

機械学習

2-1

工学院情報通信系

中原 啓貴

後半の内容

- 機械学習を全体的に俯瞰する
- 教師あり、教師なし、強化学習
- データの評価、事前準備
- Pythonを用いた演習
 - MAC or Windows PCを使います

参考書籍(必ずしも購入しなくてよい)



- Sebastian Raschka, [第2版]Python 機械学習プログラミング 達人データサイエンティストによる理論と実践 (impress top gear)



- 中井 悦司, ITエンジニアのための機械学習理論入門

評価方法

- **前半（熊澤先生担当） + 後半（中原担当）**
 - 各担当で 50 点
- **後半の評価**
 - 期末試験（11月22日実施予定）… 20点
 - レポート（期日は後日アナウンス）… 30点

スケジュール

- 10/29 1 データサイエンスと機械学習
- 11/1 2 最小二乗法 古典的な回帰分析、
- 11/5 3 機械学習ライブラリscikit-learnを使った分類問題
- 11/8 4 データ前処理
- 11/12 5 最尤推定法 確率を用いた推定理論、
推定量の一致性と不偏性
- 11/15 6 次元削減によるデータ圧縮
- 11/19 休講
- 11/22 7 アンサンブル学習 + 8 クラスタ分析
- 11/26 期末試験

データを知識に変える「知能機械」

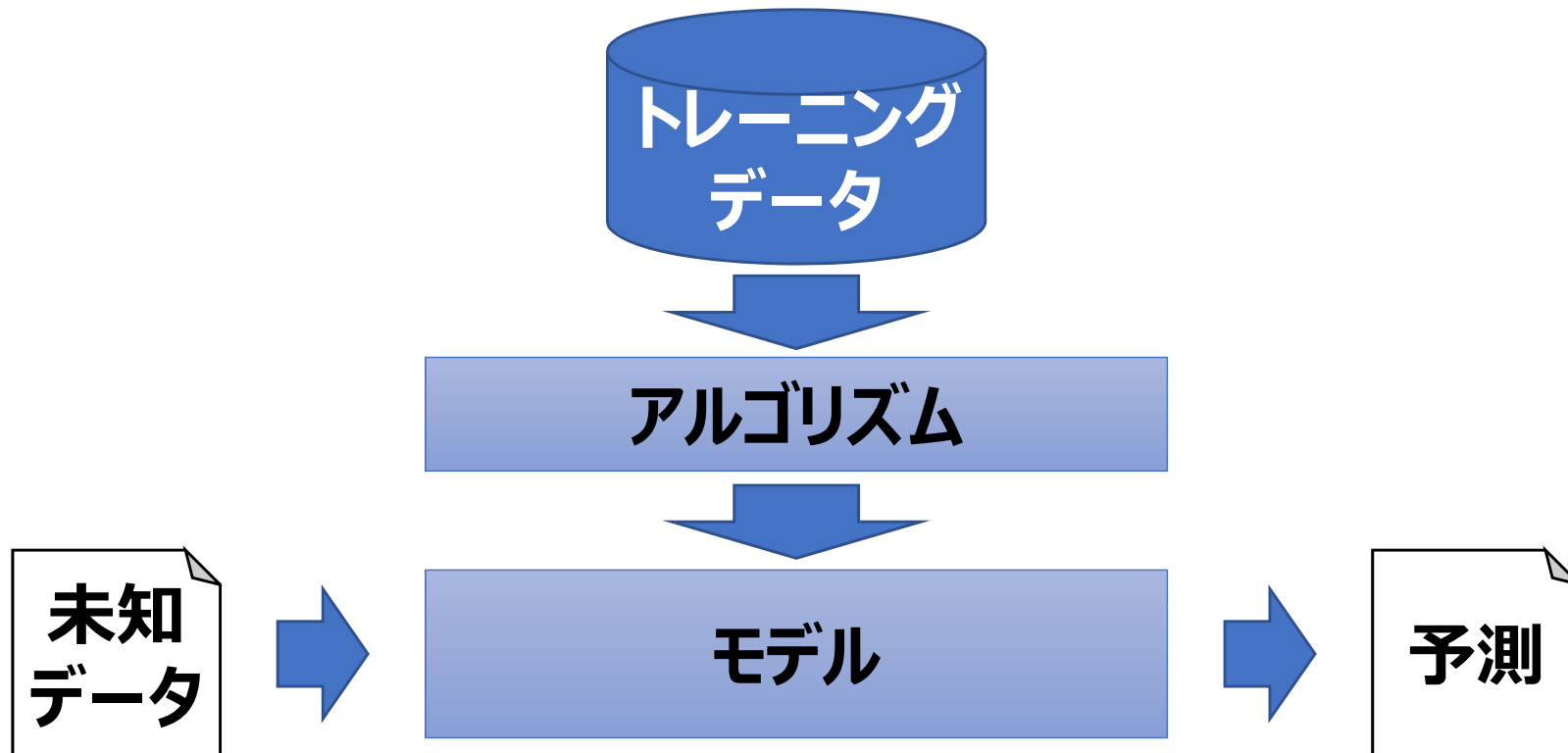
- **機械学習(Machine Learning)…人工知能の一分野**
- **人→データを分析してルールを導出、モデルを構築**
- **機械学習→データからモデルを引き出す効率的な手段**
- **Gmailなどのスパムフィルタ**
- **Siriなどの文字・音声認識**
- **GoogleなどのWeb検索エンジン**
- **Amazonのおすすめサービス**

3 種類の機械学習

- 教師あり学習(Supervised learning)
 - 分類
 - 回帰 (分類クラスラベルの回帰と考えれば, 分類とも言える)
- 教師なし学習(Unsupervised learning)
 - クラスタリング
- 強化学習(Reinforcement learning)
 - (教師なし学習に分類される場合もある)

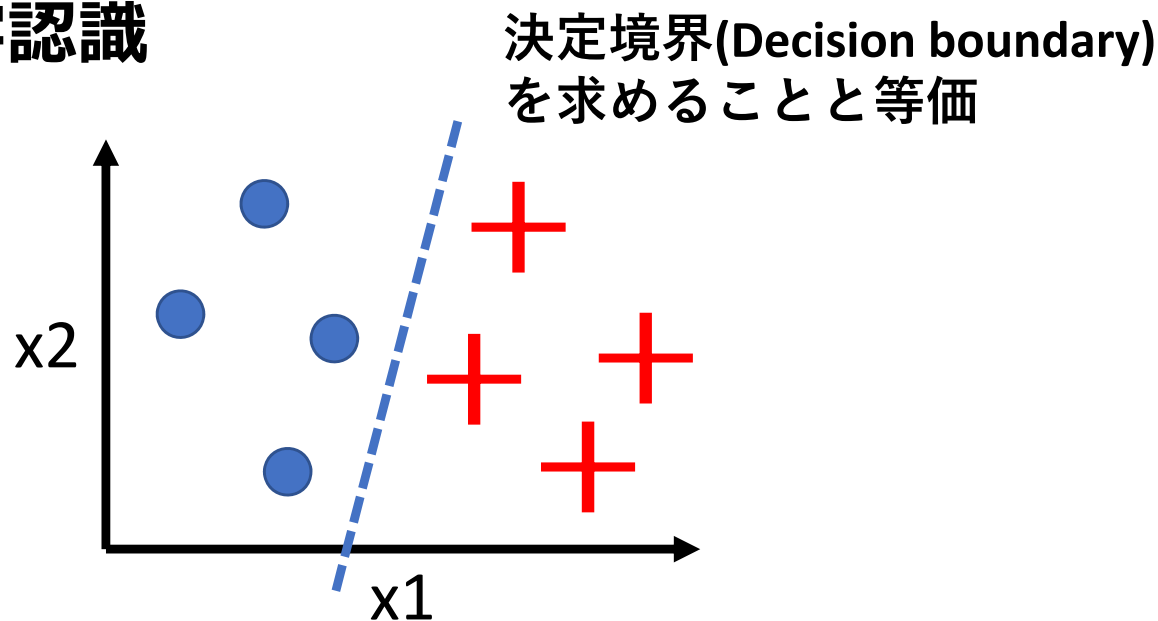
「教師あり学習」

- ラベル付けされたトレーニングデータからモデルを学習
 - トレーニングデータ：訓練データ＋学習データ
- 離散値のクラスラベルを持つ教師あり学習
→ **分類(Classification)**



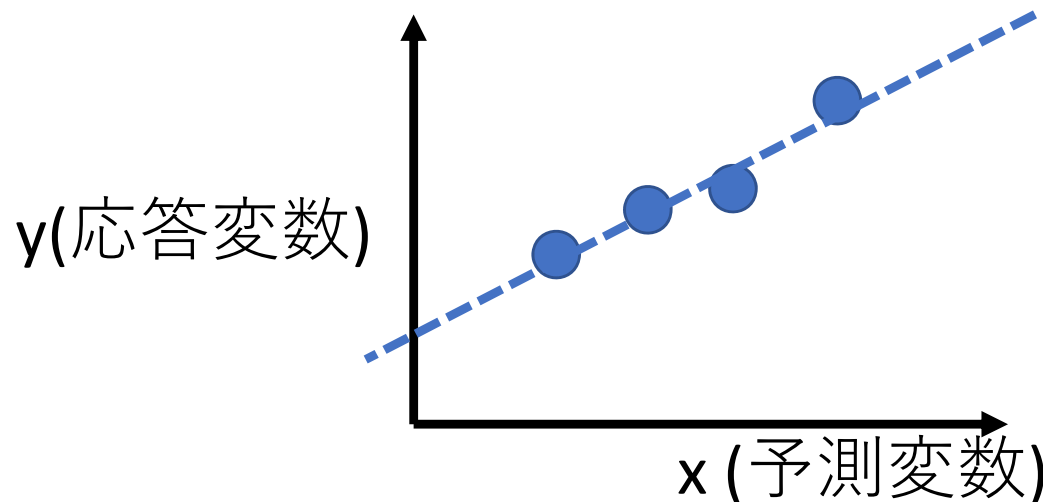
クラスラベルの予測

- クラスラベル：離散的で順序性のない値
- 過去の観測に基づき、インスタンス（サンプル）の所属として解釈
- 2値分類（Binary classification）
 - スпамメールフィルタ（スパムorハム）
- 他クラス分類(Multi-class classification)
 - 手書き文字認識



連続値の予測：回帰分析

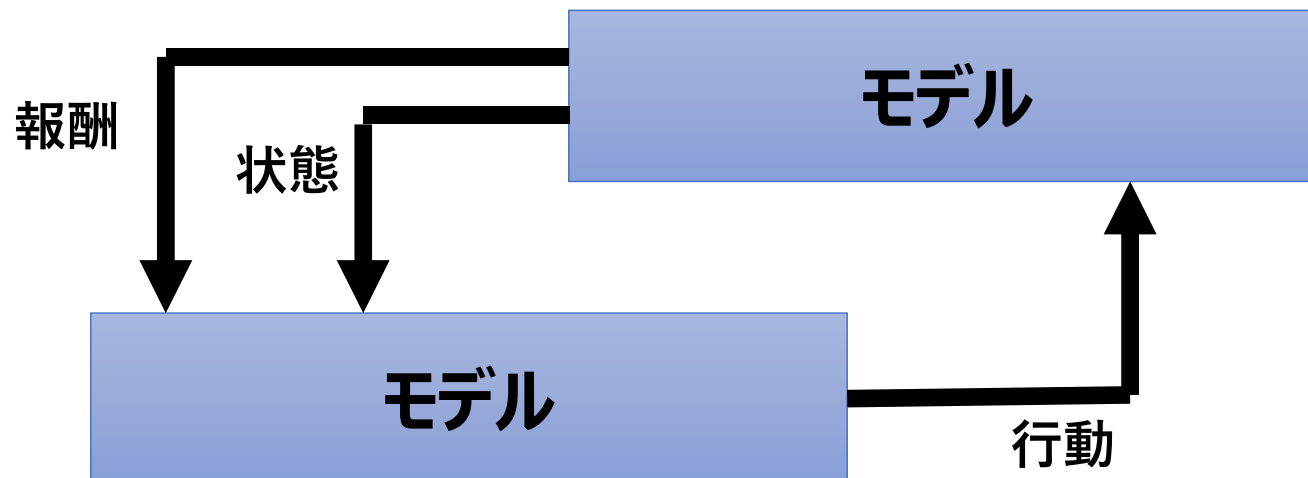
- トレーニングデータ：複数の予測変数(Predictor variable)と連続値の応答変数(Response variable)
- 予測に使用する変数：説明変数(Explanatory variable)
- 予測したい変数：結果変数(outcome)
- 例：線形回帰(Linear regression)



距離（主に 2 乗平均誤差）
が最も短くなる直線を
データに当てはめる
→説明変数に対する
結果変数を回帰（予測）

「強化学習」

- 環境とのやり取りに基づいて性能を改善するシステム（エージェント）を開発すること
 - 報酬：現在の状態に関する情報
- 教師あり学習ともいえる
 - ただし、正解ラベルや値ではなく「報酬（評価関数）」を用いる
- 例：ボードゲーム

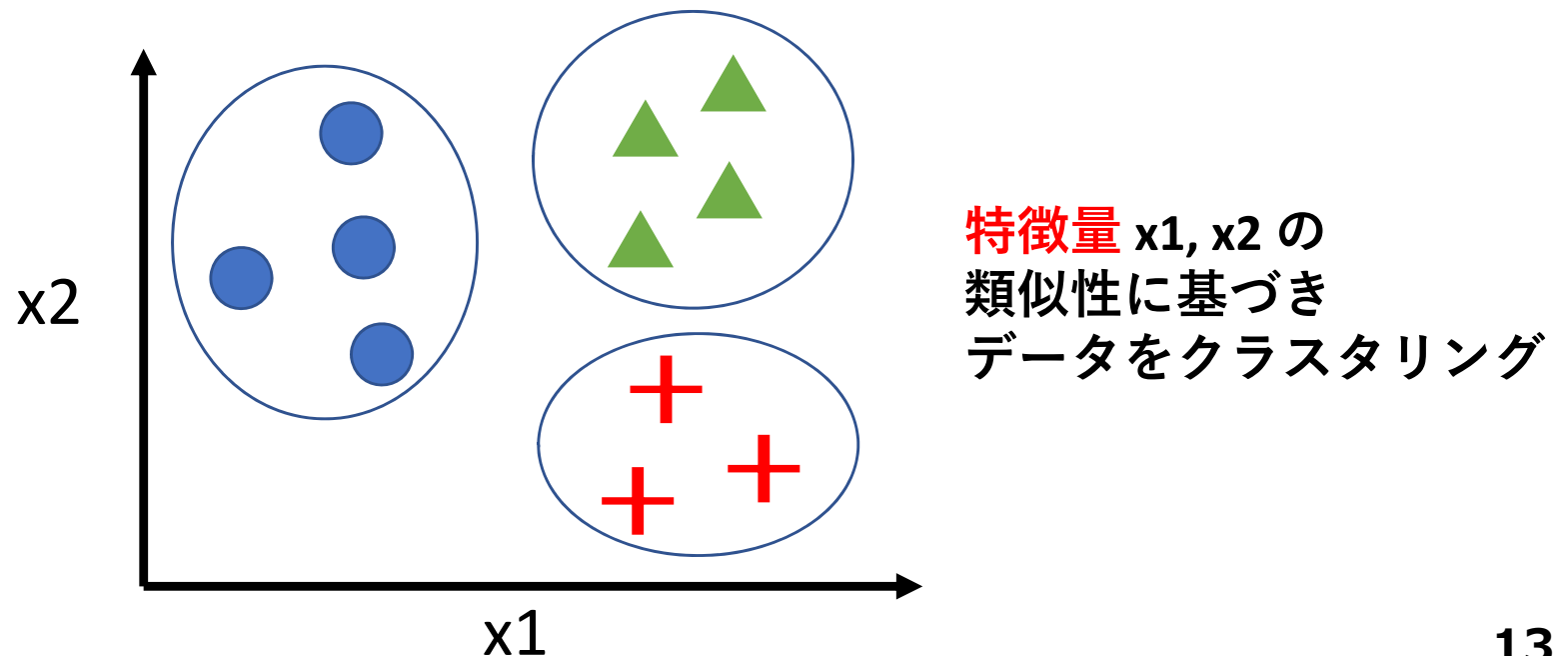


「教師なし学習」

- 教師あり学習→正解が事前に既知
 - 強化学習は正解の度合いを定義
- 教師なし学習
 - データ構造や正解ラベルが不明なデータを解析
 - 意味のある情報を抽出

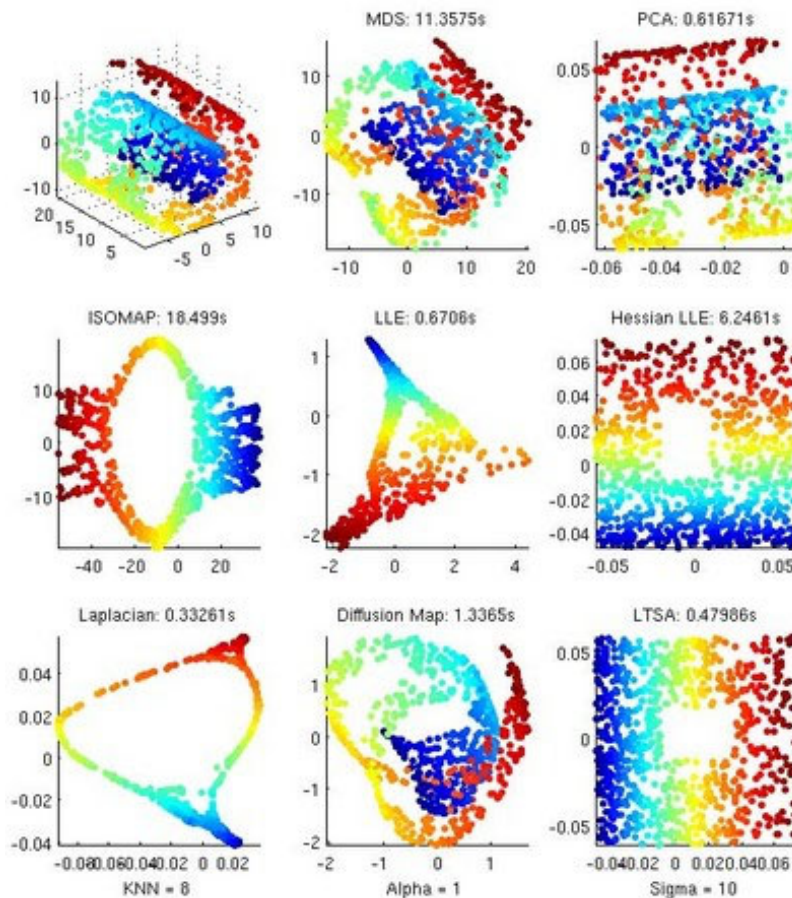
教師なし学習の例: クラスタリング

- 大量の情報を意味のあるグループ（クラスタ）として構造化
- 探索的データ解析の一種
- 例：マーケティング分析



教師なし学習の例:次元削減 (次元圧縮ともいう)

- 特徴量の前処理で多用
- データからノイズを除去して低次元の部分空間に圧縮

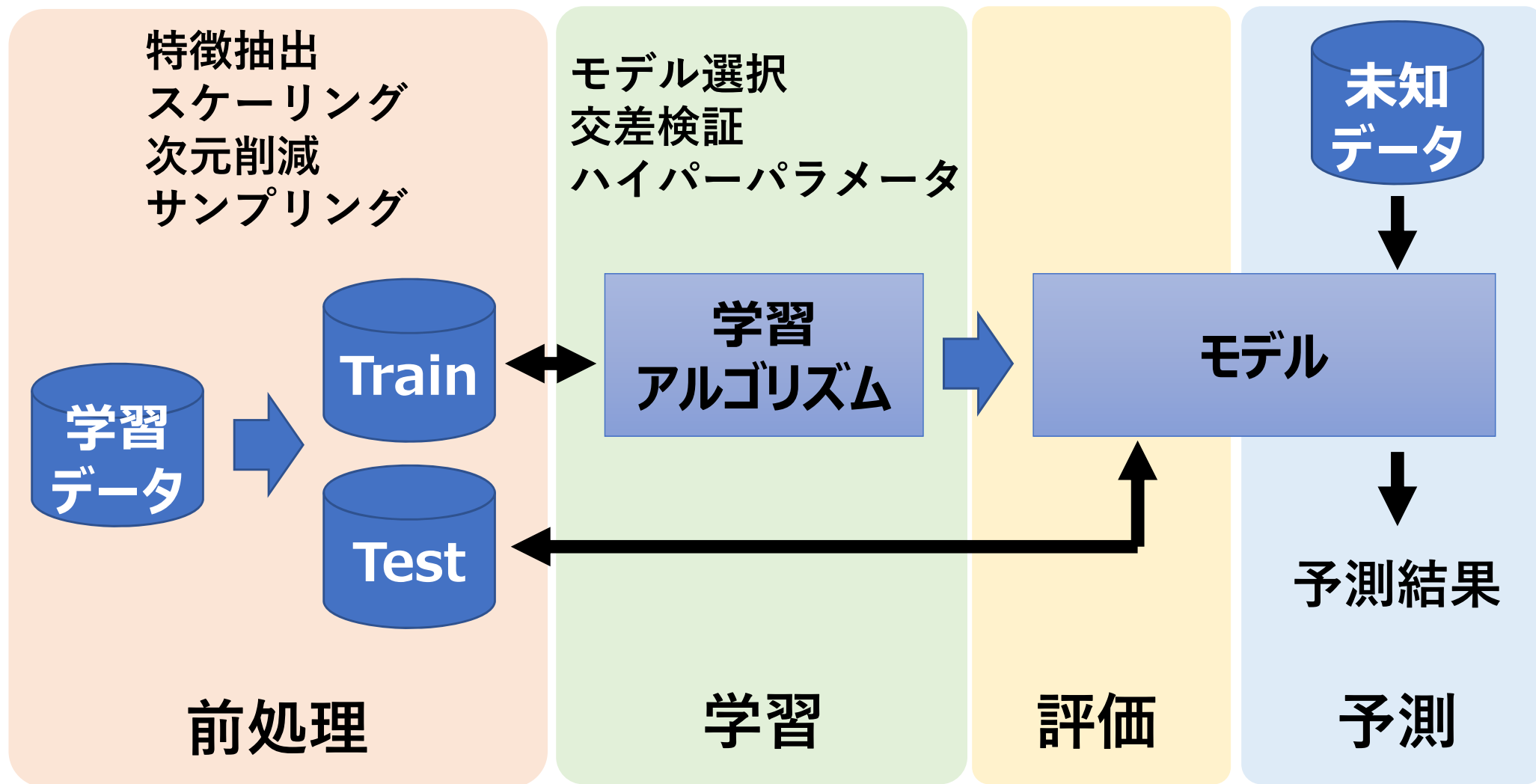


Source:

MANI fold Learning Matlab Demo

Todd Wittman @ University of Minnesota

一般的な機械学習のフロー



予測モデルの選択

- **ノーフリーランチ定理**

- 「あらゆる問題で性能の良い汎用最適化戦略は理論上不可能であり、ある戦略が他の戦略より性能がよいのは、現に解こうとしている特定の問題に対して特殊化（専門化）されている場合のみ」
- →問題領域の知識を可能な限り使用して最適化すべき

- **ハイパーパラメータ**

- 機械学習で学習されない、機械学習アルゴリズム上の（ハイパーな）パラメータ

Pythonの準備と演習

Python

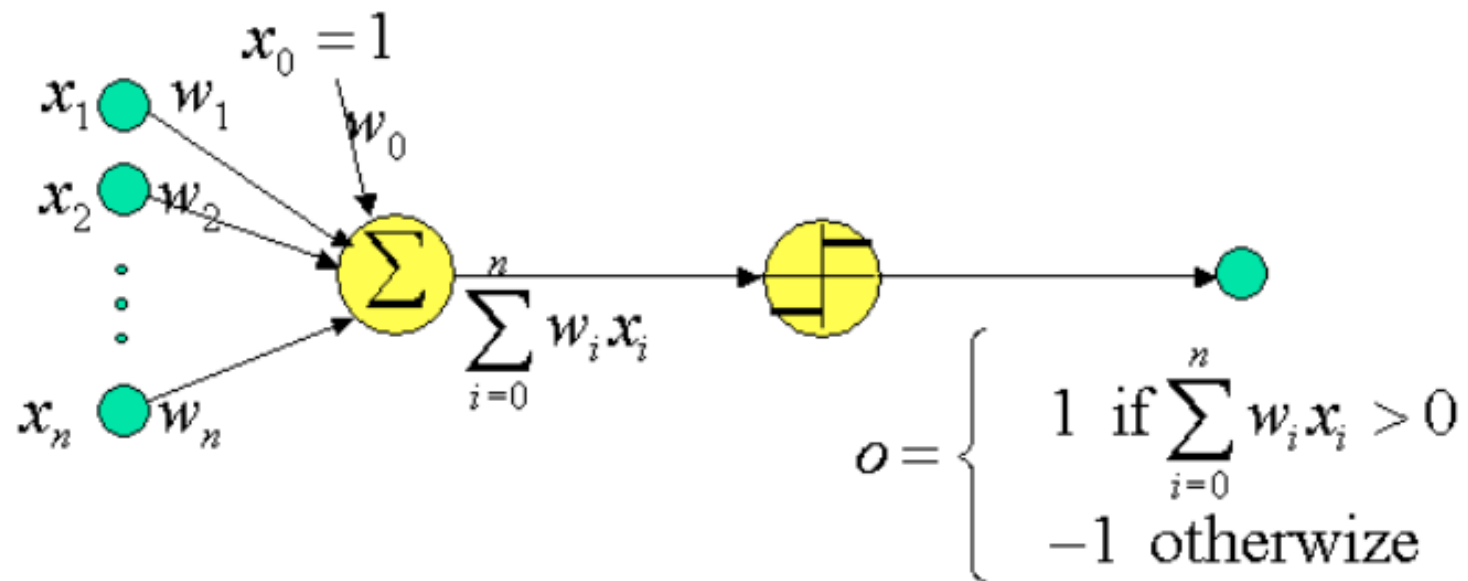
- 1991年に開発された汎用プログラミング言語
- シンプルで扱いやすい、ライブラリが豊富
 - C言語と比べて少ないコード行数で書ける
- if文等のブロックを字下げ（インデント）で指定
 - 誰が書いても同じコードになる
- 機械学習（特にDeep Learning）で実績あり

環境準備

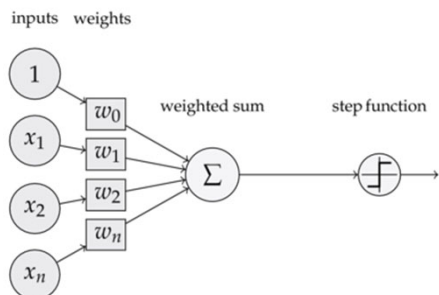
- 以下を参考にインストール
 - Python (ver3), numpy, scipy, matplotlib, pandas, scikit-learn (sklearn)
- Mac OS
 - 【決定版】MacでPythonを使って『機械学習』を学ぶための環境構築
 - https://qiita.com/yoshizaki_kkgk/items/4663148a2b3ca078ddbc
- Windows OS
 - 【決定版】WindowsでPythonを使って『機械学習』を学ぶための環境構築
 - https://qiita.com/yoshizaki_kkgk/items/1057ed4dcc36ed9be7f5

演習：パーセプトロンによる2値分類

- ニューラルネットワークの原始的なモデル

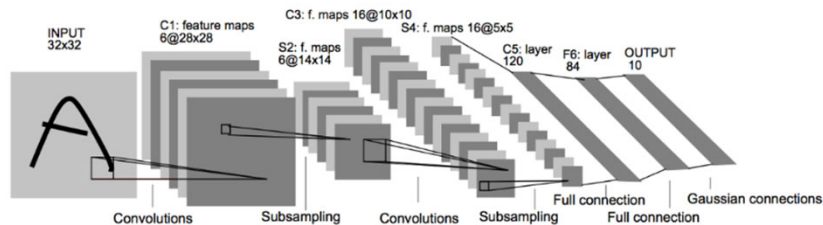


DNNの歴史



Perceptron

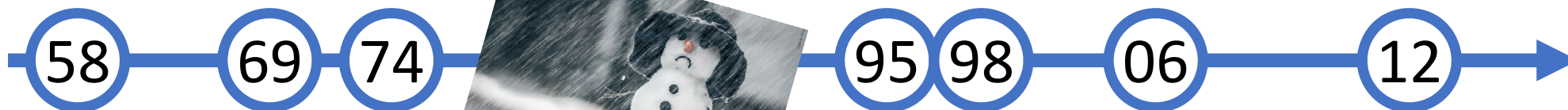
Back-propagation



Convolutional Neural Network



Google Brain Project



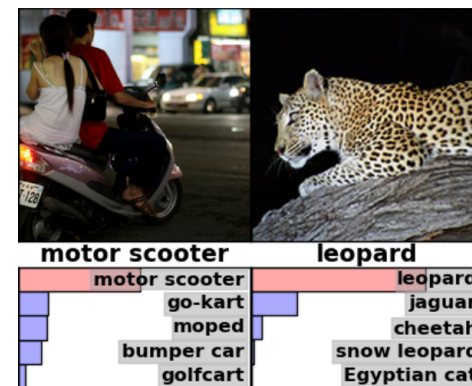
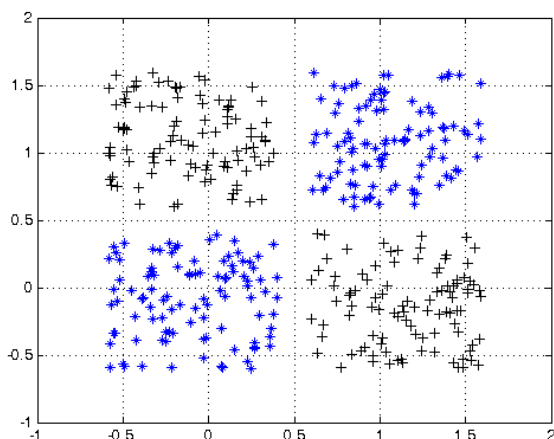
XOR問題



SVM

Restricted Boltzmann Machine

AlexNet が世界一に



パーセプトロン学習を実行

ソースコードを置いている場所

<https://github.com/HirokiNakahara>

- ランダムに生成したデータを使って分類
- 端末(Terminal)を起動

```
$git clone https://github.com/HirokiNakahara/MachineLearning_Lecture
```

→ カレントディレクトリに"MachineLearning_Lecture"がクローンされる

```
$jupyter notebook
```

→ ブラウザ(jupyter notebook)が起動する

jupyter notebook上で"MachineLearning_Lecture/lec2_1/perceptron.ipynb"
を開く

結果を見るだけならGithub上からも可能

The screenshot shows the GitHub interface for the repository 'HirokiNakahara / MachineLearning_Lecture'. The file 'lec2_1 / perceptron.ipynb' is selected, showing its content in a Jupyter Notebook format. The code is as follows:

```
In [1]: %matplotlib inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error

In [2]: # データ生成
from sklearn.datasets import make_classification

X, Y = make_classification(random_state=12,
                          n_features=2,
                          n_redundant=0,
                          n_informative=1,
                          n_clusters_per_class=1,
                          n_classes=2)

fig = plt.figure()
plt.figure(figsize=(8, 7))
plt.title("make_classification : n_features=2 n_classes=2")
plt.scatter(X[:, 0], X[:, 1], marker='o', c=Y, s=25, edgecolor='k')
plt.xlabel("Feature 1")
plt.ylabel("Feature 2")
```

まとめ

- ガイダンス
- 機械学習を俯瞰
- 環境構築
- パーセプトロンによる 2 値分類
- 宿題：
- スライド資料を参考に環境をインストールする
- パーセプトロンを動作させて確認