & \text{maximize } F_D(\alpha) = \mathbf{1}_n^\tau \alpha - \frac{1}{2} \alpha^\tau \mathbf{H} \alpha \\ & \text{subject to } \alpha^\tau \mathbf{y} = 0, \quad \mathbf{0} \leq \alpha \leq C \mathbf{1}_n \end{aligned} \tag{7}$$

- The difference between this optimization problem and (4), is that here the coefficients α_i , $i = 1, \dots, n$, are each bounded above by C ; this upper bound restricts the influence of each observation in determining the solution.
- This constraint is referred to as a *box constraint* because α is constrained by the box of side C in the positive orthant. The feasible region for the solution to this problem is the intersection of hyperplane $\alpha^\tau \mathbf{y} = 0$ with the box constraint $\mathbf{0} \leq \alpha \leq C \mathbf{1}_n$. If $C = \infty \rightarrow$ hard-margin separable case.
- If α^* solves (7) then $\beta^* = \sum_{i \in sv} \alpha_i^* y_i \mathbf{x}_i$ yields the optimal weight vector.

Content

1. Introduction
2. Linear support vector machines
3. Nonlinear support vector machines
4. Multiclass support vector machines
5. Other issues
6. Challenges for kernel methods and SVMs

Nonlinear support vector machines

- What if a linear classifier is not appropriate for the data set?
- Can we extend the idea of linear SVM to the nonlinear case?
- The key to constructing a nonlinear SVM is to observe that the observations in \mathcal{L} only enter the dual optimization problem through the inner products $\langle \mathbf{x}_i, \mathbf{x}_j \rangle = \mathbf{x}_i^\tau \mathbf{x}_j$, $i, j = 1, 2, \dots, n$.

$$F_D(\boldsymbol{\alpha}) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j (\mathbf{x}_i^\tau \mathbf{x}_j)$$

Nonlinear support vector machines

Nonlinear transformations

- Suppose we transform each observation, $x_i \in \Re^r$, in \mathcal{L} , using some nonlinear mapping $\Phi: \Re^r \rightarrow \mathcal{H}$, \mathcal{H} is an $N_{\mathcal{H}}$ -dimensional feature space.
- The nonlinear map Φ is generally called the *feature map* and the space \mathcal{H} is called the *feature space*.
- The space \mathcal{H} may be very high-dimensional, possibly even infinite dimensional. We will generally assume that \mathcal{H} is a *Hilbert space* of real-valued functions on with inner product $\langle \cdot, \cdot \rangle$ and norm $\|\cdot\|$.
- Let $\Phi(x_i) = (\phi_1(x_i), \dots, \phi_{N_{\mathcal{H}}}(x_i))^{\tau} \in \mathcal{H}$, $i = 1..n$. The transformed sample is $\{\Phi(x_i), y_i\}$, where $y_i \in \{-1, +1\}$ identifies the two classes.
- If substitute $\Phi(x_i)$ for x_i in the development of the linear SVM, then data would only enter the optimization problem by way of the inner products $\langle \Phi(x_i), \Phi(x_j) \rangle = \Phi(x_i)^{\tau} \Phi(x_j)$. The difficulty in using nonlinear transform is computing such inner products in high-dimensional space \mathcal{H} .

Nonlinear support vector machines

The “kernel trick”

- The idea behind nonlinear SVM is to find an optimal separating hyperplane in high-dimensional feature space \mathcal{H} just as we did for the linear SVM in input space.
- The “kernel trick” was first applied to SVMs by Cortes & Vapnik (1995).
- **Kernel trick:** Wonderful idea that is widely used in algorithms for computing inner products $\langle \Phi(x_i), \Phi(x_j) \rangle$ in feature space \mathcal{H} .
- **The trick:** instead of computing the inner products in \mathcal{H} , which would be computationally expensive due to its high dimensionality, we compute them using a nonlinear kernel function, $K(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$ in input space, which helps speed up the computations.
- Then, we just compute a *linear* SVM, but where the computations are carried out in some other space.

Nonlinear support vector machines

Kernels and their properties

- A *kernel* K is a function $K : \Re^r \times \Re^r \rightarrow \Re$ such that $\forall x, y \in \Re^r$

$$K(x, y) = \langle \Phi(x), \Phi(y) \rangle$$

- The kernel function is designed to compute inner-products in \mathcal{H} by using only the original input data \rightarrow substitute $\langle \Phi(x), \Phi(y) \rangle$ by $K(x, y)$ wherever. Advantage: *given K, no need to know the explicit form of Φ .*
- K should be symmetric: $K(x, y) = K(y, x)$, and $[K(x, y)]^2 \leq K(x, x)K(y, y)$.
- K is a *reproducing kernel* if $\forall f \in \mathcal{H}: \langle f(\cdot), K(x, \cdot) \rangle = f(x)$ (8),
 K is called the *representer of evaluation*. Particularly, if $f(\cdot) = K(\cdot, x)$ then $\langle K(x, \cdot), K(y, \cdot) \rangle = K(x, y)$.
- Let x_1, \dots, x_n be n points in \Re^r . The $(n \times n)$ -matrix $\mathbf{K} = (K_{ij}) = (K(x_i, x_j))$ is called *Gram (or kernel) matrix* wrt x_1, \dots, x_n .

Nonlinear support vector machines

Kernels and their properties

- If for any n -vector \mathbf{u} , we have $\mathbf{u}^\top \mathbf{K} \mathbf{u} \geq 0$, \mathbf{K} is said to be *nonnegative-definite* with nonnegative eigenvalues and K is nonnegative-definite kernel (or Mercer kernel).
- If K is a Mercer kernel on $\mathbb{R}^r \times \mathbb{R}^r$, we can construct a unique Hilbert space \mathcal{H}_K , say, of real-valued functions for which K is its reproducing kernel. We call \mathcal{H}_K a (real) *reproducing kernel Hilbert space* (RKHS). We write the inner-product and norm of \mathcal{H}_K by $\langle \cdot, \cdot \rangle_{\mathcal{H}_K}$ and $\|\cdot\|_{\mathcal{H}_K}$.
- Ex: inhomogeneous polynomial kernel of degree d (c, d : parameters)

$$K(\mathbf{x}, \mathbf{y}) = (\langle \mathbf{x}, \mathbf{y} \rangle + c)^d, \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^r$$

- If $r = 2, d = 2, \mathbf{x} = (x_1, x_2)^\top, \mathbf{y} = (y_1, y_2)^\top,$

$$K(\mathbf{x}, \mathbf{y}) = (\langle \mathbf{x}, \mathbf{y} \rangle + c)^2 = (x_1 y_1 + x_2 y_2 + c)^2 = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle$$

$$\Phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1 x_2, \sqrt{2c}x_1 x_2, \sqrt{2c}x_1, \sqrt{2c}x_2, c)$$

Nonlinear support vector machines

Examples of kernels

- Here $\mathcal{H} = \mathbb{R}^6$, monomials have degree ≤ 2 . In general, $\dim(\mathcal{H}) = \binom{r+d}{d}$ consisting of monomials with degree $\leq d$.
- For 16x16 pixels, $r = 256$. If $d = 2$, $\dim(\mathcal{H}) = 33,670$; $d = 4$, $\dim(\mathcal{H}) = 186,043,585$.

TABLE 11.1. Kernel functions, $K(\mathbf{x}, \mathbf{y})$, where $\sigma > 0$ is a scale parameter, $a, b, c \geq 0$, and d is an integer. The Euclidean norm is $\|\mathbf{x}\|^2 = \mathbf{x}^\top \mathbf{x}$.

| Kernel | $K(\mathbf{x}, \mathbf{y})$ |
|--------------------------------|--|
| Polynomial of degree d | $(\langle \mathbf{x}, \mathbf{y} \rangle + c)^d$ |
| Gaussian radial basis function | $\exp \left\{ -\frac{\ \mathbf{x}-\mathbf{y}\ ^2}{2\sigma^2} \right\}$ |
| Laplacian | $\exp \left\{ -\frac{\ \mathbf{x}-\mathbf{y}\ }{\sigma} \right\}$ |
| Thin-plate spline | $\left(\frac{\ \mathbf{x}-\mathbf{y}\ }{\sigma} \right)^2 \log_e \left\{ \frac{\ \mathbf{x}-\mathbf{y}\ }{\sigma} \right\}$ |
| Sigmoid | $\tanh(a\langle \mathbf{x}, \mathbf{y} \rangle + b)$ |

Examples of translation-invariant (stationary) kernels having the general form

$$K(\mathbf{x}, \mathbf{y}) = k(\mathbf{x} - \mathbf{y}), k: \mathbb{R}^r \rightarrow \mathbb{R}$$

sigmoid kernel is not strictly a kernel but very popular in certain situations

If no information, the best approach is to try either a Gaussian RBF, which has only a single parameter (σ) to be determined, or a polynomial kernel of low degree ($d = 1$ or 2).

Nonlinear support vector machines

Example: String kernels for text (Lodhi et al., 2002)

- A “string” $s = s_1 s_2 \dots s_{|s|}$ is a finite sequence of elements of a finite alphabet \mathcal{A} .
- We call u a *subsequence* of s (written $u = s(\mathbf{i})$) if there are indices $\mathbf{i} = (i_1, i_2, \dots, i_{|u|})$, $1 \leq i_1 < \dots < i_{|u|} \leq |s|$, such that $u_j = s_{i_j}, j = 1, 2, \dots, |u|$.
- If the indices \mathbf{i} are contiguous, we say that u is a *substring* of s . The length of u in s is $l(i) = i_{|u|} - i_1 + 1$.
- Let $s = \text{“cat”}$ ($s_1 = c, s_2 = a, s_3 = t, |s| = 3$). Consider all possible 2-symbol sequences, “ca,” “ct,” and “at,” derived from s .
 - $u = \text{ca}$ has $u_1 = c = s_1, u_2 = a = s_2, u = s(\mathbf{i}), \mathbf{i} = (i_1, i_2) = (1, 2), l(\mathbf{i}) = 2$.
 - $u = \text{ct}$ has $u_1 = c = s_1, u_2 = t = s_3, \mathbf{i} = (i_1, i_2) = (1, 3)$, and $l(\mathbf{i}) = 3$.
 - $u = \text{at}$ has $u_1 = a = s_2, u_2 = t = s_3, \mathbf{i} = (2, 3)$, and $l(\mathbf{i}) = 2$.

Nonlinear support vector machines

Examples: String kernels for text

- If $D = \mathcal{A}^m = \{\text{all strings of length at most } m \text{ from } A\}$, then, the feature space for a string kernel is \Re^D .
- Using $\lambda \in (0, 1)$ (*drop-off rate* or *decay factor*) to weight the interior gaps in the subsequences, we define the feature map $\Phi_u: \Re^D \rightarrow \Re$

$$\Phi_u(s) = \sum_{\mathbf{i}: u=s(\mathbf{i})} \lambda^{l(\mathbf{i})}, u \in \mathcal{A}^m$$

$\Phi_u(s)$ is computed as follows: identify all subsequences (indexed by \mathbf{i}) of s that are identical to u ; for each such subsequence, raise λ to the power (\mathbf{i}) ; and then sum the results over all subsequences.

- In our example above, $\Phi_{ca}(cat) = \lambda^2$, $\Phi_{ct}(cat) = \lambda^3$, and $\Phi_{at}(cat) = \lambda^2$.
- Two documents are considered to be “similar” if they have many subsequences in common: the more subsequences they have in common, the more similar they are deemed to be.

Nonlinear support vector machines

Examples: String kernels for text

- The kernel associated with the feature maps corresponding to s and t is the sum of inner products for all *common* substrings of length m

$$K_m(s, t) = \sum_{u \in \mathcal{D}} \langle \Phi_u(s), \Phi_u(t) \rangle = \sum_{u \in \mathcal{D}} \sum_{\mathbf{i}: u=s(\mathbf{i})} \sum_{\mathbf{j}: u=s(\mathbf{j})} \lambda^{l(i)+l(j)}$$

and it is called a *string kernel* (or a *gap-weighted subsequences kernel*).

- Let $t = \text{"car"}$ ($t_1 = c$, $t_2 = a$, $t_3 = r$, $|t| = 3$). The strings “cat” and “car” are both substrings of the string “cart.” The three 2-symbol substrings of t are “ca,” “cr,” and “ar.” We have that $\Phi_{ca}(\text{car}) = \lambda^2$, $\Phi_{cr}(\text{car}) = \lambda^3$, $\Phi_{ar}(\text{car}) = \lambda^2$, and thus $K_2(\text{cat}, \text{car}) = \Phi_{ca}(\text{cat}), \Phi_{ca}(\text{car}) = \lambda^4$.
- We normalize the kernel by removing any bias by document length

$$K_m^*(s, t) = \frac{K_m(s, t)}{\sqrt{K_m(s, s)K_m(t, t)}}$$

Nonlinear support vector machines

Optimizing in feature space

- Let K be a kernel. Suppose observations in \mathcal{L} are *linearly separable* in the feature space corr. to K . The dual opt. problem is to find α and β_0 to

$$\begin{aligned} & \text{maximize } F_D(\alpha) = \mathbf{1}_n^\tau \alpha - \frac{1}{2} \alpha^\tau \mathbf{H} \alpha \\ & \text{subject to } \alpha \geq 0, \alpha^\tau \mathbf{y} = 0 \end{aligned} \tag{9}$$

where $\mathbf{y} = (y_1, y_1, \dots, y_1)^\tau$, $\mathbf{H} = (H_{ij}) = y_i y_j K(x_i, x_j) = y_i y_j K_{ij}$.

- Because K is a kernel, the $\mathbf{K} = (K_{ij})$ and so \mathbf{H} are nonnegative-definite \rightarrow the functional $F_D(\alpha)$ is convex \rightarrow unique solution. If α and β_0 solve this problem, the SVM decision rule is ($f^*(x)$ is optimal in feature space)

$$\text{sign}\{f^*(x)\} = \text{sign}\{\beta_0^* + \sum_{i \in sv} \alpha_i^* y_i K(x, x_i)\}$$

- In the *nonseparable* case, the dual problem of the 1-norm soft-margin optimization problem is to find α to

$$\begin{aligned} & \text{maximize } F_D(\alpha) = \mathbf{1}_n^\tau \alpha - \frac{1}{2} \alpha^\tau \mathbf{H} \alpha \\ & \text{subject to } \alpha^\tau \mathbf{y} = 0, \mathbf{0} \leq \alpha \leq C \mathbf{1}_n \end{aligned}$$

Nonlinear support vector Machines

Example: E-mail or spam?

- 4,601 messages: 1,813 spam e-mails and 2,788 non-spam e-mails. There are 57 variables (attributes).
- Apply nonlinear SVM (R package `libsvm`) using a Gaussian RBF kernel to the 4,601 messages. The solution depends on the cost C of violating the constraints and σ^2 of the Gaussian RBF kernel. After applying a trial-and-error method, we used the following grid of values for C and $\gamma = 1/\sigma^2$:
 - $C = 10, 80, 100, 200, 500, 1,000,$
 - $\gamma = 0.00001(0.00001)0.0001(0.0001)0.002(0.001)0.01(0.01)0.04.$
- Plot the 10-fold CV misclassification rate against γ listed above, where each curve (connected set of points) represents a different value of C .
- For each C , we see that the CV/10 misclassification curves have similar shapes: a minimum value for γ very close to zero, and for values of γ away from zero, the curve trends upwards.

Nonlinear support vector Machines

Example: E-mail or spam?

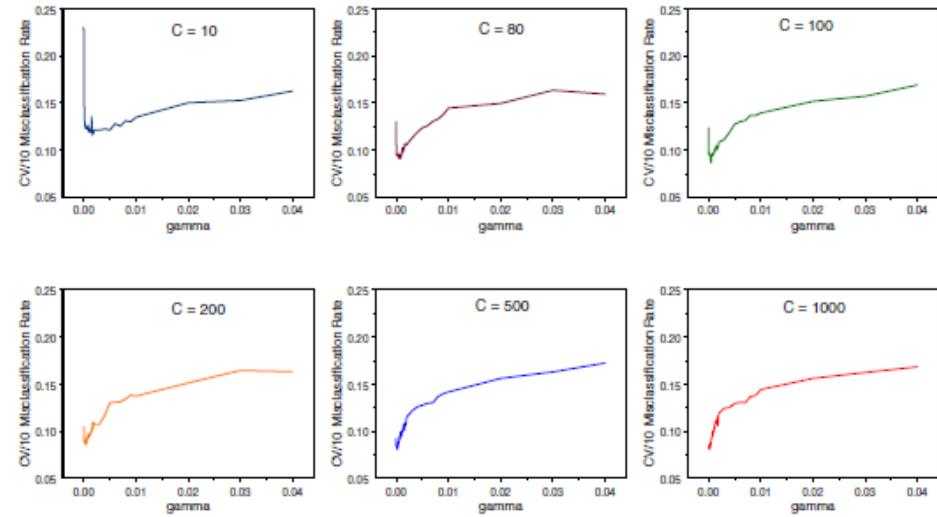
- We find a minimum CV/10 misclassification rate of **8.06%** at $(C, \gamma) = (500, 0.0002)$ and $(1,000, 0.0002)$. The level of the misclassification rate tends to decrease as C increases and γ decreases together.

- misclassification rate of **6.91%** at $C = 11,000$ and $\gamma = 0.00001$, at corresponding to classification rate:

- 0.9043, 0.9478, 0.9304, 0.9261, 0.9109,
 - 0.9413, 0.9326, 0.9500. 0.9326, 0.9328.

is better than LDA and QDA.

- 931 support vectors (482 e-mails, 449 spam).



Initial grid search for the minimum 10-fold CV misclassification rate using $0.00001 \leq \gamma \leq 0.04$. The curves correspond to $C = 10$ (dark blue), 80 (brown), 100 (green), 200 (orange), 500 (light blue), and $1,000$ (red). Within this initial grid search, the minimum CV/10 misclassification rate is 8.06%, which occurs at $(C, \gamma) = (500, 0.0002)$ and $(1,000, 0.0002)$.

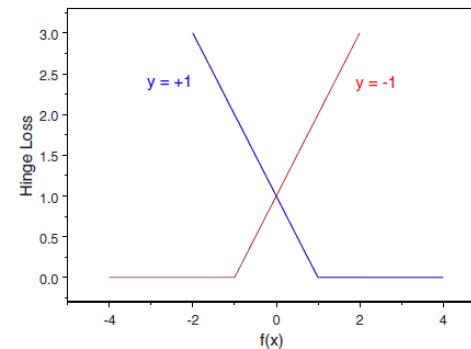
Nonlinear Support Vector Machines

SVM as a Regularization Method

- **Regularization** involves introducing additional information in order to solve an ill-posed problem or to prevent overfitting. This information is usually of the form of a penalty for complexity.
- Let $f \in \mathcal{H}_K$, the reproducing kernel Hilbert space associated with the kernel K , with $\|f\|_{\mathcal{H}_K}^2$ the squared-norm of f in \mathcal{H}_K .
- Consider the classification error, $y_i - f(\mathbf{x}_i)$, where $y_i \in \{-1, +1\}$. Then

$$|y_i - f(\mathbf{x}_i)| = |y_i(1 - y_i f(\mathbf{x}_i))| = |1 - y_i f(\mathbf{x}_i)| = (1 - y_i f(\mathbf{x}_i))_+$$

$i = 1 \dots n$, $(x)_+ = \max(x, 0)$. The quantity $(1 - y_i f(\mathbf{x}_i))_+$, which could be zero if all \mathbf{x}_i are correctly classified, called ***hinge loss function***. The hinge loss plays a vital role in SVM methodology (related to the misclassification function).



Nonlinear Support Vector Machines

SVM as a Regularization Method

- Want to find $f \in \mathcal{H}_K$ to minimize a penalized version of the hinge loss. Specifically, we wish to find $f \in \mathcal{H}_K$ to

$$\text{minimize} \quad \frac{1}{2} \sum_{i=1}^n (1 - y_i f(\mathbf{x}_i))_+ + \lambda \|f\|_{\mathcal{H}_K}^2 \quad (10)$$

- The tuning parameter $\lambda > 0$ balances the trade-off between estimating f (first term: measures the distance of the data from separability) and how well f can be approximated (second term: penalizes overfitting).
- After the minimizing f has been found, the SVM classifier is $C(\mathbf{x}) = \text{sign}\{f(\mathbf{x})\}$, $\mathbf{x} \in \mathbb{R}^r$.
- (10) is nondifferentiable, but every $f \in \mathcal{H}$ can be written as sum

$$f(\cdot) = f^{\parallel}(\cdot) + f^{\perp}(\cdot) = \sum_{i=1}^n \alpha_i K(\mathbf{x}_i, \cdot) + f^{\perp}(\cdot)$$

where $f^{\parallel} \in \mathcal{H}_K$ is the projection of f onto the subspace \mathcal{H}_K of \mathcal{H} and f^{\perp} is in the subspace perpendicular to \mathcal{H}_K ; that is, $\langle f^{\perp}(\cdot), K(\mathbf{x}_i, \cdot) \rangle_{\mathcal{H}} = 0$.

Nonlinear support vector machines

SVM as a regularization method

- We write $f(\mathbf{x}_i)$ via the reproducing property

$$f(\mathbf{x}_i) = \langle f(\cdot), K(\mathbf{x}_i, \cdot) \rangle = \langle f^{\parallel}(\cdot), K(\mathbf{x}_i, \cdot) \rangle + \langle f^{\perp}(\cdot), K(\mathbf{x}_i, \cdot) \rangle$$

- We have $f(\mathbf{x}) = \sum_{i=1}^n \alpha_i K(\mathbf{x}_i, \mathbf{x})$ (11)

is independent of f^{\perp} as the second term is zero in rhs. We have

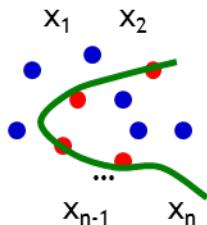
$$\|f\|_{\mathcal{H}_K}^2 \geq \|\sum_i \alpha_i K(\mathbf{x}_i, \cdot)\|_{\mathcal{H}_K}^2 \quad (12)$$

(equality iff $f^{\perp}=0$). This important result is known as the *representer*, says that *the minimizing f can be written as a linear combination of a reproducing kernel evaluated at each of the n data points* (Kimeldorf and Wahba, 1971). Problem (10) is equivalent to find β_0 and $\boldsymbol{\beta}$ to

$$\text{minimize } \frac{1}{n} \sum_{i=1}^n (1 - y_i (\beta_0 + \boldsymbol{\Phi}(\mathbf{x}_i)^T \boldsymbol{\beta})_+) + \lambda \|\boldsymbol{\beta}\|^2 \quad (13)$$

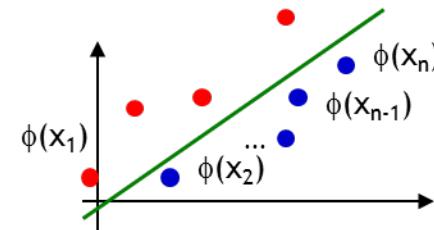
Kernel methods: math background

Input space X



inverse map ϕ^{-1}
 $\phi(x)$

Feature space F



kernel function $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$

$$k(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$$

Kernel matrix $K_{n \times n}$

kernel-based algorithm on K
(computation on kernel matrix)

Linear algebra, probability/statistics, functional analysis, optimization

- Mercer theorem: Any positive definite function can be written as an inner product in some feature space.
- Kernel trick: Using kernel matrix instead of inner product in the feature space.
- Representer theorem (Wahba): Every minimizer of $\min_{f \in \mathcal{H}} \{C(f, \{x_i, y_i\}) + \Omega(\|f\|_H)\}$ admits

a representation of the form $f(\cdot) = \sum_{i=1}^m \alpha_i K(\cdot, x_i)$

Content

1. Introduction
2. Linear support vector machines
3. Nonlinear support vector machines
4. Multiclass support vector machines
5. Other issues
6. Challenges for kernel methods and SVMs

Multiclass support vector machines

Multiclass SVM as a series of binary problems

■ **One-versus-rest:**

Divide the K -class problem into K binary classification subproblems of the type “ k th class” vs. “not k th class,” $k = 1, 2, \dots, K$.

■ **One-versus-one:**

Divide the K -class problem into comparisons of all pairs of classes.

TABLE 11.3. Summary of support vector machine (SVM) “one-versus-one” classification results for data sets with more than two classes. Listed are the sample size (n), number of variables (r), and number of classes (K). Also listed for each data set is the 10-fold cross-validation (CV/10) misclassification rates corresponding to the best choice of (C, γ) . The data sets are listed in increasing order of LDA misclassification rates (Table 8.7).

| Data Set | n | r | K | SVM-CV/10 | C | γ |
|--------------------|--------|-----|-----|-----------|--------|--------------------|
| Wine | 178 | 13 | 3 | 0.0169 | 10^6 | 8×10^{-8} |
| Iris | 150 | 4 | 3 | 0.0200 | 100 | 0.002 |
| Primate scapulae | 105 | 7 | 5 | 0.0286 | 100 | 0.0002 |
| Shuttle | 43,500 | 8 | 7 | 0.0019 | 10 | 0.0001 |
| Diabetes | 145 | 5 | 3 | 0.0414 | 100 | 0.000009 |
| Pendigits | 10,992 | 16 | 10 | 0.0031 | 10 | 0.0001 |
| E-coli | 336 | 7 | 8 | 0.1280 | 10 | 1.0 |
| Vehicle | 846 | 18 | 4 | 0.1501 | 600 | 0.00005 |
| Letter recognition | 20,000 | 16 | 26 | 0.0183 | 50 | 0.04 |
| Glass | 214 | 9 | 6 | 0.0093 | 10 | 0.001 |
| Yeast | 1,484 | 8 | 10 | 0.3935 | 10 | 7.0 |

Multiclass support vector machines

A true multiclass SVM

- To construct a true multiclass SVM classifier, we need to consider all K classes, $\Pi_1, \Pi_2, \dots, \Pi_K$, simultaneously, and the classifier has to reduce to the binary SVM classifier if $K = 2$.
- One construction due to Lee, Lin, and Wahba (2004).
- Provide a unifying framework to multiclass SVM when there are either equal or unequal misclassification costs.

Content

1. Introduction
2. Linear support vector machines
3. Nonlinear support vector machines
4. Multiclass support vector machines
5. Other issues
6. Challenges for kernel methods and SVMs

Other issues

Support vector regression

- The SVM was designed for classification. Can we extend (or generalize) the idea to regression?
- How would the main concepts used in SVM — convex optimization, optimal separating hyperplane, support vectors, margin, sparseness of the solution, slack variables, and the use of kernels — translate to the regression situation?
- It turns out that all of these concepts find their analogues in regression analysis and they add a different view to the topic than the views we saw previously.
- *ε -insensitive loss functions*
- *Optimization for linear -insensitive loss*

Other issues

Optimization algorithms for SVMs

- Main problem when computing SVMs for very large datasets is that storing the entire kernel in main memory dramatically slowdown computation. With large data sets, however, a more sophisticated approach is required.
- *Gradient ascent*: Start with an estimate of the α -coefficient then successively update α one α -coefficient by steepest ascent algorithm.
- *Chunking*: Start with a small subset; train an SVM on it, keep only support vectors; apply the resulting classifier to the remaining data.
- *Decomposition*: Similar to chunking, except that at each iteration, the size of the subset is always the same.
- *Sequential minimal optimization (SMO)*: An extreme version of the decomposition algorithm.

Content

1. Introduction
2. Linear support vector machines
3. Nonlinear support vector machines
4. Multiclass support vector machines
5. Other issues
6. Challenges for kernel methods and SVMs

Some challenges in kernel methods

Scalability and choice of kernels etc.

- The **choice of kernel function**. In general, there is no way of choosing or constructing a kernel that is optimal for a given problem.
- The **complexity of kernel algorithms**. Kernel methods access the feature space via the input samples and need to store all the relevant input samples.
Examples: Store all support vectors or size of the kernel matrices grows quadratically with sample size → scalability of kernel methods.
- Incorporating **priors knowledge** and invariances in to kernel functions are some of the challenges in kernel methods.
- **L1 regularization** may allow some coefficients to be zero → hot topic
- **Multiple kernel learning** (MKL) is initially (2004, Lanckriet) of high computational cost → Many subsequent work, still ongoing, has not been a practical tool yet.

予測的データ解析手法（2）： 決定木

担当：磯貝 孝

イントロダクション

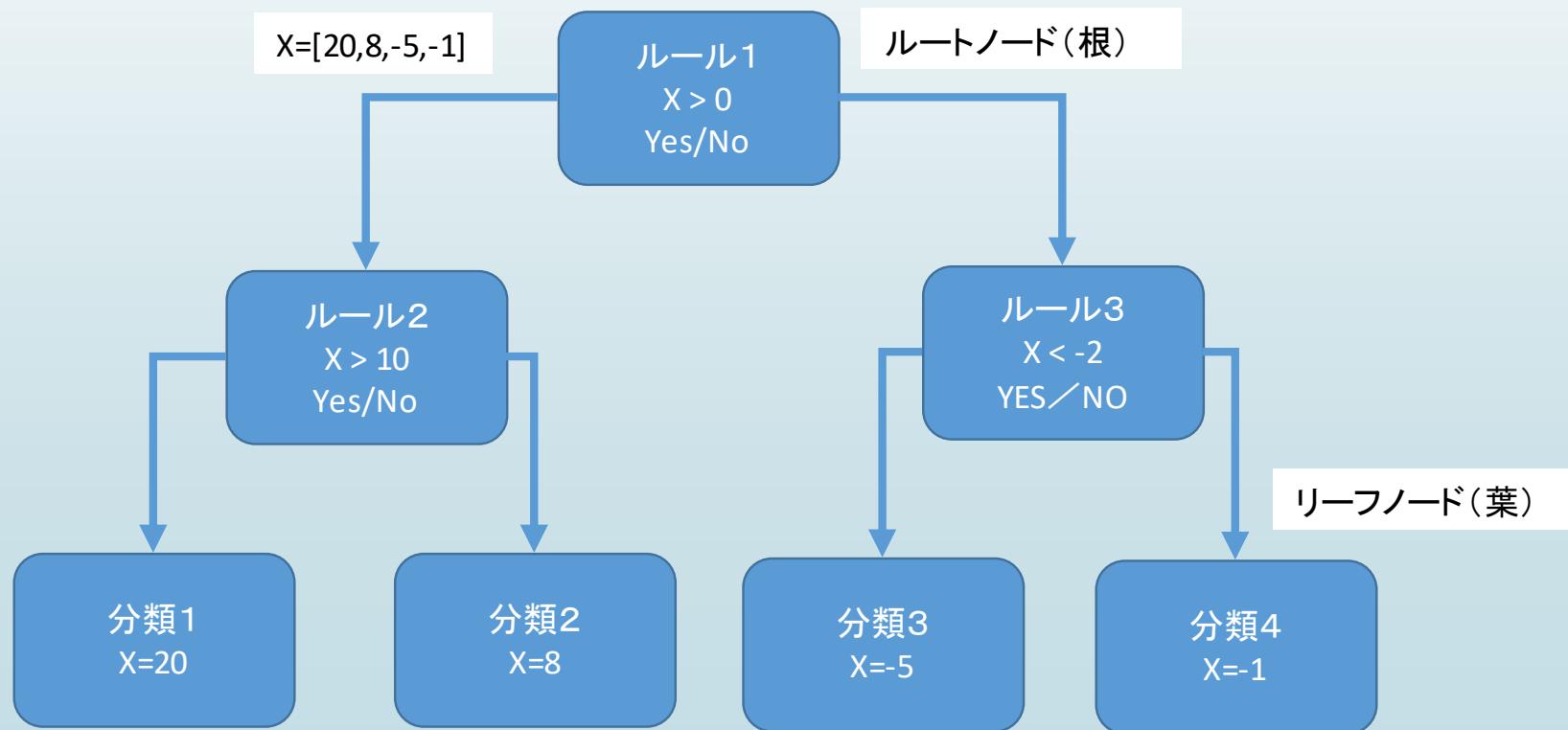
- 決定木：分類問題（個体のグループ分け、複数選択肢の中からの意思決定など）に関する伝統的な解法の一つ。
- 個体に関する特徴データ（例：人間一身長、体重、年齢）があれば、分析者が特別なモデルを構築せずとも、個体の分類を行うことができる。
- 分類に関するデータの蓄積が必要であり、教師あり学習の一つ。
 - モデルのパラメータの推定などの必要はなし（ノンパラメトリックな手法）
- RやPythonなど、主な統計処理ソフトではもれなく実装されているので、分析の現場で応用しやすい。

決定木、分類木. . .

- 使われる場面によって、異なる呼び名で呼ばれている
 - 分類木 (classification tree) . . . 分類問題の解（分類ルール）が欲しい時に用いる。回帰モデルを前提にする場合（対象の数値を推定）は、回帰木 (regression tree) とも呼ばれる。
 - 決定木 (decision tree) . . . 意思決定を行う上で、関係する複数の状態に関して場合分けを行い、ベストな選択肢を提示する（例：食品の生産高を決める上で、気温、天気に関して場合分けを行って、生産量をコントロールする）
 - 分類木と回帰木をまとめて決定木と呼ぶ、という整理もある
- 呼び方は異なっても（呼び方自体にも使う人でばらつきあり）、基本的な枠組みはすべて同じ。
 - ソフトウェアを使用する場合も、同じように分析できる
- 以下では、便宜上、すべて決定木と呼ぶ。

決定木のかたち

- 視覚的には、「木」のかたちをした「有向グラフ」として表現できる
 - 木の先端（「根」）から下に降りて次々と枝分かれ（2分岐）していく <分岐構造>
 - 向きは上から下と決まっている（向きのあるグラフ = 有向グラフ <ネットワーク>）
 - 最後は「葉」（分類結果）

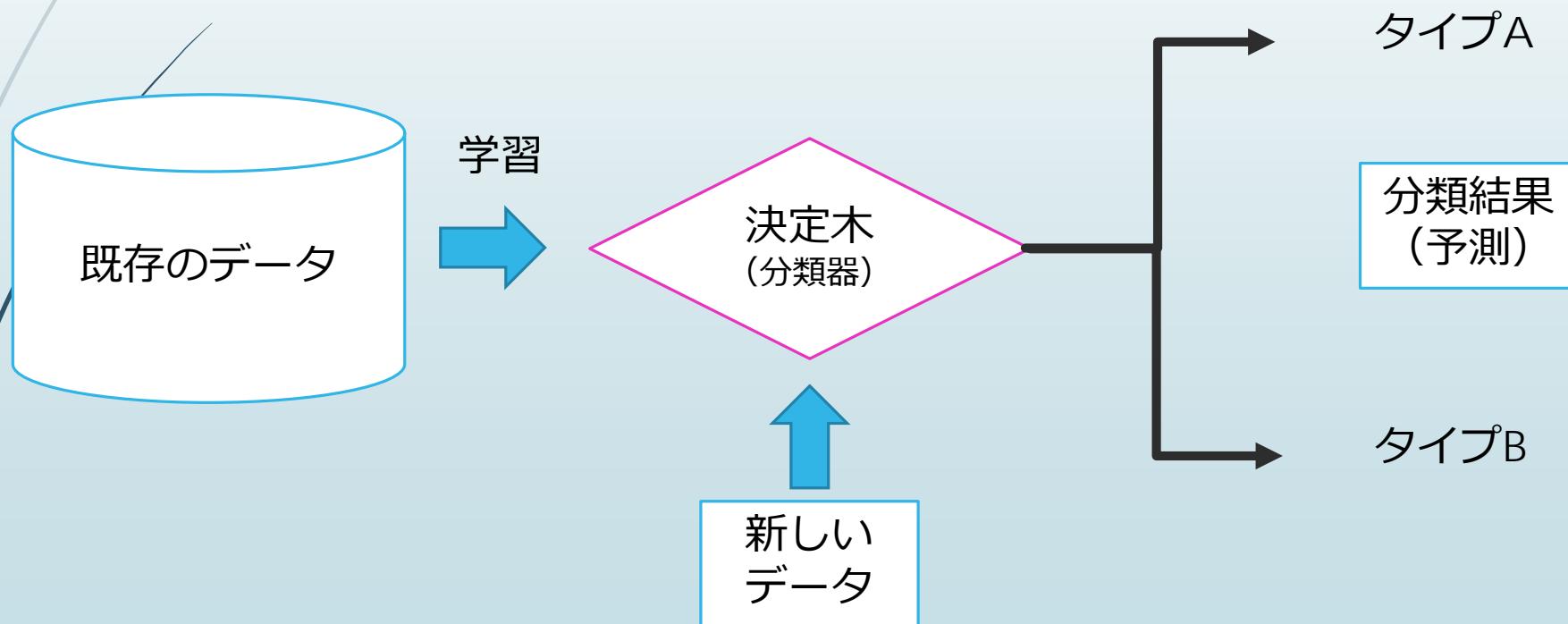


決定木の基本的な考え方

- 複雑な問題（分類、予測）の答えを得るのに、「単純な規則に基づいて選択肢を絞り込んでいく」というアプローチをとる
- 問題の複雑性を単純な選択肢の集まりに圧縮する
- データから学ぶ
 - 「銀行の融資先の信用査定」などの複雑かつ経験値が必要な問題でも、専門的なノウハウを持つ人の判定結果や過去のデフォルト事例などのデータから、単純な選択肢の連結として「知識」を構成できる（機械学習）
- 選択肢を「木」構造として表現する点が最重要ポイント
 - どのようにして「木」を成長させるか、「木」の大きさやかたちをどうコントロールするか
- 最終的な分類ルールは、人間にとてわかりやすい（ブラックボックス的ではない）
 - 同じ機械学習の手法である深層学習などの手法と大きく異なる特徴

機械学習としての決定木

- データから決定木を学習させる、すなわち決定木を作成する
- 得られた決定木を使って、新しいデータに関する予測（意思決定）を行う



対象、分類ルールはどんなものか

- 対象となるデータは、集合の分類、意思決定の場合分け、回帰分析など、明確に分離・分解できるものなら何でもよい
 - 決定木を何につかうのか、予めはっきりさせておく
- データの特徴に関して、場合分け（分類ルール）を順次記述していくので、データを適切に表す特徴を集めておく
- 分類ルール（条件判定）に特に制限はない：質的なもの、量的なもの、論理的なもの、何でもよい
 - ルールは、アルゴリズムにより自動的に生成される
 - アルゴリズムの選択や設定条件により同じ問題について生成されるルールは異なることがある
 - 特徴の組み合わせによって、計算が複雑になってしまうこともある（特徴量に数値に関する条件をたくさん含む場合など ⇒ 計算時間が長くなる、過学習が起こりやすい）

決定木の特徴（ノンパラメトリック）

- ノンパラメトリックな機械学習という時の、ノンパラメトリック
 - パラメーターを含まない、分類ルールの記述の集合体
- パラメトリックなモデル：確率変数を含むモデルを構築して、データからモデルのパラメータ (a,b) を推定し、予測などに役立てる
 - $Y = a^*X + b$
 - 決定木では、こういう形での「パラメトリック」なモデルを構築することはしない
- 対象となるデータに関して、確率分布を想定したモデルの枠組みを明示的に考えることはしない（非確率的モデル）
- 重要なのは、「分類に関するルール」を記述し、新たなデータに対してこのルールを適用することで、分類に関する答えが得られるようにすること

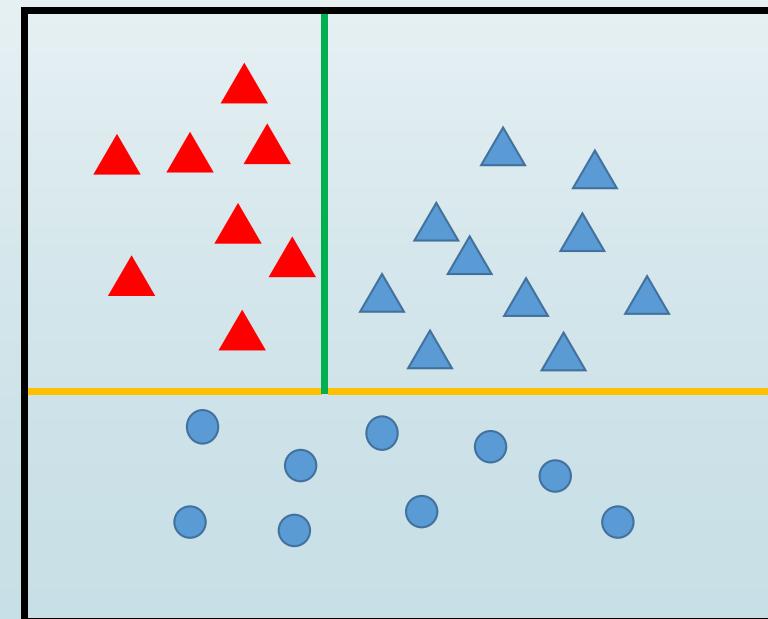
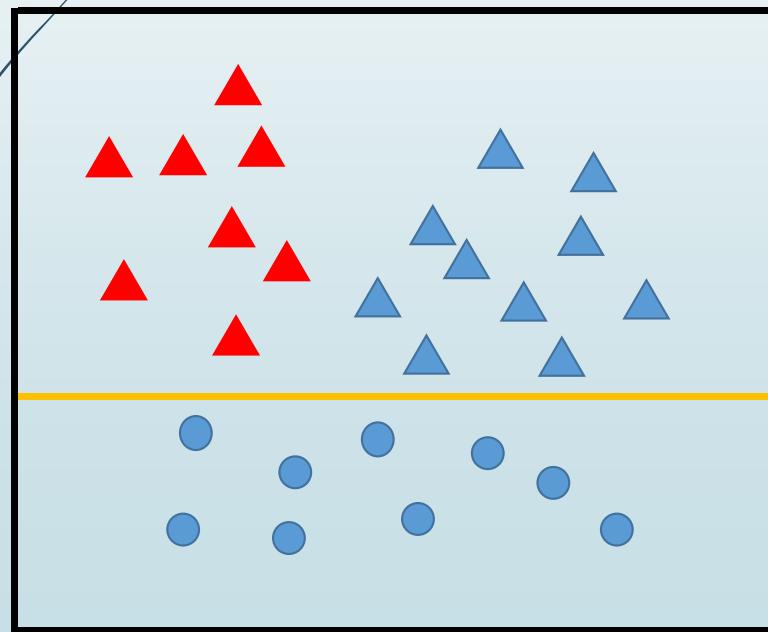
決定木の生成（分類ルールの決定）

- データ = 3つのカテゴリー（青丸、青三角、赤三角）
- ルールの決定（前提：一度に一つの属性に関する条件判定しか使えない）
 - ルール1：属性1に注目（属性1に関する判定）
 - ルール2：属性2に注目（属性2に関する判定）

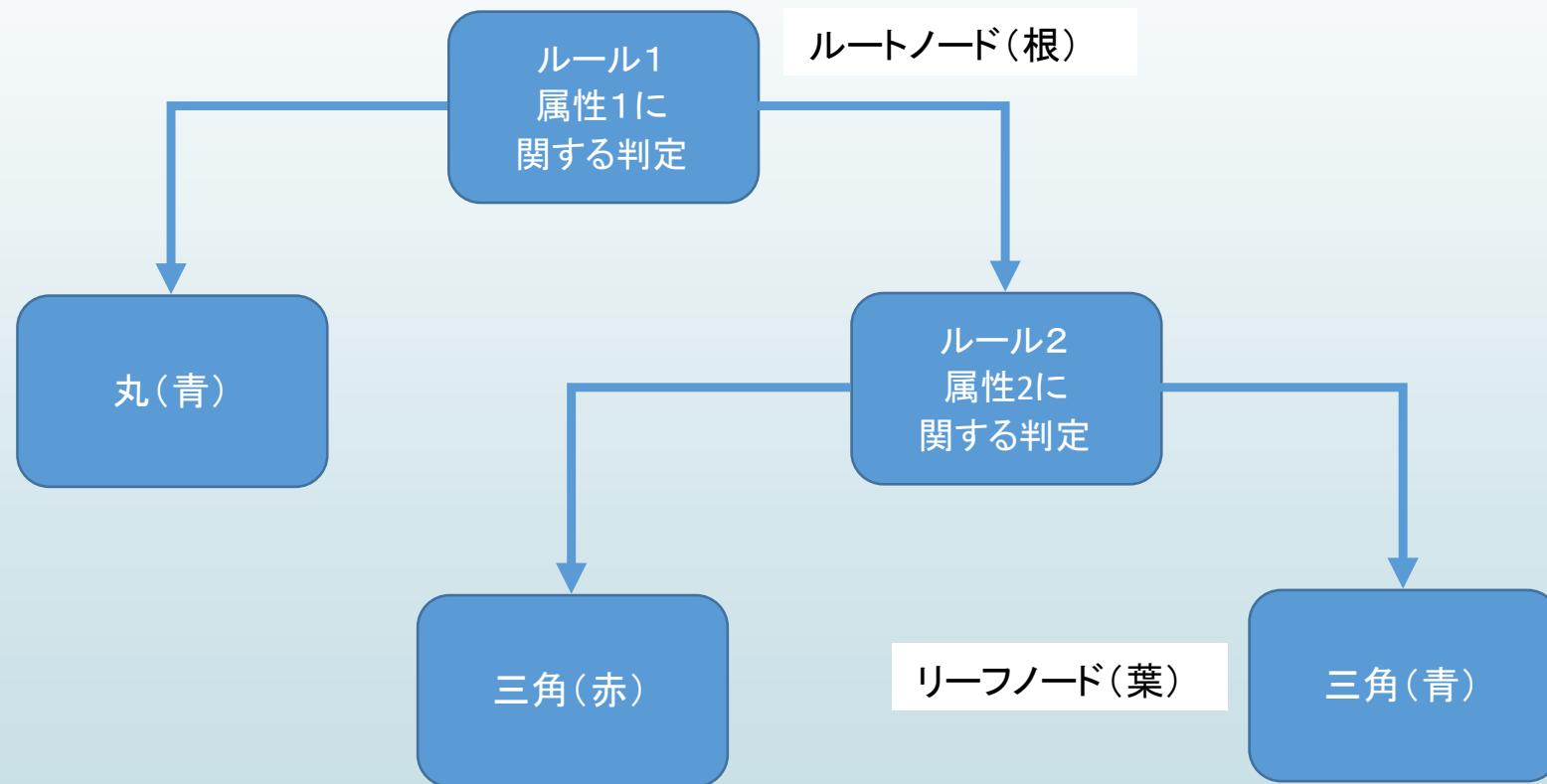
＜うまく分離できた＞

属性1

属性2

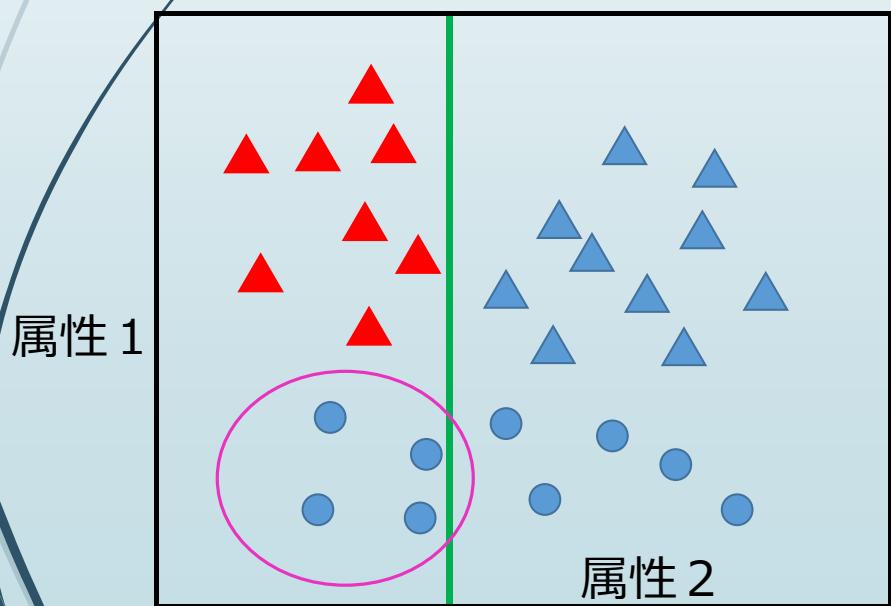


決定木として表現する



分類ルールの決め方（組み合わせ）

- ルール1を生成する際に、属性2に注目して下図のように区分することもできた
- 先ほどの例とは違って、ルール1ではデータ集合を完全に場合分けできなくなる
 - 青丸の集合が二つに分かれてしまった（本来は一つ）
 - 赤三角と青三角は、属性2ベースのルール1で正しく分離できている



ルール1を決める時に、属性1と属性2のどちらを選ぶかが重要
⇒ どういう基準で属性を選べばいい?
⇒ 「もっともきれいに分割できる」基準?

決定木の生成アルゴリズム

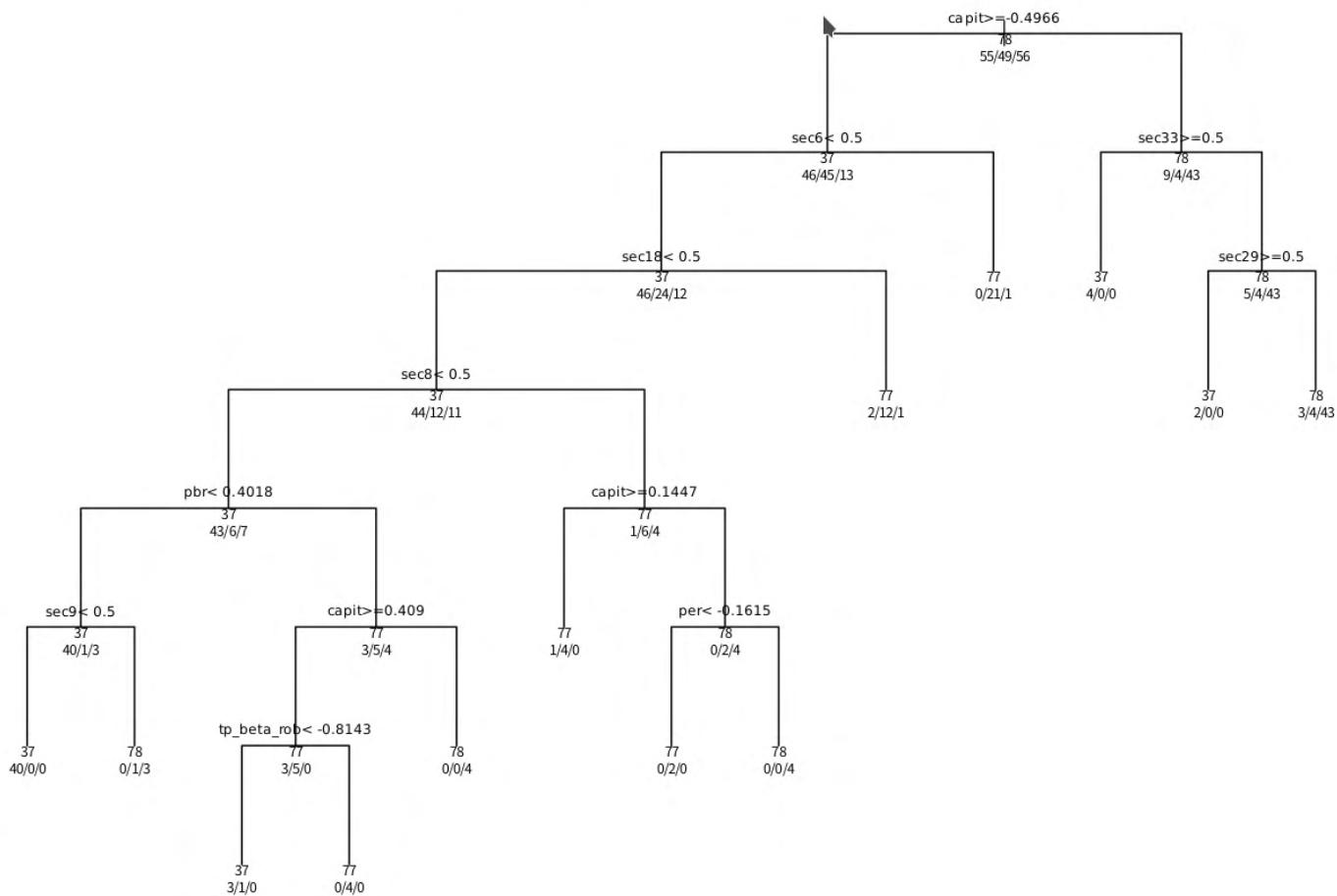
- CART、C5.0など、決定木の実装に関するアルゴリズムが複数存在する
 - RやPythonで使用する各種ライブラリは、これらのアルゴリズムを実装したもの
 - Rの例：パッケージ「rpart」（CARTアルゴリズムの実装）
- 決定木生成のアルゴリズム・・・分岐ルールをデータからどう構成するか
- CART (Classification and Regression Tree) ・・・代表的なアルゴリズム
 - 目的変数を「最もよく」分類する基準として、ジニ・インデックス（Gini impurity）を用いて、どの特徴（変数）に着目して分割するかを決める
 - 目的変数、説明変数・・・カテゴリ変数でも連続変数でもOK
 - 木の複雑さをCP(complex parameter)でコントロールする（過学習対策）

Gini impurity indexによるカテゴリー分割

- $Gini = 1 - \sum_{i=1}^C (p_i)^2$: p_i は、分割Cにおけるクラスiの比率（確率）
- このGiniをすべての変数について計算して、最小の値をとる変数で分割する
 - Giniはすべてが一つのグループに分割される場合に、最小値0となる
- 分割が終わったら、枝分かれした先で同じ分割ルールの探索を行う
- 分割をどこまで進めるのか？
 - 分割をたくさん行って、複雑な木（大きな木）を作れば、現在のデータをよりよく説明できる分類器ができる
 - そうした現在のデータにあてはりのよい（過ぎる）分類器は、新しいデータに対してうまく適合できないことが多い（予測は外れる）・・・過学習の問題
- CP（複雑さの指標）によって、複雑さを上げても（枝の数を増やしても）説明力がそれほど上がらなくなつたところで、木の成長を止める（分割停止）

R: rpartの実行例

Classification Tree BID: 15 GID: 37 77 78



```

do rpart
Call:
rpart(formula = frmla, data = dat, method = "class", control = cntr,
      xval = cpxval, cp = 0)
n= 585

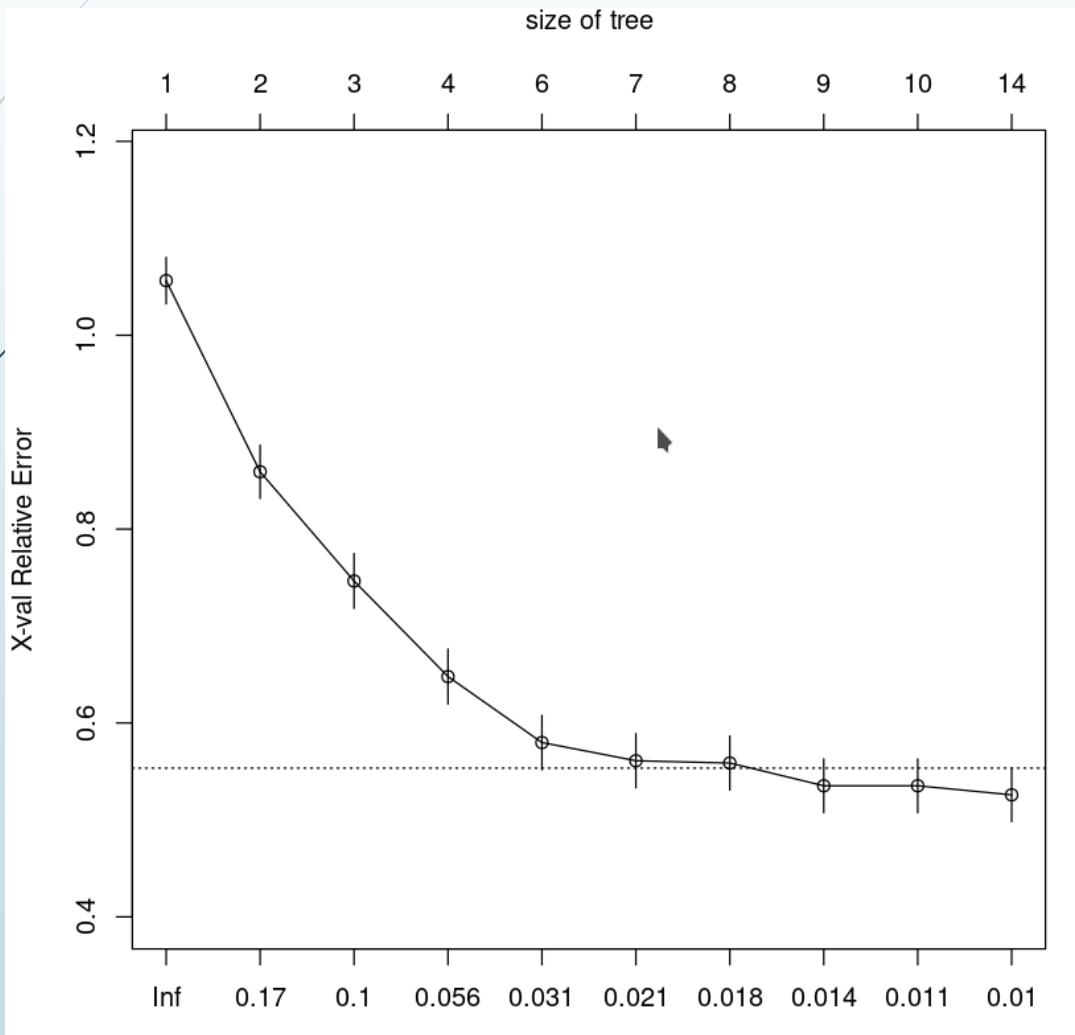
          CP nsplit rel error     xerror        xstd
1 0.19483568      0 1.0000000 1.0563380 0.02392135
2 0.14553991      1 0.8051643 0.8591549 0.02747736
3 0.07511737      2 0.6596244 0.7464789 0.02828015
4 0.04107981      3 0.5845070 0.6478873 0.02834303
5 0.02347418      5 0.5023474 0.5798122 0.02804267
6 0.01877934      6 0.4788732 0.5610329 0.02790933
7 0.01643192      7 0.4600939 0.5586854 0.02789110
8 0.01173709      8 0.4436620 0.5352113 0.02768943

Variable importance
          capit      pbr      mk1 tp_beta_rob      per      mk3
            21         19       11        11        9        7
overseas      mk2      mk4      us_beta    sec17      sec6
            6         4        4         3        1        1

Node number 1: 585 observations, complexity param=0.1948357
predicted class=23 expected loss=0.7282051 P(node) =1
  class counts: 159 134 158 134
  probabilities: 0.272 0.229 0.270 0.229
left son=2 (186 obs) right son=3 (399 obs)
Primary splits:
  capit < 0.3032493 to the right, improve=47.59008, (0 missing)
  pbr   < -0.3959386 to the right, improve=41.38048, (0 missing)
  mk1   < 0.5      to the right, improve=39.23801, (0 missing)
  sec5  < 0.5      to the left,  improve=23.74067, (0 missing)
  mk3   < 0.5      to the left,  improve=21.21772, (0 missing)
Surrogate splits:
  pbr   < -0.073964 to the right, agree=0.742, adj=0.188, (0 split)

```

木の複雑さとCPの例

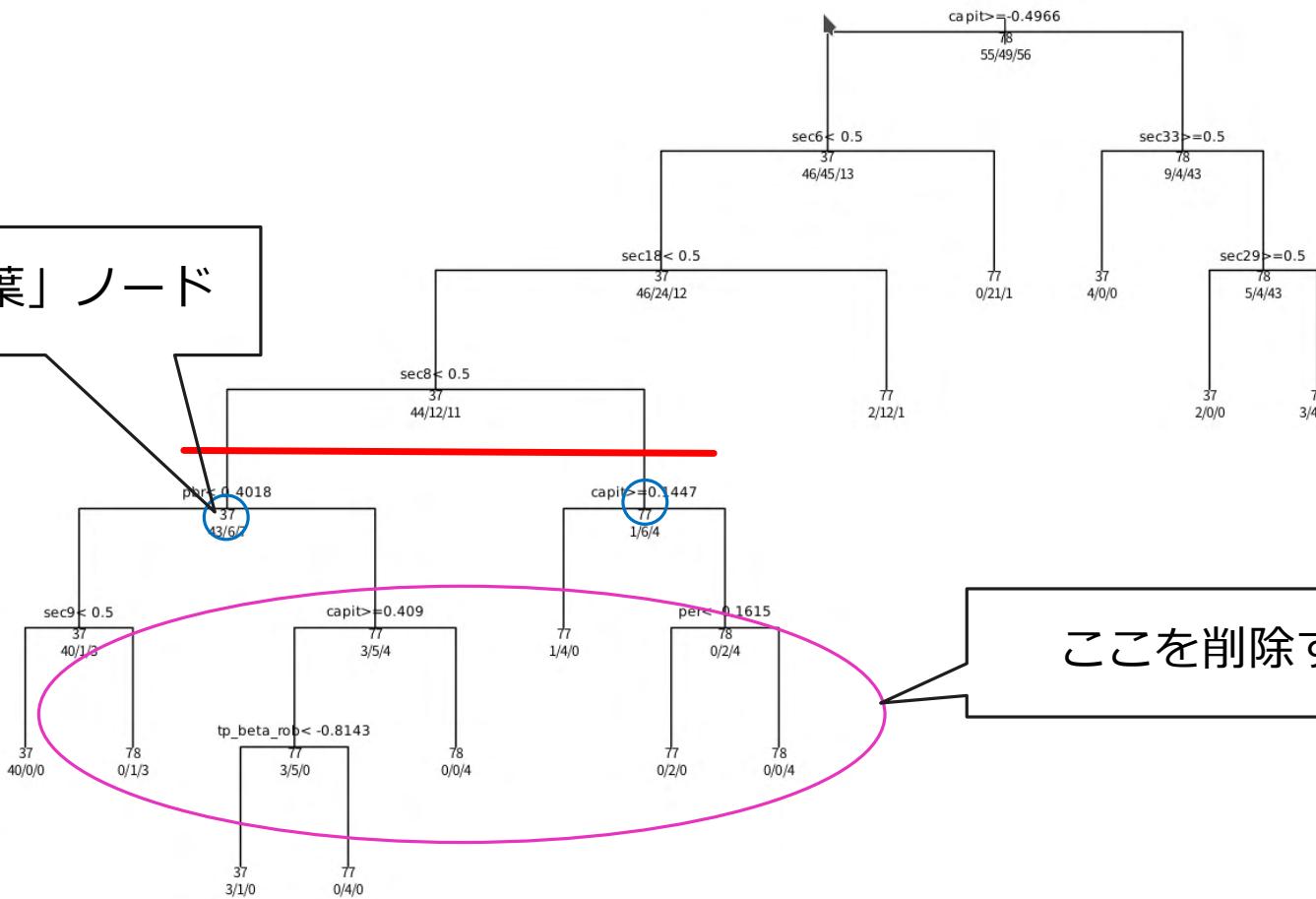


- いったん大きな木を作つてから、CPを見ながら、適当なところで木をカットする
- この処理を木の「刈り込み」(post-pruning)と呼ぶ
- ライブラリでは、ここまでをほぼ自動的に処理してくれることが多い
 - こまかに制御は手動で行ったほうがより適切な木が得られる

木の刈り込み

新しい「葉」ノード

Classification Tree BID: 15 GID: 37 77 78



C5.0、エントロピー

- 別の決定木生成アルゴリズムであるC5.0では、分割の良さを図る指標として「エントロピー」(measure of disorder、無秩序の指標)を用いる

- $E(S) = \sum_{i=1}^C -p_i \log_2(p_i)$

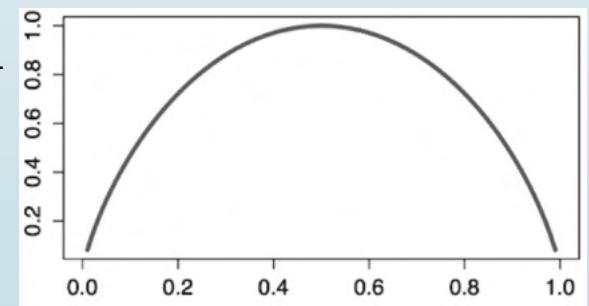
- このエントロピーEが最小（情報量が多い、有益）になる分割を選択する
- エントロピーは、全要素が単一集合となるときに最小となる（下図）
- 具体的には、分割による情報利得（information gain）を

$$IG = E(\text{分割前}) - E(\text{分割後})$$

として計算する

- 分割によるIG、すなわちエントロピーの削減幅が最も大きい分割を選択する

エントロピー



シェア

データ（特徴の一部）が欠損している場合 の扱い

- メインの分岐に代わって、代理分岐（surrogate split）と呼ばれる「2番手」が代わりに処理してくれる枠組みが用意されている
 - 代理がないと、データが欠損している場合、その先に進めなくなるので、答えが得られないという問題に陥る
 - こういう柔軟な枠組みが用意されているところも決定木の優れたところ
 - ソフトウェアを使う場合、代理分岐は自動的に構成されるのが普通（ユーザーがいちいち意識しなくてもよい、どこで代理が使われるかなども明示される）



決定木のメリット

- わかりやすい
 - 分類ルールが木構造で表現されるので、人間にとってとてもわかりやすい
 - 機械学習の手法は、ブラックボックス的なものが多いとされる中で（例：深層学習）、この「わかりやすさ」は非常に重要なポイント
 - 確率分布などを用いていない分、シンプル
- 適用対象の幅が広い
 - カテゴリー分割、回帰分析を含めて、非常に応用範囲が広い
 - 大規模なデータセット、欠損値があるデータセットでも大丈夫
- 実装プログラムが豊富
 - 伝統的手法なので、Rを含めて非常に多くの統計プログラムで標準装備されている

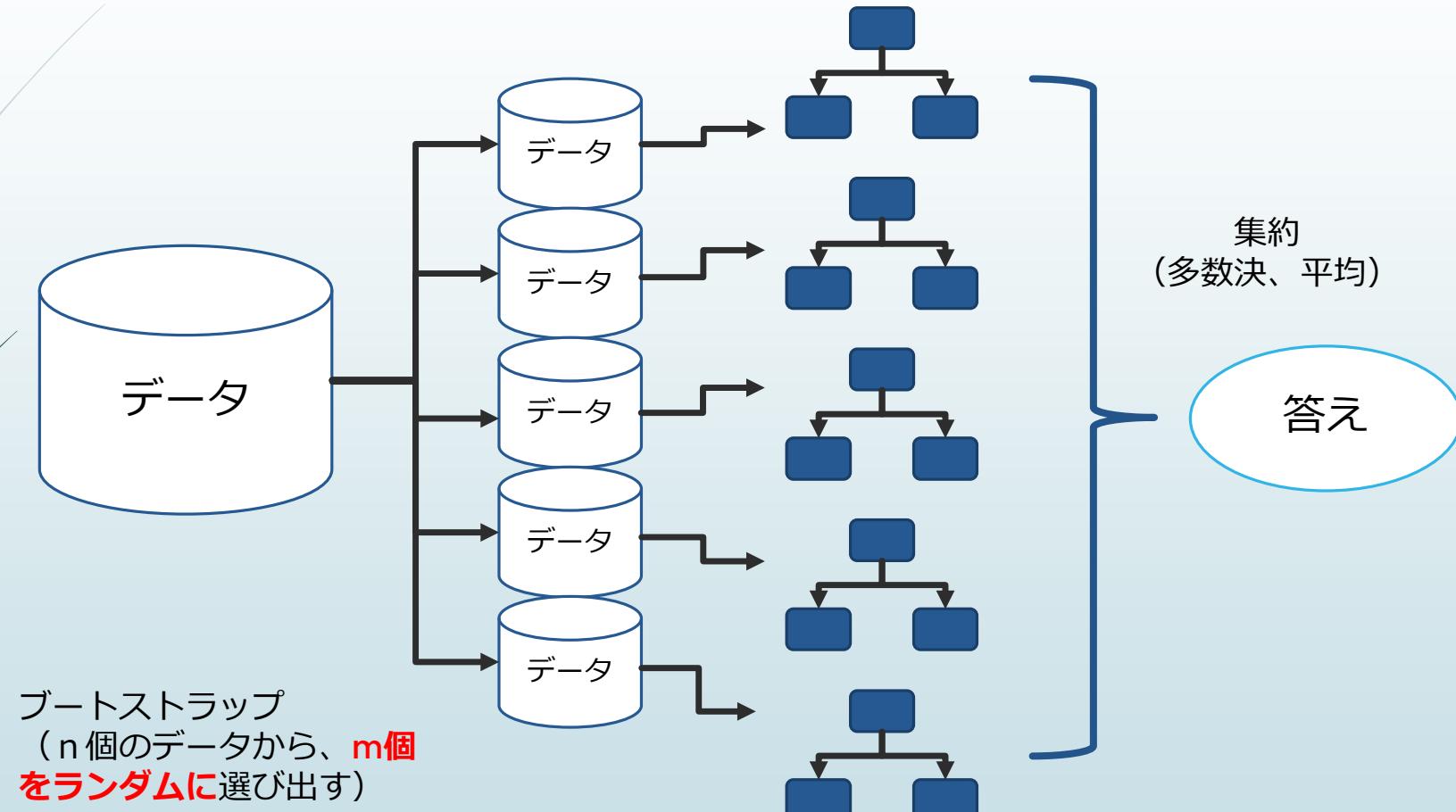
決定木の問題点：過学習

- 過学習とは、学習のやり過ぎ（やらせ過ぎ）を指す
- 学習教材（データ）に対して、あまりも細かく分類ルールを設定してしまう
- 現在のデータに対しては、非常にあてはまりの良い説明（分類ルール）が得られる（好成績）
- しかし、未知のデータ（本番の応用例）では、あてはまりがガタ落ちとなってしまう（使い物にならない）
- 決定木では、この過学習の問題が生じやすい（SVMなどに比べて）
- CPによる複雑性の制御、木の刈り込みを行うなどして、一応の対策をとっているが、最適な木の構築は非常に難しい
 - オーバー・アンダーフィッティング問題がどうしても生じやすい
- より抜本的な対策としては、ランダム・フォレストやブースティングなどの手法が取られることもある

過学習への対策：アンサンブル学習

- アンサンブル学習：単一では精度の低い分類（学習）器を複数用いて精度を上げる試み ⇒ バギング、ランダム・フォレスト、ブースティング
- バギング・・・ブートストラップを用いてデータをリサンプリングすることで、複数のデータセットを作成して、それぞれ決定木を作成する
 - ブートストラップ・・・袋から取り出して、もどして、また取り出す（復元抽出）
 - 機械学習の用語としては、バギング（袋）と呼ばれる
 - ランダムに作成されたデータのサブセット群は、元のデータ（母集団）の性質を保持している（統計理論）
- 複数の決定木を使って予測を行い、結果を**多数決**（カテゴリー）、**平均**（回帰）するなどして、単一の木を使った場合よりも精度のよい答えを得る

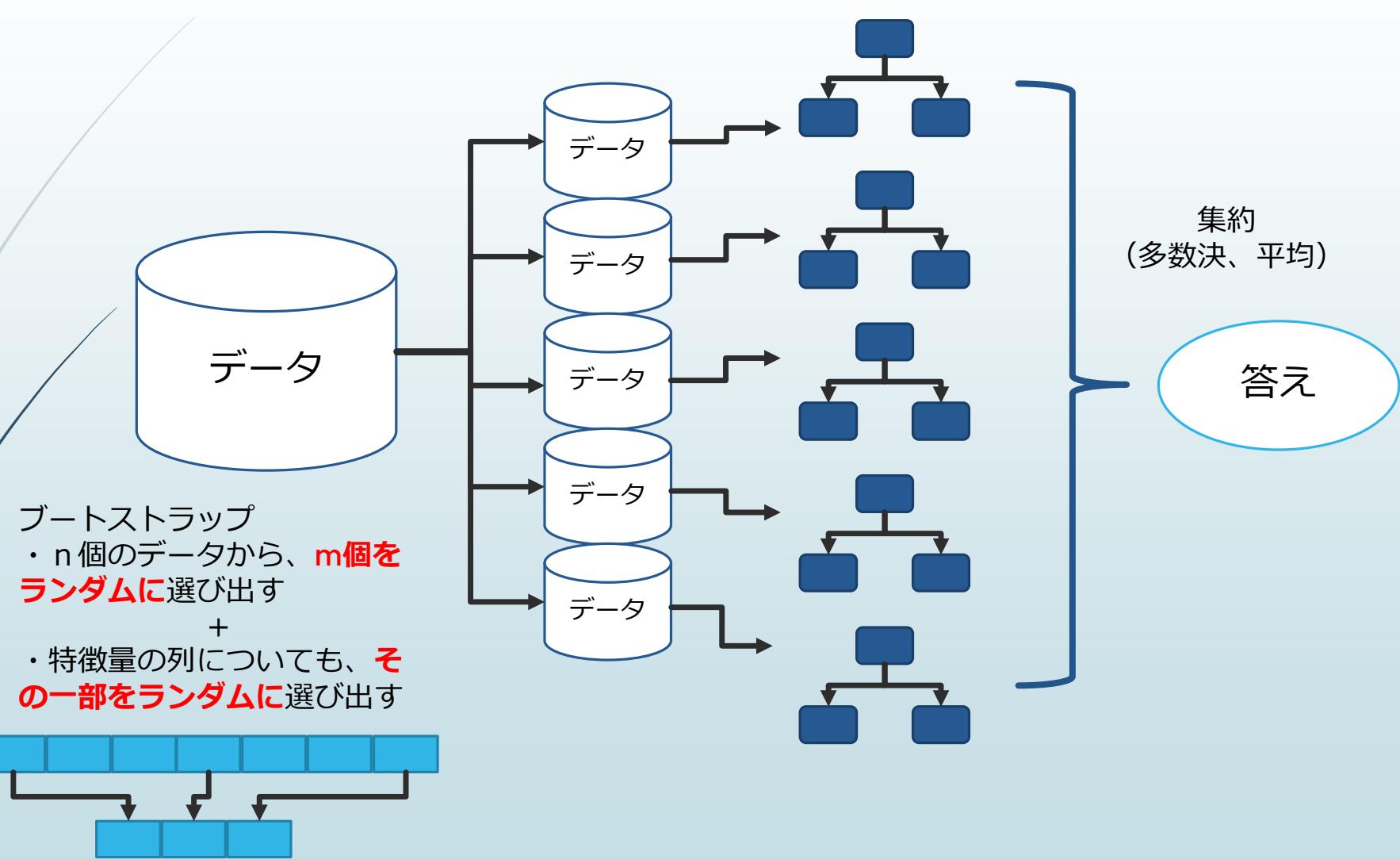
バギング



ランダム・フォレスト

- ランダム・フォレスト： バギング+決定木のさらなる拡張
- アンサンブル学習・・・ブートストラップを用いてデータをリサンプリングすることで、複数のデータセットを作成して機械学習にかける（バギングと同様）
- さらに、特徴量に関する次元圧縮を行う
 - 特徴量（データの列）についても、リサンプリングを行う
 - データサイズ（個数）、データの次元（列の数）の双方について、圧縮されたサブデータセットが作成される
 - バギングのみの場合よりさらに多様性のある決定木が作成できる（はず）
- 多数の試行（**ランダム**なデータセットで複数の決定木＜森＞）を繰り返し、結果を**多数決**（カテゴリー）、**平均**（回帰）するなどして、单一の木を使った場合よりも安定した答えを得る
 - ランダム+森 ⇒ ランダム・フォレスト

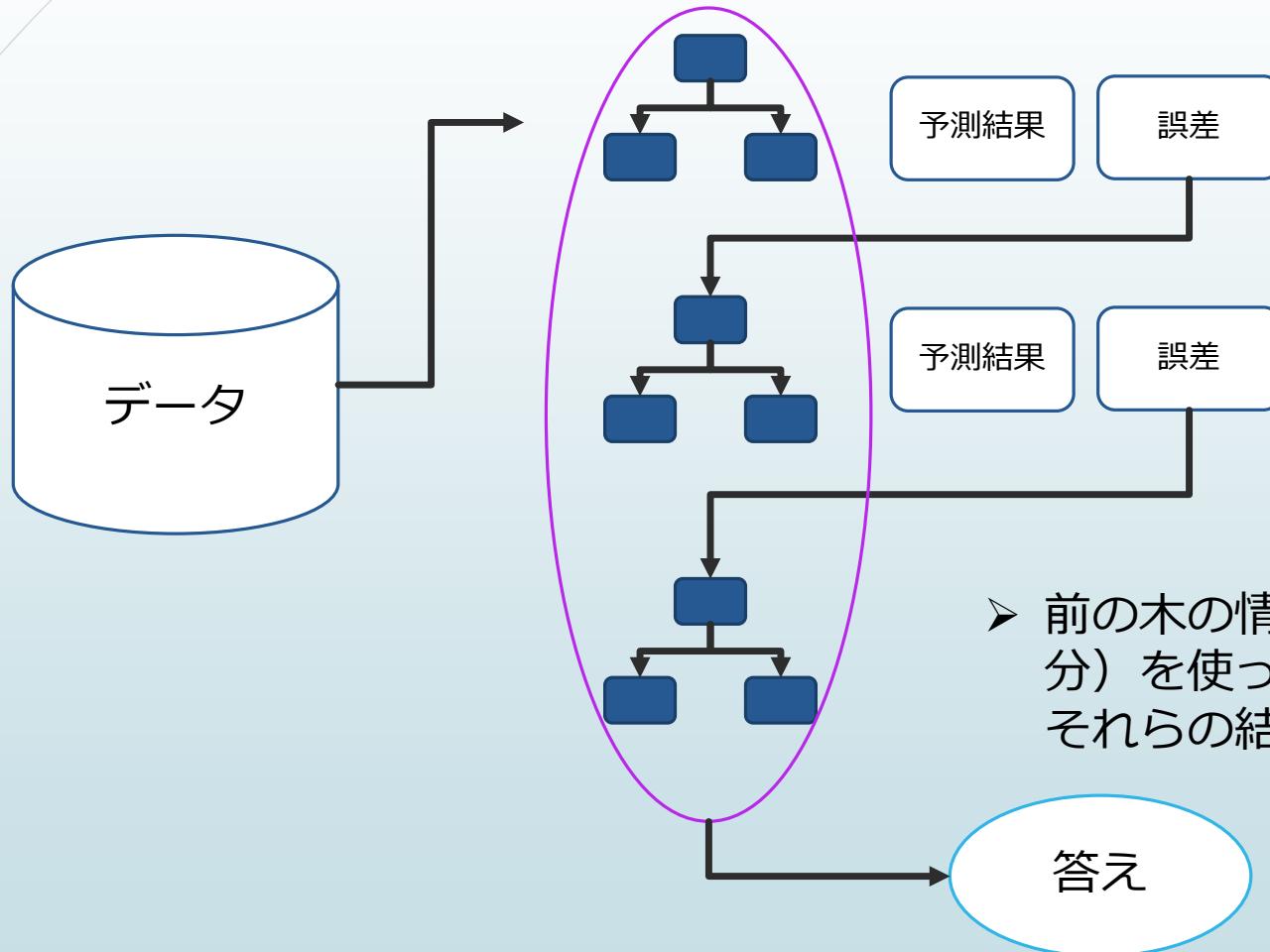
ランダム・フォレスト



ブースティング

- ブースティング： 単純な学習器（決定木）を多数組み合わせて、予測精度の高い学習器を生み出す
- アンサンブル学習（ランダム・フォレスト）は、学習器（決定木）を複数並列させるアプローチ ⇔ ブースティングは、複数の学習器を直列させるアプローチ
- モデルを逐次更新していく手法
 - ひとつ前の学習器の結果（誤差）を次の学習器の入力に反映させることで、精度の向上を目指す（逐次更新）
 - モデルの信頼度、データの重みなどのパラメータが更新されていく
 - ランダム・フォレストのように、最後に各段階の結果を統合する処理あり
- ブースティングは、ある種の損失関数の最小化問題として解くことができる（数理最適化問題）
 - 予め決められた損失関数の各特徴に関する勾配情報（微分）を使ったアルゴリズムで最適化問題を解く手法（勾配ブースティング）が有名
 - XGBoostとして実装されており、RやPythonで利用可能（かなり強力な手法として近年様々な場で使われている）

ブースティング (XGBoost) のイメージ



➤ 前の木の情報（誤差：説明できなかった部分）を使って、新たな木を逐次的に作成し、それらの結果を総合して最終的な答えを得る

参考文献

1. 「Rによる機械学習」ブレッドランツ著、長尾高宏訳、翔泳社、2017
2. 「Pythonで学ぶ統計的機械学習」金森敬文著、オーム社、2018
3. 「見て試してわかる機械学習アルゴリズムの仕組み 機械学習図鑑」、秋庭伸也 他著、翔泳社、2019



Q&A



予測的データ解析手法（3）： ベイジアン分類

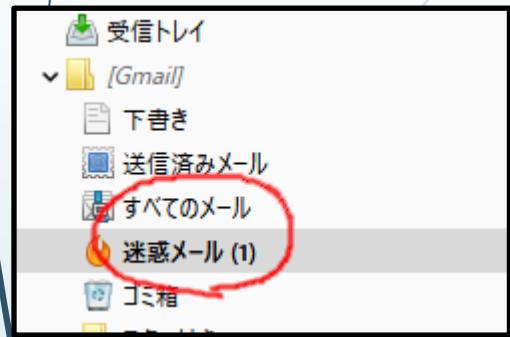
担当：磯貝 孝

イントロダクション

- 機械学習の基本的なアプローチ
 1. 教師あり学習・・・正解あり
 2. 教師なし学習・・・正解は必ずしも明らかでない
- 「ベイジアン分類」は、教師あり学習の一つの手法
- 「ベイジアンフィルター」、「ナイーブ（単純）ベイズ分類器」などと呼ばれることがある
- 代表的な応用例・・・スパム（迷惑）メールの振り分け、文書の自動分類など
- 基本技術・・・「ベイズの定理」の応用
 - 分類のほか、ベイズの定理を複雑な変数間の因果関係の記述に用いる「ベイジアンネットワーク」（グラフィカルモデルの一つ）という応用も存在する

迷惑メールのフィルタリング

▶ Thunderbirdの迷惑メール振り分け機能



迷惑メールフォルダ
にメールが自動的に
移動される

The screenshot shows the Thunderbird preferences window with the '迷惑メール' tab selected. A red circle highlights this tab. Below it, there are several configuration options:

- 既定の迷惑メールフィルターの動作を設定します。アカウントごとの迷惑メールフィルターの設定は [アカウント設定] で行います。
- 迷惑メールであると手動でマークしたときに次の処理を実行する(W):
 - "迷惑メール" フォルダーへ移動する(O)
 - メッセージを削除する(D)
- 迷惑メールと判断したメッセージを既読にする(M)
- 迷惑メール適応フィルターのログを有効にする(E)

A confirmation dialog box titled '判別基準データ消去の確認' (Confirmation to delete learning data) is displayed at the bottom right. It contains the question '本当に学習フィルターの判別基準データをリセットしますか?' (Do you really want to reset the learning filter's classification data?) with 'OK' and 'キャンセル' (Cancel) buttons. A red circle highlights the '判別基準データのリセット(R)' button in the dialog.

ベイジアン分類を理解するには

- ベイジアン、ベイズ統計とは何か？
- そもそもベイズの定理って何？
- 確率、条件付き確率との関係

⇒ このあたりのことを大まかに理解しておくと、理解がしやすくなる

- 実際の運用は、ソフトウェアを使うことで難しい数式などを理解しなくても大丈夫（研究者は別）
- ベイズ統計的な手法は、機械学習の様々な場面でよく用いられている
- 以下では、やや遠回りになるが、（せっかくの機会なので）ベイズ統計と普通の統計との違い、ベイズの定理の意味と応用について説明してから、ベイジアン分類に進む

ベイズ、ベイジアン

- 機械学習、データマイニングなどで、ベイジアン、ベイズ統計など、「ベイズ」という言葉がよく聞かれるようになった
- これらは、かなり昔から存在する考え方・統計理論で、それ自体は目新しいものではない
- 近年、よく使われるようになった背景としては、データの活用に便利な側面があること、モデルの推定がコンピュータの処理能力の向上、推定アルゴリズムの進歩でかなり改善したことが背景にある
- ベイズは人の名前・・・トマス・ベイズ (Thomas Bayes) 、18世紀のイギリスの学者
- 確率に関して、ベイズの定理（後述）を示し、「直観的信頼度」の概念を導入した（大雑把な理解）

統計学における二つの大きな流れ

- ベイズ統計学（ベイジアン）と伝統的な頻度主義（普通の統計学）
- この二つは、問題への対処法がかなり違う
 - 確率分布など、基本的な事柄が決定的に違うわけではない
 - 正規分布は、ベイジアンでも伝統的な統計学でも何ら変わらない
- 我々が、中学・高校（？）や大学の教養課程（？）で習うのはほとんどの場合、普通の統計学
 - 例：回帰分析・計量経済学で、家計消費とGDPの関係をモデル化して推定する、など
 - 家計消費とGDPの関係式を設定して（モデル化）、パラメータ（係数）を「推定」する
 - 係数の妥当性をt値などの統計量で確認する
- ベイズ統計、にはほとんど出会わなかつたはず

統計学の位置づけ、ベイジアン

- 「日本の大学には統計学部がない」などの指摘は、データサイエンスの重要性を語る際によく聞かれる言葉
 - 最近は、少し変化もみられているが、大きくは変わっていない
- 大学でも、統計を真正面から勉強するという機会はあまり多くなかったかも
- 伝統的な統計学ですら正直よく理解できていない、というのはよくあること
- なのに、いきなりベイズ統計とかベイジアンとか、言われてもなかなか理解しにくい
- どこが違うのか、一番の違いを知りたい
 - 人によって強調するポイントは違ったりしますが、視点は大きくは同じはず

頻度主義統計とベイジアン（ベイズ統計）

- モデル構築、パラメータ推定の側面から比較してみる
- $Y = a^*X + b$ のようなモデルを考えてみる
 - Y : 家計消費
 - X : 所得 (GDP)
 - YとXに関する時系列データがたくさんあるとして、モデルのパラメータ a, b を特定したい
 - Xがどのくらい増えたら、Yがどのくらい増えるか、などの予想をしてみたい
- まず普通の統計学のアプローチでは、XやYは確率変数（確率的に変動します）とされ、 a や b は、パラメータ、すなわち定数（何らかの値）と考える
 - データX、Yを用いて、 a や b の「値」を推定する (**点推定**)

ベイジアンの場合

- $Y = a^*X + b$ のようなモデル
- ベイジアンの考え方では、パラメータの a, b ともに確率変数としてばらつきをもつ（分布しています）
 - 普通の統計学では、あくまで固定値です（分布なし）
- 普通の統計学では、最尤法や回帰係数の公式などの手法を用いて、 a, b の正しい答えを特定します
- ベイジアンでは、大事なのは パラメータの分布 です
- パラメータの分布をまず求めて、その後に分布を適切に要約する値を求めます
 - 時に両者のアプローチは、表面的には違っても、結局同じことをしている、という場合も多い

事前確率、事後確率

- ベイジアンの手法では、必ず事前確率、事後確率という言葉が出てきます
 - ここが非常にわかりにくい部分、なぜそれが必要なのか？
- ベイジアンの考え方の基礎である「**直感的信頼度**」に関係します
- 例えば、パラメータa, bに関する直感的信頼度とは
 - 分析者の主観（以前の実験の結果や事象に関する個人的信頼度）などの事前知識に基づいて決めるもの
 - 頻度または傾向に基づく固定値ではない
 - 新たな情報（データ）が集められると変わる ⇒ 確率の**更新**

確率の更新

- 分析者がパラメータ a, b に関する確率分布を自分で決める
⇒ 事前確率（分布）
- データ (X, Y) が得られた
- X, Y を用いて、パラメータ a, b に関する確率分布をより確からしいものに更新する
⇒ 事後確率（分布）
- 更新されたパラメータ（確率分布から得られた代表的な値）を使って、新たな事象 Y の発生確率（条件付き確率）も予測できます
- これが、ベイズ流（ベイジアン）のデータを用いたパラメータの更新となります
 - 普通の統計学では、事前確率などは一切想定せず、得られた全てのデータを使って、固定値であるパラメータを求めて、それで終わります

ベイジアン・アプローチにおける確率更新の流れ



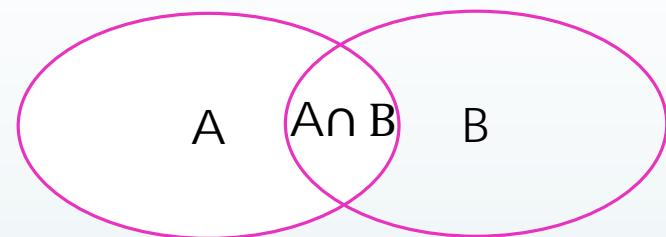
*データが新たに入手できれば、パラメータの分布に関する直感的信頼度が更新される

更新をどのように行うのか？

- 事前確率、データ、事後確率、この流れを具体的にどう処理すればいいか？
- ここで、「ベイズの定理」の出番
- ベイズの定理は、条件付き確率の公式から直接導かれるもの
- ベイズの公式は、データに基づく「直感的信頼度」の更新の具体的処理の流れとしてみることができる（ベイジアンの考え方の基礎）
- パラメータに関する事後確率（分布）が得られたら、分布から中央値や期待値などの具体的な値を割り当てることができる
 - 事後確率がきちんと計算できるか、がベイジアンの最重要点
 - 解析的に計算できる場合もある（そのようにモデルを限定することもある）

ベイズの定理

$$P(B|A) = \frac{P(A, B)}{P(A)}$$



条件付き確率 (Aであるという条件のもとでのBである確率)

$$P(B|A)P(A) = P(A, B)$$

$$P(B|A)P(A) = P(A, B)$$

AとBの入れ替え $P(A, B) = P(B, A)$

$$P(A|B)P(B) = P(B|A)P(A)$$

両辺を $P(B)$ で割る

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

ベイズの定理

$$P(A|B) \propto P(B|A)P(A)$$

\propto : 比例する、 $P(B)$ は定数 <計算困難なことが多い>

ベイズの定理、事前確率→事後確率

$$P(A|B) \propto P(B|A)P(A)$$

- A : モデルのパラメータなど、関心がある対象
- B : データ
- 事前確率: $P(A)$ パラメータに関する事前確率（主観的に設定する）
- 事後確率: $P(A | B)$ データが得られた後で更新されたパラメータの確率分布
- ベイズの定理を適用することで、事前確率から事後確率への更新が可能となる
- $P(B | A)$ は、尤度、パラメータを既知としてデータの生じる確からしさ
 - $P(B | A)$ は、確率に近いイメージだが、足しあげても 1 にならない（確率ではなく尤度と呼んでいる）

事後確率と尤度

事後確率

\propto

尤度

\times

事前確率

- 事前確率を設定する
- データが得られたら、すべてのデータについて尤度を計算する
 - 確率密度の計算は、既にわかっているものと仮定している（事象のモデルに依存）
- 二つの掛け算から事後確率（分布）を得る
 - 多数のパラメータの値の組が得られるイメージ（分布）
 - 事後確率が解析的に表現できることもある（数式で表現可能）
 - 事後確率が容易に計算できないことが多い（多くの場合はこれ、要シミュレーション）

事後確率の計算の難しさ

- 事前分布はどう設定するのか ⇒ 情報がなければ無情報分布とか
 - 頻度主義の観点からは、ベイジアンのここが最も批判されやすい
- 事後分布が解析的に容易に処理できないこともある
 - MCMC（マルコフチェイン・モンテカルロ）などの手法を用いて、事後分布からサンプリングを行うことで、たくさんのパラメータの組を得て、それで事後確率分布の代用とする
 - PCの能力向上、サンプリングアルゴリズムの改善により能力がかなり向上している
- ベイズ統計を処理できる各種ソフトウェアも揃っており、実際の応用はかなり精度が上がっている

機械学習との相性の良さ

- データが得られれば、パラメータの更新ができる、新しいモデルによる予測ができるようになる
- 機械学習の基本的な枠組み（データから学習）として使いやすい
- パラメータが点推定でなく、分布なので確からしさの評価がやりやすい
 - ベイジアン信頼区間と従来の統計学での信頼区間
 - ベイジアンだと、パラメータが分布なので、信頼区間を評価しやすい（95%信頼区間など）
- ただし、基本的な考え方についてきちんと理解していないと、何をやっているのかわからなくなりやすい

ベイズ統計の技術的な難しさ

- 事前分布の選択
- 事後確率の計算方法の選択
 - 解析的な解決：特定の確率分布をイメージして、最尤法を直接適用してパラメータの推定値を求めることがあります
 - モンテカルロ：多くの場合、事後分布が複雑になるので、MCMCなどでサンプリングを行って、密度推定を行う
 - そこからMAP推定（最大事後確率推定、maximum a posteriori, MAP）などの基準で、パラメータの推定値を特定します
 - この流れが結構難しい
 - 特にMCMCの収束の確保など、技術的に扱いが難しいところもあります

伝統的な統計学とベイジアン

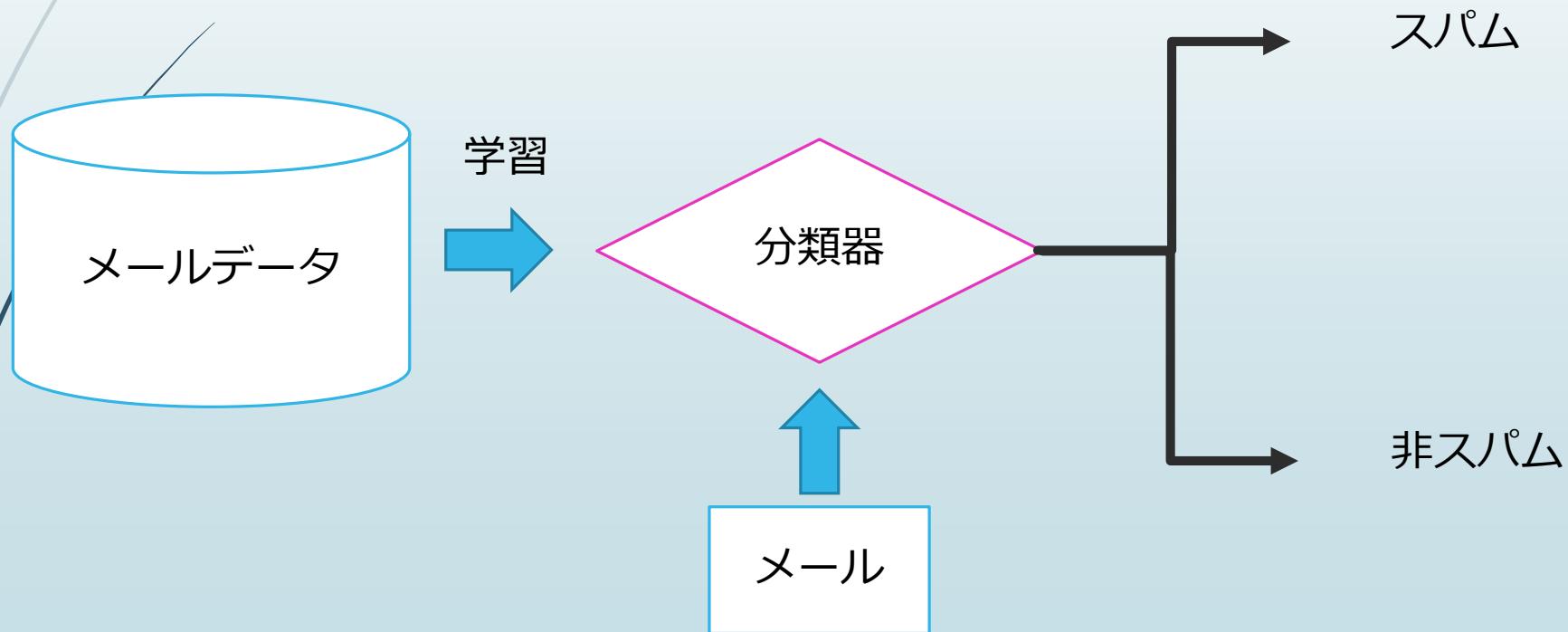
- どちらがより優れている、というような対立軸には本来ない
- 都合のよい方を見極めてつかいこなす、というのが、データサイエンス的には望ましいアプローチだと思われます
- 頻度主義のアプローチとベイジアンのアプローチは、結構共通する部分も多いので、それぞれ参考になります
- モデリングを行う場合、データ追加によるモデルの更新（オンライン学習的なもの）を想定する場合、ベイジアンアプローチは特に便利
- 一方で、モデルの推定を含めた扱いは、やや難しくなることが多いかも

ベイジアン分類（ベイジアンフィルター）

- ベイジアン分類：ベイズの定理を分類問題の解を得るのに応用する
- 単純（ナイーブ）ベイズフィルター、などとも呼ばれる
- 「迷惑メールのフィルター」などに応用されている
- 既存のメールから、ある種の単語やフレーズに着目して、そのメールが迷惑メールかどうか、判定する処理
 - 機械学習の一種（教師あり学習）
 - データ（メールの判定結果）が増えれば、精度の向上が期待できる
- テキスト処理の場合、形態素解析（単語品詞分解）、単語辞書登録、単語文書行列の作成（重みの設定など）などの事前処理が必要になることがあります（詳細省略）

簡単なベイジアン分類の例

- メールの中に「セール」という単語を含むか否かに着目して、スパムメール、非スパムメールに分類する



ベイズの定理の応用

$$P(\text{スパムメール} | \text{「セール」あり}) = \frac{P(\text{「セール」あり} | \text{スパムメール}) * P(\text{スパムメール})}{P(\text{「セール」あり})}$$

事後確率

尤度

事前確率

- メールの本文中に、「セール」という言葉が含まれているとき、そのメールがスパムメールかどうかを判定したいとする
- 事前確率：主観的な予想として、受信メールの20%（メール100通中、20通がスパムメール）はスパムメールだと想定した $\Rightarrow P(\text{スパムメール}) = 0.2$
- 過去の受信メールの中身を調べて、実際に「セール」という言葉が含まれていたメールについて、それがスパムだったかどうか（自分で）判定しておく
 \Rightarrow データ
- このデータから尤度を計算して、スパムメールに関する事前確率を事後確率に更新する

過去に受信したメールのデータ

頻度
(事前確率)

| | | 単語「セール」 | | 合計 |
|-----|------|---------|------|-----|
| | | 含む | 含まない | |
| スパム | スパム | 5 | 15 | 20 |
| | 非スパム | 2 | 78 | 80 |
| 合計 | | 7 | 93 | 100 |

尤度

| | | 単語「セール」 | | 合計 |
|-----|------|---------|----------|-----------|
| | | 含む | 含まない | |
| スパム | スパム | 5 / 20 | 15 / 20 | 20 / 20 |
| | 非スパム | 2 / 80 | 78 / 80 | 80 / 80 |
| 合計 | | 7 / 100 | 93 / 100 | 100 / 100 |

事後確率（スパムかどうか）

$$P(\text{スパムメール} | \text{「セール」あり}) = \frac{P(\text{「セール」あり} | \text{スパムメール}) * P(\text{スパムメール})}{P(\text{「セール」あり})}$$

$$P(\text{スパムメール} | \text{「セール」あり}) = \frac{P(5/20) * P(20/100)}{P(7/100)} = \frac{0.25 * 0.2}{0.07} = 0.71$$

- 事後確率（「セール」を含むメールがスパムメールである確率）は、71%という結果が得られた
- 事前に閾値を設定しておいて（例えば、70%とか90%とか）、それに応じてスパム判定を行うことができる

単語の数を増やす

- 「セール」に加えて、「大幅値引き」も判定に使いたい
⇒ 二つの単語の出現の有無をそれぞれデータから抽出しておく（ケース分けが必要）
(セール、大幅値引き) : 1 (有、有) 、 2 (有、無) 、 3 (無、有) 、 4 (無、無)
- ベイズの公式を拡張する

$P(\text{スパムメール} \mid \text{'セール'} \text{ と } \text{'大幅値引き'} \text{ の有無})$

$$= \frac{P(\text{'セール'} \text{ と } \text{'大幅値引き'} \text{ の有無} \mid \text{スパムメール}) * P(\text{スパムメール})}{P(\text{'セール'} \text{ と } \text{'大幅値引き'} \text{ の有無})}$$

単語の数を増やす（2）

- P （「セール」と「大幅値引き」の有無 | スパムメール）の計算
 - 単語数が2なら、ケース分けで対応可能
 - 単語数が増えると、ケース分けが複雑なため計算が難しくなる
- 計算の簡略化：「クラス条件の独立性」が成立しているという前提
 - P （単語1の有無 | スパムメール）* P （単語2の有無 | スパムメール）
 - 単語が3つなら、 P （単語1の有無 | スパムメール）* P （単語2の有無 | スパムメール）* P （単語3の有無 | スパムメール）
 - 各ケースの尤度は、尤度表を拡張して計算しておく
 - 単語間の独立性の前提で複雑なケース分けをしなくても、尤度（ベイズの公式の分子）が計算可能
 - 分母の計算は、 P （単語1）+ P （単語2）のように積算する
- 分母、分子の割り算で、事後確率を計算する
- 判定に使用する閾値は、いろいろなケースを試して適切な水準を決める

単純（ナイーブ）ベイズ分類？

- なぜ、単純（ナイーブ）と呼ばれるのか
- クラス条件の独立性の仮定（それぞれの言葉＜特徴＞を独立事象と想定しているから）
- この仮定のおかげで、事後確率が（簡単に）計算できる
- 実際には、「セール」と「大幅値引き」は独立ではなく、相互依存している可能性がある ⇒ もしそうなら、事後確率の計算は厳密でないことになる
 - なので、ナイーブ
- ただし、実用面では、結構使えるという評価が多い

ベイジアン分類の問題点

- クラス条件の独立性の仮定のほかにも、技術的な問題点が存在する
- 複数クラス（単語）で、尤度を計算する場合に、観察された単語の発生数が 0 となっているため、尤度を 0 と計算してしまうことがある
- この場合、ベイズの定理の分子の尤度全体も掛け算の結果、0 になってしまう
- 事後確率に 0 % という結果が出てしまう
 - 予想としては、0 % の確率ということは現実的でない
- 頻度表を作成する場合、0 の部分に 1 などの小さな数字を加えて、尤度全体が 0 になってしまうのを回避する
 - ラプラス推定量
 - ある程度、現実的な答えが得られることが多い

参考文献

1. 「Rによる機械学習」ブレッドランツ著、長尾高宏訳、翔泳社、2017
2. 「Rで楽しむベイズ統計」奥村晴彦ほか著、技術評論社、2017
3. 「ベイズ統計学」松原望著、創元社、2017



Q&A



記述的データ解析手法（1）： クラスタリング

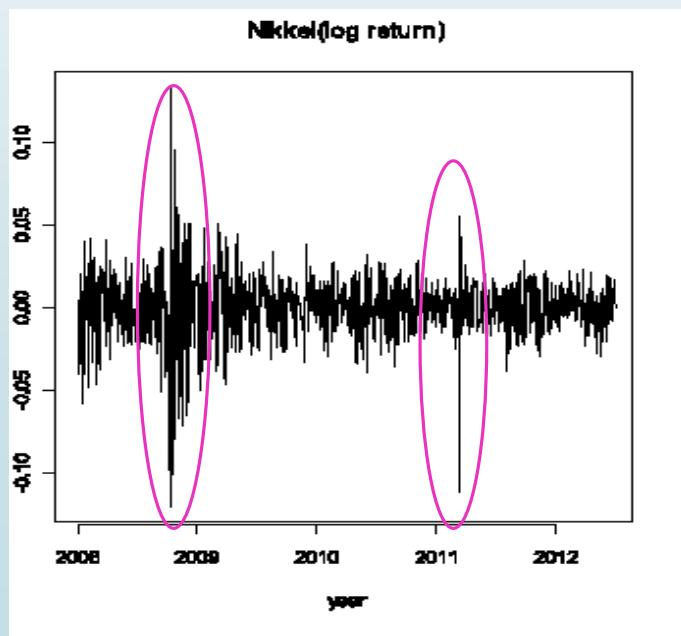
担当：磯貝 孝

教師あり学習と教師なし学習

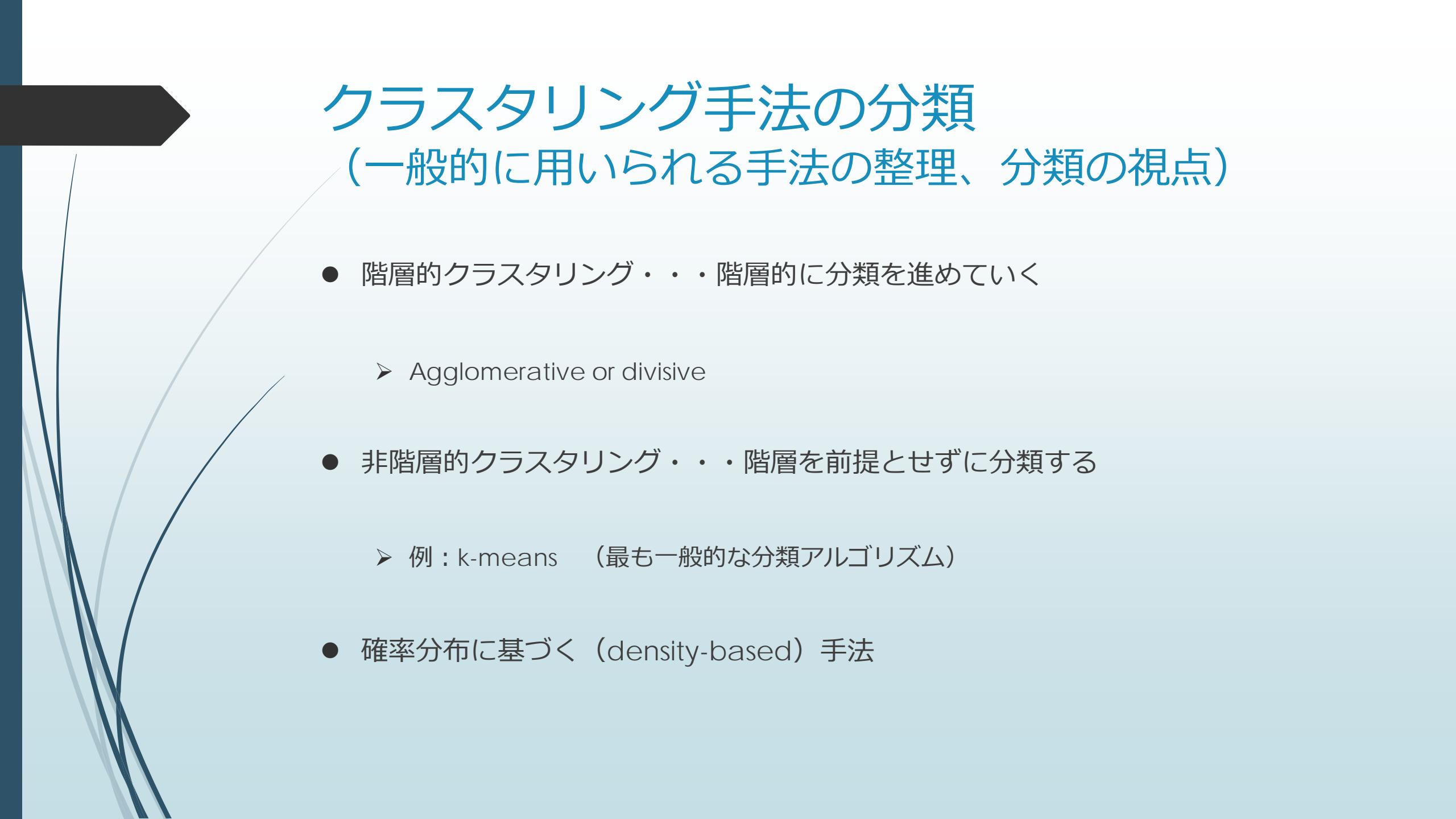
- 機械学習の基本的なアプローチ
 1. 教師あり学習・・・正解あり
 2. 教師なし学習・・・正解は必ずしも明らかでない
- 「クラスタ」 (cluster、群れ・集団)
 - クラスタリング（分類処理）は、教師なし学習の代表的な手法
 - 類似の個体をまとめて少数のグループを形成し、グループ毎の違いに注目する
 - 直接問題の「正解」を得るのではなく、複雑な問題を整理して様々な知見を得ることを目指す
- クラスタの意味づけは、分析者が行う（自動的には得られない）

クラスタリング – 言葉の使われ方

- データマイニングでは、「グループ分け」とほぼ同義で使われることがほとんど
- 分野によっては、「グループが形成されている」という状況を表現するのに、クラスタリングという言葉を使うことがある



*ボラティリティ・クラスタリングの例
(株価の大きな変動が特定の時期に集中して生じている)



クラスタリング手法の分類

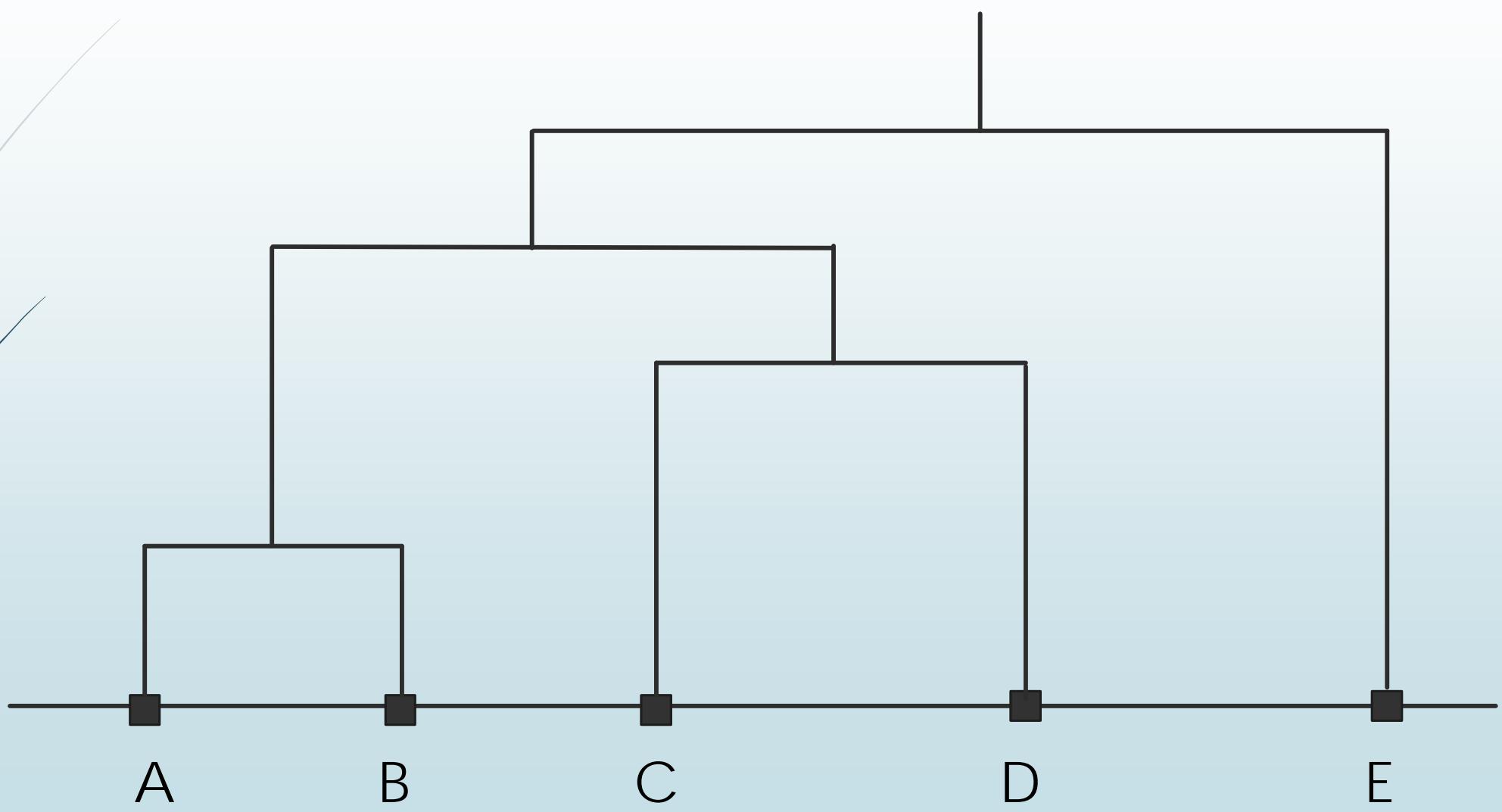
(一般的に用いられる手法の整理、分類の視点)

- 階層的クラスタリング・・・階層的に分類を進めていく
 - Agglomerative or divisive
- 非階層的クラスタリング・・・階層を前提とせずに分類する
 - 例：k-means (最も一般的な分類アルゴリズム)
- 確率分布に基づく (density-based) 手法

階層的クラスタリング

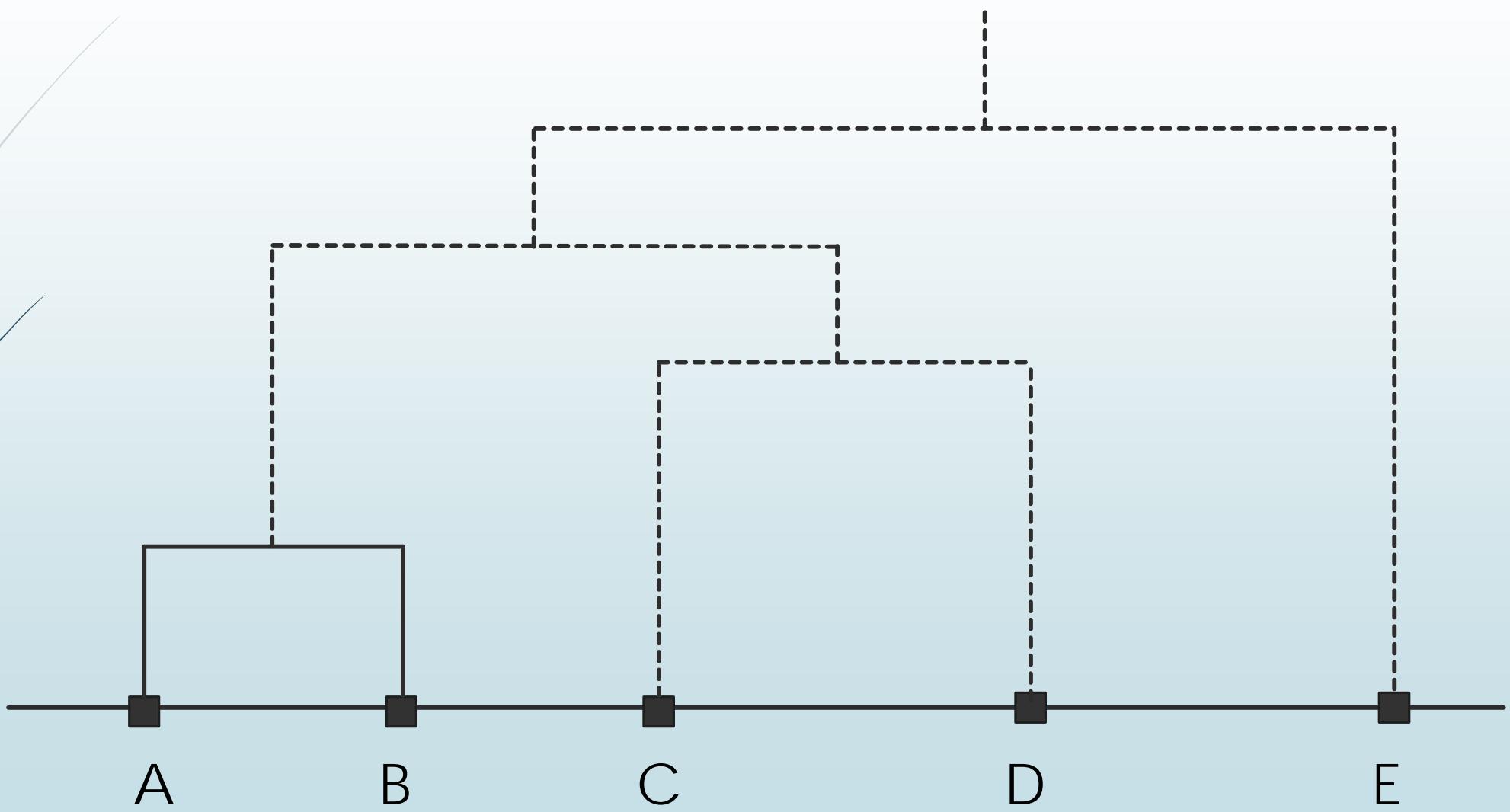
- ボトムアップ (Agglomerative) 、トップダウン (divisive)
 - Agglomerative ··· 凝集型、似通った個々のメンバーを集めてグループを形成するアプローチ
 - ✓ デンドログラムと呼ばれる系統図を描きながらグループを形成していく
 - ✓ 最後に全体が一つのグループにまとまる
 - Divisive ··· 分割型、全体をひとつのグループとみて、グループの分割を繰り返してグループを作成する
 - ✓ 最終的に個々のメンバーからなる最小グループができあがる
 - ✓ 実際には、大きな集合の分割には計算量の問題が生じることが多いので、凝集型が使われることが多い
- 階層的クラスタリングの場合、凝集型、分割型とともに、どこかでストップする必要あり（グループ数の決定）

デンドログラム（完成図）



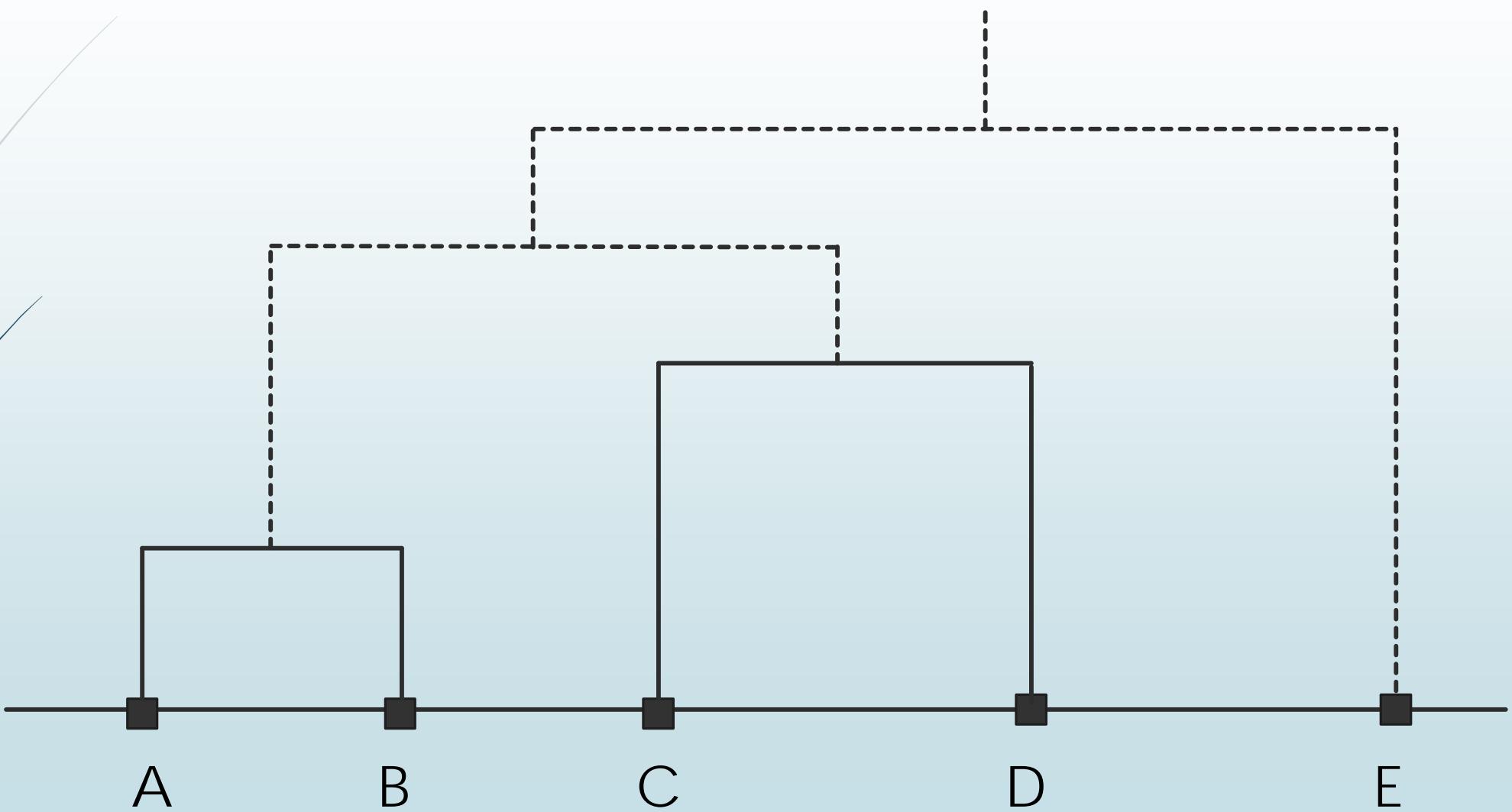


デンドログラム（1）

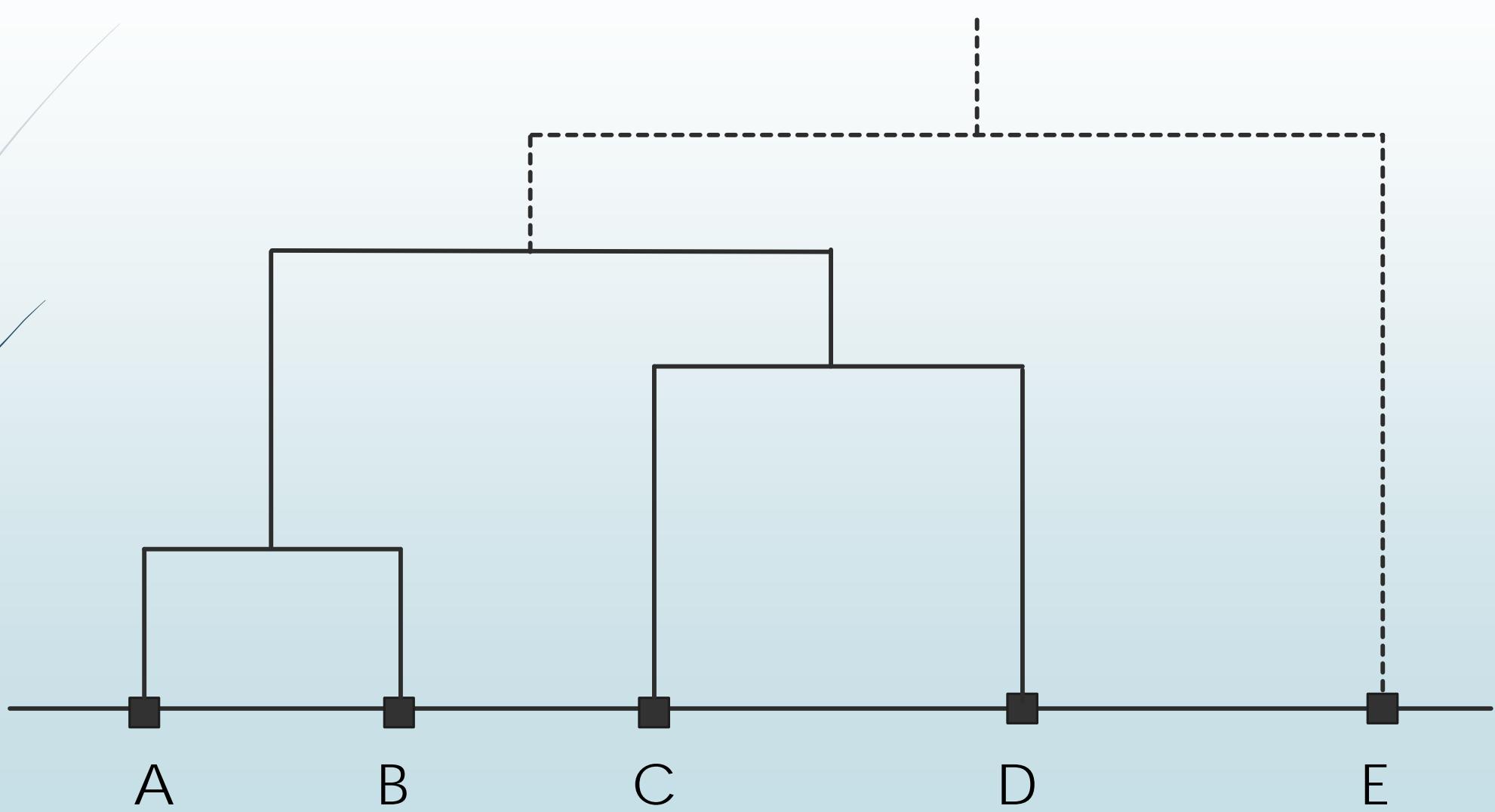




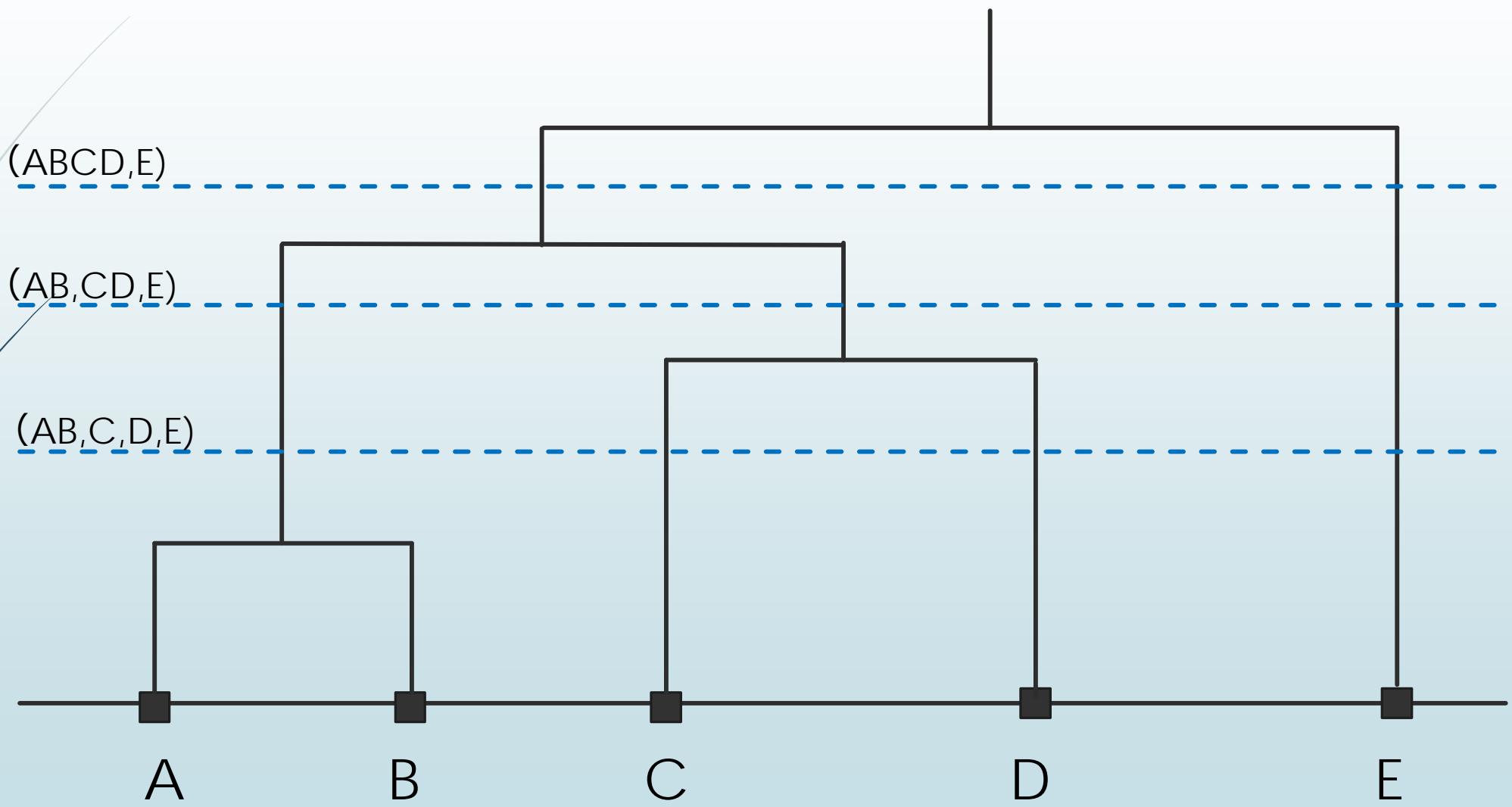
デンドログラム（2）

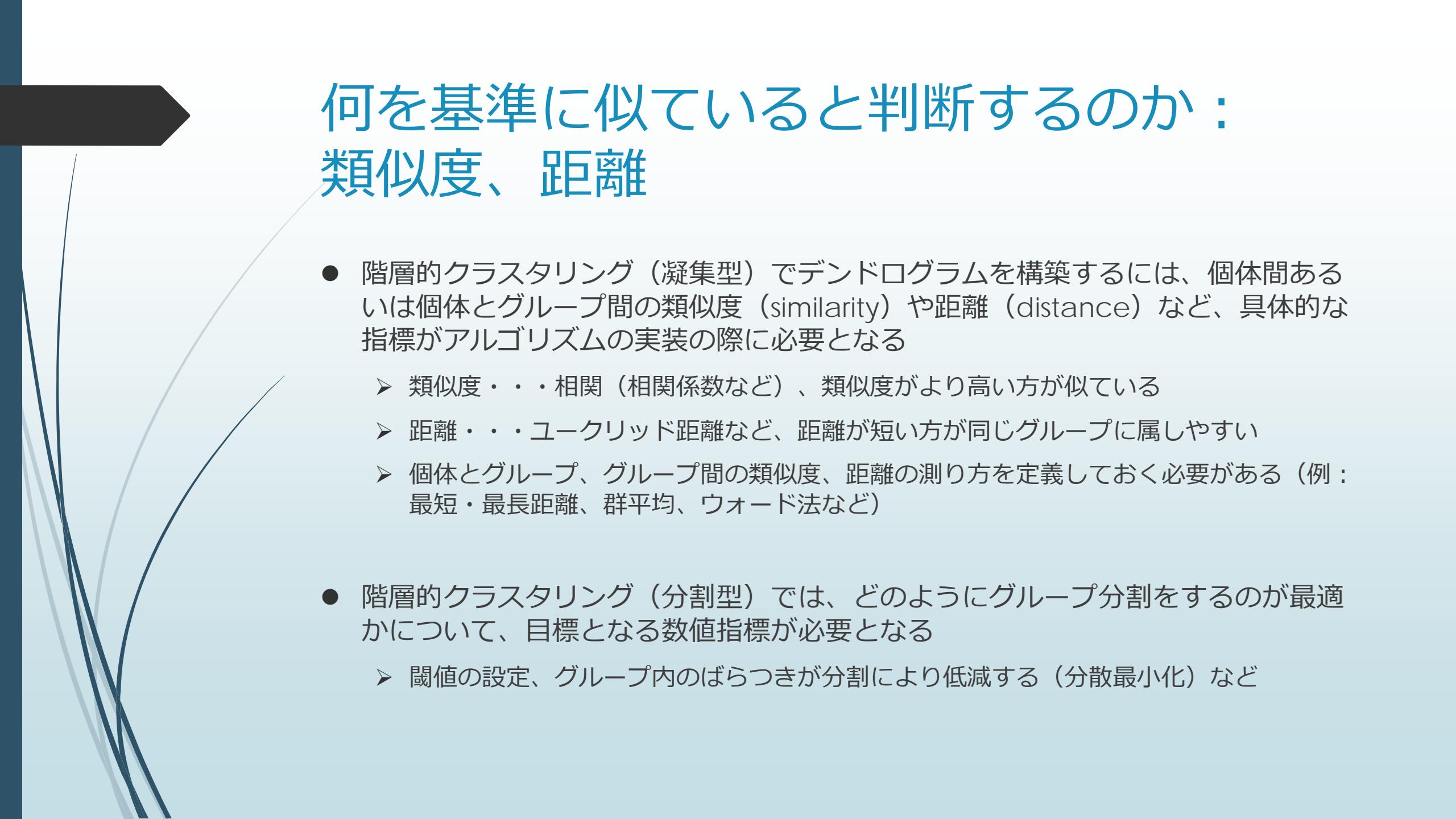


デンドログラム（3）



デンドログラム：クラスタ数の決定





何を基準に似ていると判断するのか： 類似度、距離

- 階層的クラスタリング（凝集型）で дендрограммを構築するには、個体間あるいは個体とグループ間の類似度（similarity）や距離（distance）など、具体的な指標がアルゴリズムの実装の際に必要となる
 - 類似度・・・相関（相関係数など）、類似度がより高い方が似ている
 - 距離・・・ユークリッド距離など、距離が短い方が同じグループに属しやすい
 - 個体とグループ、グループ間の類似度、距離の測り方を定義しておく必要がある（例：最短・最長距離、群平均、ウォード法など）
- 階層的クラスタリング（分割型）では、どのようにグループ分割をするのが最適かについて、目標となる数値指標が必要となる
 - 閾値の設定、グループ内のはらつきが分割により低減する（分散最小化）など

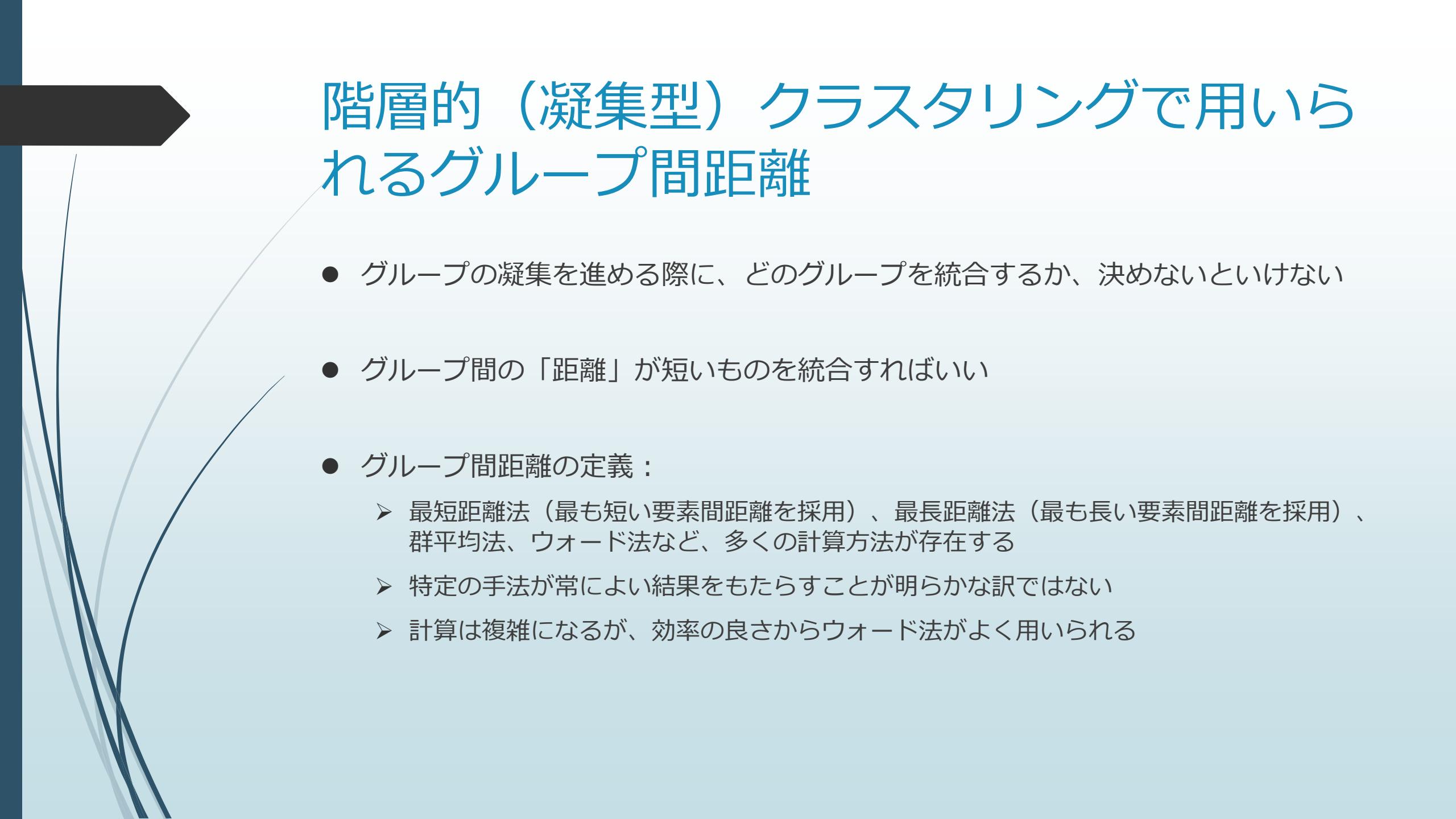
要素間の距離

- ユークリッド距離（直線距離）

平面上の2点 $(x_1, y_1), (x_2, y_2)$ の間の距離 d

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

- 一般的に用いられる距離指標
- データの属性によって次元数は（2よりも）大きくなったりする
- 相関係数（ピアソンの相関係数など）を用いたりすることもある（相関クラスタリング）
- どのような距離指標をクラスタリングに用いるかは事前の検討が必要



階層的（凝集型）クラスタリングで用いられるグループ間距離

- グループの凝集を進める際に、どのグループを統合するか、決めないといけない
- グループ間の「距離」が短いものを統合すればいい
- グループ間距離の定義：
 - 最短距離法（最も短い要素間距離を採用）、最長距離法（最も長い要素間距離を採用）、群平均法、ウォード法など、多くの計算方法が存在する
 - 特定の手法が常によい結果をもたらすことが明らかな訳ではない
 - 計算は複雑になるが、効率の良さからウォード法がよく用いられる

群平均法

$$d(C_1, C_2) = \frac{1}{|C_1||C_2|} \sum_{x_1 \in C_1} \sum_{x_2 \in C_2} d(x_1, x_2)$$

- ✓ 二つのクラスタの要素の全ての組み合わせについて、距離を測りその平均をクラスター間距離とする

ウォード法

$$d(C_1, C_2) = L(C_1 \cup C_2) - L(C_1) - L(C_2)$$

$$L(C_i) = \sum_{x \in C_i} D(x, G_{C_i})^2$$

- ✓ $L(C_i)$ は、クラスタの重心 G_C からの距離（ユークリッド）の2乗和
- ✓ C_1 と C_2 が似ているほど、 d は小さくなる

階層的（凝集型）クラスタリングの特徴

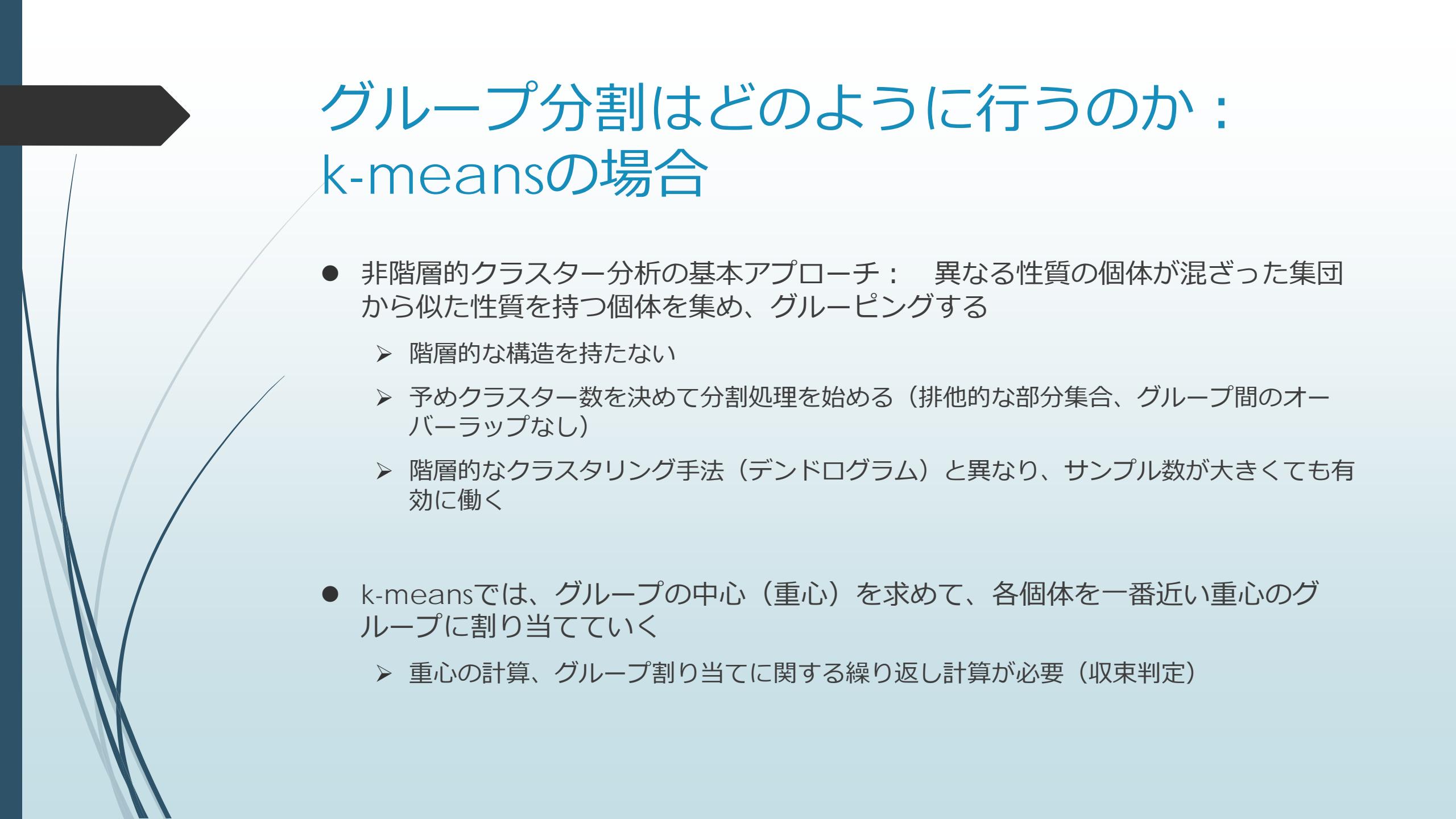
- グループの数を予め決めておく必要がない
 - k-meansなどの（非階層的）手法では、グループ数を決めておく必要がある
 - デンドログラムが出来上がれば、グループの数は後から決めることができる
- 計算過程がわかりやすい、初期状態に依存して結果が変わるということもない
 - k-meansなどの手法では、クラスタリング結果が初期状態に依存する場合がある
- わかりやすくて便利だが、要素数が非常に多くなると計算が複雑になって扱いにくくなる
⇒ その場合は、非階層的なクラスタリング手法を用いる

グループの数はどう決めたらいいのか

- 分類対象のデータに対して、あらかじめグループ数に関する何らかの知見があれば、積極的に活用する
 - 分析者の視点をクラスタリング結果（グループ数）に反映させることができる
 - 統計的な指標を用いてクラスタ数の大まかな情報を得ることもできる（例：クラスタ数を変えて、クラスタ内距離の2乗和の変化を観察し、変化が落ち着くところを探す）
 - 決定的な情報にはなりにくいことが多い
- 問題に応じて様々なヒューリスティックな方法を考え出すことも有益

非階層的クラスタリング

- グループの階層を前提とせずにクラスタリングを行う
 - 代表的な手法が「k-means」
 - ✓ Centroid (重心) -baseのグループ構成アプローチ
 - ✓ 一般的な分類問題に対してよく用いられる
 - 分割・グレーピングの妥当性を評価する何らかの数値的基準が必要となる
 - ✓ within distance, between distance
 - 非階層的クラスタリングに分類される手法を用いた場合も、同一アルゴリズムの再帰的な応用により、階層構造を持たせることは可能
 - ✓ 階層的、非階層的などのクラスタリング手法の分類は、便宜的・相対的なもの
 - 確率分布に基づくクラスタリングなども、分類としては非階層的クラスタリングの手法とみなせる



グループ分割はどのように行うのか： k-meansの場合

- 非階層的クラスター分析の基本アプローチ： 異なる性質の個体が混ざった集団から似た性質を持つ個体を集め、グルーピングする
 - 階層的な構造を持たない
 - 予めクラスター数を決めて分割処理を始める（排他的な部分集合、グループ間のオーバーラップなし）
 - 階層的なクラスタリング手法（デンドログラム）と異なり、サンプル数が大きくても有効に働く
- k-meansでは、グループの中心（重心）を求めて、各個体を一番近い重心のグループに割り当てていく
 - 重心の計算、グループ割り当てに関する繰り返し計算が必要（収束判定）

K-means法の考え方

- 階層構造を前提としない分割最適化
- k 個の分割に関する評価関数 L

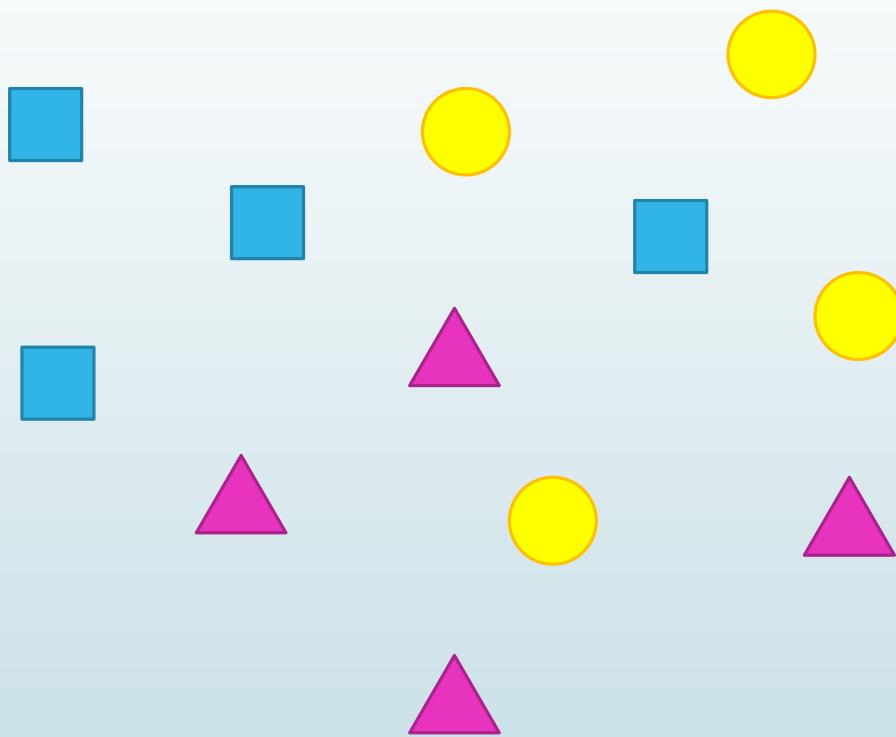
$$L(C) = \sum_{i=1}^k \sum_{x_i \in C_i} d(x, C_i)^2$$

- ✓ クラスタの重心点 C_i をクラスタの「代表」（クラスターの平均）とみなす
- ✓ この評価関数を最小化するような分割を求める

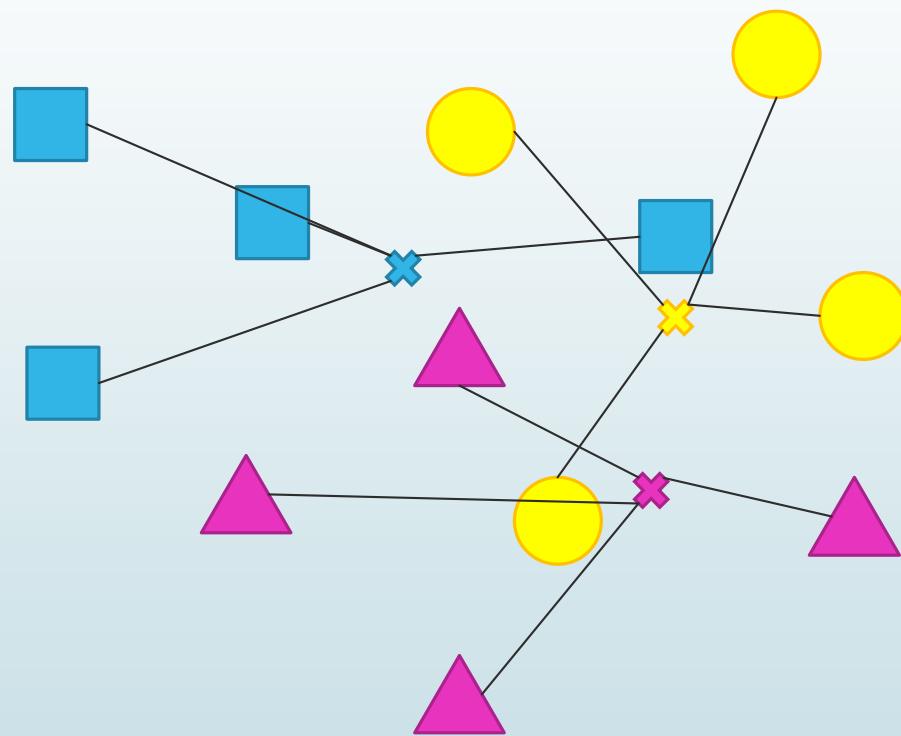
K-means法のアルゴリズム

1. 初期状態のセット：
グループ数 k を決めて、各個体をランダムに k 個のグループに割り当てる（均等配分）
 2. 各グループの重心を計算する
 3. 各要素を一番近い重心のグループに移動する（グループの重心が変わる）
 4. 各グループの重心を更新
 5. 重心の移動がなければクラスタリング終了、変化あれば 3 に戻って計算を続ける
-
- 明らかに初期状態に依存する結果・・・初期状態を何度も変えて結果の平均をとるなどの対処が必要

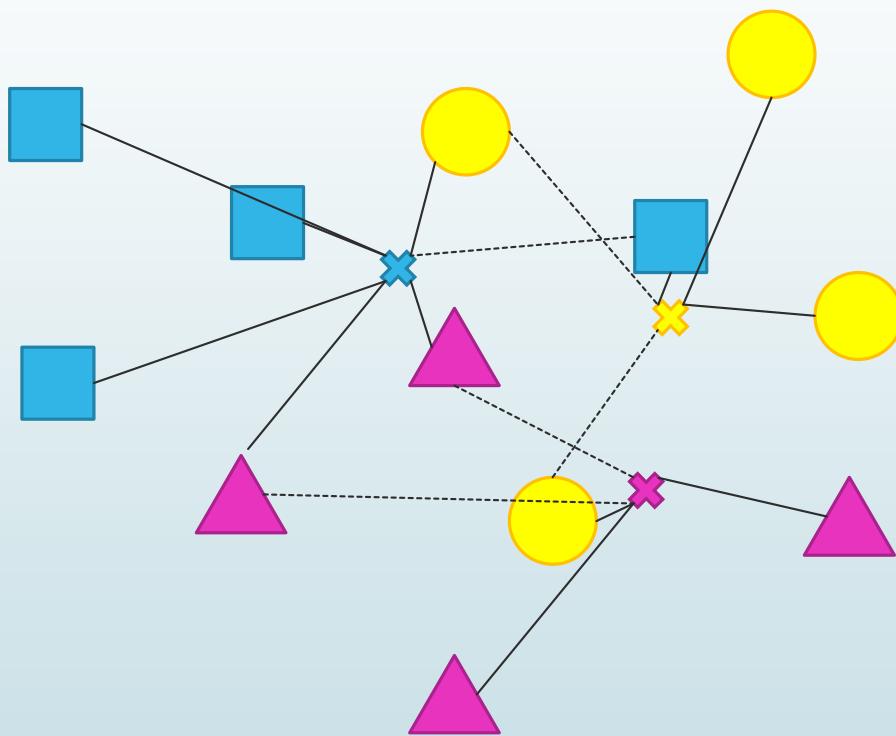
K-means法によるクラスタリング（1）



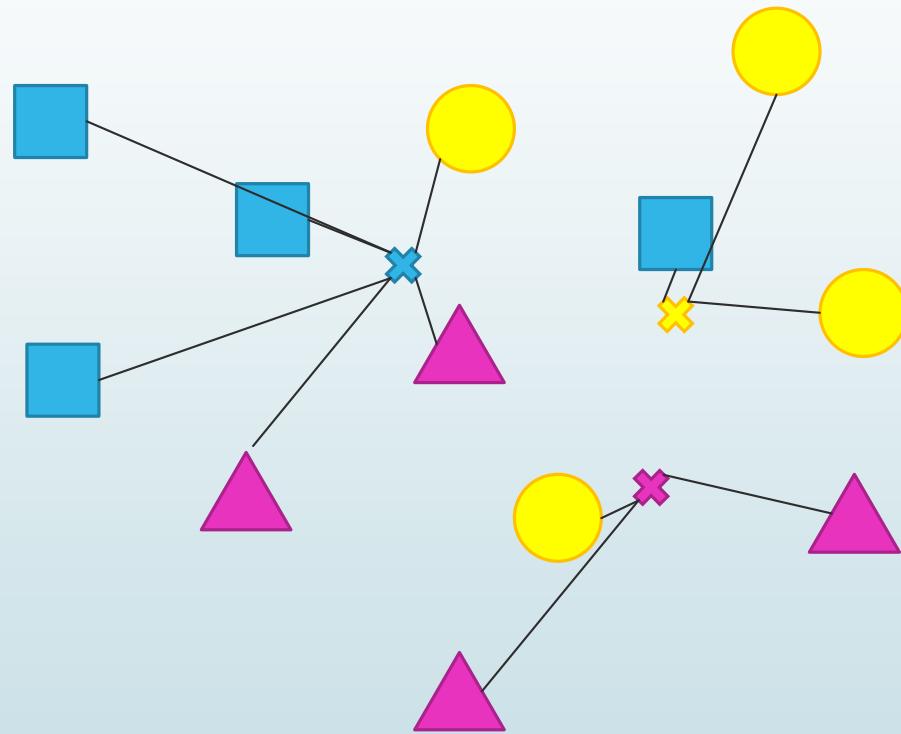
K-means法によるクラスタリング（2）



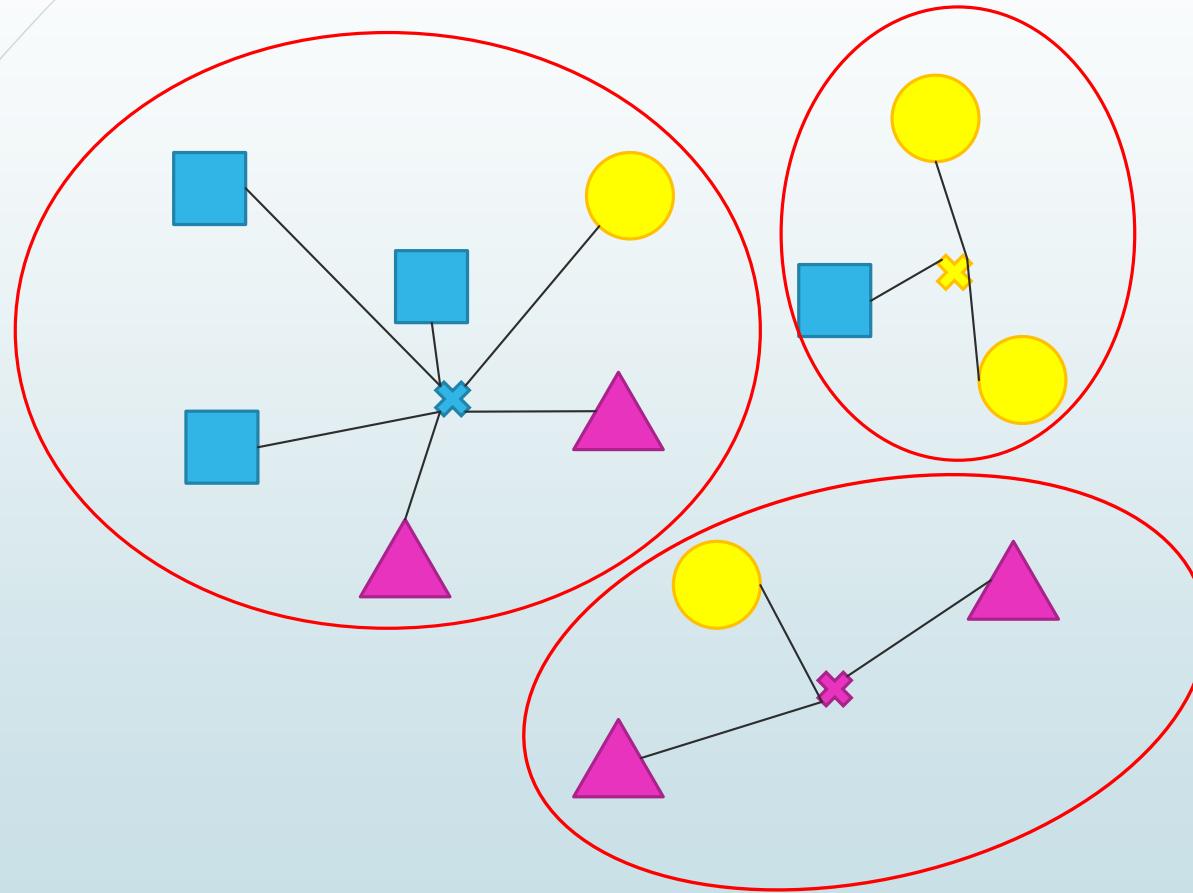
K-means法によるクラスタリング（3）



K-means法によるクラスタリング（4）

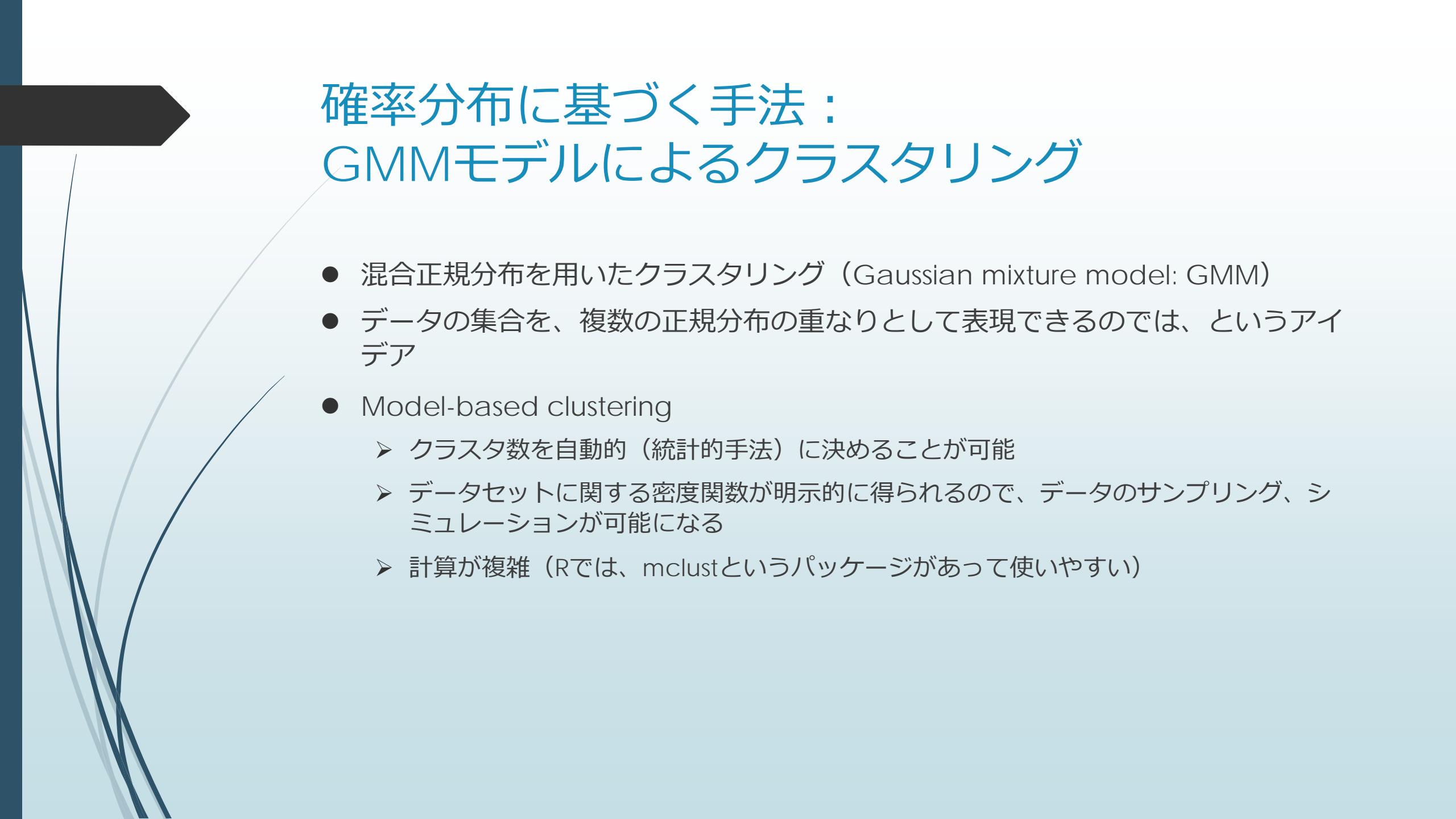


K-means法によるクラスタリング（5）



K-means法の問題点

- グループ数を予め決めておかないといけない・・・答えの一部を予め知っていることになる?
 - 統計的なアプローチ（クラスタ内距離二乗和、クラスタ間分散とクラスタ内分散の比＜F値＞など）、視覚的アプローチ（エルボー、シルエット図など）、いろいろな手法が提案されている
 - 実際には、分析者の主觀に依存して決めることになることが少なくない
 - 得られた結果で何が言えるか、の方が重要
- 初期状態に結果が依存してしまう・・・何度も計算をやり直して最適な結果を選ぶ、結果の平均をとるなどの対応が必要
 - グループの個体数に極端な差がある場合などでも、うまく動作しないことがある



確率分布に基づく手法： GMMモデルによるクラスタリング

- 混合正規分布を用いたクラスタリング (Gaussian mixture model: GMM)
- データの集合を、複数の正規分布の重なりとして表現できるのでは、というアイデア
- Model-based clustering
 - クラスタ数を自動的（統計的手法）に決めることが可能
 - データセットに関する密度関数が明示的に得られるので、データのサンプリング、シミュレーションが可能になる
 - 計算が複雑（Rでは、mclustというパッケージがあって使いやすい）



どうして正規分布を使うのか？

中心極限定理

- 基本的な考え方：複数の個体が形成するグループを正規分布で表現する
- 中心極限定理： n 個の標本平均の確率分布は、 n が十分に大きければ平均 μ 、分散 $\frac{\sigma^2}{n}$ の正規分布で近似できる
 - データセットに関する真の分布は、正規分布ではないかもしれないが、ある程度のデータ量があれば、正規分布で表現しても統計学的に問題ない
 - データの数が 100 個以下など、非常に少ない場合、i.i.d.とは考えられない場合などでは、問題が生じる可能性がある

混合正規分布を用いたクラスタリング

混合正規分布の確率密度関数

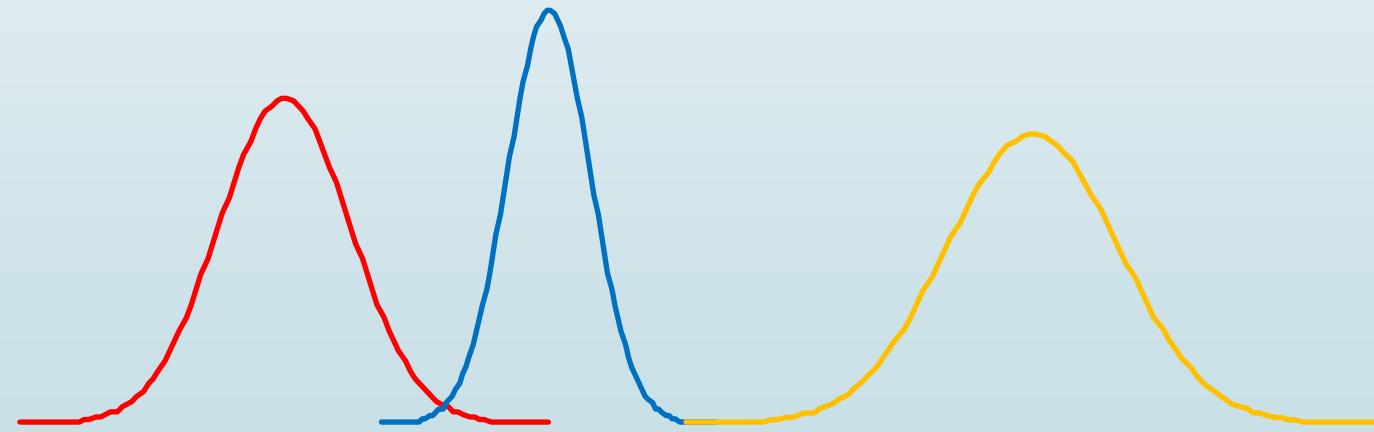
$$p(x) = \sum_{k=1}^K \pi_k N(x|\mu_k, \Sigma_k), 0 \leq \pi_k \leq 1, \sum_{k=1}^K \pi_k = 1$$
$$N(x|\mu_k, \Sigma_k) = \frac{1}{(2\pi)^{\frac{m}{2}} |\Sigma_k|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)\right)$$

mは変数の数

- ✓ π_k は混合比率（クラスタリングではk番目のグループに属する割合）
- ✓ 正規分布は、平均（ベクトル）と分散（分散共分散行列）の二種類のパラメータしか知らない
- ✓ データ x について、3種類のパラメータを推定する（最尤法）

正規分布の合成（線形結合）

- 例：平均、分散が異なる3つの正規分布で、3つのグループを表現する
- 実際のデータの確率密度は、3つの異なる正規分布の混合（線形結合）で計算される
- 多峰型の複雑な確率分布の表現に便利（様々な応用が可能）



EMアルゴリズムによるパラメータ推定

- 混合正規分布の密度関数は明示的にわかっているので、データ x から尤度関数を構成できる
 - 最尤法での推定については、データのラベリング（各データ要素がどのグループに属するかについての情報）ができていれば、パラメータ推定できる
 - 最初は、この情報は得られないので、適当な初期値を設定（k-meansと同様）し、尤度計算を始める
- EMアルゴリズムを用いた推定
 - Eステップ：設定したグルーピングの下での最大尤度をもたらす平均、分散共分散行列の推定
 - Mステップ：求めたパラメータの下で最大尤度をもたらすグルーピングの決定
 - 収束するまで反復計算

GMMのモデル推定、クラスタ数の決定

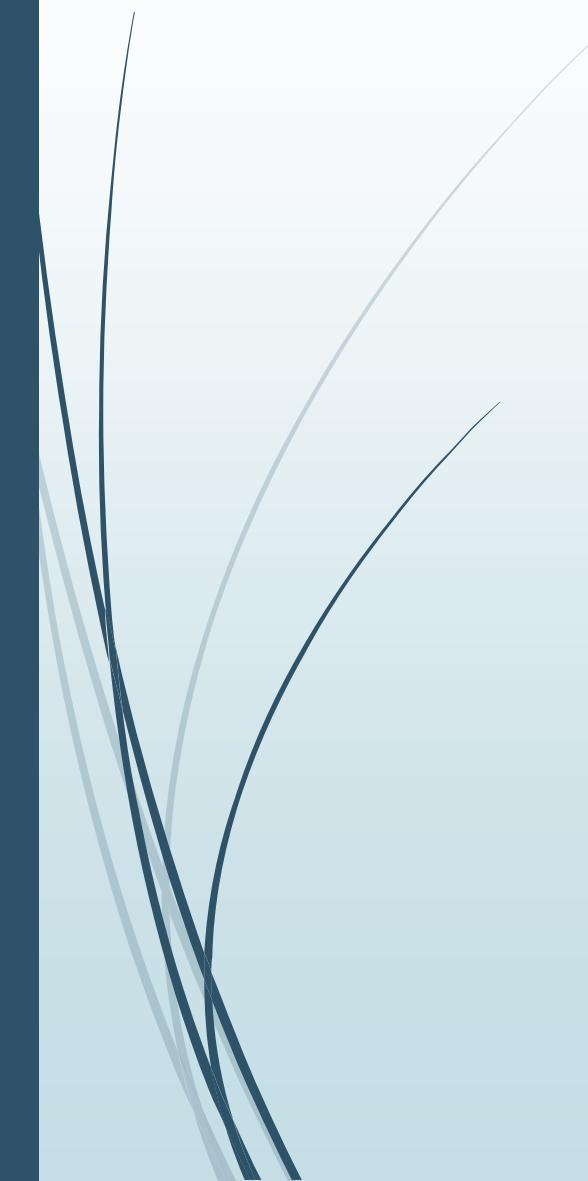
- モデルの設定に際しては、分散共分散行列 Σ_k に一定の制約を加えて計算負荷を軽くすることが多い
 - 予め得られているデータに関する知見を活かす
 - 要素が空間上に球状に分布している（等分散、無相関）、楕円状に分布（相関あり）、分散の大きさがグループで共通／異なる、などの情報をモデルに組み込んで、推定すべきパラメータの数をなるべく少なくする
 - 最も制約の緩いモデルでは答えが収束しない場合もある
- クラスタ数Kは、複数のモデルを推定し、BICなどの統計的な基準で妥当性の高いものを選択する
 - 完全に自動で決まるというわけではなく、目安として活用する

さらに進んだクラスタリング手法

- そのほか、より複雑な手法、多面的な応用例も多数存在する
 - 伝統的なk-meansの改良手法（クラスタ数の決定をある程度自動化）
 - 確率分布を用いた手法では、DP（Dirichlet process mixture model）なども用いられる
 - ネットワークのクラスタリングなども様々な研究が進められている
- いずれの手法でも共通していることは、クラスタリングの結果は、それ自体で答えを示しているものではなく、そこから分析者が何らかの知見を得て、新たな視点を提示する、ということ
 - 教師あり学習のアプローチとは大きく異なる
- クラスタリング結果を別の分析と組み合わせて、説得力を増す、分析の厚みを出す、というのが望ましい方向性



Q&A



記述的データ解析手法（3）：

グラフの解析

（ネットワーク理論を応用した分析例の紹介）

担当：磯貝 孝

グラフ理論・ネットワーク理論

- グラフ理論：ノード（V<vertex>、頂点・ノード）の集合とエッジ（E<edge>、リンク・辺）の集合で構成されるグラフ（G<graph>）に関する理論

$$G = (V, E)$$

- リンクは、頂点同士を結ぶもの（直線、曲線）
 - リンクに向き（頂点Aから頂点Bに向かうなど）があれば、「有向グラフ」、向きがなければ、「無向グラフ」
 - 同一頂点を結ぶリンクを考える場合もある（表現したい対象の特性による）
 - リンクに重みを考えない場合（頂点がつながっているのかどうかのみ）は重みなしグラフ、重み（つながりの強さ）を考える場合は、「重み付き」グラフ
-
- データ分析上は、「ネットワーク理論」もグラフ理論とほぼ同義の言葉として使われることが少なくない



何を分析するのか？

- 対象となる「個々の主体」（頂点、ノード）の相互関係（リンク）に関心がある
- 個体情報をまとめてグループとしての全体傾向を探る、という分析の方向性とは異なる
 - 回帰モデルなどの場合、関心のある変数（消費関数における所得データなど）を特定して、従属変数（消費額）との関係性を示したい（モデルの構築）
 - グラフ理論は、個体間の関係（例：SNS上の人々のつながり方、金融資産＜個別銘柄＞の収益率の相関）
- データマイニングにおけるグラフ理論の応用は、多数の個体の相互関係の整理・有益な情報の抽出・・・全体としてどのような構造を有しているか、重要なノードはどれなのか、など
 - 現実世界のデータをまずノードとリンクからなるグラフ（数値データ）として再現し、可視化や種々の演算を行って、情報を取り出す
 - 知見の獲得が主な目的（クラスタリングや各種データマイニングの手法を積極的に応用する）

分析のアプローチ

- 理論モデルに重点を置く分析
 - ランダムネットワーク理論など、ネットワークの発生・拡張の過程を数学的な理論モデルで再現する (Erdős-Rényi、Barabasi Albertモデルなど)
- ネットワークの特性に注目する分析
 - スケールフリー、スモールワールドなど、異なるデータの間で共通してみられる特徴を発見し、どうしてそういう構造を持ち得るのか、を考える
- ノードの探索、最短経路の発見など、検索・最適化への応用
- 多数の個体をグルーピングし、グループによる違いを観察する（今回のテーマ）
 - ネットワーク上のクラスタリングを行い、大規模なネットワークを複数のサブネットワークに分割する
 - 共同体検出<community detection>とも呼ばれる
 - グラフ分割を行う様々なアルゴリズムが存在する

グラフ・ネットワークの「隣接行列」表現 (adjacency matrix representation)

- ネットワークの情報は、隣接行列に変換・保持できる
 - ネットワークの行列表現：「隣接行列」による数値データ化
 - 相関ネットワーク：ノード（銘柄）、リンク（つながりの有無・強さ・方向）

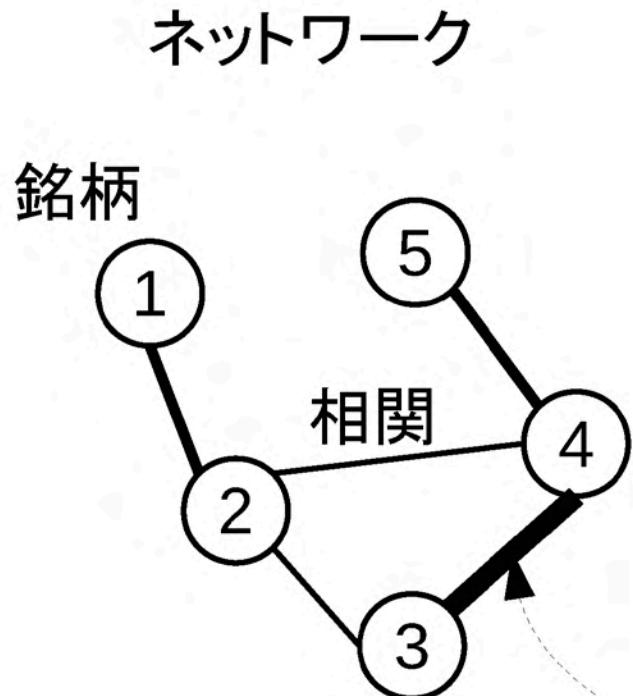
| | A | B | C | D |
|---|---|---|---|---|
| A | 0 | 1 | | |
| B | | 0 | | |
| C | 1 | | 0 | |
| D | | | 1 | 0 |

有向グラフ（上下非対称の隣接行列）

| | A | B | C | D |
|---|---|---|---|---|
| A | 0 | 1 | | |
| B | 1 | 0 | | |
| C | | | 0 | 1 |
| D | | | 1 | 0 |

無向グラフ（上下対称の隣接行列）

例：株価の相関構造のネットワーク表現 (重み付きのネットワーク)



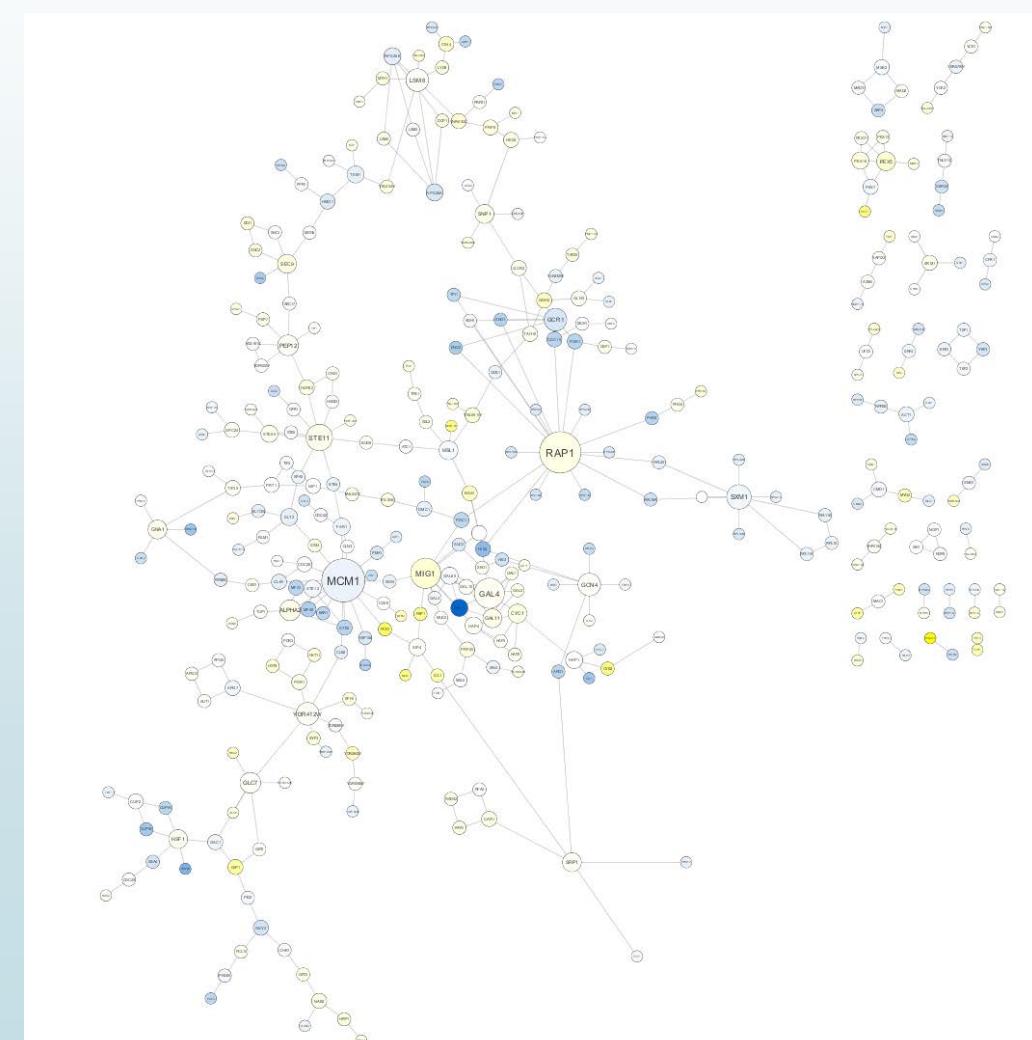
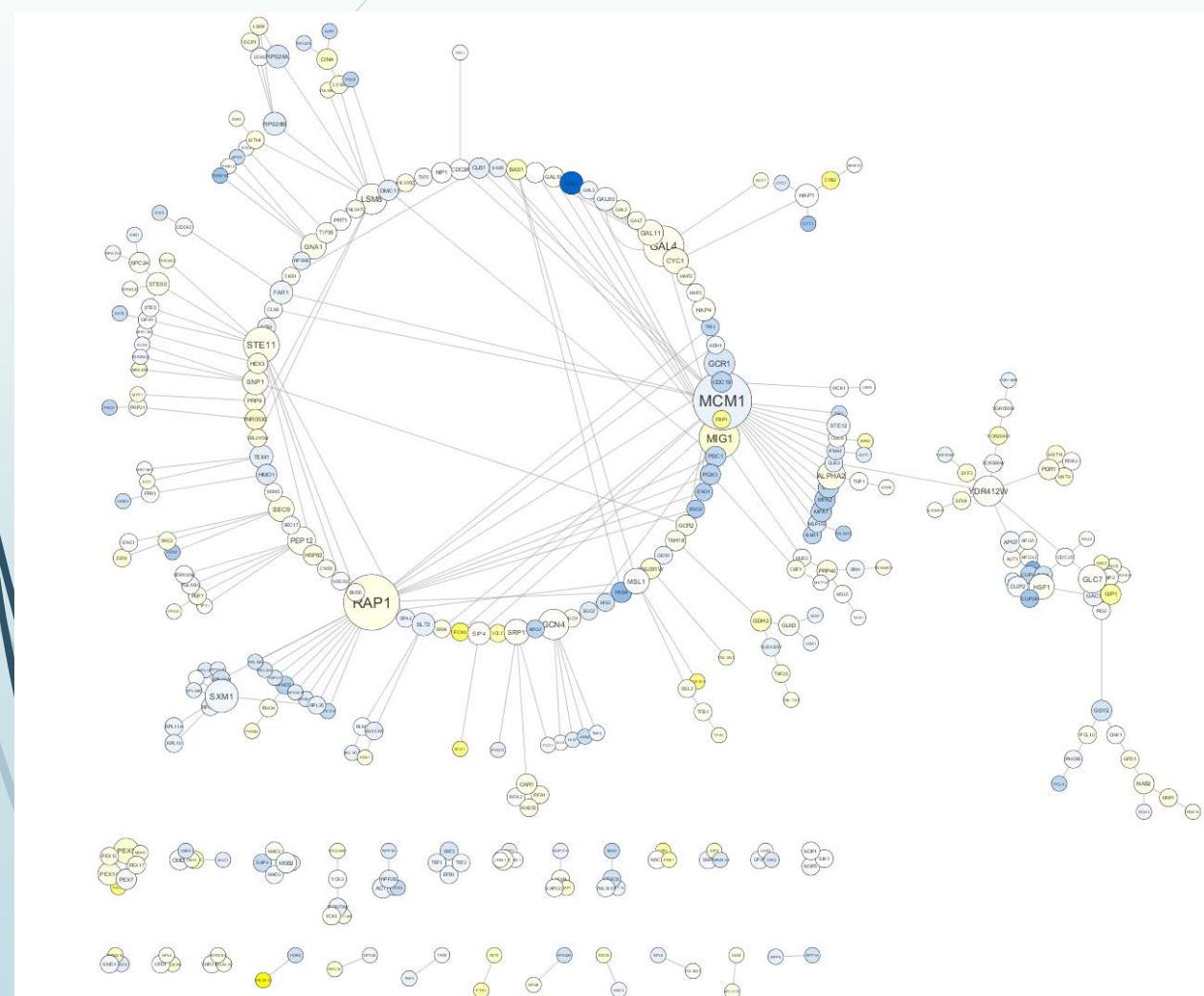
隣接行列
(Adjacency matrix)

| | 1 | 2 | 3 | 4 | 5 |
|---|-----|-----|-----|-----|-----|
| 1 | 0 | 0.6 | 0 | 0 | 0 |
| 2 | 0.6 | 0 | 0.2 | 0.3 | 0 |
| 3 | 0 | 0.2 | 0 | 0.7 | 0 |
| 4 | 0 | 0.3 | 0.7 | 0 | 0.5 |
| 5 | 0 | 0 | 0 | 0.5 | 0 |

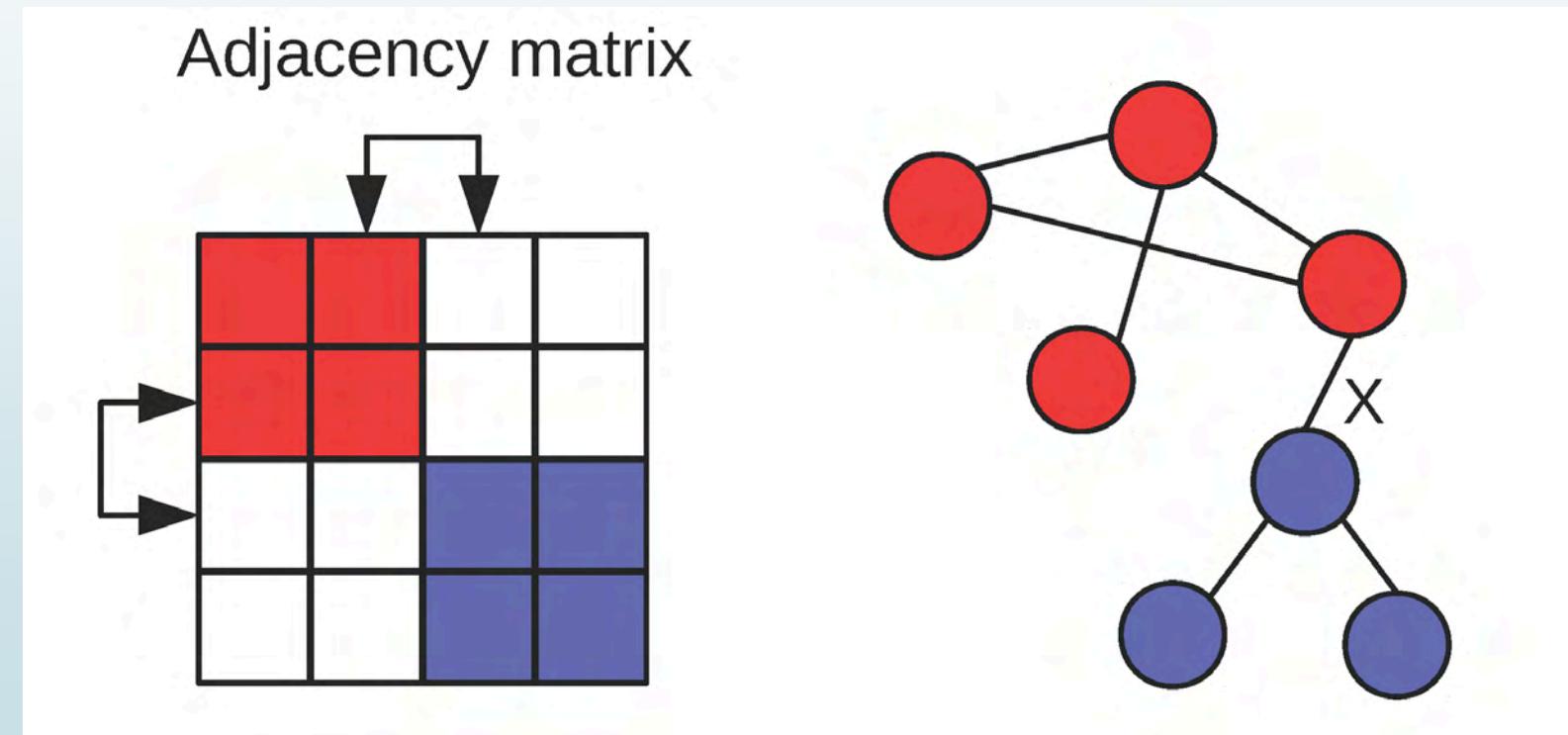
ネットワークの可視化

- 個体の連結状況は隣接行列で完全に表現できるが、ネットワークのサイズが大きくなると、全体像の理解が困難になる
- ネットワーク分析では、情報整理、全体構成の直感的な理解などの目的で、視覚化を行うことが多い（ほぼ必須）
 - R(igraph), Python(Matplotlib), Cytoscapeなどネットワークを視覚化できるプログラムでは、元となるデータ+レイアウトアルゴリズムの選択で、具体的な図の表示（視覚化）を行う
 - Spring-embedded、Kamada – Kawai、Fruchterman – Reingoldなど、非常に多くのレイアウトアルゴリズムが存在する
 - 全く同一の隣接行列でも、レイアウトアルゴリズムが異なれば、見た目は全くことなる・・・どのレイアウトを選ぶか？（見た目、好み、センス？）

同一データに別のレイアウトを用いた例



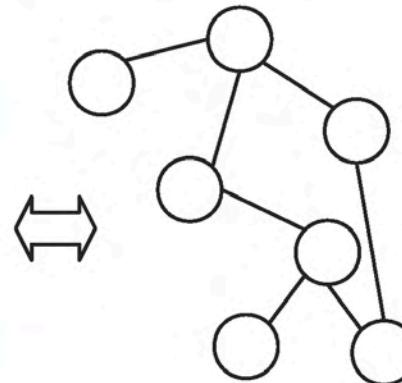
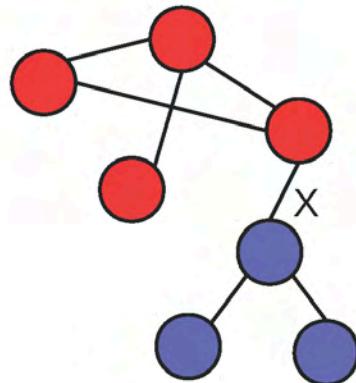
ネットワークの分割・グルーピング (隣接行列の並べ替え)



ネットワークの分割

- ネットワークの中の特定のリンク (X) をカットして複数のサブネットワーク (グループ) を生成する (共同体検出)
 - 最適な分割を見つけるアルゴリズムと評価基準が必要
 - 分割の評価基準としてModularity (Q :分割の良さを示す数値) などの数値指標が用いられる (Q を最大化するような分割を見つける) . . . グラフスペクトルクラスタリングの応用
- ✓ 隣接行列をブロック化していくことと同値
 - 確率的ブロックモデルなどの確率的アプローチ (ノードの結合確率をモデル化) など、ほかにも様々な分割手法が存在する

Modularity の考え方



| | | | | |
|---|---|---|---|---|
| 0 | | 0 | | |
| | 0 | 1 | | |
| 0 | 1 | 0 | | |
| | | | 0 | |
| | | | | 0 |

| | | | | |
|---|---|---|--|---|
| 0 | | | | |
| | 0 | 0 | | |
| 0 | 0 | 1 | | |
| | 1 | 0 | | |
| | | | | 0 |

Adjacency matrix

Distance measure = **Modularity (Q)**

Qの定義

$$Q = \frac{1}{2W} \sum_{i=1}^n \sum_{j=1}^n B_{ij} \delta(C_i, C_j)$$

$$w_i = \sum_{j=1}^n w_{ij}, 2W = \sum_{i=1}^n w_i, B_{ij} = (A_{ij} - \frac{w_i w_j}{2W})$$

A_{ij} :隣接行列, w_i, w_j : ノード i, j の重みの和,
 $\delta(\)$: if $C_i = C_j$ (同一クラス) 1, それ以外 0, $-1 < Q < 1$

- Qの最大化 : 最適分割ではなく、あくまでも近似解しか得られない
- Modularity の定義には様々なバリエーションが提案されている

隣接行列のブロック化 (Qがより大きくなるように並び替える)

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | | |
| 2 | | | | | | | | | |
| 3 | | | | | | | | | |
| 4 | | | | | | | | | |
| 5 | | | | | | | | | |
| 6 | | | | | | | | | |
| 7 | | | | | | | | | |
| 8 | | | | | | | | | |
| 9 | | | | | | | | | |

Q_0

| | A | | | B | | | C | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1 | 9 | 6 | 4 | 7 | 3 | 5 | 8 | 2 |
| 1 | | | | | | | | | |
| A | | | | | | | | | |
| 9 | | | | | | | | | |
| 6 | | | | | | | | | |
| 4 | | | | | | | | | |
| B | | | | | | | | | |
| 7 | | | | | | | | | |
| 3 | | | | | | | | | |
| 5 | | | | | | | | | |
| C | | | | | | | | | |
| 8 | | | | | | | | | |
| 2 | | | | | | | | | |

$Q_1 (> Q_0)$

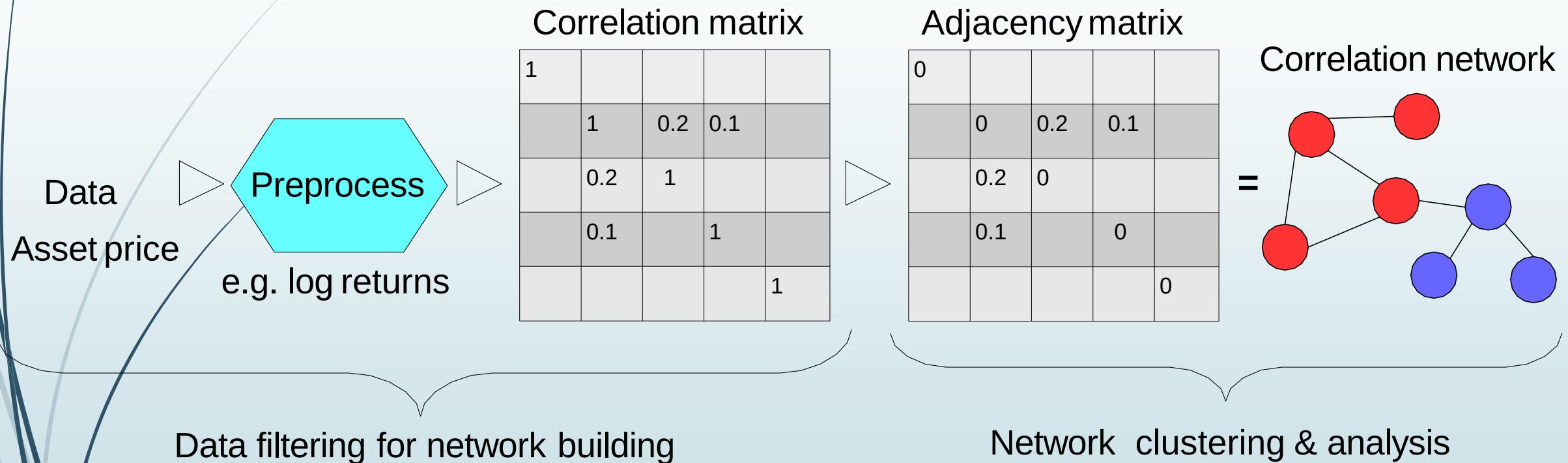
確率的ブロックモデル（ベイジアンアプローチ）

- Qの計算による分割を用いない別の分割手法
- 確率的ブロックモデルによるネットワーク分割（クラスタリング）の考え方
 - 潜在的なグループの存在を仮定して、グループラベル（ベクトル:例「1,2,3,...」）の分布を考える（事前分布の想定）
 - 既にネットワークの情報（隣接行列）は得られているので、メンバーの相関関係に関する情報をデータとして、ベイズの定理を応用して、グループラベルの事後分布を求める
 - 更新されたグループラベルをクラスタ分類のラベルとして採用する
 - グループ数そのものも推定すべきパラメータにできるので、いくつのグループに分割するのが適当か、という答えも確率的に得られる
 - 隣接行列上のブロックを確率的な推論で求める感じなので確率的ブロックモデル？
 - 階層的なクラスタリングなど、複雑な設定も可能（統計パッケージも存在します）

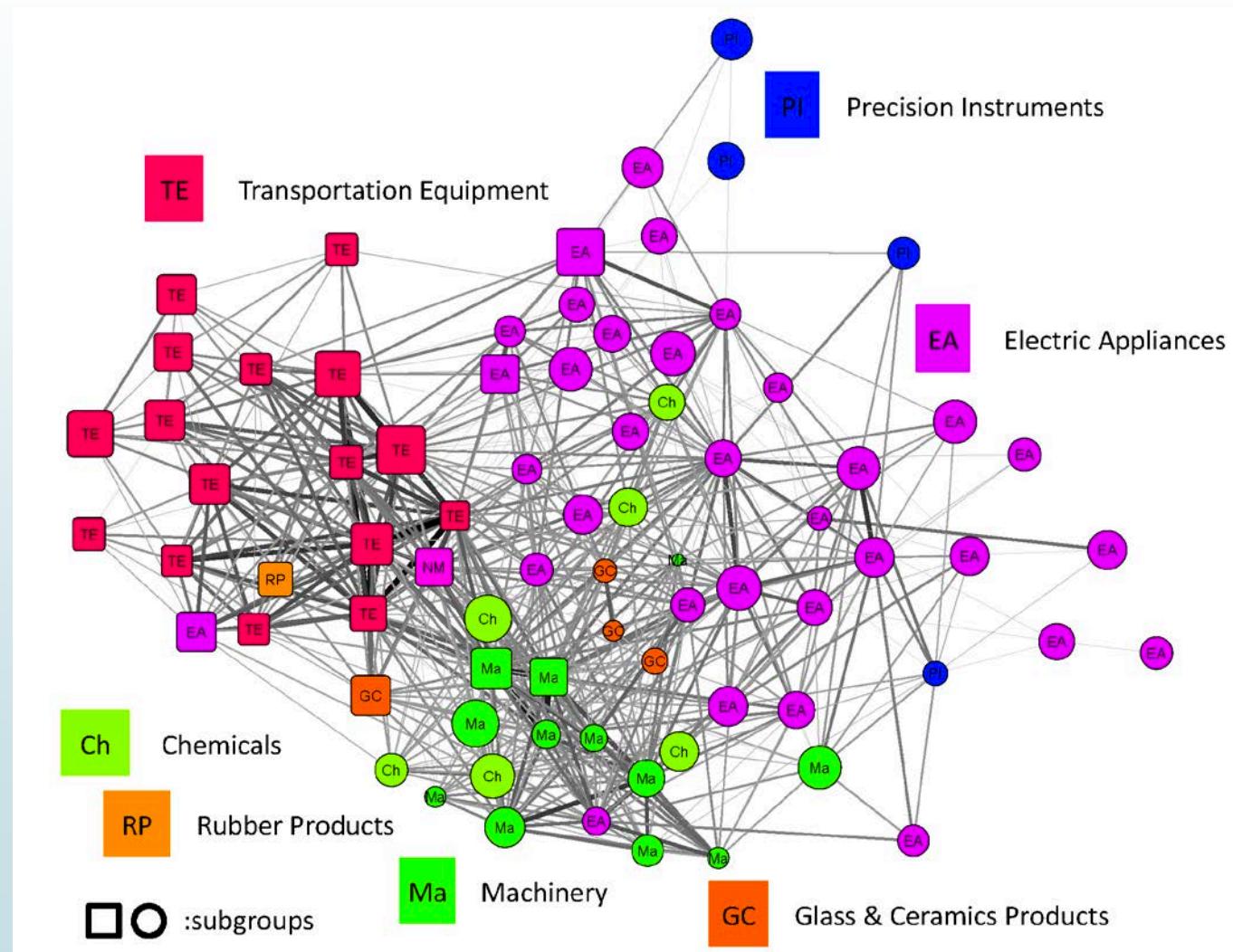
ネットワーク分析の具体例 (金融分野での応用)

- 資産収益率の変動（例：株価の価格変動＜日次対数収益率＞）に銘柄間でどのような相関構造が存在するのか、はポートフォリオ投資・リスク管理の観点から重要な情報
 - 株式の場合、銘柄数が多いので相関構造の全体像を把握するのは結構たいへん
 - 正確な相関の計算自体に時系列モデルを用いた複雑な計算が必要
- 株価の相関をネットワーク分析してみる
 - 銘柄をノードとみて、リンクの太さを相間に例える・・・相関ネットワーク表現
 - ネットワークを分割して、グルーピングしてみる（分析対象の要約、次元圧縮）
 - 相関構造の時間変動も観察してみた

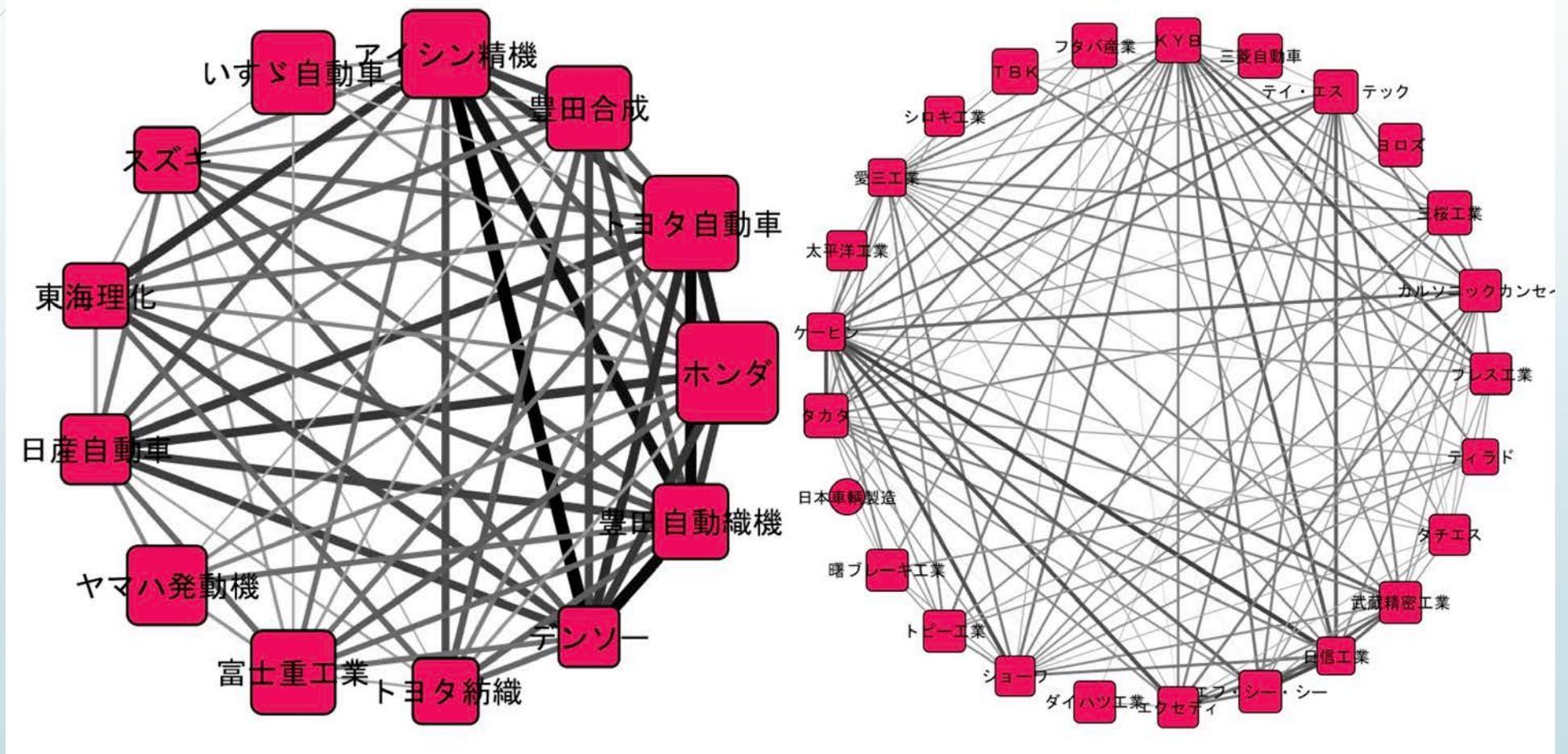
金融データをネットワークデータに変換して分析する



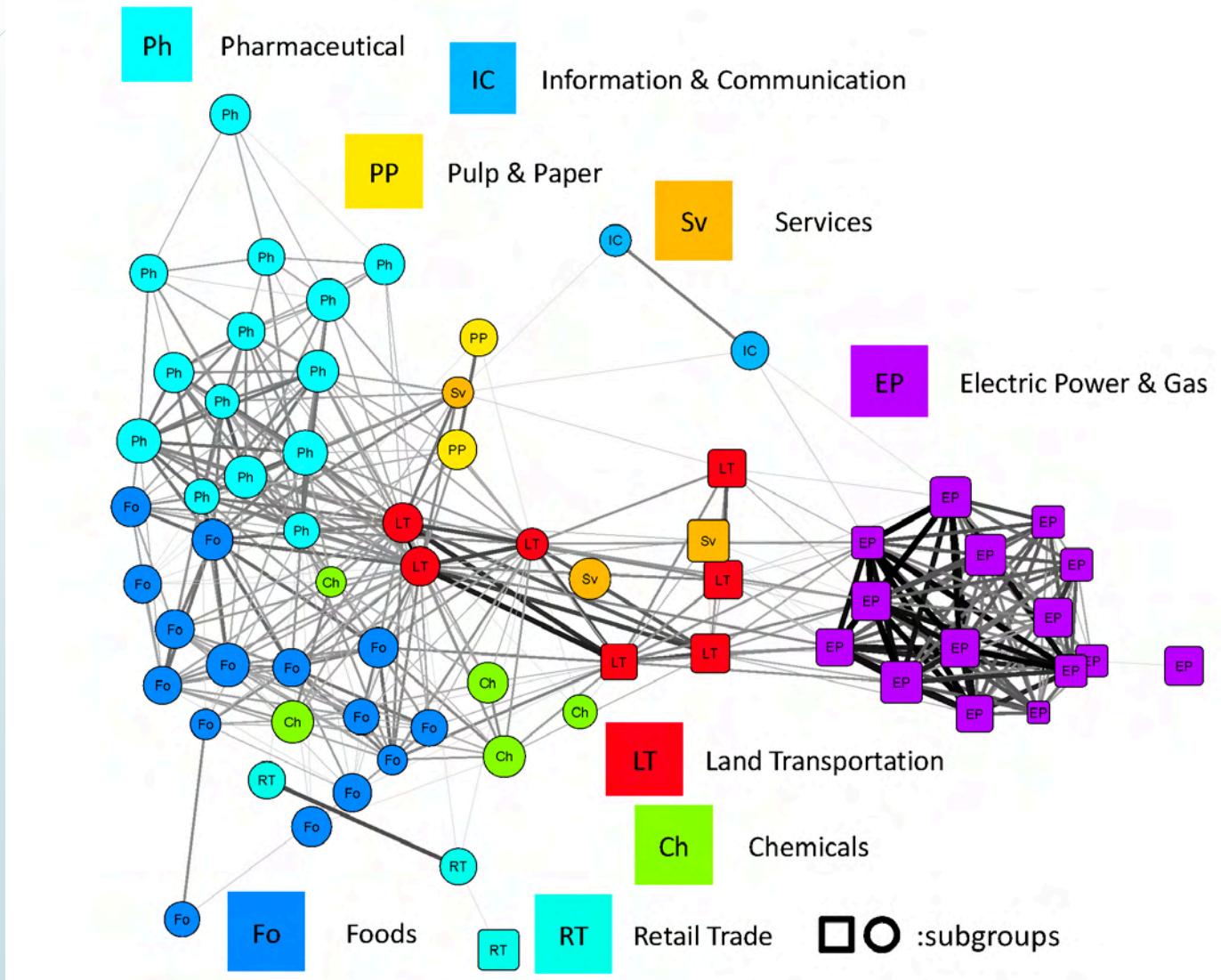
発見されたサブグループの一つ



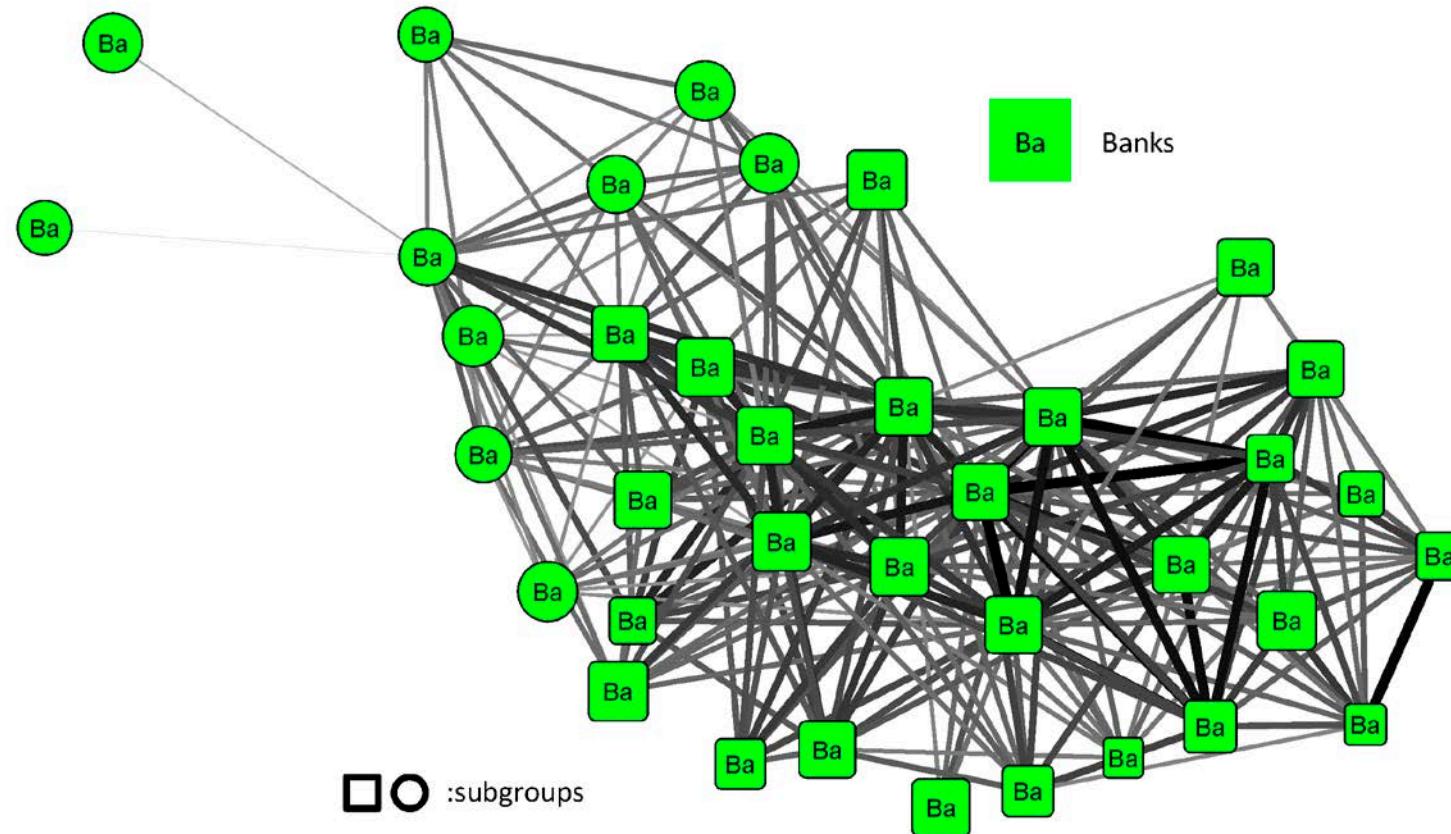
輸送用機器に関する二つのカテゴリー



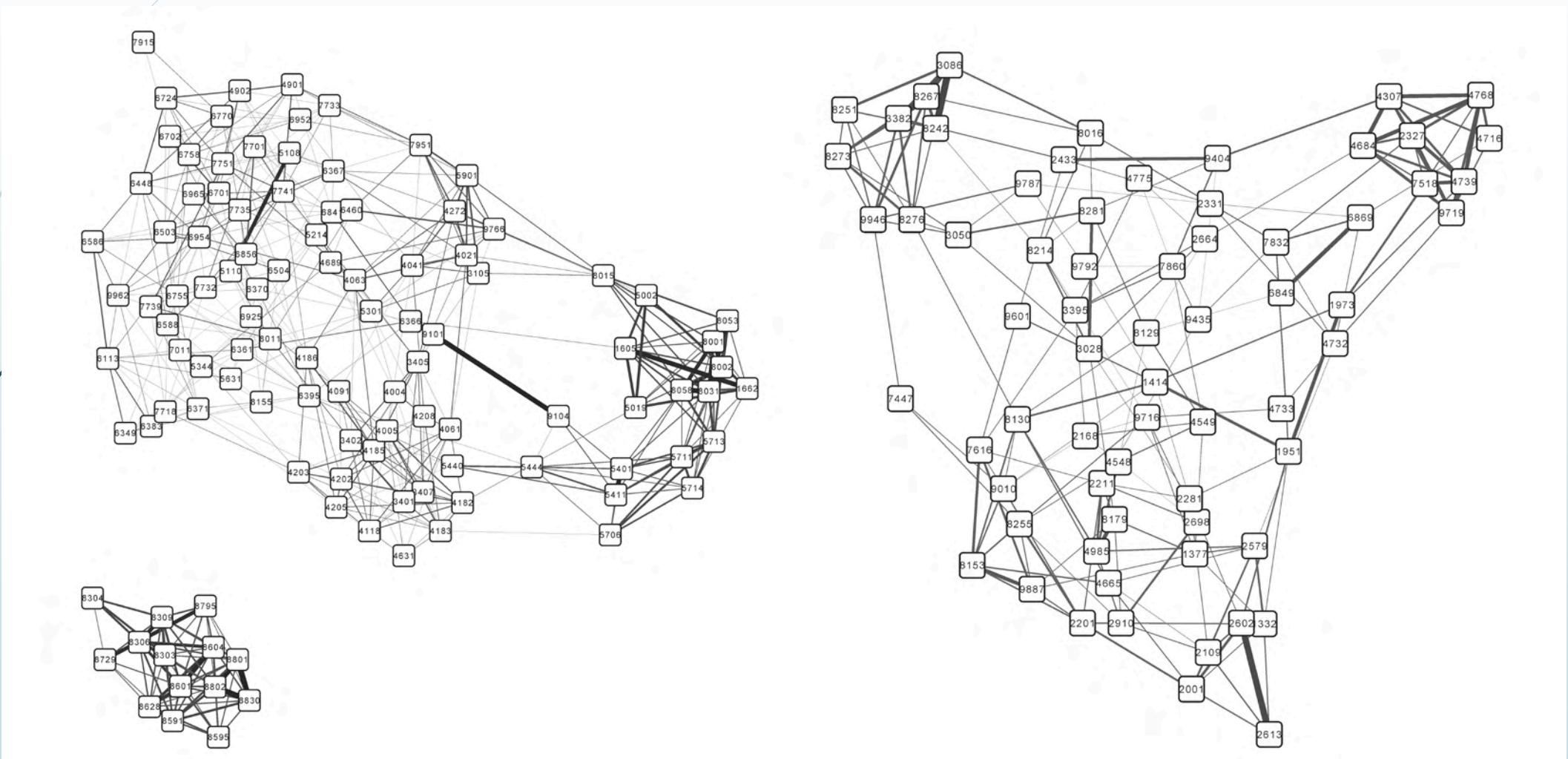
別のサブグループ



非常にまとまりの強いグループの例 (地域金融機関)



相関の代わりに偏相関を用いた例



ネットワーク・トポロジー (形状の特徴に関する量的指標の例)

Density: $D(\mathbf{A})$ – ネットワークの密度（ノード間の全体的なリンクの強さ）

$$D(\mathbf{A}) = \frac{\sum_i \sum_{j>i} \mathbf{A}_{ij}}{n(n-1)/2} = \frac{\text{mean}(\mathbf{k})}{n-1}$$

Connectivity (ノードの次数) $k_i = \sum_{j \neq i} \mathbf{A}_{ij}$; \mathbf{k} is a vector of connectivity

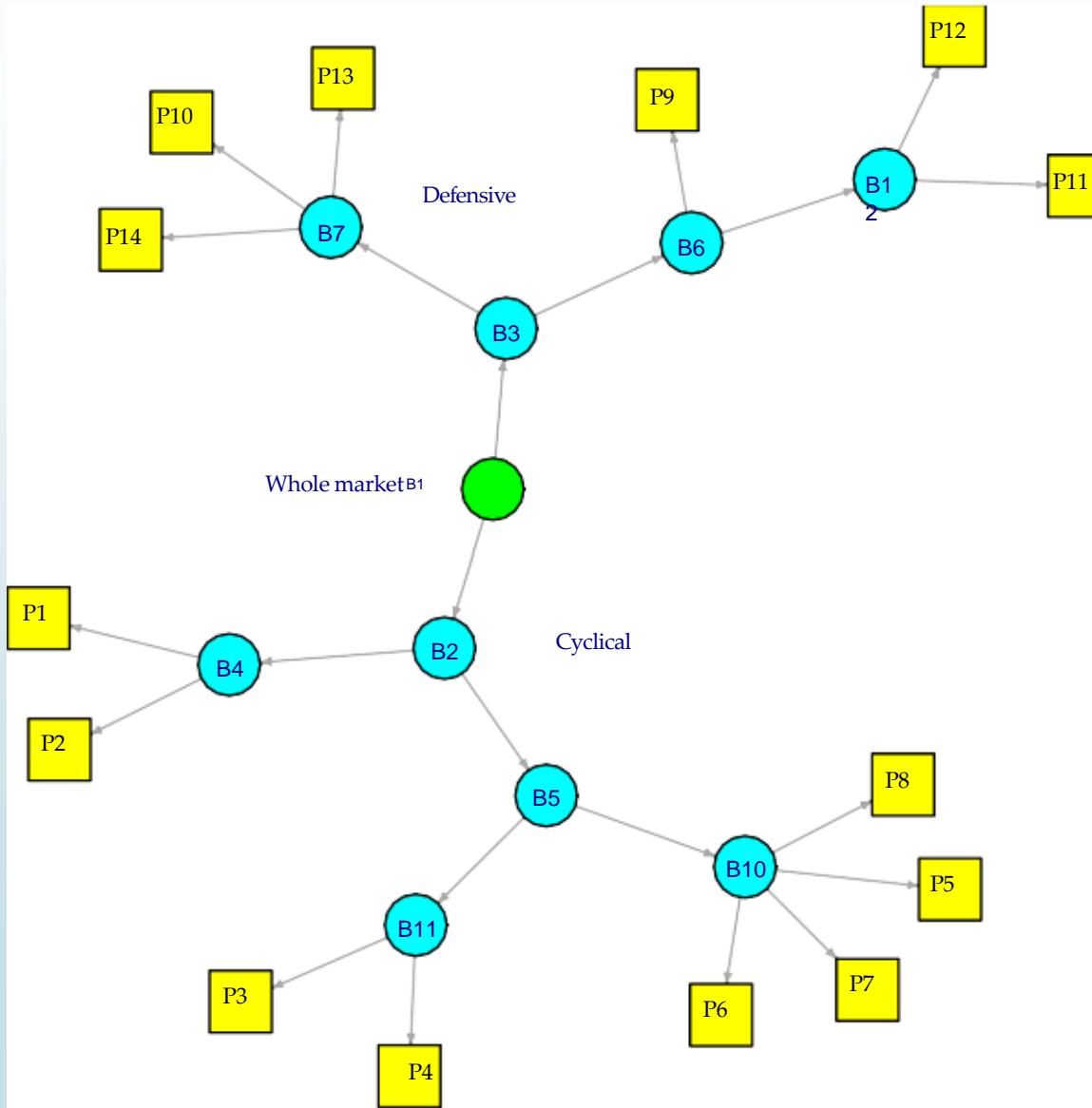
Centralization: $C(\mathbf{A})$ – ネットワークの中心性（すべてのノードがつながる中心的なノードが存在する度合い）

$$C(\mathbf{A}) = \frac{n}{n-2} \left(\frac{\max(\mathbf{k})}{n-1} - \frac{\text{mean}(\mathbf{k})}{n-1} \right) \approx \frac{\max(\mathbf{k})}{n} - D(\mathbf{A})$$

Heterogeneity: $H(\mathbf{A})$ – ネットワークの異質性（ノード間のリンクの強さがどの程度違っているか）

$$H(\mathbf{A}) = \frac{\sqrt{\text{var}(\mathbf{k})}}{\text{mean}(\mathbf{k})} = \sqrt{\frac{n \sum_i k_i^2}{(\sum_i k_i)^2} - 1}$$

再帰的なクラスタリング

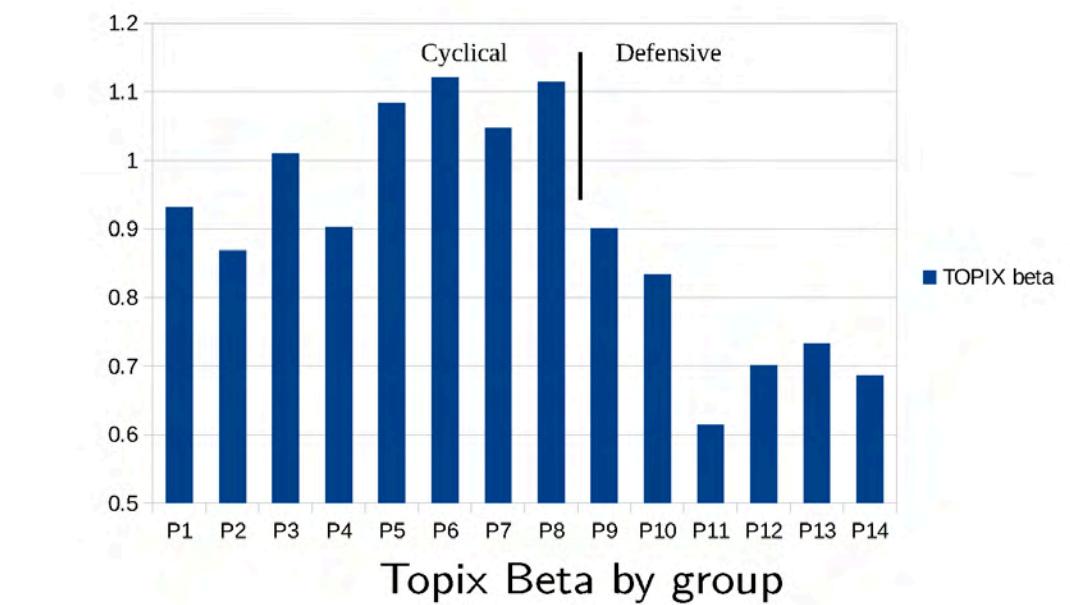
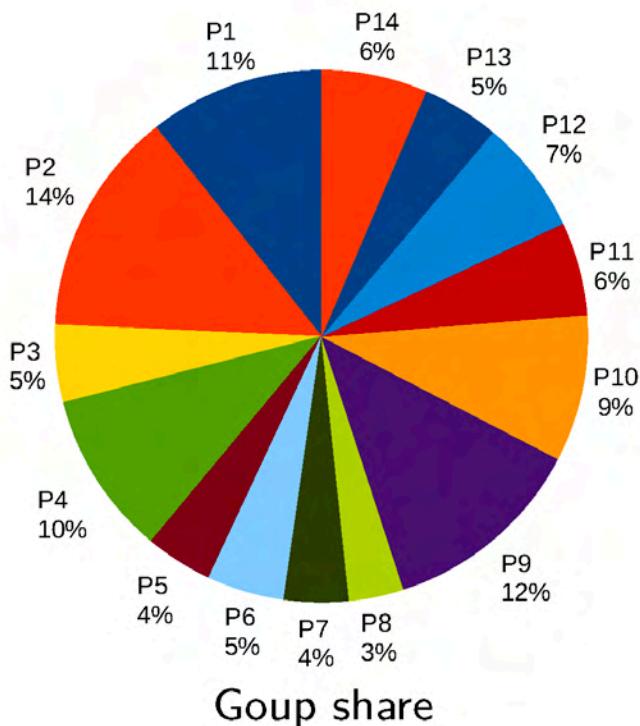


分析結果（1：業種情報との対比）

| | | TOPIX beta | Exchange rate Beta (Dollar/Yen) | Company size (market Capitalization Index) | Overseas sales Ratio | Sector info | | | | |
|-----------|-----|------------|------------------------------------|---|----------------------|-------------|---------------------------------------|----|---------------------------------------|-----------------------------------|
| Cyclical | P1 | (141) | 0.93 | 0.93 | 35.3 | 47.1 | Electric appliances | 17 | Service | 12 Machinery 10 |
| | P2 | (181) | 0.87 | 0.92 | 35.4 | 36.6 | Construction | 18 | Machinery Other financial business | 13 Wholesale trade 10 |
| | P3 | (62) | 1.01 | 1.16 | 63.5 | 34.6 | Securities | 21 | Business | 13 Real Estate 11 |
| | P4 | (132) | 0.90 | 0.95 | 43.0 | 42.9 | Electric appliances Transportation | 20 | Chemicals | 18 Wholesale trade 16 |
| | P5 | (54) | 1.08 | 1.19 | 72.9 | 61.6 | Equipment | 39 | Electric appliances | 20 Machinery 11 |
| | P6 | (62) | 1.12 | 1.15 | 81.3 | 66.5 | Electric appliances | 47 | Machinery | 23 Chemicals 10 |
| | P7 | (52) | 1.05 | 1.10 | 79.1 | 50.8 | Chemicals | 19 | Iron & Steel Transportation | 17 Nonferrous metals 13 |
| | P8 | (44) | 1.11 | 1.21 | 92.6 | 61.9 | Electric appliances | 30 | Equipment | 18 Chemicals 9 |
| Defensive | P9 | (164) | 0.90 | 0.99 | 64.6 | 38.5 | Banks Information & Communication | 26 | Construction | 11 Chemicals 9 |
| | P10 | (118) | 0.83 | 0.89 | 81.5 | 31.5 | Foods | 17 | Retail trade | 15 Wholesale trade 14 |
| | P11 | (75) | 0.61 | 0.60 | 27.3 | 30.0 | Retail trade | 19 | Information & Communication | 17 Wholesale trade 17 |
| | P12 | (92) | 0.70 | 0.72 | 40.8 | 27.1 | Retail trade Electric power & Gas | 17 | Wholesale trade | 16 Foods 8 |
| | P13 | (62) | 0.73 | 0.71 | 75.6 | 25.5 | Pharmaceutical | 26 | Foods | 16 Information & Communication 15 |
| | P14 | (85) | 0.69 | 0.62 | 64.8 | 21.9 | Retail trade | 53 | Service | 19 Information & Communication 8 |

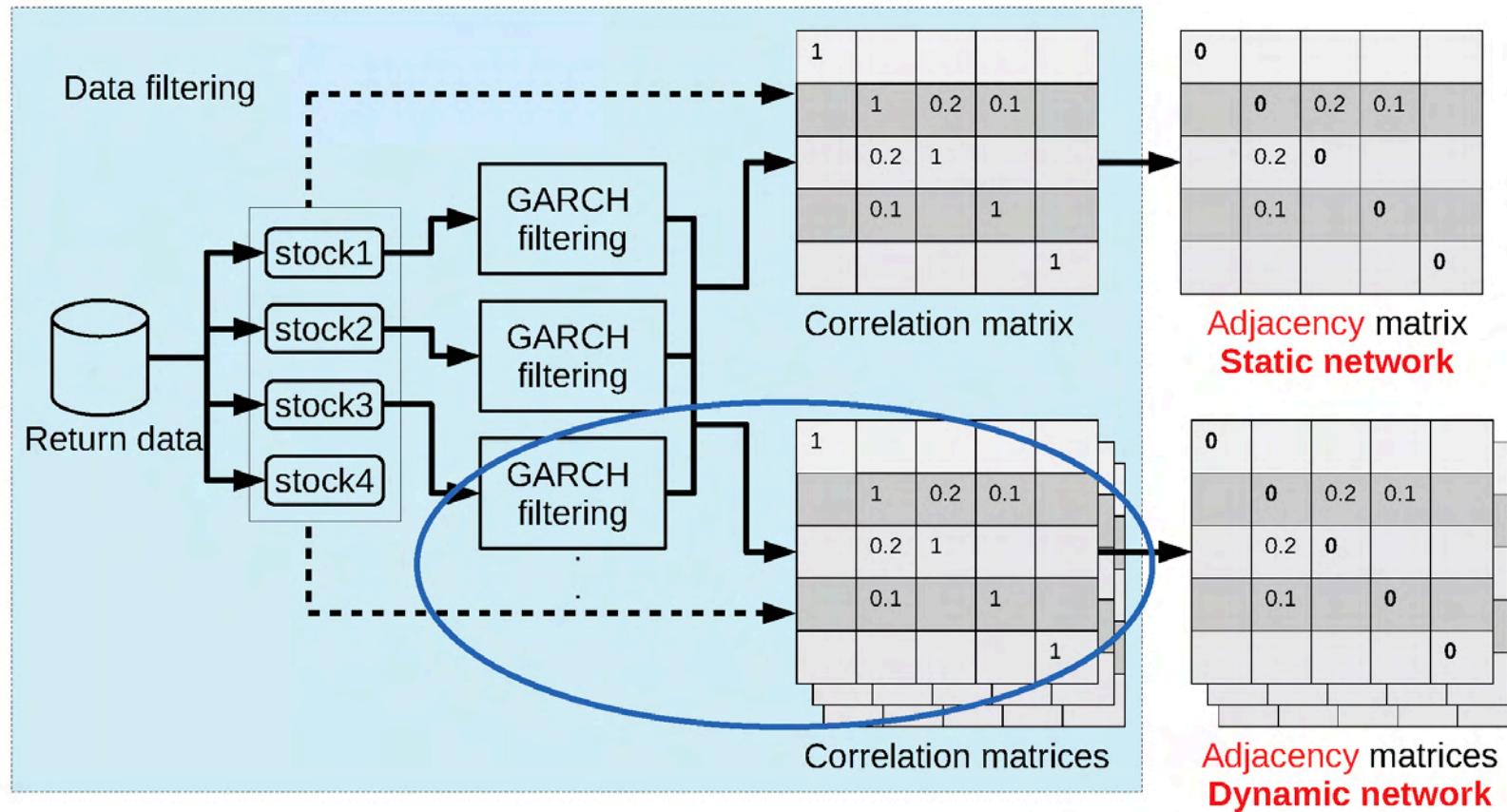
分析結果（2：ベータ値ほか）

| | TOPIX beta | Exchange rate beta (Dollar/Yen) | Company size (market capitalization Index) | Overseas sales ratio |
|-----------|------------|------------------------------------|--|----------------------|
| Cyclical | (728) | 1.01 | 1.08 | 62.89 |
| Defensive | (596) | 0.74 | 0.75 | 59.10 |



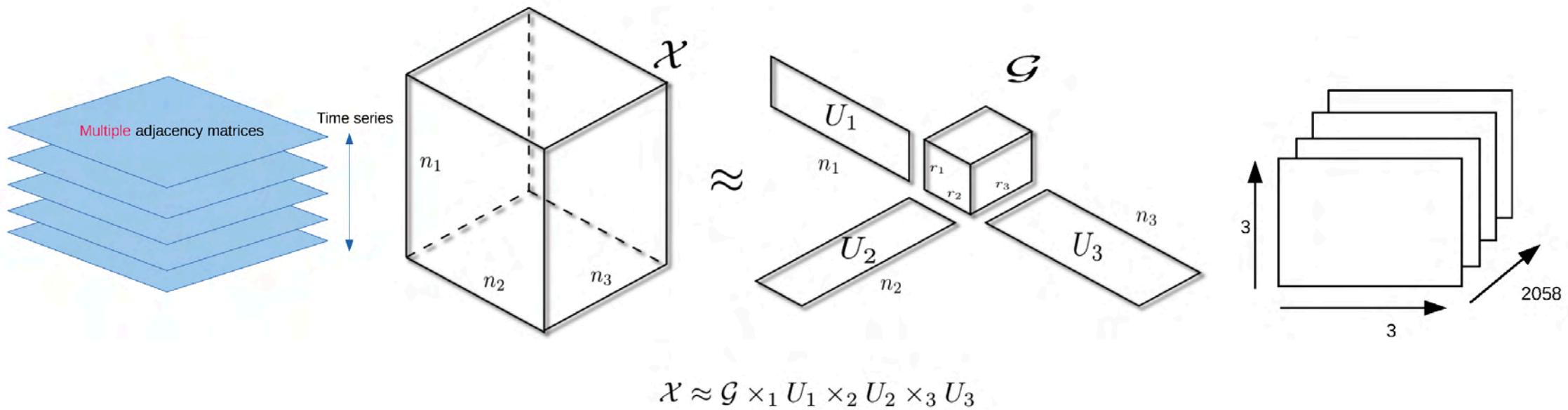
相関構造の動的変化の検出 (動的なネットワーク分析の応用)

Data processing flow for dynamic correlation network



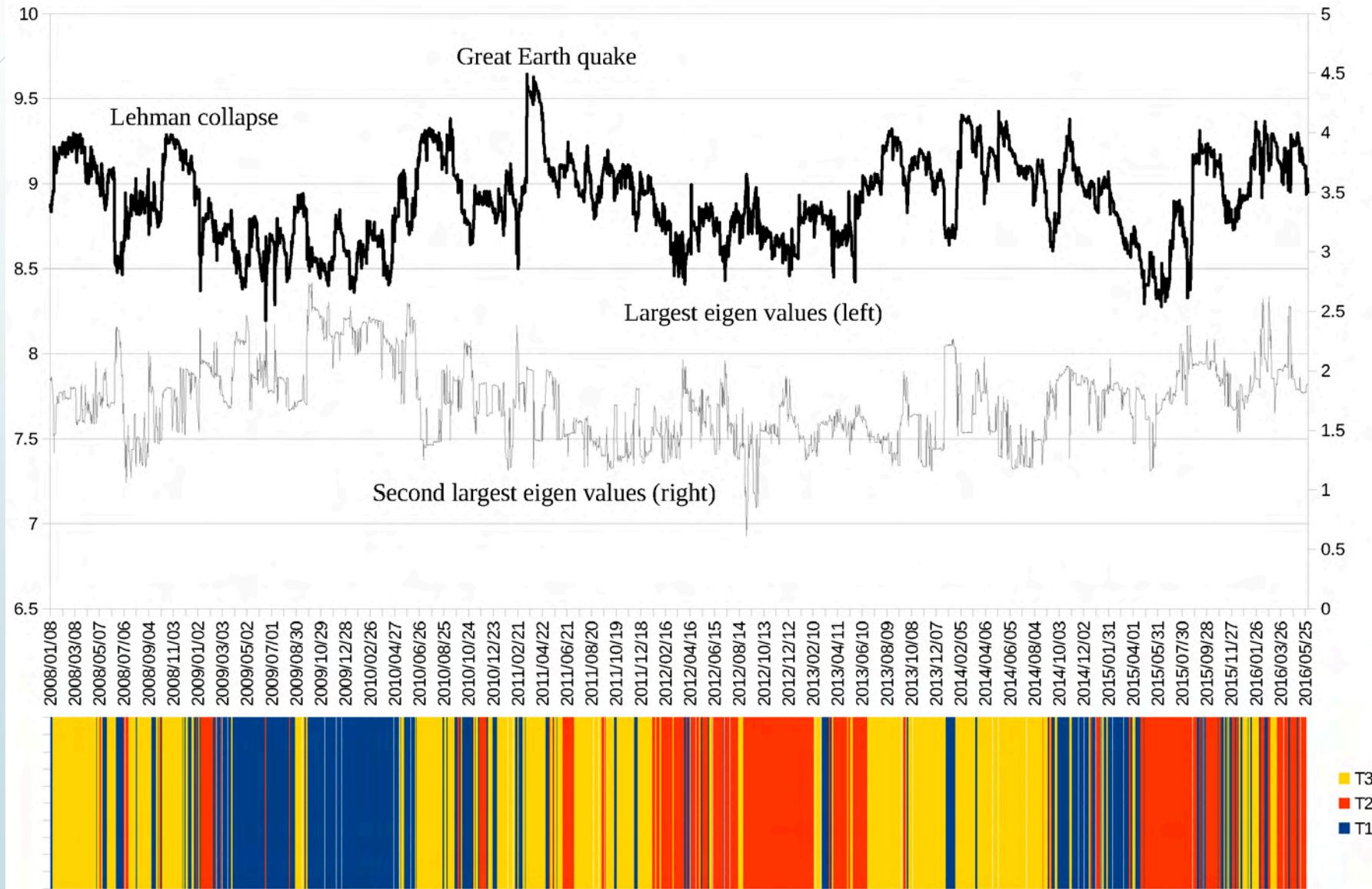
時間軸における相関ネットワークの分類

Multiple adjacency matrices \Rightarrow time periods clustering

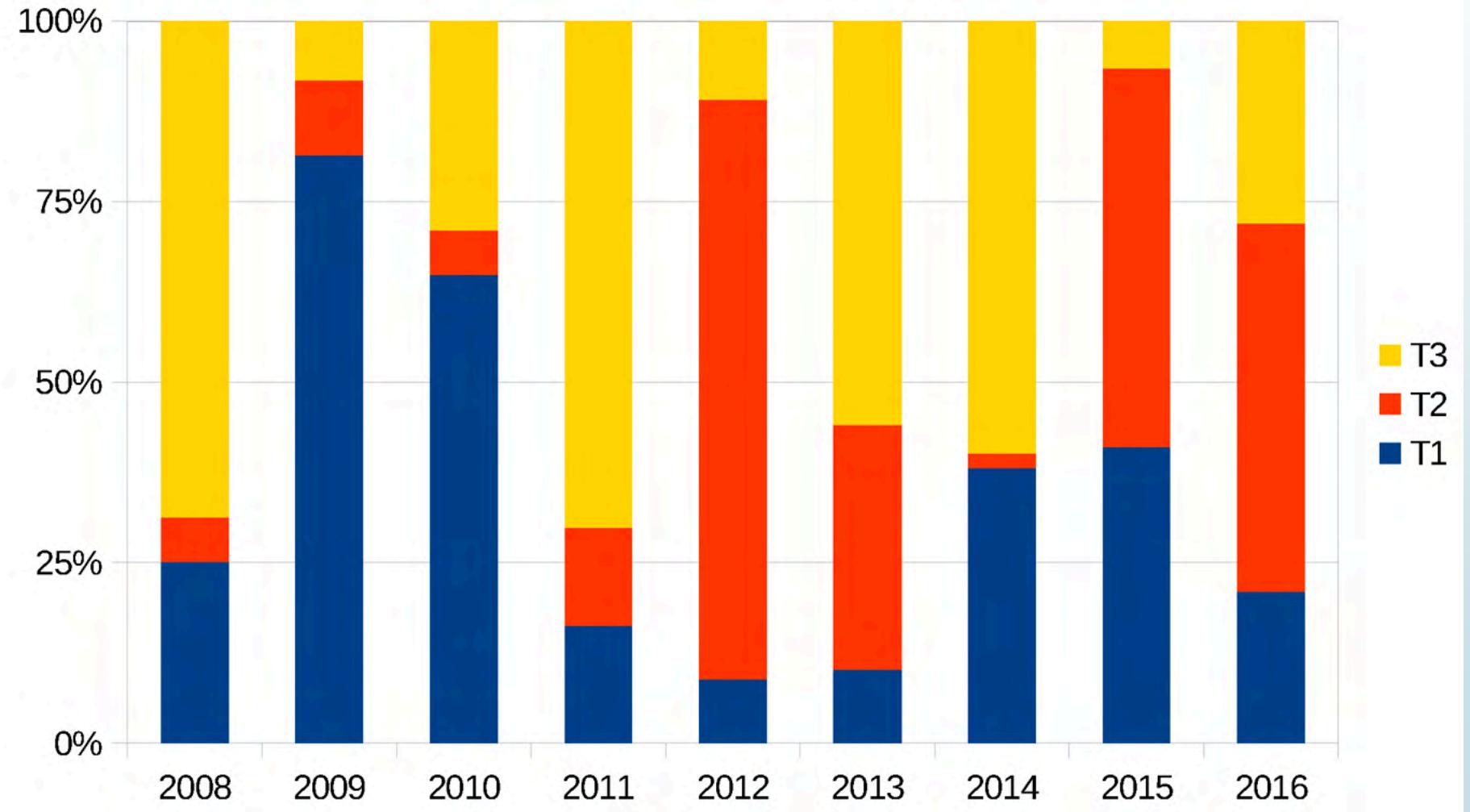


- Tucker decomposition:
 - ▶ Decompose $\chi \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ into factor matrices with orthogonal columns $U_k \in \mathbb{R}^{n_k \times r_k}$ ($k = 1, 2, 3$) and core tensor $\mathcal{G} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$
 - ▶ Reduced rank ($r_2 = r_3 = 3 \leq 14$) factor dimension
- $\mathcal{G} \times_1 U_1$ as feature vector for k -means clustering
 - ▶ Set class number at $k = 3$ (three periods categorization)

分類結果（1）

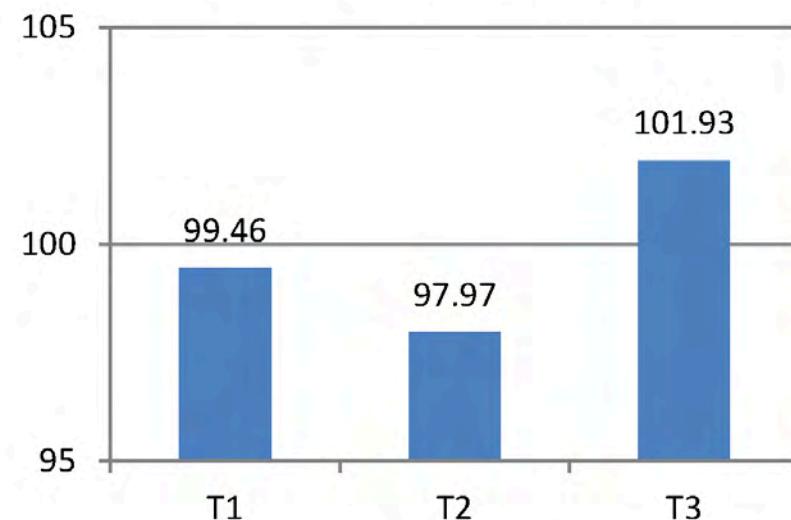


分類結果（2）



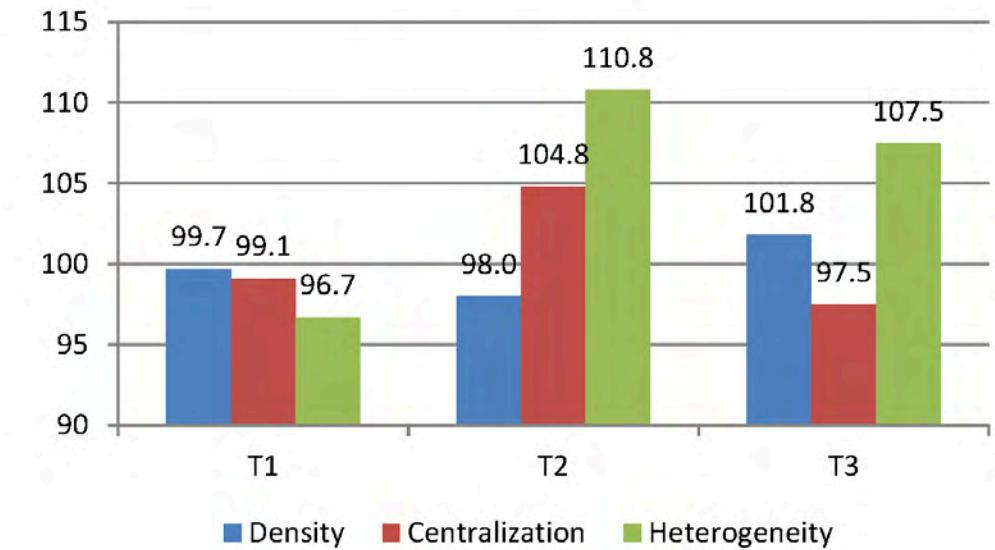
3つの部分期間の比較分析

**Largest eigen value
(whole period=100)**



Note: The largest eigenvalue of each adjacency matrix A_t at time t . The average values are calculated for each period of T1, T2 and T3.

**Network topologies
(whole period=100)**

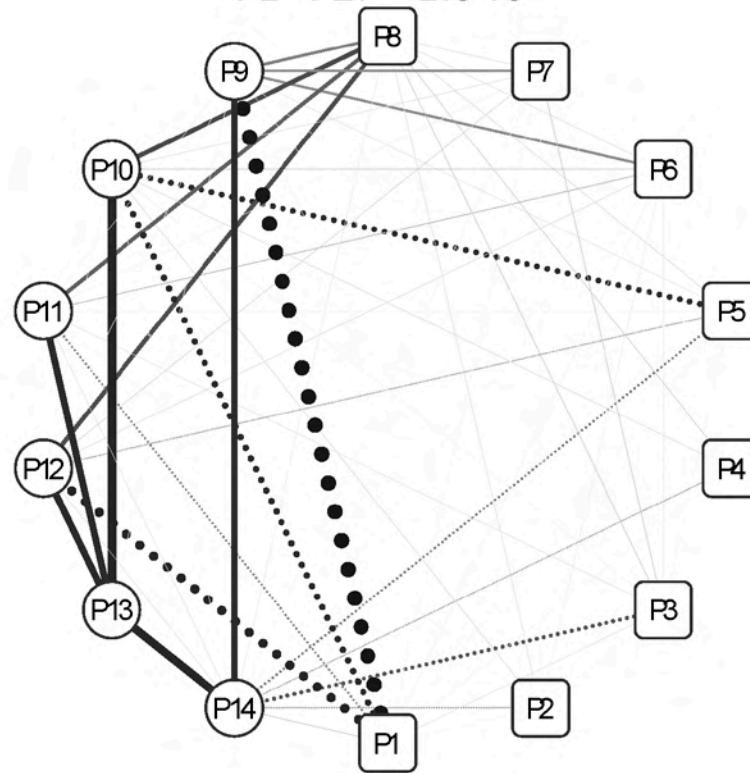


部分期間の比較

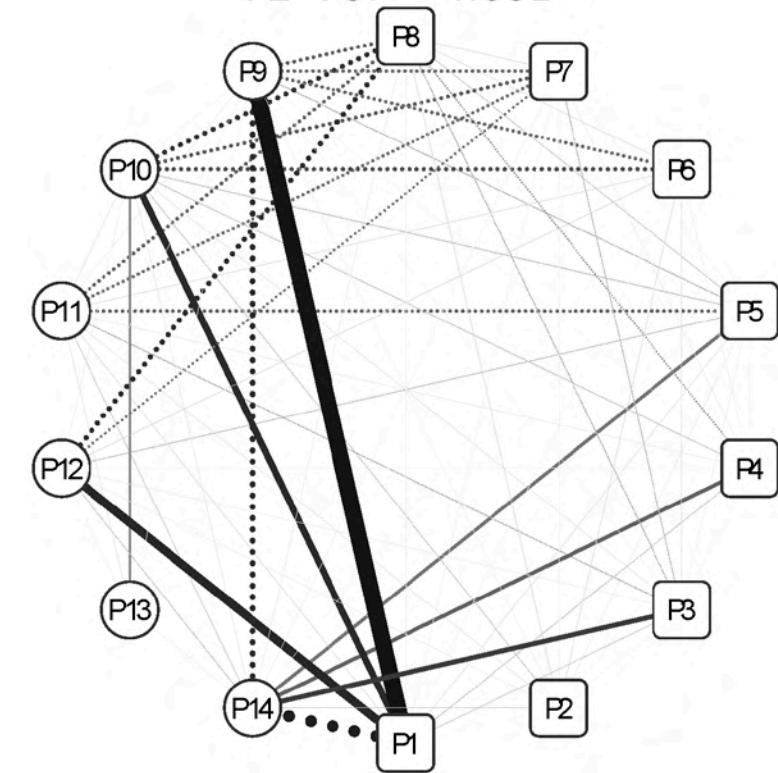
(複数のグループ間の相関でどこが変化していたかの特定)

実線 (相関の上昇); 点線 (相関の低下); <リンクウェイトの合計>

T1-T2: <1.943>



T2-T3: <-4.331>



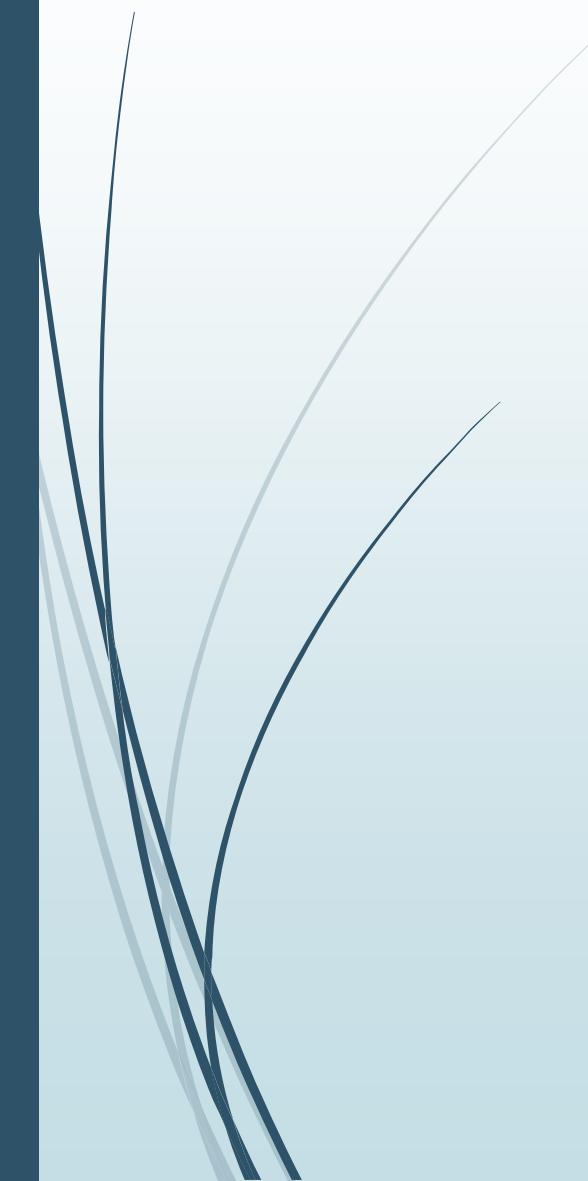


参考文献

1. 「複雑ネットワーク 基礎から応用まで」増田・今野著、近代科学社、2015
2. 「関係データ学習」石黒・林著、講談社、2014



Q&A



記述的データ解析手法（2）： 特徴選択と次元削減

担当：磯貝 孝

データマイニングにおける特徴量

- 分析対象のデータに対して、そのデータを特徴づける性質を示す数値的な量的・質的な特徴量を特定する
 - 複数の特徴量となるのが普通 ⇒ 特徴ベクトルを構成する
- 個体を特徴づけるためのデータ：様々な特徴量が考えられる（観察可能なもの、観察困難なもの）
 - A君の成績（国語、算数、理科、社会、英語の点数）
 - B君の健康状態（健康診断の各項目の値）
 - C君の信用度（職業、年収、クレジット履歴、クレジットスコア）
 - ✓ 関心のある状態と関係ない特徴を選んでも意味のあるモデルの構築・分析結果は得られない

特徴選択

- 特徴選択 (feature selection) : 特徴集合のうち意味のある部分集合だけを選択する
 - モデルに関して本当に必要な（意味のある）項目に絞り込むことで、モデルの精度、汎用性の向上が期待できる
 - 無関係な特徴を選んでモデルに加えても有害なノイズが増えるだけ
 - 次元の呪い（データの次元が高くなると、パラメータ推定、類似度の計算など様々な面で困難さが増す）
 - 特徴ベクトルの次元数を減らす努力
 - ✓ PCAなど数学的な手法を用いて、データを圧縮して新たな特徴ベクトルを生成するアプローチは、特徴選択とは異なる

特徴ベクトル

- 回帰分析、クラスタリングなど、データを機械学習のアルゴリズムで分析するには、個体のデータを数値化する必要がある
- 特徴空間、特徴ベクトル
 - データが持つ特徴を軸とするn次元の空間（特徴空間）
 - 各データは特徴空間上の1点としてあらわされる
 - 特徴空間の次元数は、選択した特徴の種類の数と同じ
 - ✓ 2次元（特徴量は2種類）はx y平面、3次元（同3種類）はx y z空間で表現できるが、4次元以上は簡単には図に示せない
 - 例：A君の成績 (90, 70, 60, 80, 50)
 - ベクトル表示なので、特徴ベクトルと呼ばれる
 - ✓ 実際に分析にかける際は、全体の平均からの乖離や偏差値などに変換する場合が多い（正規化、変数変換）

データを特徴ベクトルに変換する利点

- 構造化されていないデータを特徴ベクトルという型に変換（構造化）することで、様々な演算が可能になる
 - 線形代数をはじめとする数学的手法の応用、統計学的な分析処理（確率分布のフィッティング）・推論などが可能となる
 - ✓ テキスト分析などは、特徴ベクトル化（単語文書行列）が必須の例（元の文字データのままでの分析は困難）
 - 分析の入力は、すべて特徴ベクトルということになるので、結果も特徴ベクトルの作り方次第という側面がある
 - 特徴ベクトルの構成方法、特徴量の選択に関する正解は存在しない
 - ✓ どんな特徴を選べばいいのか、は分析者の視点・分析の目的などにも大きく依存する
 - ✓ データのアベイラビリティとは直接関係ない（必要なデータが常に得られているかは事前にはわからない）

時系列分析の場合

- 時系列データの場合、データの発生時点（前後関係）が分析上重要なものとなる
 - データを発生順と無関係にばらばらにしてしまうと、情報が失われる
 - 定常性の仮定や自己相関など、時系列分析に特有の条件などにも注意が必要
- 時系列分析の場合、特徴ベクトルは、例えば日付毎の変数（特徴）の並びとみなせる
 - 時系列モデルの場合、特徴ベクトルという用語は実際あまり使われない
 - 例： $y = ax + b$ のような線形モデルで、従属変数y（ある日の商品の売り上げ）、独立変数x（その日の値段、来客数、天気）
 - ✓ モデルのパラメーター推定や従属変数の予測・シミュレーションなどを行う場合、誤差 ($y - \hat{y}$) に関する統計的な前提（正規性、分散均一性など）が重要な役割を果たす
 - ✓ 時系列の概念がないクロスセクションデータの分析と同じアプローチをとることはできない

次元の呪い (curse of dimensionality)

- 考えられる限りのデータを入れてモデルを作つてみる
 - データの次元数が大きくなると、データの組み合わせパターンが飛躍的に増え、問題の複雑度が急激に増すため、汎化性能（未知のデータに対する予測力）の著しい低下が生じてしまう
 - 特徴ベクトルの次元が増えすぎると、要素間の距離の大小の差がなくなっていく
 - ✓ 高次元の特徴ベクトルでクラスタリングしようとしても、いい結果が得られない
 - 得られたサンプルデータだけでは十分な学習ができなくなる
 - ✓ データの追加で次元数の増加のデメリットを補うことは困難
- 次元の削減、すなわち特徴量の選択（数の削減）が必要になる
 - Deep Learning (Neural network) の場合でも、同じような状況が生じ得る

特徴量の選択：どんな基準が考えられるか

- まず何を分析したいのかを明確にしておく
 - 時系列予測が目的なら何を予測するのか、クラスタリングなら何を分類したいのか
 - データのアベイラビリティの確認・・・得られる全体集合（ユニバース）をはっきりさせておく
- 選択の基本的なアプローチ：特徴量のランキング、部分集合の選択
 - 単変量選択・フィルタリング・・・特定の指標（カイ二乗検定やANOVA (Analytics of Varianceなど)）を用いて、特徴量（変数）の順位付けを行い、モデルの有意な性能向上をもたらす変数をみつける

変数の部分集合を構成する

- 最適な変数の組み合わせを探索するアプローチ
 - 変数増加法・減少法、両者の組み合わせ
 - 回帰分析でよく用いられる方法：ステップワイズ回帰（変数をひとつずつ増やしながら当てはまりのよい変数だけを追加していく<あるいは減らしていく>）
 - ✓ 比較的よい部分集合を見つけられる
 - ✓ 自動処理のままだと分析者の知見を反映しにくい
- モデルベースの変数選択
 - 機械学習や回帰モデルなどで、モデルの推定・アルゴリズムの中に変数選択のプロセスがある程度含まれている
 - ✓ 決定木、Lasso回帰・Ridge回帰<過学習を防ぐ目的で正則化という仕組みでモデルにおける過度な複雑さを回避>など

オーバーフィッティング

- オーバーフィッティング：特徴量の選択をモデル推定に用いるデータに過度に最適化してしまう
- 特徴量を増やしすぎると、モデルの（インサンプル・データへの）オーバーフィッティングが生じてしまい、予測性能の大幅な低下などの弊害が生じる
 - アウトオブサンプルでのモデルのフィットが極端に悪くなる
 - モデルの性能の良さ（見た目）を意図的に強調したい、などの動機がなければ、この問題は極力回避すべき
- K-foldクロスモデルバリデーション（データをk個のサブセットに分けて、各データセットでモデルフィッティングと予測性能評価を繰り返す）などでのチェックが必要
 - 金融機関などが用いる信用スコアモデルなどの構築では必須の対応
 - 新しい未知の類似データ（新規顧客）が来てもそれなりにきちんとした信用判定ができるか、という問題

分析者の主觀と変数選択の結果の妥当性

- 選ばれた変数の部分集合は、それなりに意味があるはず
- さて、実際に選ばれた変数を含むモデルをみてどう感じるか（不自然を感じないか）
- モデルの説得力の問題
 - 住宅ローンの審査モデルを作ったとして、どうしてその変数が選ばれるのか、理論モデル（経済学的）や実務感覚（現場の知見）などと整合的になっているか
 - 統計的な指標のわずかな差異よりも、モデルの可読性、人間に対する説得力が重要と判断される場合もある
 - 「なぜ、そうなるのか説明できませんが、こちらのモデルの方が上です」と言われて、納得するか（させられるか）？

回帰分析における重共線性問題

- 従属変数に確かに影響しそうな説明変数がモデルに含まれているが、どの説明変数の係数をみても有意とはいがたい（十分なt値が出ていない、係数の標準誤差が大きい、理論が示す符号とは逆方向、など）
 - 回帰モデルの場合、変数を増やせば、モデルの決定係数（修正 R^2 ）は高くなる・・・従属変数の変動をより細かく説明できるので
 - 推定結果の見た目（決定係数）をよくするために変数を増やす？
- 何が起きているのか？
 - 複数の独立変数（特徴量）の間に強い相関関係を有するものが含まれている（多重共線性）
 - 本来一つの変数だけで説明できる変動を複数の変数でシェアしている・・・係数の符号が逆になったり、各変数の説明力が低下してしまう
 - ✓ きちんと変数選択を行う必要がある・・・モデルを構築する際にできれば相関の強そうな変数についてチェックしておく（分析者の知見を十分活かす）

圧縮処理により新たな特徴量・特徴ベクトルを作り出すアプローチ

- 代表的な手法として、固有値分解・特異値分解を用いた次元圧縮法がある
 - 線形代数的なアプローチ、確率的要素はなし
 - ✓ 固有値分解は、PCA（主成分分析）と基本的に同じ
 - この処理によって得られる新たな特徴ベクトルは、元の特徴ベクトルを変換・圧縮したものとなっている（数字としては全く異なる値をとる）
 - ✓ どの程度の次元圧縮をするかは、もとのデータの次元のサイズや、変換後の特徴ベクトルを用いた分析結果の良さなどをもとに判断する
 - 強い非線形構造をもったデータに対しては効果が期待できないかもしれない
- テンソル分解など、さらに複雑な分解・圧縮方法もある
 - より高度な確率的手法も多数存在する

固有値分解・特異値分解

- 線形代数的なアプローチ（非確率論的手法）
 - Eigen value decomposition (EVD) . . . 正方行列（縦横のサイズが同じ、分散共分散行列など）の分解、スペクトル分解とも呼ばれる
 - ✓ PCA (principal component analysis)
 - Singular value decomposition (SVD) . . . 普通の行列（縦横のサイズが違ってもいい、単語文書行列など）
 - EVDはSVDの特殊なケースとして考えることができる
- 行列の低ランク近似 . . . 行列AについてEVD、SVDで分解した上で行列のランク数を低めて情報圧縮を試みる

固有値分解 (EVD)

- 固有値、固有ベクトル
 - 固有値： 正方行列A

$$Ax = \lambda x$$

スカラー λ (固有値) 、列ベクトル x (固有ベクトル)

- 固有値分解：

$$\begin{aligned} A &= U\Sigma U^t \\ UU^t &= I \end{aligned}$$

- ✓ Σ は A の固有値を対角成分を持つ対角行列 (同じ固有値が存在する場合あり)
- ✓ U は固有ベクトルの集まり (行ベクトル) ··· 正規直交基底
- ✓ $AU = U\Sigma U^t U = U\Sigma$ ··· 部分空間表現 (固有ベクトルによって張られる空間)

低ランク近似によるデータの次元圧縮

- 特徴量の変化に大きく寄与する部分だけを取り出して、変数要約・ノイズ除去を行いたい
 - データの特徴ベクトルを線形代数処理などによりモデルで扱いやすい特徴ベクトルに変換する（主要な要素のみを自動的に絞り出すイメージ）
 - 回帰分析、クラスタリングなどの精度向上が期待できる
- 基本的なアイデア：もともとのデータ（A）が持つ固有値の大きいもの（データとしての分散が大きく、意味がありそう）だけを残して、固有値の小さいものについては、ゼロに置き換える
 - フロベニウスノルム（全成分の二乗和の平方根）などで圧縮の度合いを判断する
 - 分析者の知見を反映して決めることがある

低ランク近似（固有値分解）

- $A = U\Sigma U^t$ の Σ には、対角成分に固有値が並んでいる
 - Σ の対角成分を大きい順に左上から右下に並べておく
 - ✓ MatLab, R などで普通に eig(A), eigen(A) のように計算すると、大きい順に並んだ結果が得られる
 - N 個の固有値のうち、最大値から k 個だけを採用してあとはすべて 0 に置き換える
 - ✓ $A_k = U_k \Sigma_k U_k^t \quad \dots \quad A_k$ は A の低ランク (k) 近似
 - さらに、 $A_k U_k = U_k \Sigma_k U_k^t U_k = U_k \Sigma_k \dots$ 低ランク近似を用いた部分空間表現
 - ✓ この表現を使うと実際に k 次元の特徴ベクトルでクラスタリングを行うことができる（部分空間クラスタリング、例：文書集合<単語文書行列>のクラスタリング）

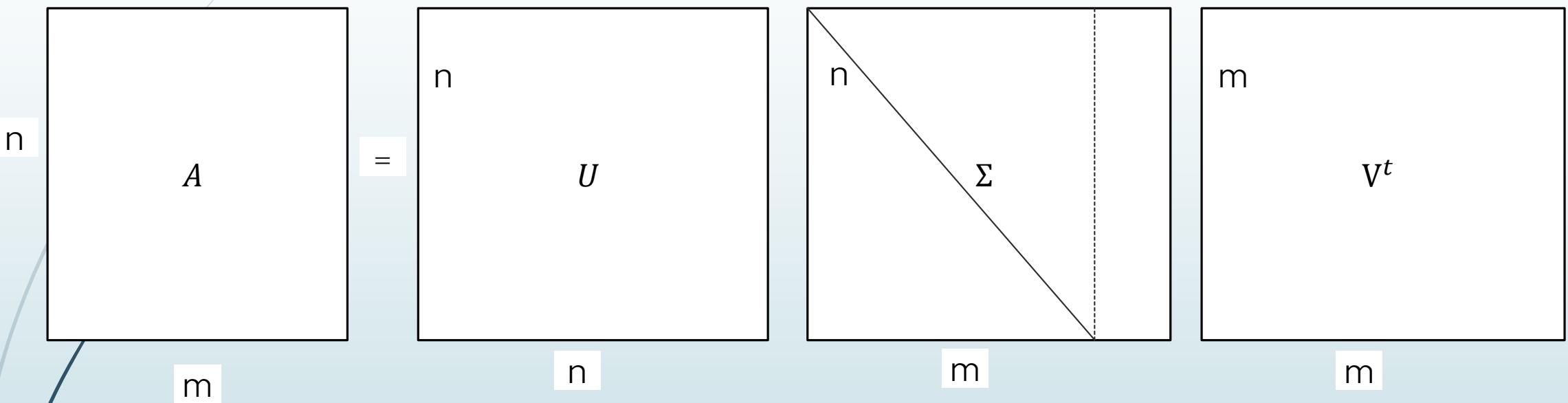
特異値分解 (SVD)

- 固有値分解の場合、行列Aの縦横サイズが同じでないといけない
 - 分散共分散行列などの場合はいいが、一般的なデータを考えると、正方行列でない場合も多い
 - 縦横サイズが違う場合の分解はどうすればよいか？
- 特異値分解 (SVD) · · · 固有値分解の一般化
$$A = U\Sigma V^t$$
$$UU^t = I, VV^t = I$$

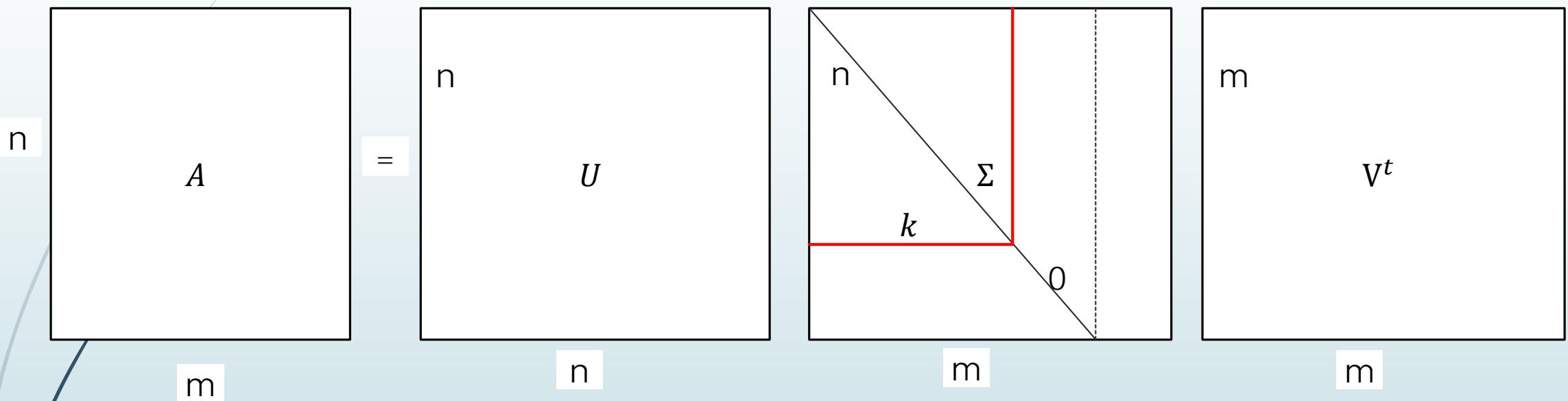
A は $n \times m$ 、 U は $m \times m$ 、 Σ は $m \times n$ 、 V は $m \times n$ の行列

 - Σ は対角成分に特異値が並んだ対角行列（特異値は $A A^t$ の固有値の平方根）

SVDのイメージ



SVDによる低ランク近似

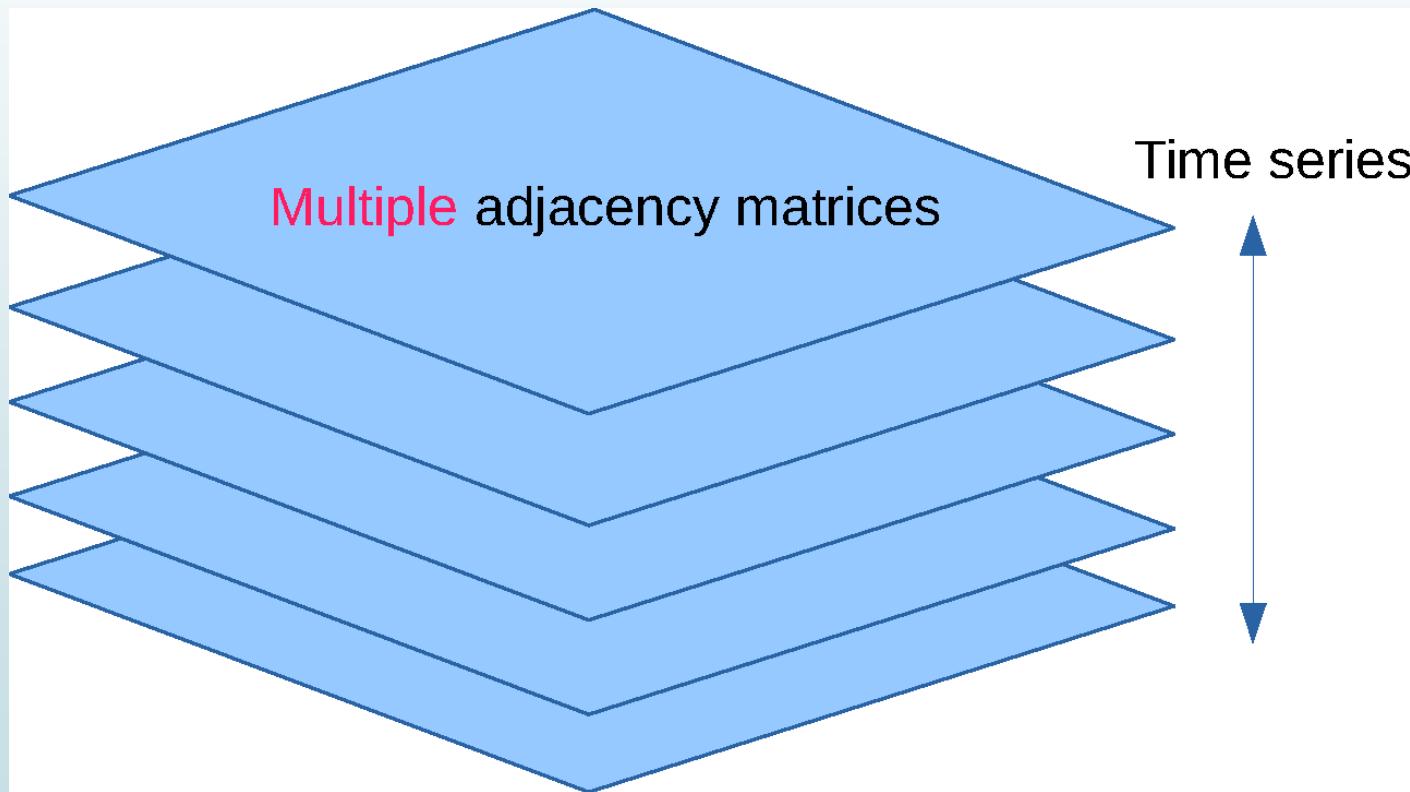


$$A_k = U_k \Sigma_k V^t_k$$
$$U_k A_k = \Sigma_k V^t_k, A_k V_k = U_k \Sigma_k$$

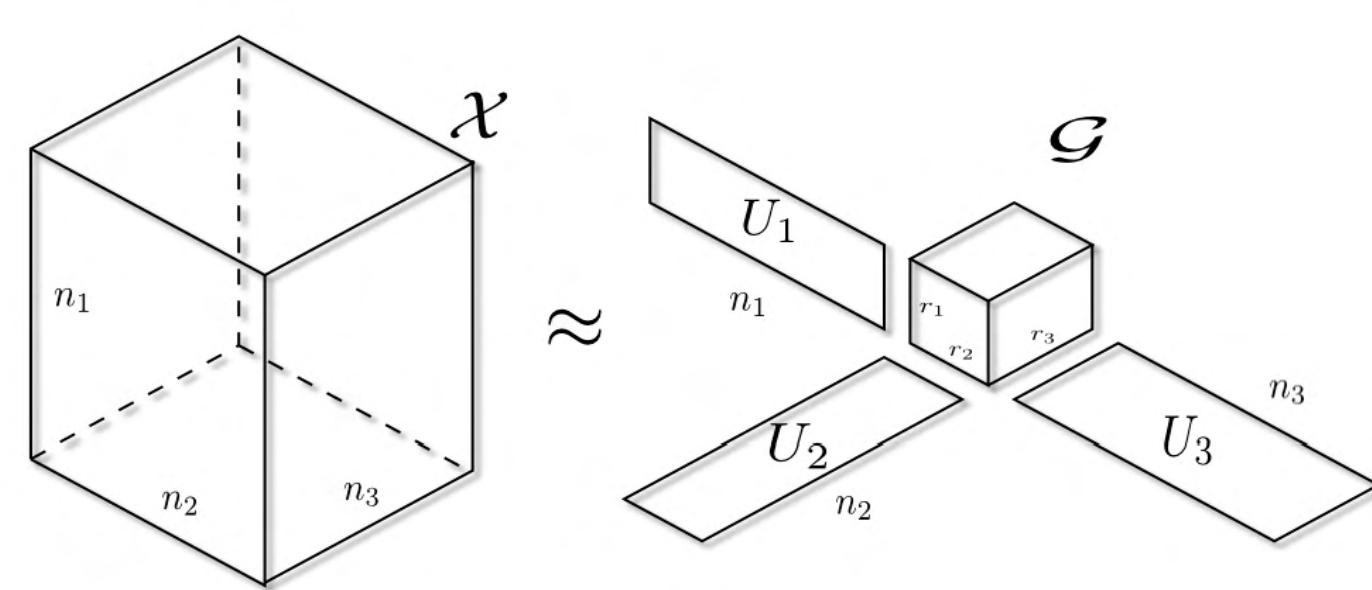
次元圧縮の目的でのテンソルの応用

- テンソル (Tensor) : n次元の多次元配列（厳密な定義ではなく、簡略なイメージ）
 - 物理学で用いる概念（運動量）まで詳しく立ち入らない・・・特徴ベクトルのデータ圧縮に関する部分では、多次元配列の並びと理解しておく
 - 例：行列が何枚も積み重なったもの・・・一つの行列がある時点におけるデータ行列で、それが時系列を構成する、など
- 四則演算や数学関数、線形代数計算などが可能
 - 特異値分解のような分解、部分空間への射影などにも応用可能
 - Rやpythonなどでもライブラリで通常必要となる計算は十分行える
 - どんなふうに使うのか？

テンソルのかたちをイメージする： ネットワークの隣接行列の時系列の例



テンソル分解

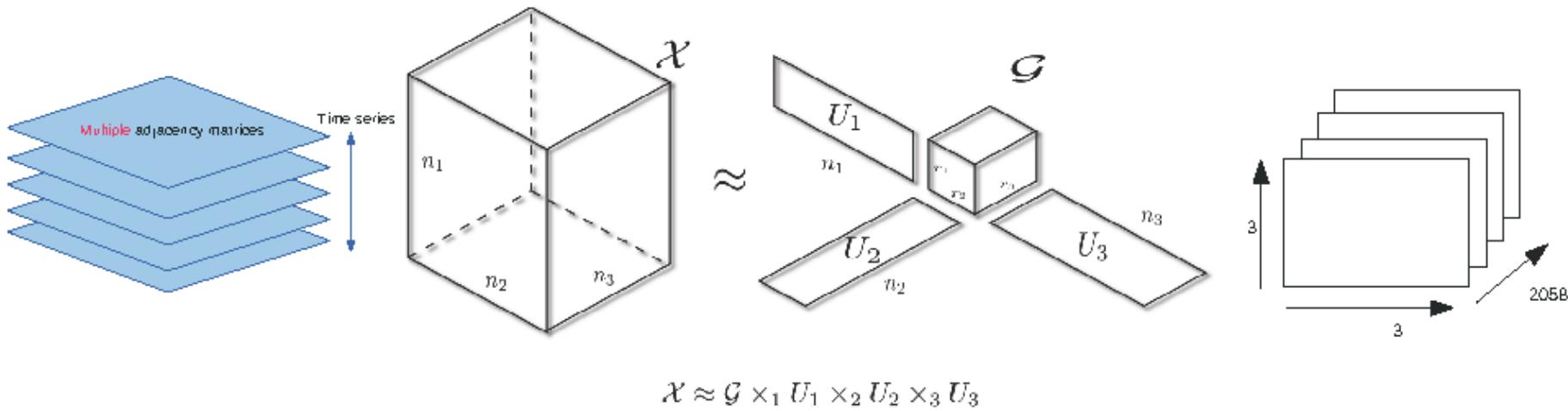


低ランクテンソル分解

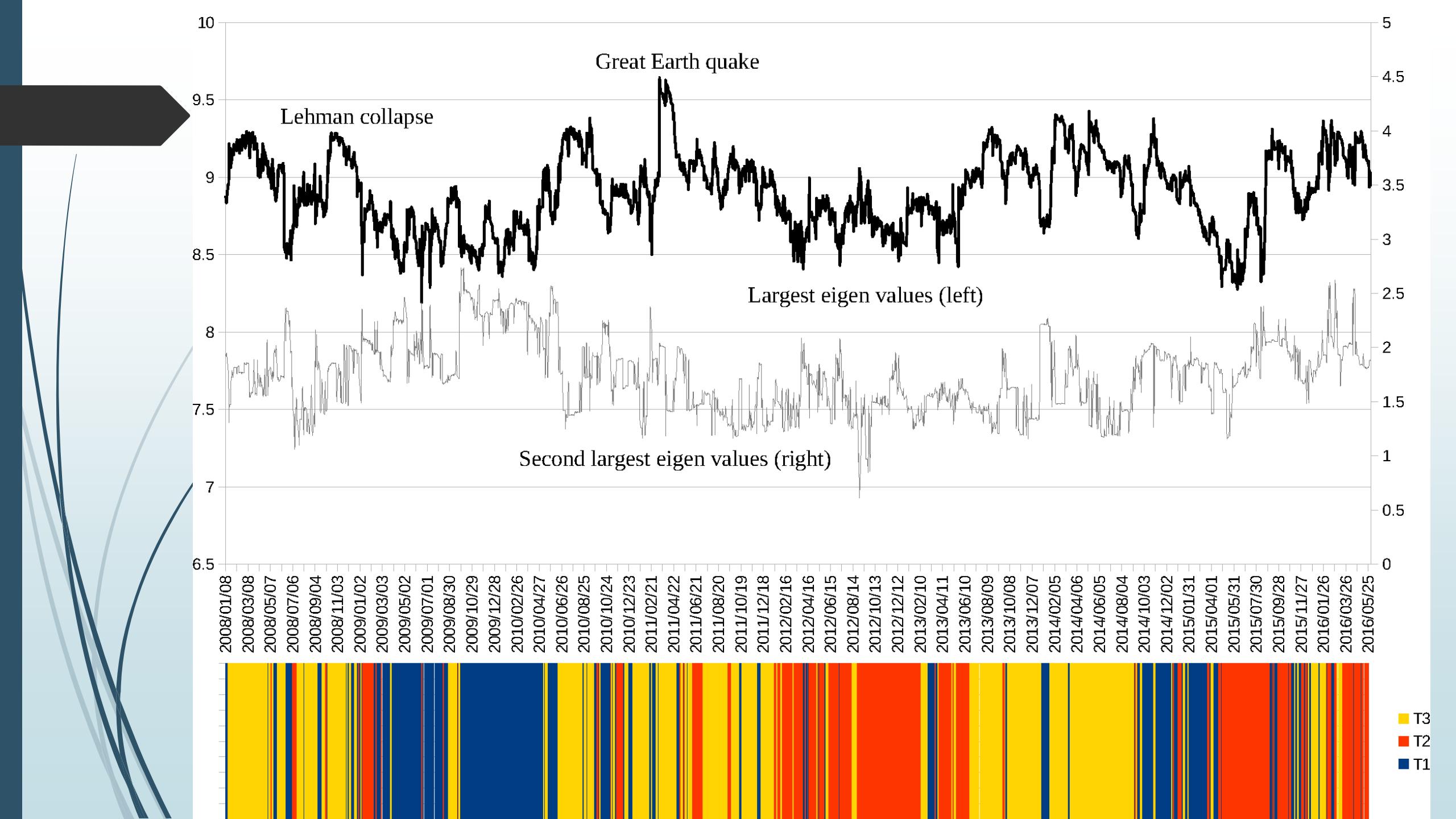
- テンソルの分解については、様々な手法が存在する
 - 手法によって得られる分解結果は異なる、目的によって方法を選ぶ
 - 伝統的でよく用いられる手法（例：Tucker decomposition）
- 特異値分解の場合と同様に、例えば（3次元データの場合）3つの軸を示す基底要素とコアテンソル（SVDの Σ に相当）に分けられる
 - SVDの低ランク近似の場合と同様に、コアテンソルの次元を下げて（意味のある部分のみ残す）、データの圧縮を行うことができる
 - SVDの場合と同様に関心のある軸に関して部分空間を構成し、特徴ベクトルを構成してクラスタリングを行うこともできる（subspace clustering）

Low rank Tensor decomposition

Multiple adjacency matrices and time periods clustering



- Tucker decomposition:
 - ▶ Decompose $\chi \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ into factor matrices with orthogonal columns $U_k \in \mathbb{R}^{n_k \times r_k}$ ($k = 1, 2, 3$) and core tensor $\mathcal{G} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$
 - ▶ Reduced rank 3 ($r_2 = r_3 = 3$) from 14 (factor dimension)
- $\mathcal{G} \times_1 U_1$ as feature vector for k -means clustering
 - ▶ Set class number as 3 (three periods categorization)





参考文献

1. 「Python機械学習クックブック」Chris Albon著、中田秀基訳、オーライリー、2019

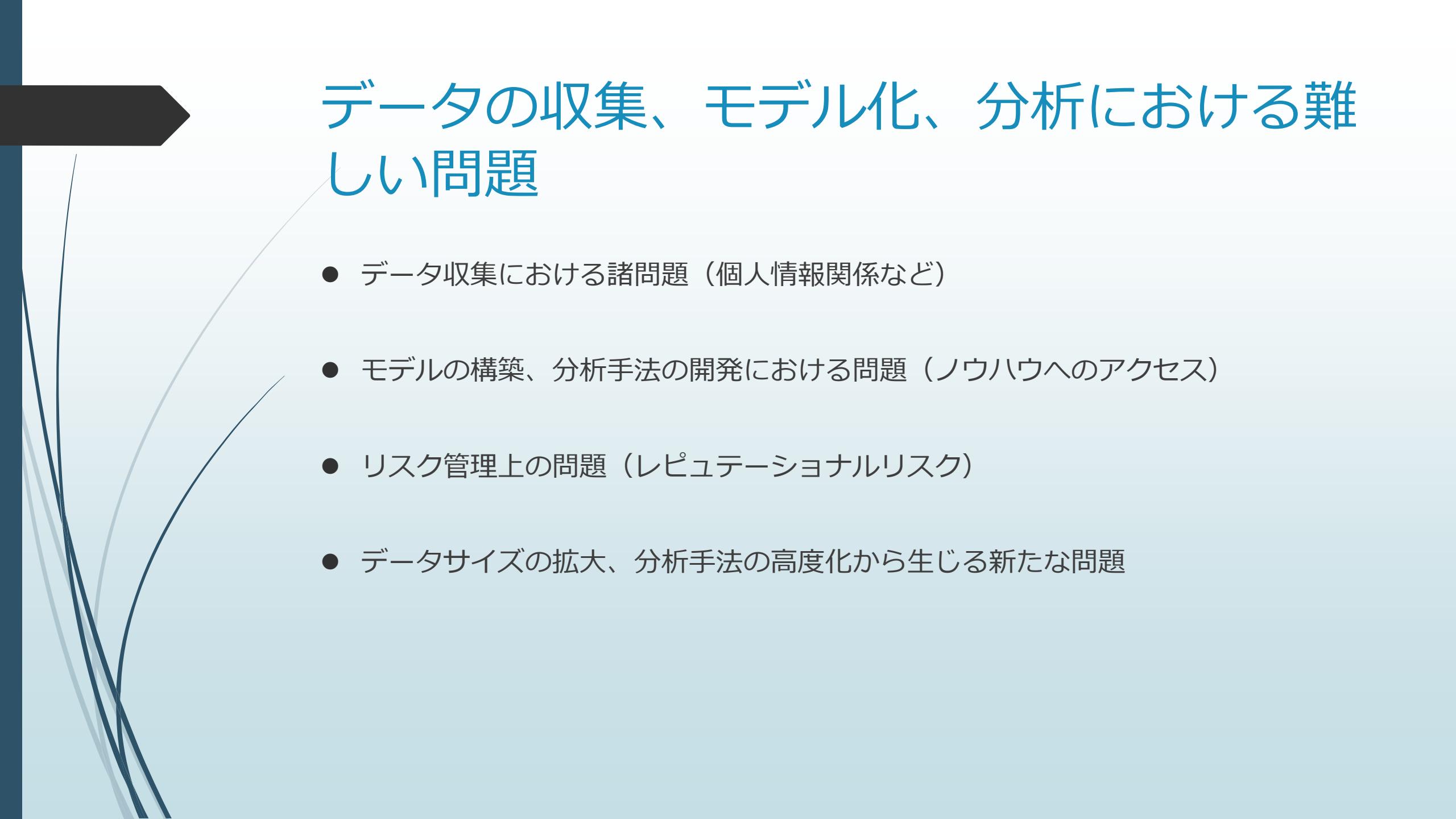


Q&A



データ科学と倫理問題

担当：磯貝 孝

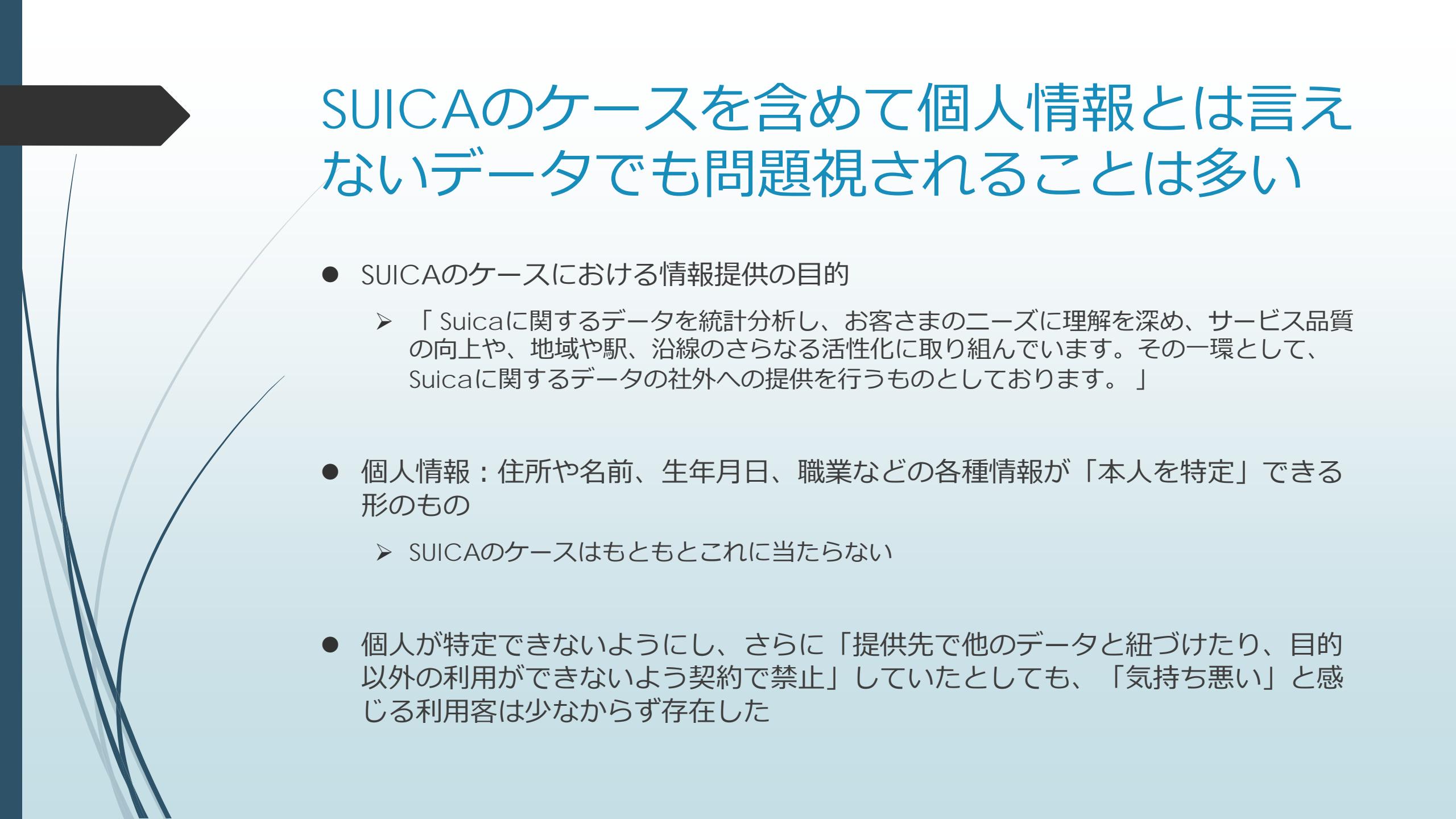


データの収集、モデル化、分析における難しい問題

- データ収集における諸問題（個人情報関係など）
- モデルの構築、分析手法の開発における問題（ノウハウへのアクセス）
- リスク管理上の問題（レピュテーショナルリスク）
- データサイズの拡大、分析手法の高度化から生じる新たな問題

データの活用と個人情報保護

- SUICAの利用情報（分析用データ）をJR東日本が社外提供することがわかった（2013年6月）
 - 「Suica利用データから氏名、電話番号、物販情報等を除外し、生年月日を生年月に変換した上、さらに、Suica ID番号を不可逆の別異の番号に変換したデータ」
 - 利用目的は、マーケティング分析
 - 多くの利用者から、個人情報の保護、プライバシーの保護や消費者意識に対する配慮に欠けているのではないかとして批判や不安視する声があがった
 - 同社は、7月25日には販売中止を宣言
 - HP上で「Suicaに関するデータの社外への提供について」の説明文と除外要望受付フォームを用意して対応した
 - ✓ 有識者会議資料「Suicaに関するデータの社外への提供について」
(https://www.jreast.co.jp/information/aas/20151126_torimatome.pdf)

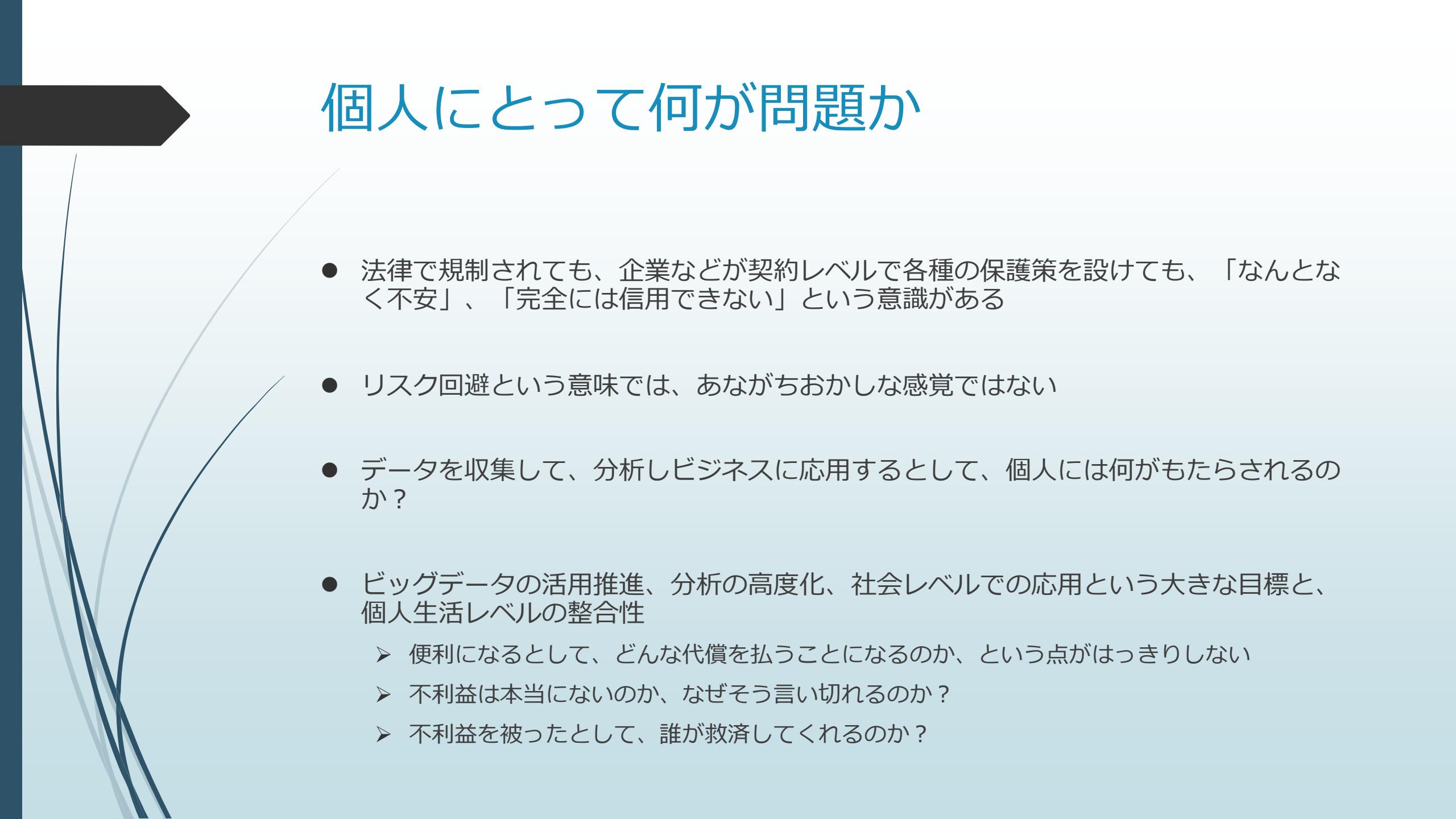


SUICAのケースを含めて個人情報とは言えないデータでも問題視されることが多い

- SUICAのケースにおける情報提供の目的
 - 「Suicaに関するデータを統計分析し、お客様のニーズに理解を深め、サービス品質の向上や、地域や駅、沿線のさらなる活性化に取り組んでいます。その一環として、Suicaに関するデータの社外への提供を行うものとしております。」
- 個人情報：住所や名前、生年月日、職業などの各種情報が「本人を特定」できる形のもの
 - SUICAのケースはもともとこれに当たらない
- 個人が特定できないようにし、さらに「提供先で他のデータと紐づけたり、目的以外の利用ができないよう契約で禁止」していたとしても、「気持ち悪い」と感じる利用客は少なからず存在した

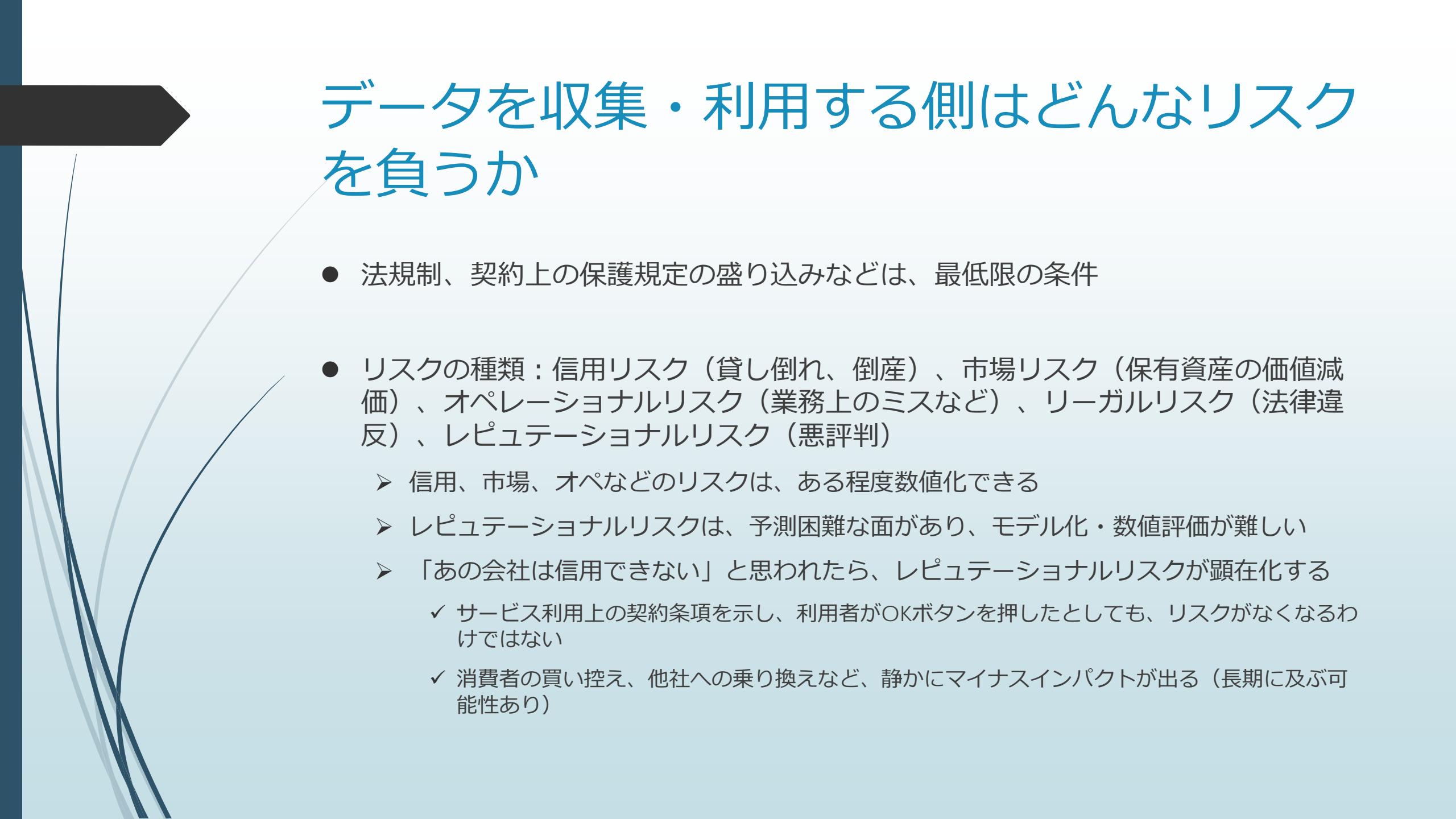
改正個人情報保護法での扱い

- 改正個人情報保護法（平成27年）におけるビッグデータの扱い
 - 匿名加工情報（特定の個人を識別したり、元の個人情報を復元することができないようしたもの）を加工する事業者は、本人の同意を得る
 - 匿名加工情報の加工に関する基準、加工方法に関する漏えい防止措置、作成した匿名加工情報に関する公表義務、自ら取り扱う場合の識別行為の禁止義務などが課された
- その後も様々な対応が進められているが、個人の行動、身体に関する情報についてのデータ化、分析上の利用については引き続き抵抗感を持つ人も多い
 - フィンテック関係でも、スマホ決済などの拡充が日本でも進んでいる（決済関連の情報収集も進んでいる様子）



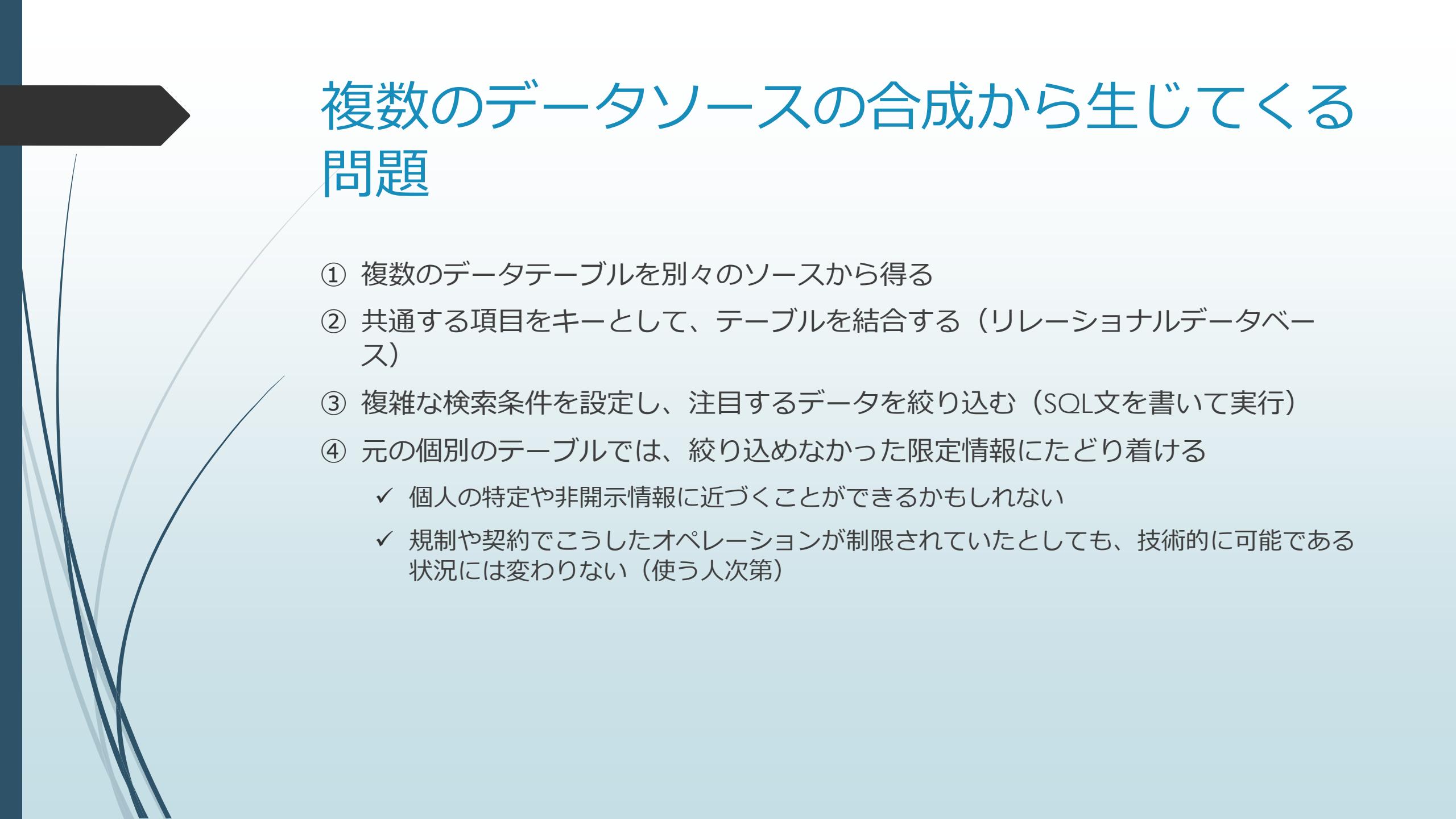
個人にとって何が問題か

- 法律で規制されても、企業などが契約レベルで各種の保護策を設けても、「なんとか不安」、「完全には信用できない」という意識がある
- リスク回避という意味では、あながちおかしな感覚ではない
- データを収集して、分析しビジネスに応用するとして、個人には何がもたらされるのか？
- ビッグデータの活用推進、分析の高度化、社会レベルでの応用という大きな目標と、個人生活レベルの整合性
 - 便利になるとして、どんな代償を払うことになるのか、という点がはっきりしない
 - 不利益は本当にはないのか、なぜそう言い切れるのか？
 - 不利益を被ったとして、誰が救済してくれるのか？



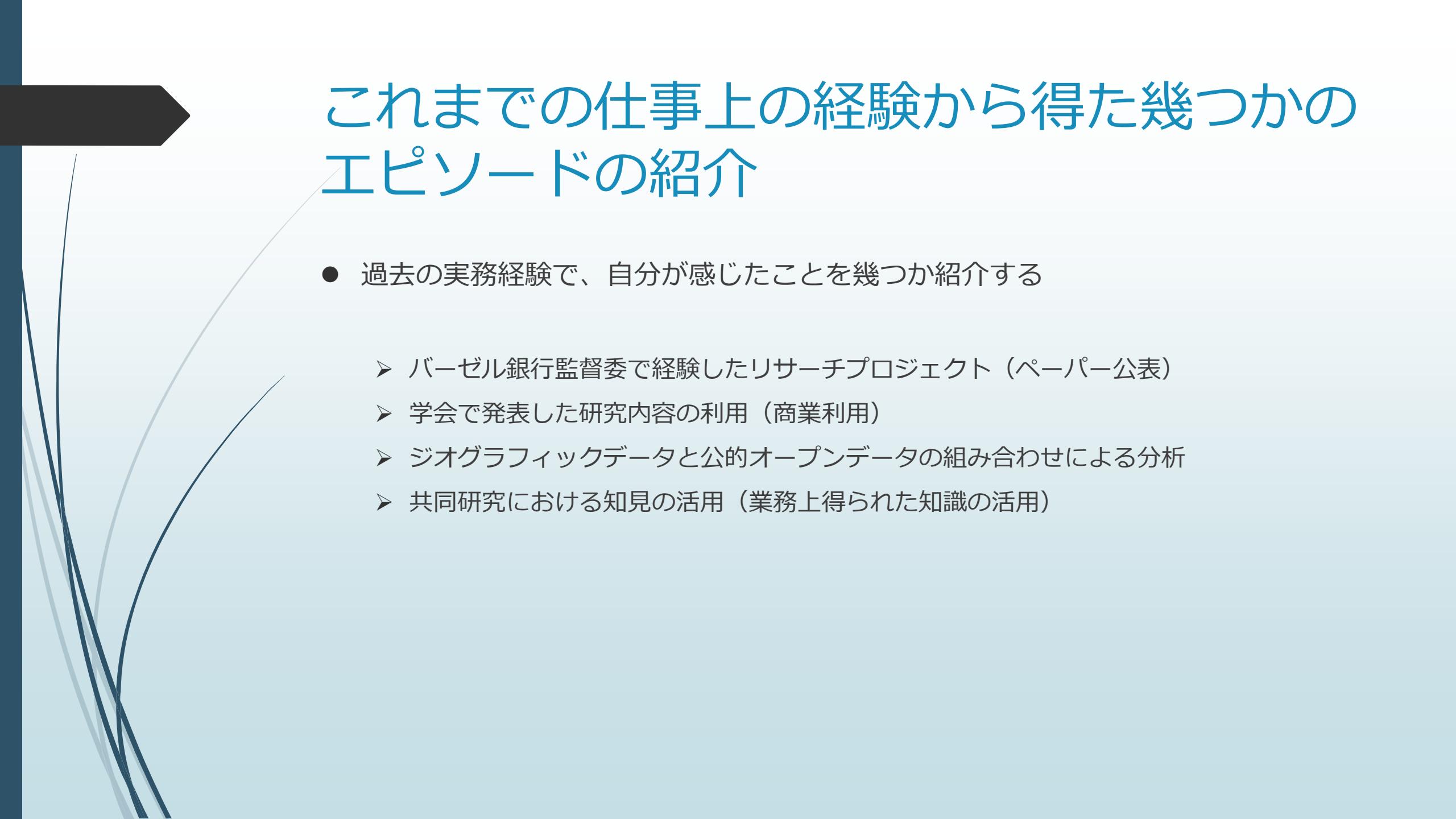
データを収集・利用する側はどんなリスクを負うか

- 法規制、契約上の保護規定の盛り込みなどは、最低限の条件
- リスクの種類：信用リスク（貸し倒れ、倒産）、市場リスク（保有資産の価値減価）、オペレーションリスク（業務上のミスなど）、リーガルリスク（法律違反）、レビューションリスク（悪評判）
 - 信用、市場、オペなどのリスクは、ある程度数値化できる
 - レビューションリスクは、予測困難な面があり、モデル化・数値評価が難しい
 - 「あの会社は信用できない」と思われたら、レビューションリスクが顕在化する
 - ✓ サービス利用上の契約条項を示し、利用者がOKボタンを押したとしても、リスクがなくなるわけではない
 - ✓ 消費者の買い控え、他社への乗り換えなど、静かにマイナスインパクトが出る（長期に及ぶ可能性あり）



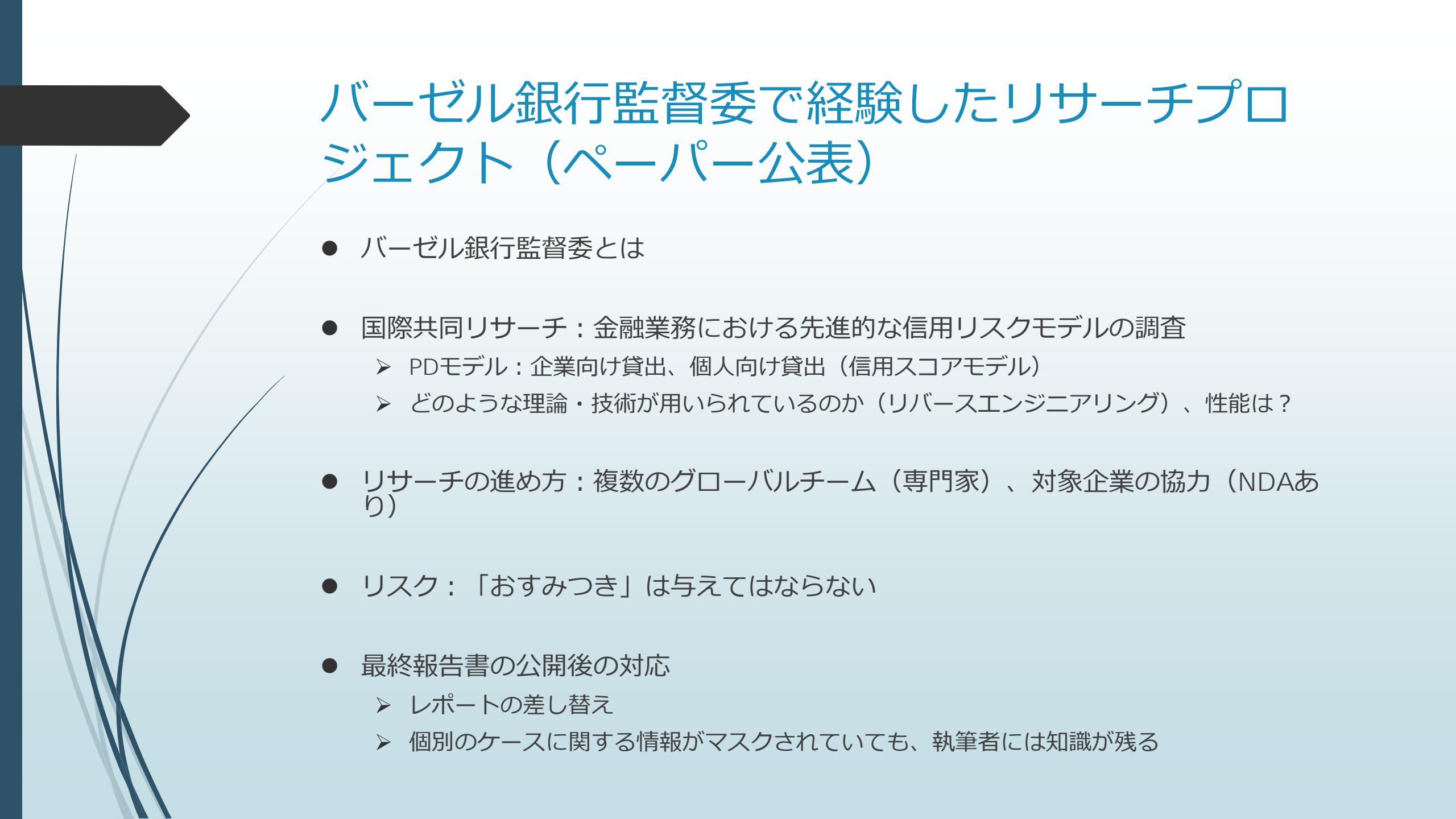
複数のデータソースの合成から生じてくる 問題

- ① 複数のデータテーブルを別々のソースから得る
- ② 共通する項目をキーとして、テーブルを結合する（リレーションナルデータベース）
- ③ 複雑な検索条件を設定し、注目するデータを絞り込む（SQL文を書いて実行）
- ④ 元の個別のテーブルでは、絞り込めなかった限定情報にたどり着ける
 - ✓ 個人の特定や非開示情報に近づくことができるかもしれない
 - ✓ 規制や契約でこうしたオペレーションが制限されていたとしても、技術的に可能である状況には変わりない（使う人次第）



これまでの仕事上の経験から得た幾つかのエピソードの紹介

- 過去の実務経験で、自分が感じたことを幾つか紹介する
 - バーゼル銀行監督委で経験したリサーチプロジェクト（ペーパー公表）
 - 学会で発表した研究内容の利用（商業利用）
 - ジオグラフィックデータと公的オープンデータの組み合わせによる分析
 - 共同研究における知見の活用（業務上得られた知識の活用）



バーゼル銀行監督委で経験したリサーチプロジェクト（ペーパー公表）

- バーゼル銀行監督委とは
- 國際共同リサーチ：金融業務における先進的な信用リスクモデルの調査
 - PDモデル：企業向け貸出、個人向け貸出（信用スコアモデル）
 - どのような理論・技術が用いられているのか（リバースエンジニアリング）、性能は？
- リサーチの進め方：複数のグローバルチーム（専門家）、対象企業の協力（NDAあり）
- リスク：「おすみつき」は与えてはならない
- 最終報告書の公開後の対応
 - レポートの差し替え
 - 個別のケースに関する情報がマスクされていても、執筆者には知識が残る

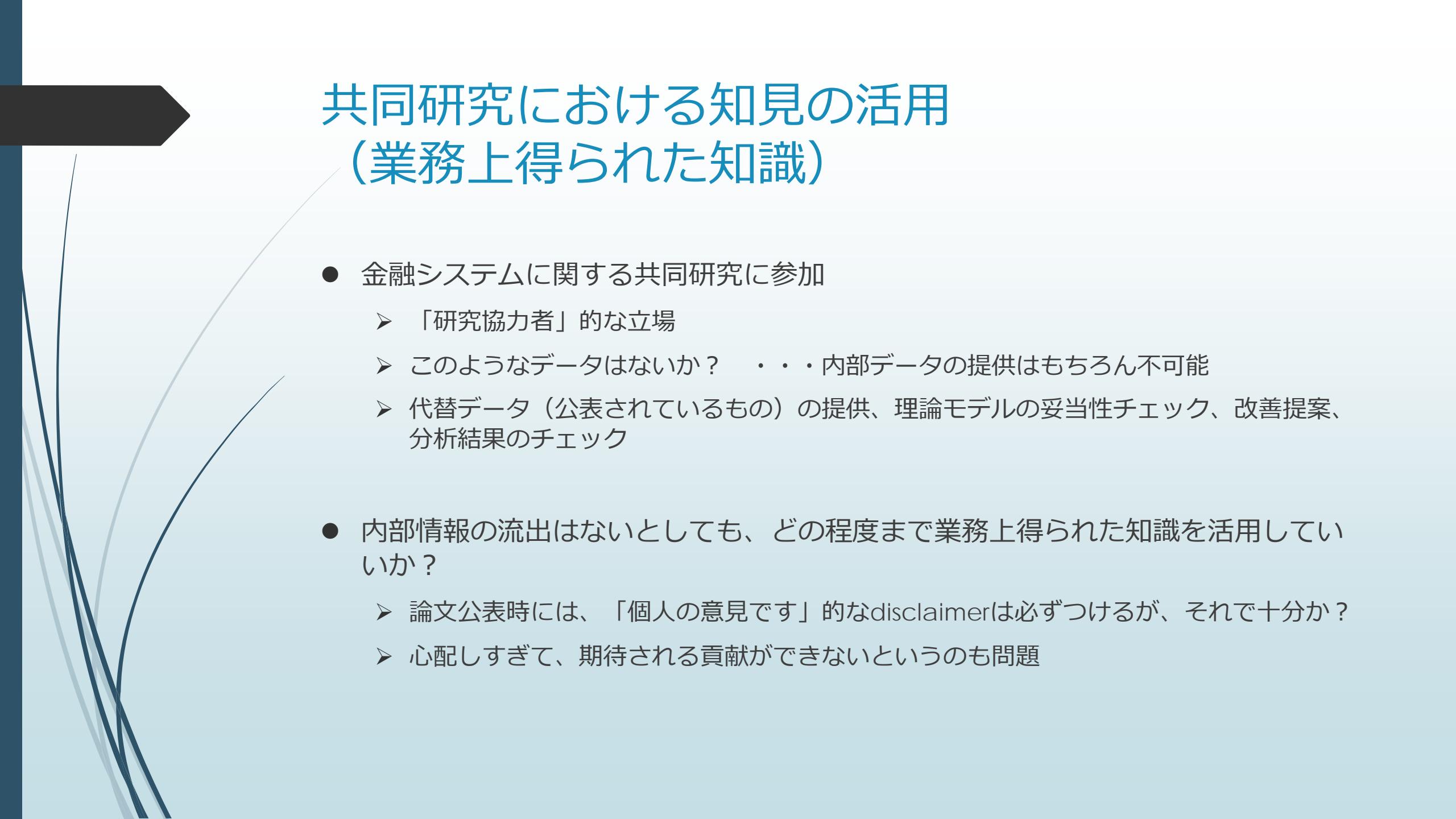
学会で発表した研究内容の利用（商業利用）

- 分析結果がある学会で発表した（論文公表済）
- 後日、ある会社からコンタクト、研究内容について詳しく知りたい
 - さらに後日、業務に応用したい、とのこと
- リスク：勤務先での兼職禁止、X氏のリサーチの結果を応用と紹介される（あるいはされない）
 - データやコードの受け渡しはなくとも、完全に無関係と言えるか



ジオグラフィックデータとオープンデータの組み合わせによる分析

- 最近は、公的機関によるデータのオープンソース化も進んできている
- メッシュデータと呼ばれる地表を細かな四角に区切ったジオロケーションデータと各種データを組み合わせることができる
 - 法人は登記上場所が特定できる
 - かなり細かいメッシュデータをもとにテーブルリンクすると、ある種の絞り込みを行った後で、特定の条件に合う少数の先を見つけられたりする
 - ✓ 各種のテーブルリンク、アルゴリズムの適用でデータの提供者が想像しないかたちでのデータ利用もできてしまう
 - データ提供者側も、加工後を意識してデータ公表の仕方を変えている



共同研究における知見の活用 (業務上得られた知識)

- 金融システムに関する共同研究に参加
 - 「研究協力者」的な立場
 - このようなデータはないか? . . . 内部データの提供はもちろん不可能
 - 代替データ（公表されているもの）の提供、理論モデルの妥当性チェック、改善提案、分析結果のチェック
- 内部情報の流出はないとしても、どの程度まで業務上得られた知識を活用しているか?
 - 論文公表時には、「個人の意見です」的なdisclaimerは必ずつけるが、それで十分か？
 - 心配しすぎて、期待される貢献ができないというのも問題



バランス感覚

- これらの「倫理問題」に関して、直接の答えとなる指針はおそらく存在しない
 - 完全なリスク回避は困難
- 企業や個人（分析者）のレベルで、バランス感覚を持って個別に対処していくしかない
- 常に「リスク感覚」を大切にすることを心がける（私個人の対処方針）



Q&A



補論：

時系列予測の話題

– 深層学習で時系列予測を行って感じたこと

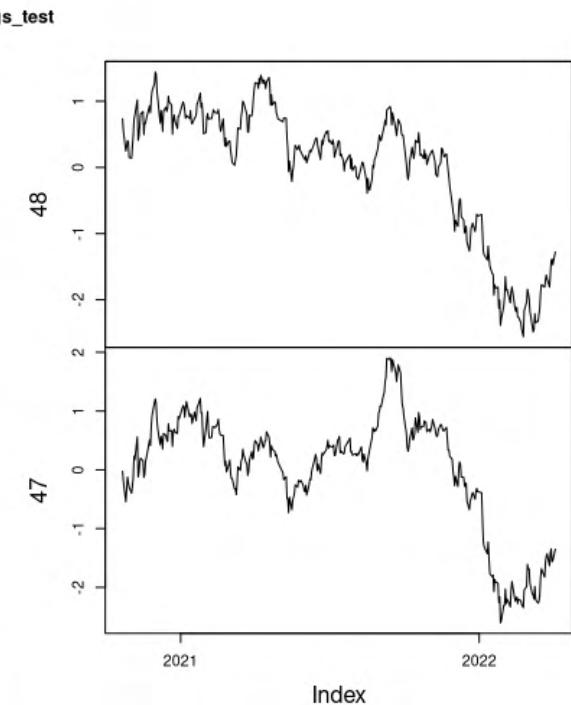
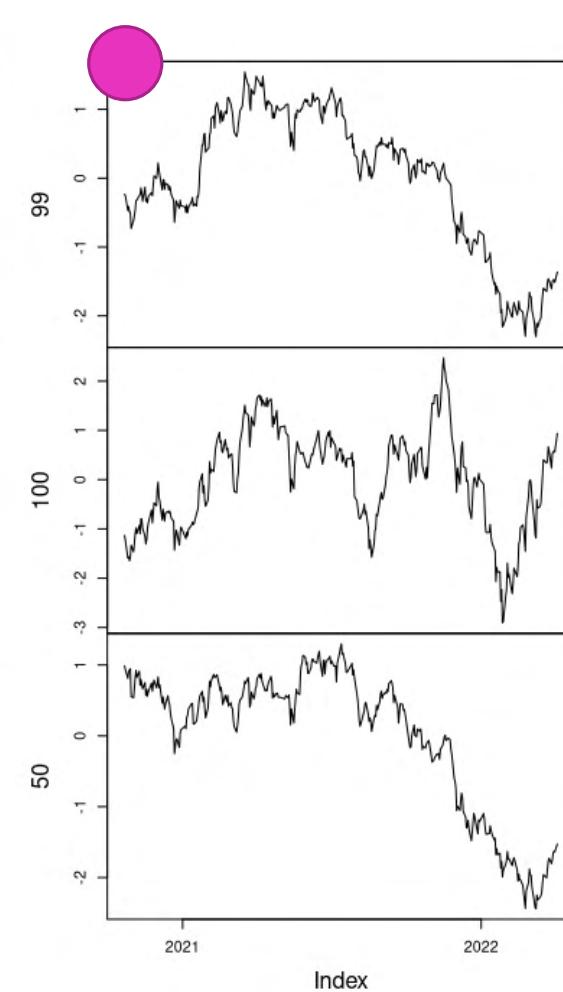
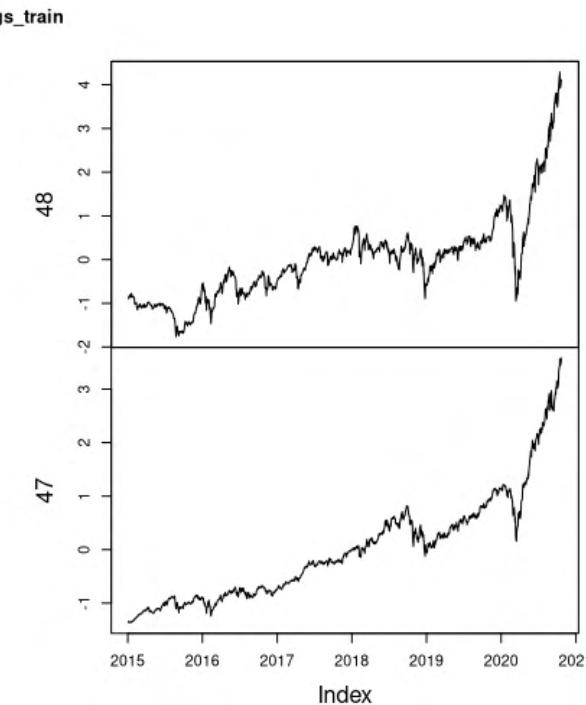
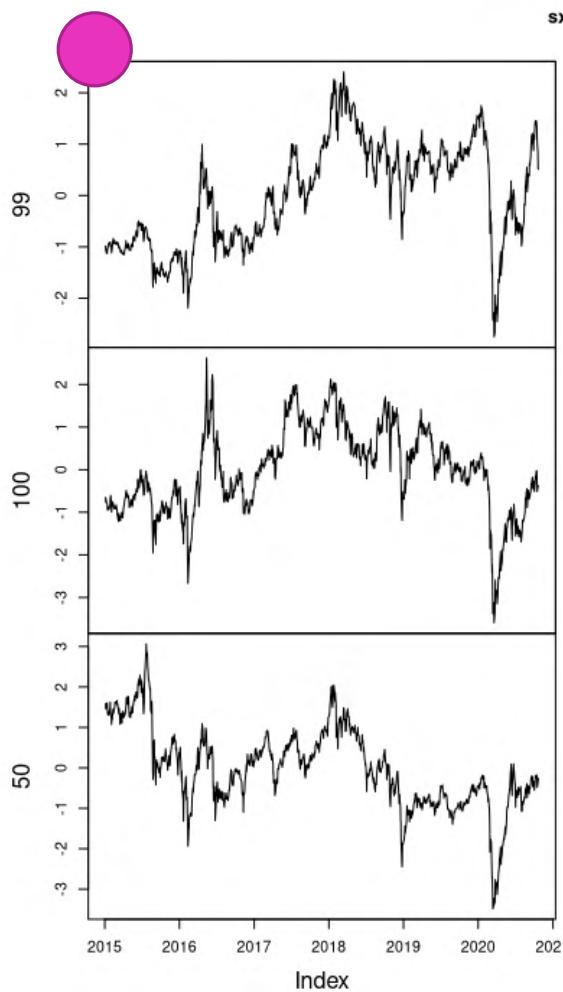
担当：磯貝 孝

時系列データの予測

- 時間経過とともに変化するデータ、例えば株価などを分析して将来予測をしてみたい（モデル分析の動機）
- 「回帰式的なアプローチ」： 感心のある変数（株価、売り上げ、気温など）に関係しそうな別の変数を探して、因果関係を探して、関係性をモデル化し、予測に応用する
- 「時系列分析的なアプローチ」： 関心のある変数の過去のデータだけを使ってモデルを構築し、将来予測に応用する ⇒ 今回はこちらの話
- データサイエンス的に近年よく使われる手法： 深層学習の応用
 - RNN、中でもLSTMは時系列データのパターン（長期的な依存関係など）を記憶・学習できるのでよく用いられる
 - RNN (recurrent neural network) 、LSTM(long short term memory)

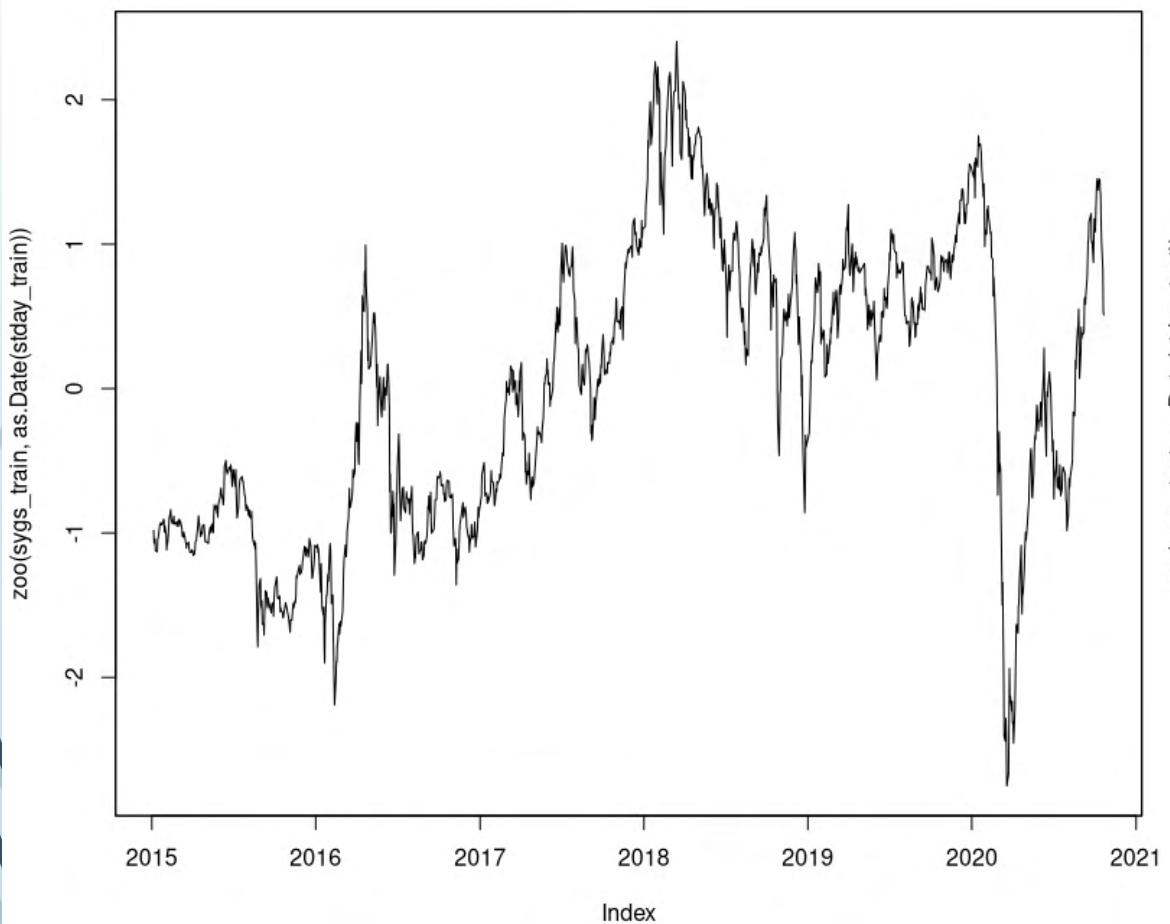
株式ポートフォリオの価値の予測

5つのポートフォリオがあって、過去の時系列データ（日次）がある。このうちの一つについて、5つの時系列データを使って**翌日の価値**を予測したい。

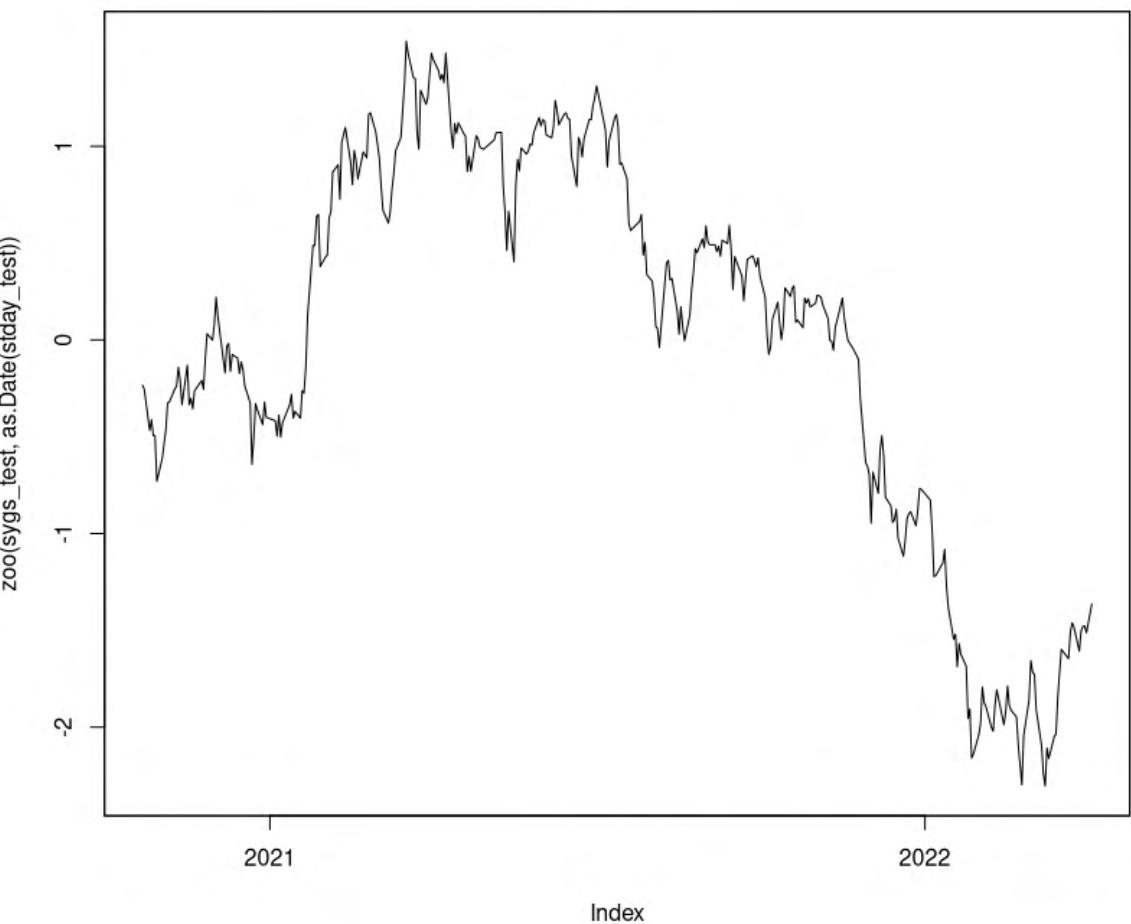


対象の時系列データを学習用と評価用に分ける

学習用データ

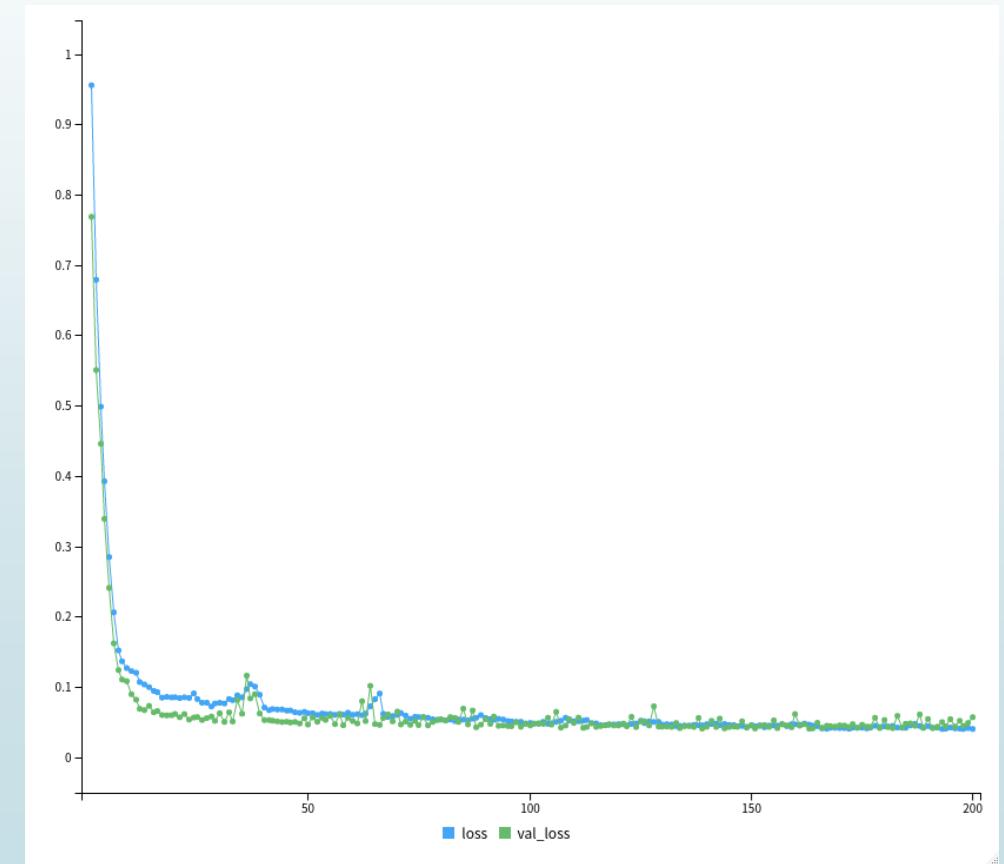


評価用データ



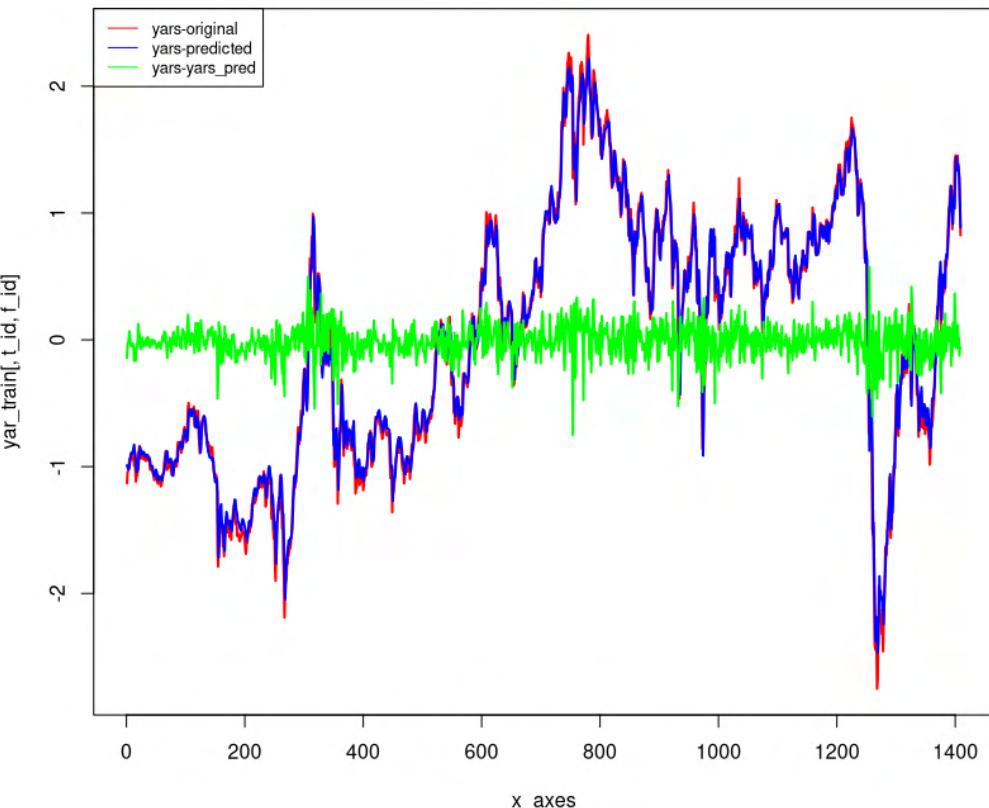
深層学習モデル（LSTMモデル）を構築して 学習させる

```
Model: "sequential_1"
Layer (type)          Output Shape       Param
=====
lstm_2 (LSTM)         (None, 5, 125)    65500
dropout_2 (Dropout)   (None, 5, 125)    0
lstm_1 (LSTM)         (None, 50)        35200
dropout_1 (Dropout)   (None, 50)        0
dense_2 (Dense)       (None, 5)         255
dense_1 (Dense)       (None, 3)         18
=====
Total params: 100,973
Trainable params: 100,973
Non-trainable params: 0
```

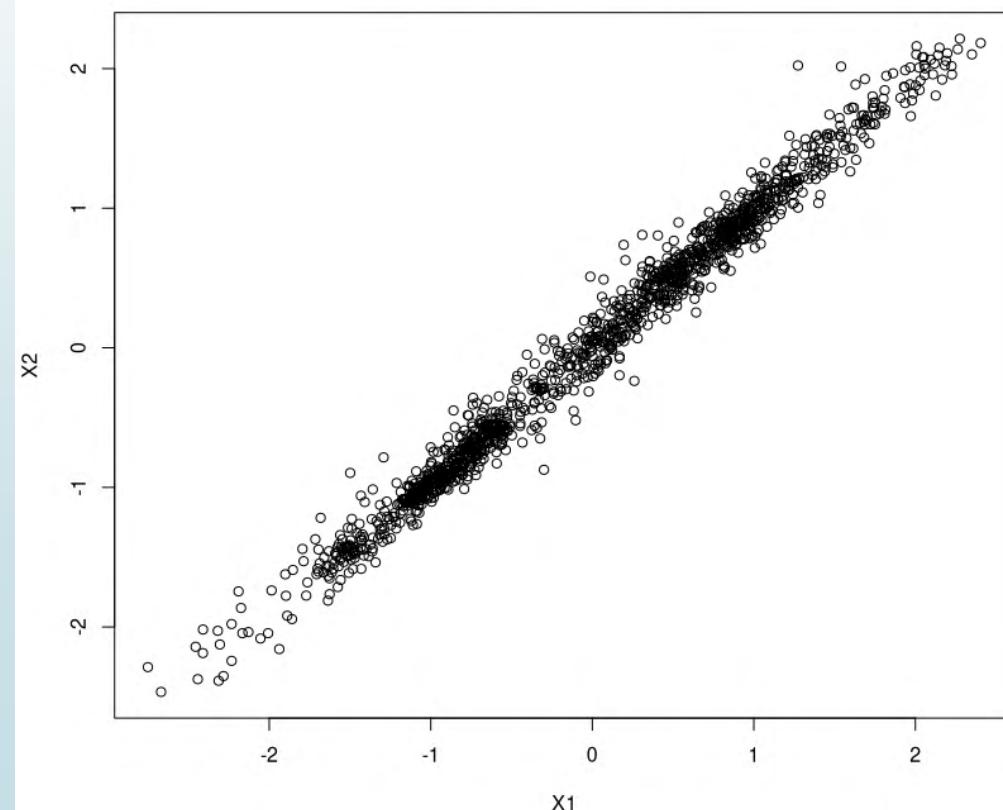


学習結果：モデルのフィット具合

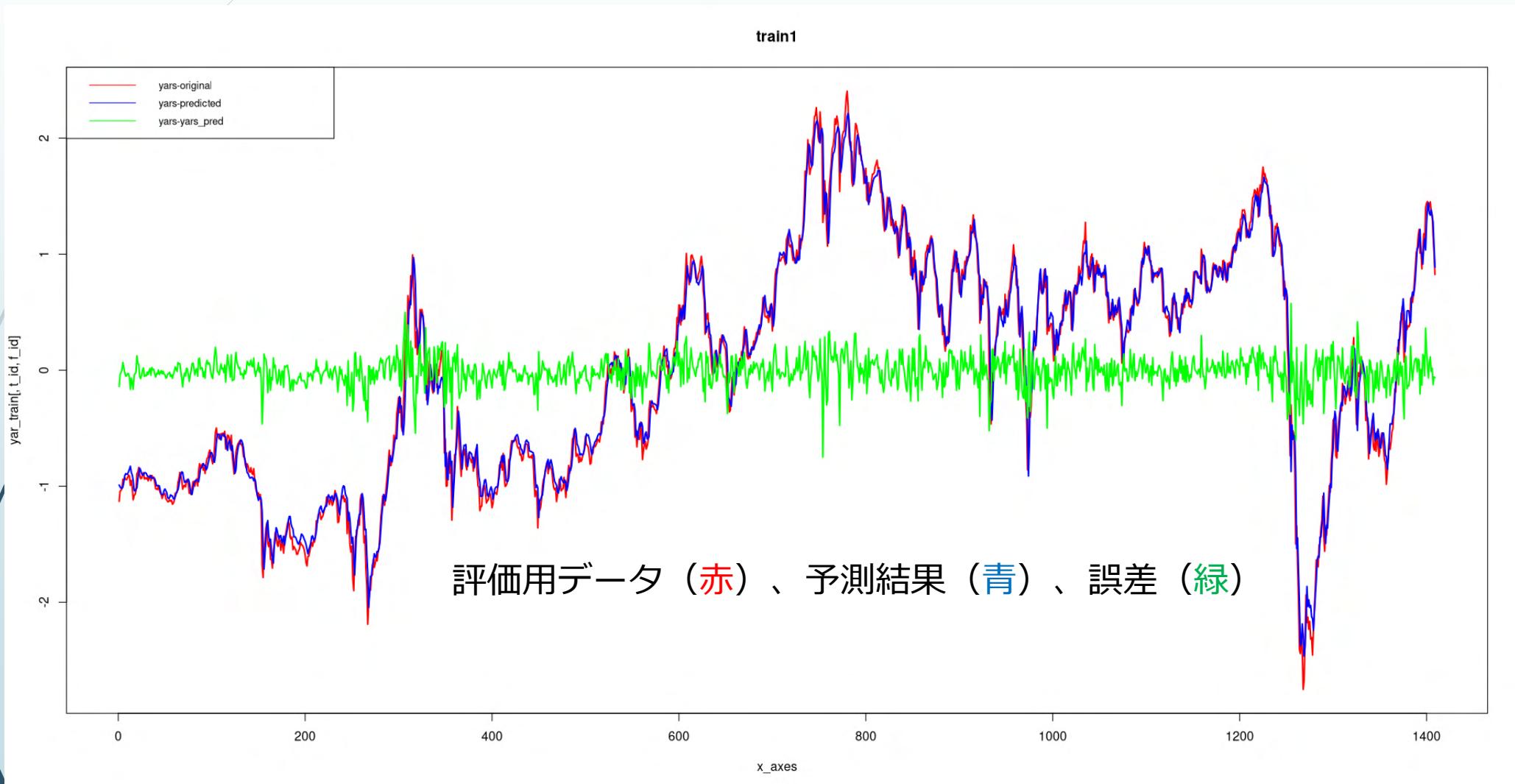
学習用データ（赤）と学習結果（青）、誤差（緑）



X1:学習用データ、X2 : 学習結果

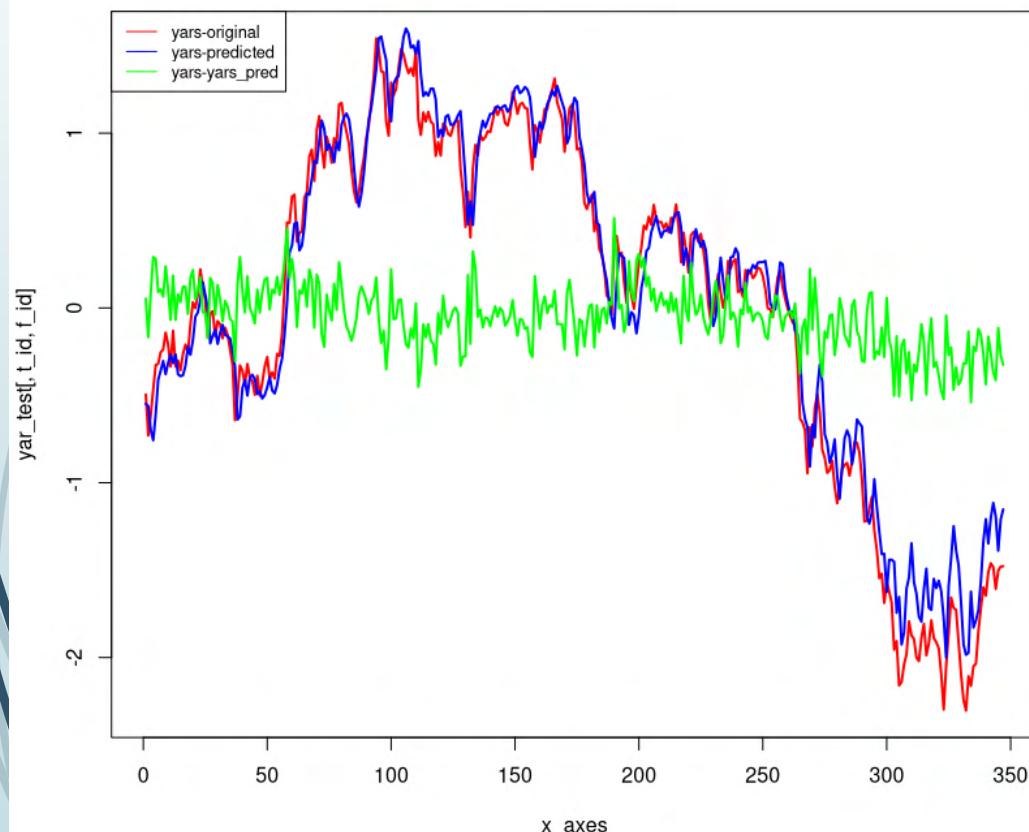


学習結果の拡大図

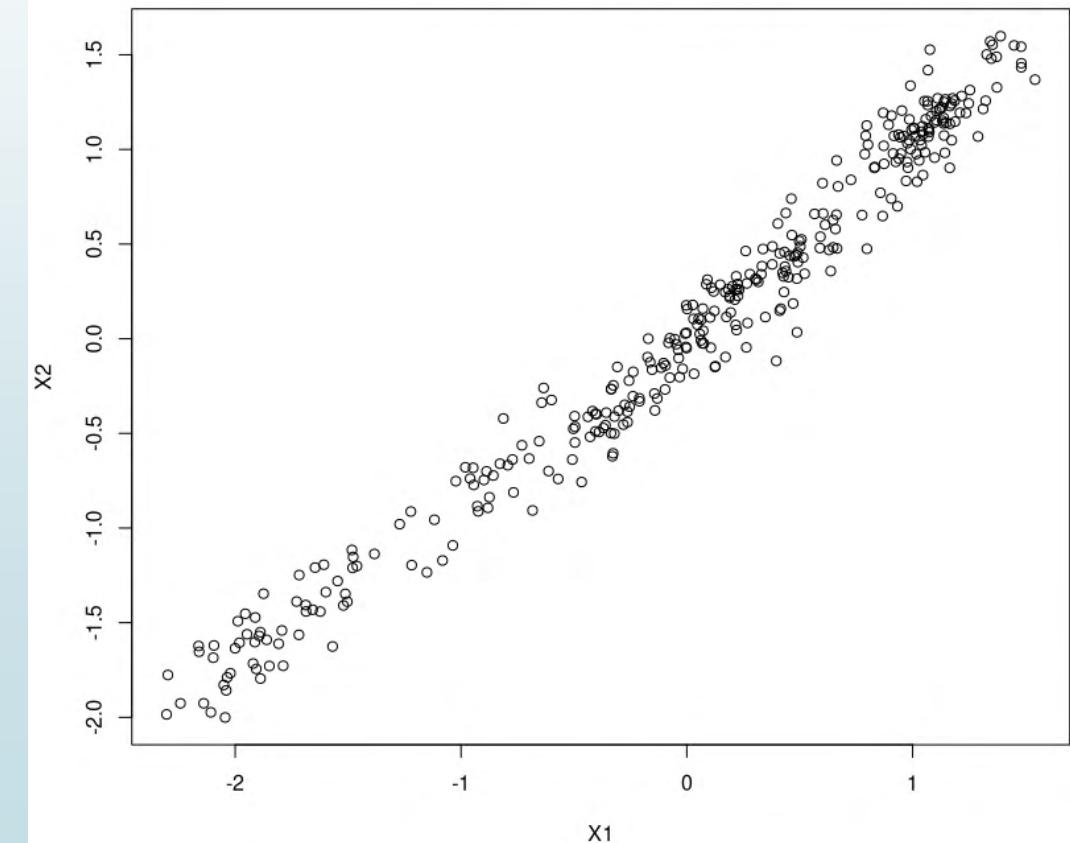


評価結果：学習用データよりも当てはまりは悪いが、 それなりにフィットしているように見える

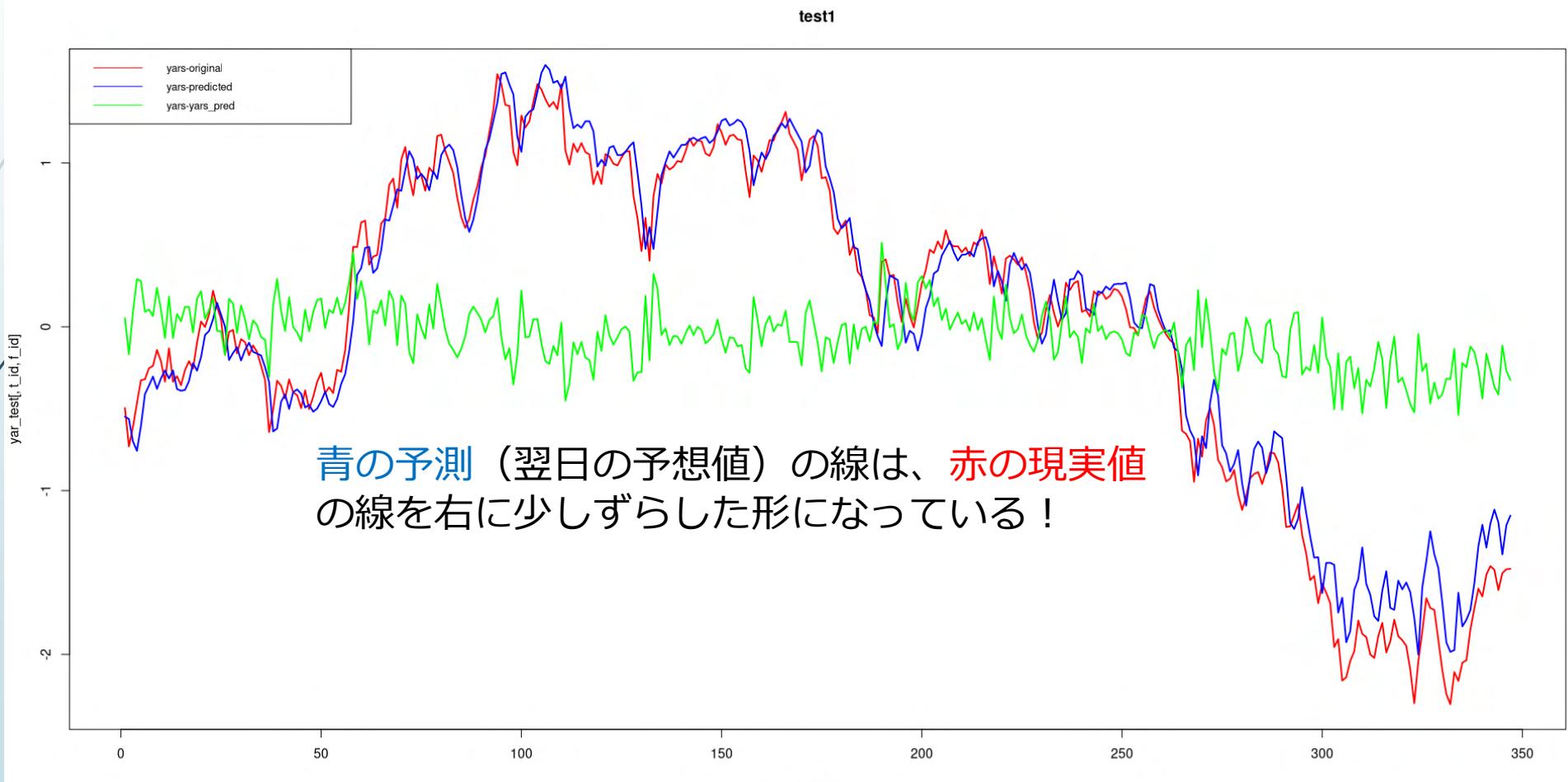
評価用データ（赤）と予測結果（青）、誤差（緑）



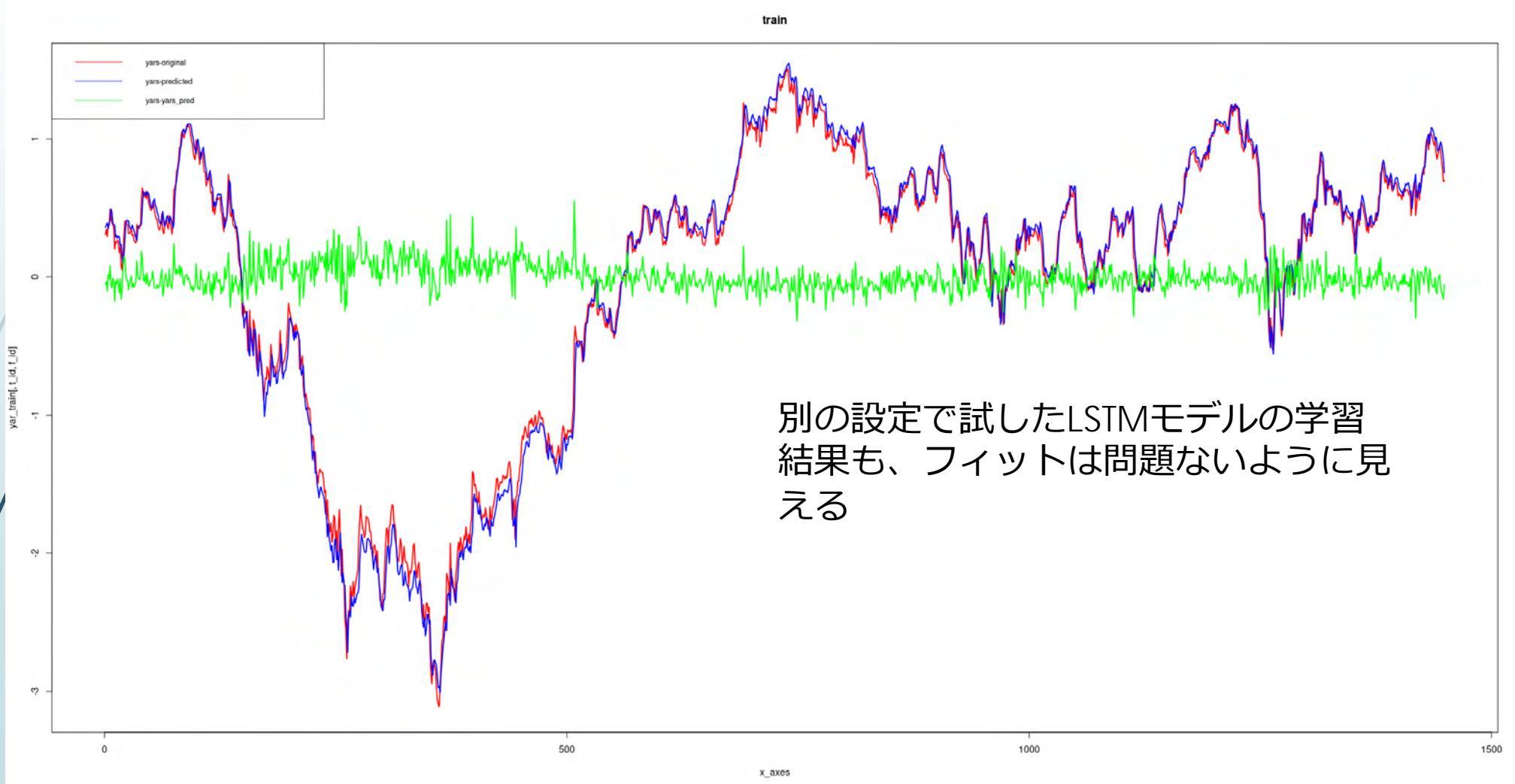
X1:評価用データ、X2：予測結果



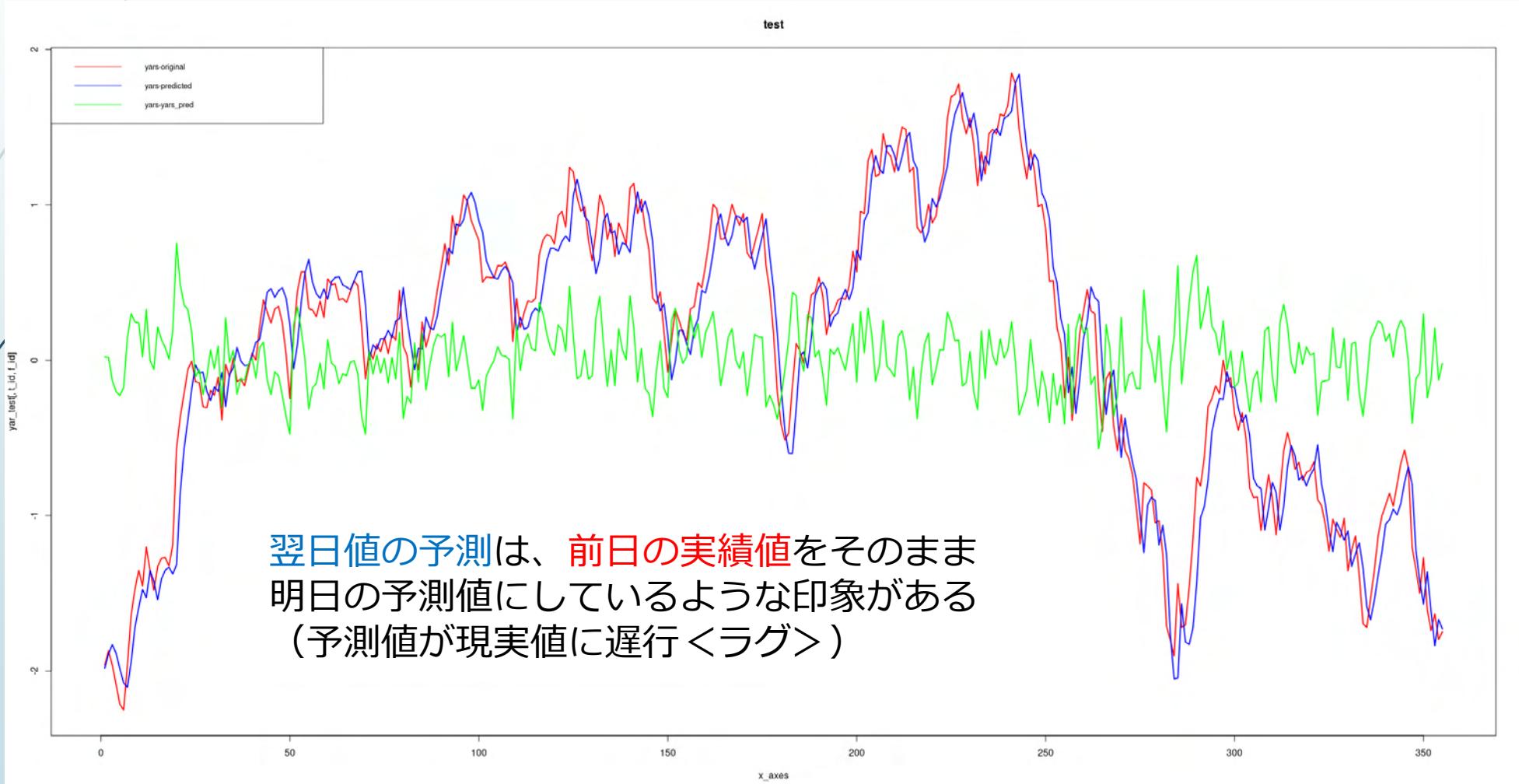
評価結果の拡大図：よく見るとグラフがX軸方向にずれている？



類似のモデルの結果



予想と現実のズレはより明確に出ている



(考察) 何が起きているのか

- ▶ 学習では、誤差(MSE, MAEなど) を最小化するようなパラメータを選ぶ(ARIMAモデルなどでも最尤法によるモデル推定はほぼ同じ概念)
- ▶ 学習の結果から得られた予測モデルは、

「明日もまた今日の如し」

のような感じになっている

- ▶ マルチングール : $E(|X_n|) < \infty, E(X_{n+1}|X_1, \dots, X_n) = X_n$

昨日までの変数 X の実現値を前提にした場合に、明日の X の期待値は今日の実現値となる（条件付き期待値）

- ▶ 変数 X がランダムウォークしている場合などで、この状態が成立することが知られている

ランダムウォーク？

- ▶ ランダムウォーク：（例） $x_t = \mu + x_{t-1} + \varepsilon_t, E[\varepsilon_t] = 0, V[\varepsilon_t] = \sigma^2$ （正規分布など）
 - ▶ 株価が上がる確率0.5、下がる確率0.5で（上下対称の場合）、上下動を繰り返していく
 - ▶ ランダム・ウォークなら、予測はそもそも困難（予測可能性がない）
 - ▶ こうしたデータに対して、深層学習モデルを応用しても結果的にマルチングールを学習しているような状況になりはしないか？
 - ▶ 今回経験した学習、予測結果もこれに近い状態であったと思われる
-
- ▶ 誤差最小化の観点では、マルチングール的な予測はおかしいものではない
 - ▶ ただし、我々が予測問題を考える際に、明日の予想は今日の値と同じです、と言われてその予測にどのような価値を見出すか、は人による？

予測可能性とエントロピー

- ▶ エントロピー指標などを応用して、そもそもどのくらいの情報量（シークエンスとしての規則性）がその時系列データに存在しているかを数値的に測定して、予測モデルへの応用可能性を事前に考えるというアプローチは有効かもしれない

- ▶ 例：Sample Entropy (SampEn), Approximate Entropy (ApEn)

$$ApEn(X) = 0.27$$

SampEn(X) = 0.22 #Low entropy value => high regularity (good predictability)

$$ApEn(\text{diff}(\log(X))) = 1.74$$

SampEn(\text{diff}(\log(X))) = 1.90 #High entropy value => low regularity (poor predictability)

$ApEn(\sin(t)) = 0.26$, $SampEn(\sin(t)) = 0.27$, $\langle t = \text{seq}(0, 10, 0.1) \rangle$ サインカーブ

$ApEn(rnorm(1000)) = 1.65$, $SampEn(rnorm(1000)) = 2.17$ 正規乱数

予測可能性に関する考察

- ▶ Xはレベルで低エントロピー（予測可能性が高そう）に見えて、対数前日差では高エントロピー（予測可能性が低そう）
- ▶ 元データにはなんらかの規則性があっても、対数前日差に変換すると、その情報はほとんどなくなっている？
- ▶ 例えば、局所的トレンドに関する情報はおそらく水準から対数前日差への変換で失われる
- ▶ LSTMで元の価格データの時系列からこうした情報を抜き出すことができても、対数前日差のモデルでは何の情報も抜き出せない（おそらく）
- ▶ 結局、価格データでの予測モデルを構築できて誤差をある程度小さくできても、線形トレンドに沿った価格予測や昨日の終値を次の日の予想価格にする（マルチングール）ような予測にしかならない
- ▶ こうした予測にどの程度の価値があるのか、予測モデルに期待するものにもよるが、価格予測モデルとしての有用性が高いと主張しにくい場面も多いはず

新しいモデルを使う場合の注意

- ▶ 従来の金融理論的なアプローチ： 統計的な前提についていろいろチェック（単位根検定、自己相関、正規性、分散変動のチェックなど）した上で、制約を満たすモデルを考える
- ▶ どうやらランダムウォーク的な動きだとみれば、予測の難しさは予め予想される
- ▶ 深層学習モデルでは、そうした理論的な制約なしにある程度自由にモデルを構築することが可能な点がメリット（何かしら答えは出る）
- ▶ 実際の活用に際しては、理論的な検討や従来のアプローチとの比較など複眼的にチェックしておいた方が安心
- ▶ 異なる分野の技術を応用する場合、伝統的なアプローチではどういう展開になりそうか、という点も考慮するのが無難か



Q&A

